UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

CAMPUS ARAGÓN

ADMINISTRACION DE UNA ESTACION DE TRABAJO (INDY SILICON GRAPHICS)

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
JOSÉ MANUEL QUINTERO CERVANTES

ASESOR: ING. JUAN GASTALDI PÉREZ

México 1998.

Contenido

III. ADMINISTRACIÓN AVANZADA

IV. SEGURIDAD

V. MANTENIMIENTO

OBJETIVOS

Debido al hecho de que el uso de estaciones de trabajo es cada vez más común en las empresas, el mercado de trabajo para profesionistas, en especial de computación que estén capacitados en el manejo de estos equipos, se empieza a incrementar. En este ámbito, uno de los problemas a los que se enfrentan las personas que se piensen dedicar a esta labor, es la falta de material que muestre el funcionamiento tanto del equipo como del sistema operativo que se utilice. En el mercado se pueden encontrar gran cantidad de libros que hablan de UNIX en forma general, explicando los conocimientos básicos del administrador; pero al tratarse de temas avanzados, el número de ellos desciende y mucho más si se habla de uno en particular (como IRIX, HP-UX, Solaris, entre otros). Realmente una de las fuentes principales y fundamentales que se pueden encontrar, son los manuales del equipo; pero hasta ellos tienen un costo, y muchas empresas invierten en el equipo pero no en la compra de los manuales, y si lo hacen, los guardan con mucho celo, y resulta muy difícil que personas y mucho más si son externas a la compañía, tengan acceso a ellos. Por estos y otros motivos, se regresa al mismo punto de partida, la falta de información específica de un equipo o sistema en particular.

La presente tesis pretende ser una fuente de información valiosa para las personas interesadas en estas labores. Se ha elegido al sistema operativo IRIX, por ser el que se encuentra corriendo sobre los equipos de la compañía Silicon Graphics, que son uno de los más avanzados en el área de visualización; rama importante que en los últimos años se empieza a explotar profundamente, no sólo en el área de diseño, análisis e investigación, sino también en la de diversión, donde se utilizan para la creación de juegos, videos, animaciones, películas, etc., por el gran potencial que ofrecen.

De lo anterior se deduce el objetivo general de la tesis, que es: ***Brindar los conocimientos, tanto básicos como avanzados, para lograr administrar en forma eficiente un equipo (en particular INDY) con el sistema operativo IRIX.***

Para lograr lo anterior se cumplirán los siguientes objetivos particulares:

- Dar los conocimiento teóricos sobre el funcionamiento de los distintos componentes de IRIX
- Dar a conocer una serie de pasos que permitan, tomando los conocimientos anteriores, realizar tareas fundamentales del administrador.
- Señalar, tomando en cuenta la experiencia adquirida, puntos de interés y de sumo cuidado que deben contemplarse para evitar fallas y pérdida de información
- Dar referencias a fuentes de información que permitan ampliar explicaciones y conceptos.

Es conveniente mencionar que el hecho de conocer un equipo, no quiere decir que se será buen administrador; ya que la experiencia para aplicar lo aprendido y enfrentarse a problemas nuevos, se adquiere a través del tiempo y siguiendo buenas costumbres, como el realizar labores de mantenimiento y prevención de errores así como el mantenerse actualizado en cuanto a los avances tecnológicos, y en particular los referentes a su equipo.

Por otra parte, dado que los conceptos y técnicas que se requieren conocer son tantos y requieren casi siempre de conocimientos previos, se darán referencias a dónde recurrir para profundizar en cada uno de ellos.

Por último, lo expuesto en la presente tesis se aplicará al sistema operativo IRIX versión 5.1 corriendo sobre una estación de trabajo Indy; pero los fundamentos y conceptos expuestos pueden ser aplicados a otros equipos; únicamente se tiene que buscar los comandos análogos que realicen dicha labor, por lo que se facilitará la administración de nuevos equipos y sistemas, si se conoce bien uno. La versión 5.1 era la más actual al momento de iniciar los estudios para el desarrollo de este trabajo; a su término, la vigente era la 6.3. Esta última, incluye algunas mejoras importantes con respecto a la estructura del sistema de archivos, permitiendo manejar capacidades de almacenamiento mayores, tanto en discos como en archivos; pero la teoría y los conceptos son prácticamente los mismos.

En forma más específica, el administrador tiene que realizar lo siguiente:

- ***Instalación y pruebas del hardware***: Para lo cual se requieren conocimientos del funcionamiento y características de los diversos componentes físicos que integran al equipo; que permitirá armarlo e instalar periféricos sin mayores dificultades.
- ***Planeación***: Determinar la distribución de los discos, los usuarios y la carga de trabajo que soportarán los equipos.
- ***Instalación del Sistema Operativo (SO)***
- ***Configuración***: realizar las adecuaciones necesarias para que funcione en óptimas condiciones.
- ***Operación***: Operaciones rutinarias que van desde encender y apagar el equipo, solucionar cualquier problema que se presente, añadir y controlar usuarios, así como educarlos para que puedan optimizar el uso de los recursos, implantar reglas de uso, instalar nuevo software, etc.
- ***Administración de recursos***: Vigilancia del uso de CPU, espacio en disco, frecuencia de utilización de aplicaciones, bases de datos, etc., para poder llevar y controlar en forma adecuada su uso.
- ***Instalación y configuración en Red***: Conectar y configurar el equipo a la red; ponerse en contacto con el administrador de redes para establecer parámetros del equipo y políticas de uso.
- ***Seguridad***: Monitorear continuamente el uso del equipo y recursos para mantenerla al máximo; Implantar medidas y activar utilerías de seguridad, lo cual implica conocer profundamente el equipo y el SO.
- ***Análisis de fallas***: Es importante llevar una bitácora del funcionamiento y fallas del equipo; porque de ahí se pueden deducir y atender a tiempo problemas mayores antes de que ocurran.
- ***Respaldos***: La información es lo más valioso que se encuentra almacenado en el equipo; de ahí la importancia que tiene la labor de respaldar, y en su caso, restaurar la información almacenada.
- ***Actualizaciones***: Se requiere realizar un análisis antes de instalar una nueva versión de cualquier software; en el cual se determinen las ventajas, la compatibilidad con los archivos y sistema actual, los problemas que pueda tener, etc.
- ***Asistencia en dudas sobre el manejo del sistema***. Debe tener los conocimientos suficientes para poder brindar la asesoría necesaria a los distintos usuarios del equipo; o por lo menos, el poder guiarlos en la búsqueda de soluciones.

FALTAN PAGINAS

De la: 1

A la: 4

INTRODUCCIÓN

Silicon Graphics (SGI) es el principal vendedor de estaciones de trabajo (WorkStations) para el diseño mecánico asistido por computadora. Durante los últimos 10 años, los Sistemas de Silicon Graphics han sido utilizados en la industria del diseño, simulación visual, exploración de energía y en el entretenimiento - donde el poder del equipo y una visualización interactiva son esenciales para acelerar el proceso de creación -.

Los equipos que ofrecen van desde las pequeñas estaciones de trabajo Indy, hasta los sofisticados equipos de multiprocesador Power Challenge. Los equipos Indy son el punto de entrada a esta tecnología y no por ello son menos potentes; ofrecen un sistema de captura digital completo que permite integrar sonido, video y elementos de 3D al trabajo que se desarrolle.

La filosofía de las estaciones de trabajo es la de tener al alcance del usuario y en un equipo de escritorio, una gran potencia de proceso. La potencia de estos equipos llega a tal grado, que son capaces de trabajar con varios usuarios (locales y remotos) al mismo tiempo y corriendo múltiples procesos, sin que disminuya en forma notable su rendimiento. Debido a esto, el sistema operativo elegido es generalmente el UNIX.; el cual ofrece un estupendo desempeño y aprovecha al máximo los recursos y potencia que ofrecen estos equipos.

En particular, el Sistema Operativo (SO) utilizado en los equipos Indy es el llamado IRIX, que es la versión particular de UNIX implementada por SGI. Es un sistema multiusuario y multitareas; permite controlar y anexar diversos dispositivos al equipo, como impresoras, discos, cámaras de video, unidades de cinta, etc.; además, permite conectar el equipo a una red, y de esta forma, compartir sus recursos con otros.

Junto con todas estas ventajas que brinda el SO vienen consigo una serie de responsabilidades para establecer, mantener y solucionar cualquier problema que se presente con su funcionamiento. A este conjunto de responsabilidades se les conoce como **administración del sistema** y son las que se detallarán aquí; dando las bases, procedimientos y puntos de especial atención, así como experiencias y consejos para obtener un documento que pueda ser de utilidad para cualquier administrador de sistemas, que es la persona encargada de llevarlas a cabo.

Aunque lo expuesto en esta tesis se encuentra enfocado al equipo Indy y SO IRIX, la teoría y los fundamentos en que se basan pueden ser utilizados para otros equipos; únicamente se tiene que encontrar las instrucciones equivalentes para llevarlas a cabo. Puede ser que los comandos sean los mismos, o que tengan otro nombre, o que se encuentren ubicados en un directorio diferente, o que los procedimientos sean un poco diferentes; pero las bases son las mismas.

Tomando en cuenta lo anterior así como los objetivos que se plantean cubrir, la tesis abarcará 5 capítulos, además de un glosario y varios apéndices de la siguiente forma:

En el capítulo 1 se discutirán los aspectos que deben tomarse en cuenta durante la instalación física del equipo, además de contemplar los conceptos y pasos para instalar el SO correctamente.

El capítulo 2 cubrirá los aspectos básicos de la administración; como el encender y apagar correctamente el equipo, crear y administrar usuarios, etc.

En el capítulo 3 se describen labores avanzadas de administración, para lo cual se exponen conceptos profundos y técnicos sobre la estructura y funcionamiento del sistema; como lo relacionado a los sistemas de archivos, optimización del SO, reconfiguración del kernel, etc.

El capítulo 4 es relacionado con la seguridad. Aquí se tratarán políticas que deben implantarse y técnicas que permitirán mantener con un grado mayor de confiabilidad, la información almacenada en los equipos.

En el capítulo 5 se describirán las labores que deben realizarse para mantener en buen estado y operando el equipo; así como prevenir, y en dado caso que ocurra un mal funcionamiento, solucionarlo en forma eficiente y rápida.

Finalmente, se remata con una serie de apéndices donde se describen ciertos aspectos fundamentales del sistema y un glosario de términos para facilitar el entendimiento de la tesis.

Como último punto y antes de empezar el tratado, es conveniente señalar cuáles son las responsabilidades del administrador; que se irán cubriendo a lo largo de este documento.

Responsabilidades del administrador

El administrador de sistemas es el encargado de administrar los recursos del equipo que tiene a sus cargo; por lo que es responsable de la instalación y configuración del Sistema Operativo, mantenerlo en buenas condiciones y repararlo cuando presente fallas. Es decir, sus responsabilidades son todas aquellas tareas que se encuentran fuera del alcance de los usuarios normales, ya sea porque se requiere de privilegios especiales dentro del sistema o porque tiene que ver directamente con el manejo del equipo (instalación o remoción de partes).

CAPÍTULO I

INSTALACIÓN

Descripción de conceptos y procedimientos para realizar la instalación de los diferentes componentes de una Estación de Trabajo.

I. INSTALACIÓN

La instalación es el proceso durante el cual son colocados todos los componentes de un equipo en condiciones que permitan su funcionamiento adecuado. Ésta la podemos dividir en dos partes:

- Instalación del hardware, en la cual se acondicionan y conectan todos los componentes físicos del equipo y
- La instalación del software, en la que son colocados dentro de la máquina todos los programas necesarios para poder operarla correctamente.

I.I. Instalación del hardware

La instalación del hardware es un proceso relativamente sencillo, pero que se tiene que realizar con cuidado. De ello depende la vida útil del equipo así como su buen funcionamiento. Existen tres puntos en los cuales se debe poner especial atención:

- Selección del lugar.
- Adecuación de las instalaciones.
- Conexión de los componentes del sistema.

I.I.I. Selección del lugar

La adecuada selección de la ubicación donde será colocado el equipo, permitirá garantizar un buen desempeño, y por lo tanto, la vida útil del equipo no se verá afectada.

La visión de un cuarto de cómputo donde se controla la humedad y temperatura (que generalmente eran bajas) está cambiando poco a poco. A medida que la tecnología avanza se crean nuevos sistemas de cómputo, que pueden operan en condiciones y a temperaturas ambientales; lo cual reduce enormemente los gastos de instalación de sofisticados sistemas para acondicionar el lugar.

Lo anterior no quiere decir que se debe descuidar este punto. El lugar seleccionado debe ser un sitio limpio, libre de polvo y con buena ventilación. Si es un cuarto cerrado y con el transcurso del tiempo se calienta, se debe instalar una unidad que baje la temperatura a condiciones normales de operación. Por lo general se puede decir que si el operador del equipo tiene calor la computadora también.

Para evitar un calentamiento excesivo del equipo se debe tomar en cuenta las siguientes indicaciones:

- Colocarlo sobre una superficie plana y firme
- No colocar carpetas ni manteles debajo de él, ya que esto interfiere con la circulación de aire por la parte inferior del gabinete y genera un calentamiento mayor
- No debe ser colocado en lugares cerrados, ni cubierto con fundas mientras esté encendido
- Procurar dejar el espacio suficiente a su alrededor para que exista una adecuada ventilación; si es colocado junto a un pared, se recomienda dejar una separación como mínimo de 15 cm. entre el equipo y la pared.

I.I.II. Adecuación de las instalaciones

Una vez seleccionado el lugar, éste se debe acondicionar. Se debe poner especial atención en tres factores:

- Sistema de alimentación de energía
- Fuente de respaldo de energía (UPS: Uninterruptible Power Supply, "Fuente de Poder Ininterrumpida") y si se requiere
- Un sistema de aire acondicionado.

Un sistema de energía regulado y aterrizado, es indispensable y permite evitar posibles daños al equipo, así como el sistema de energía ininterrumpido. Esto es debido a las características propias del sistema operativo UNIX, el cual guarda información relativa a modificaciones y actualizaciones del sistema de archivos del disco duro en la memoria RAM, y cada determinado tiempo realiza la actualización física en el disco duro; de tal forma que si se suspende la energía, la información de RAM se pierde y con ello el sistema de archivos del disco queda corrupto. Los daños que puede causar una interrupción de corriente van desde una pérdida total de la información del disco (caso extremo) hasta el caso ideal en el que no se pierde nada.

Si la energía que llega al sitio no cumple con estas características, se puede adquirir un UPS el cual cumple con las dos funciones, regular y mantener un respaldo de energía por determinado tiempo.

El último punto es el de un sistema de aire acondicionado. Como ya se mencionó, los equipos pueden estar operando a temperatura ambiente, pero si las condiciones del lugar son extremas, se debe adquirir éste. Recordar que toda inversión es en beneficio del equipo, y por ende, de la información que se encuentra dentro de él, que representa lo más valioso.

I.I.III. Conectar los componentes del sistema

Ésta es la última fase en el proceso de instalación física, en ella se debe comenzar por inspeccionar el estado en que llegó el equipo, hasta terminar con la conexión de todos los componentes; de tal forma que se encuentre listo para ser encendido.

Al llegar el equipo se debe inspeccionar el estado físico de las cajas en las que viene empaquetado, si presentan daños graves, probablemente el equipo también los tenga; por lo cual, es conveniente revisar con mayor detalle cada una de sus partes. Si presenta algún deterioro se debe reportar inmediatamente al distribuidor para que sea cambiada o reparada la parte afectada.

La Fig. I-1 muestra los componentes principales que vienen generalmente, dentro de las cajas.

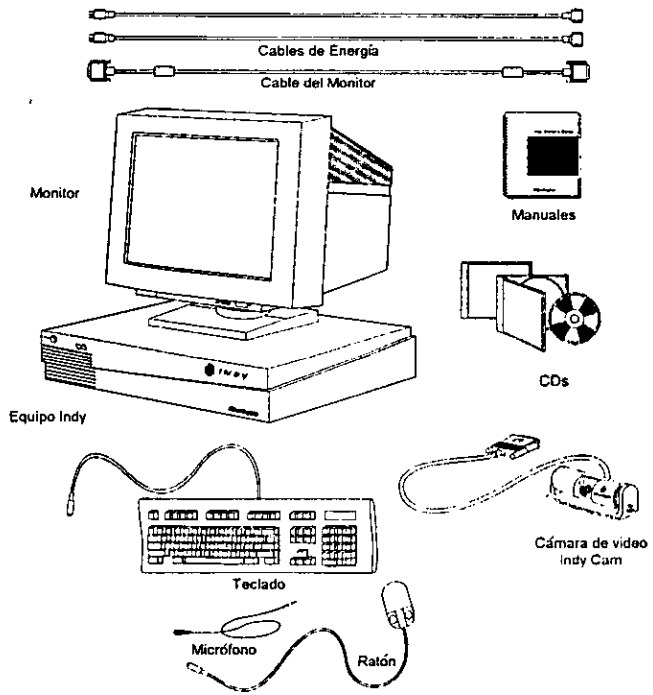


Fig. I-1 Componentes del equipo

Adicional a estos dispositivos se pueden adquirir otros periféricos que permitan mejorar el funcionamiento del sistema, como son: unidades de disco, CDROM, cinta y cartucho, controladores de discos ópticos, impresoras, plotter, modems, etc.

Es importante recordar que siempre se debe leer todos y cada uno de los documentos que acompañan al equipo, en especial, antes de realizar la instalación: el de Material Packing List, el cual contiene la lista de todos los componente que vienen en la caja, tanto manuales, equipo o documentos, y de esta manera podemos detectar piezas faltantes; el de Readme Before Installing, que describe una serie de pasos y medidas que se deben tomar antes de realizar la instalación; el de Owner's Guide, que describe las características del equipo y los pasos para instalar diversos componentes opcionales; así como cualquier hoja de actualización, que puede contener información de último momento y que no se encuentra dentro de los manuales o, descripción de erratas de los documentos.

Un buen administrador debe estar acostumbrado a leer éstos y todos los documentos disponibles para el equipo, ya que una y muy importante función que tiene, es la de mantenerse actualizado respecto a las capacidades, características y funcionamiento, tanto del hardware como del software que se encuentre disponible en el equipo.

Un error que se puede cometer, es el tratar de realizar la instalación sin tener la experiencia y el conocimiento necesario para efectuarla, y no leer los manuales. Por querer ahorrar tiempo al no leerlos, se pueden presentar problemas que la dificulten, u ocasionar daños al equipo.

I.I.III.I. Conexión física

En la Fig. I-2 se ilustra la parte posterior del equipo Indy, en ella se muestran los distintos puertos con que cuenta y los periféricos que deben ser colocados en cada uno de ellos. Se debe tener cuidado al introducir el conector para no dañarlo. Debe alinearse e introducirse firme y suavemente; si no se acopla, revisar la alineación y volver a intentarlo.

La conexión del monitor, teclado, micrófono, ratón y cámara de video no tiene mayor problema. Para facilitar la instalación, cada uno de los puertos tiene dibujado un pequeño gráfico que indica qué dispositivo debe ser conectado en él, y por su parte, el conector de cada periférico tiene grabado la figura correspondiente. Otro detalle más, tanto el puerto como el conector correspondiente están diseñados para que únicamente exista una sola posición (la correcta) en la que se pueden acoplar sin ningún problema.

Los puertos restantes son para dispositivos adicionales que pueden ser conectados al equipo cuando se adquieran, pero uno de ellos merece especial atención, el del puerto SCSI. A

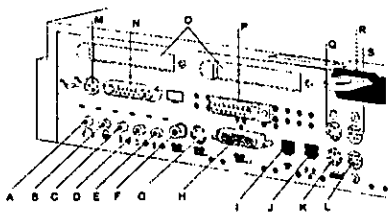


Fig. I-2 Descripción de puertos.

A. Audífonos, soporta señal estéreo.
B. Micrófono, soporta señal estéreo.
C. Entrada de audio analógico, soporta señal estéreo.
D. Salida de audio analógico, soporta señal estéreo.
E. E/S de audio digital serial.
F. Entrada de video compuesto análogo
(Formatos RCA, NTSC o PAL).
G. Entrada de video-S análogo.
H. Entrada de video digital, (cámara a color Indy Cam).
I. Puerto ISND (Integrated Services Digital Network).
J. Puerto Ethernet 10-BASE T.
K. Teclado
L. Puerto Serial 2
M. Puerto Stereo View, para colocar Lentes 3D.
N. Monitor
O. Ranuras de Expansión; tarjetas GIO (Graphics I/O).
P. Puerto Ethernet AUI.
Q. Ratón.
R. Barra de bloqueo, para candado de protección.
S. Puerto Serial 1, o Consola.

diferencia de los demás en los cuales sólo se puede conectar un dispositivo, éste es un bus[1] al cual se pueden conectar hasta 7; ver Fig. I-3.

La información viaja por el bus y todos los dispositivos que estén conectados a él, la pueden escuchar; por lo que para distinguir a quién va dirigida, se utiliza una dirección. Cada dispositivo SCSI cuenta con un método que le permita definirla; unos utilizan puentes (jumper), otros interruptores en miniatura (mini dip swichs), o también botones selectores que permiten cambiar la selección actual. Las direcciones que se utilizan son de la 0 a la 7;

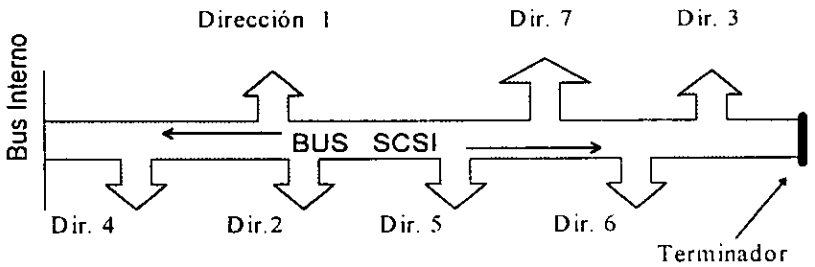


Fig. I-3 Diagrama de la arquitectura del BUS SCSI

[1] Un bus no es más que un grupo de cables paralelos sobre los cuales, varios dispositivos se pueden comunicar.

pero la 0 está reservada para direccionar a la tarjeta controladora SCSI en sí y, por acuerdo, la dirección 1 se da al disco interno donde se coloca el SO y que sirve de arranque, así como la 2 que se asigna al segundo dispositivo interno, ya que los equipos Indy vienen diseñados para dar alojamiento a un máximo de dos unidades. Por tanto, se pueden conectar hasta 5 dispositivos externos con direcciones de 3 a 7, ninguno de los cuales puede tener una dirección repetida.

Las unidades externas son conectadas una a continuación de la otra formando una cadena, ver Fig. I-4. Al final se debe conectar un terminador, que tiene la función de impedir que la información siga viajando indefinidamente por el bus. La longitud máxima permitida es de 2 mtrs. Si no es conectado ningún dispositivo externo, se debe colocar el terminador en el puerto SCSI.

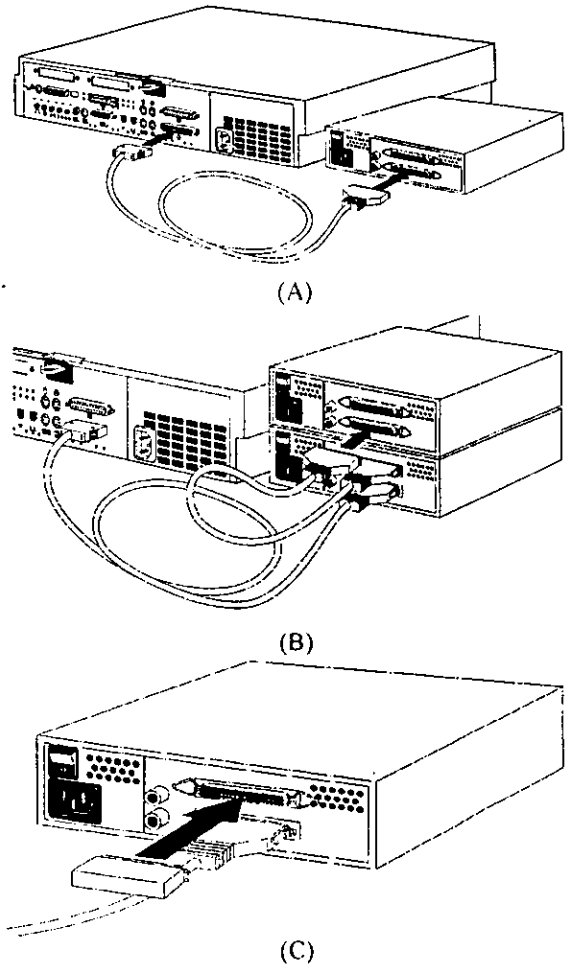


Fig. I-4 Conexión en cadena de unidades externas SCSI.

La importancia de realizar la conexión correctamente y poner al final el terminador, reside en lo siguiente: Al encender el equipo, una de sus funciones es la de revisar todos los dispositivos que se encuentren conectados al bus, así como reconocer sus direcciones para poder establecer una comunicación con cada uno de ellos, si alguno no está encendido al momento de realizar estas pruebas, el sistema no lo reconocerá después, dado que estas pruebas se realizan únicamente al encender el equipo. Por este motivo se deben encender en primer lugar, los dispositivos externos que se encuentren conectados al bus y al final el equipo. Una vez que la unidad sea reconocida por el sistema, ésta se puede apagar y encender cuando sea necesario y el sistema la seguirá reconociendo. La única forma de que

el sistema reconozca a un equipo que no estaba encendido es la de dar de baja el equipo, apagarlo y volverlo a encender, pero en esta ocasión, el dispositivo debe estar encendido.

Al igual que la computadora, cada uno de los dispositivos externos al momento de ser encendidos, realizan una serie de pruebas en forma independiente para determinar su estado y quedar finalmente en operación. Dependiendo del dispositivo, puede que la prueba sea casi instantánea o que dure algunos minutos. Esta información se puede ver en el manual de cada uno de ellos. Por este motivo se debe dar el tiempo suficiente para que todos los periféricos estén listos antes de encender la computadora. Si se desconoce el tiempo de prueba que requiere alguno, es recomendable encenderlo y esperar a que el sonido que emite sea estable, dar un lapso de más y encender el equipo.

Finalmente, una vez conectados todos los dispositivos al CPU, éstos deben ser conectados a una fuente de alimentación de energía regulada e ininterrumpible (ver. "Adecuación de las instalaciones" en el inciso anterior). Efectuar una última revisión a las conexiones y encenderlos. La regla general es:

Secuencia de Encendido.

- Si existen dispositivos externos, encenderlos.
- Dar un tiempo para que arranquen y queden en operación.
- Encender el monitor.
- Encender el CPU.

La Fig. I-5 muestra los controles para encender tanto al CPU como el monitor.

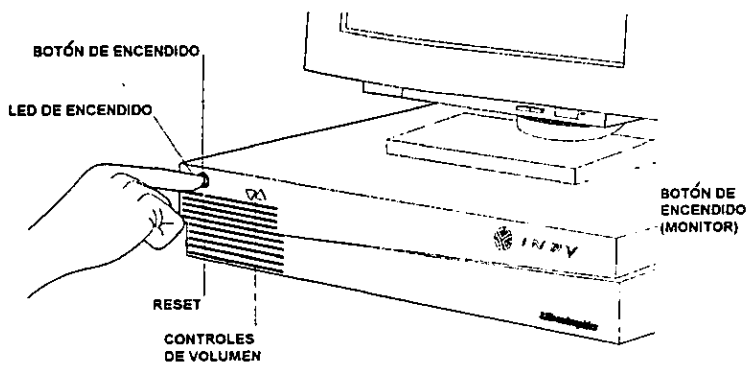


Fig. I-5 Controles frontales

I.I.III.II. Proceso de encendido

Al encender el equipo, el LED de encendido se torna en color rojo, después de un momento cambia a verde y se escucha la melodía de arranque. En seguida cambia a rojo nuevamente y en la pantalla se despliega la Fig. I-6 indicando que se están realizando las pruebas de encendido y reconocimiento de dispositivos.

Fig. I-6 Ventana que indica la ejecución de las pruebas de diagnóstico

Si se detecta algún problema se despliega una pantalla que permitirá corregirlo - que sería el caso si no se encuentra instalado el SO y que se describe en el punto siguiente..

Por otra parte, si las pruebas son pasadas satisfactoriamente, el LED cambia a verde y en la pantalla se mostrará un mensaje indicando que el sistema se empezará a dar de alta (Fig. I-7). En este punto se puede detener el proceso de arranque para utilizar algunas herramientas que permiten corregir daños o instalar el SO.

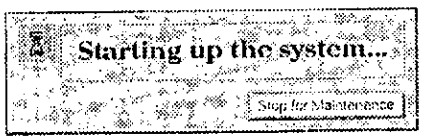


Fig. I-7 Ventana de arranque del SO y acceso al monitor PROM[2]

Después aparecerá el que indica que se está dando de alta el sistema (Fig. I-8), y a partir de este momento, se desplegarán varios más (dependiendo de la configuración) hasta que finalmente aparezca la pantalla de bienvenida (Fig. I-9).

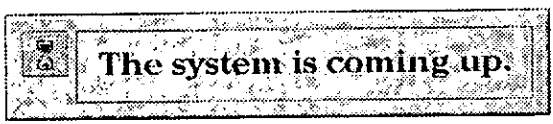


Fig. I-8 Ventana indicando que el SO se está cargando en RAM.

[2] Ver Instalación del software.

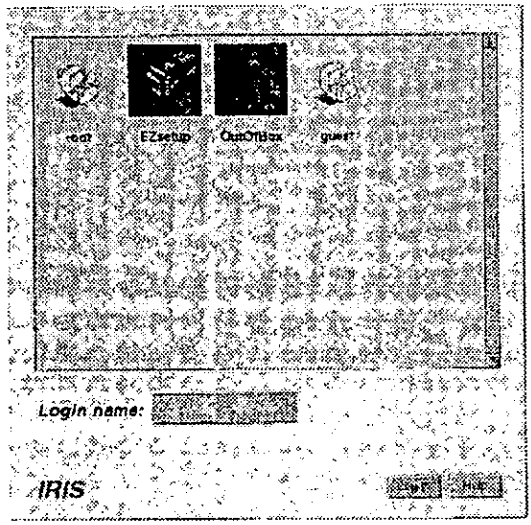


Fig. I-9 Ventana de Bienvenida (Acceso al sistema).

En este momento el sistema se encuentra operando y se puede trabajar con él. Es importante recordar que existe un procedimiento adecuado para darlo de baja y poder apagar el equipo, y que por ningún motivo se debe apagar sin realizarlo; ya que se corre el riesgo de dañar en forma irreparable la información almacenada en los discos (ver Capítulo II).

I.II. Instalación del software

La instalación de software en general la podemos clasificar de varias formas, dependiendo de donde se encuentre físicamente el dispositivo de lectura o de si es necesario dar de baja el sistema para llevarla a cabo (ver Fig. I-10).

El software que se instala en los equipos puede venir grabado en varios medios, como CD o Cinta a los cuales se les llama *medios de distribución* . Claro está que, dependiendo del medio en el que se encuentre el software, es la unidad que se requiere para leerlo. Si el dispositivo se encuentra conectado físicamente al equipo en el que se instalará el software, se dice que la instalación será *LOCAL*. Si no se cuenta con un dispositivo de lectura local y el equipo está conectado a una red, se puede utilizar la unidad de otro equipo y en ese caso estamos hablando de una instalación *REMOTA*.

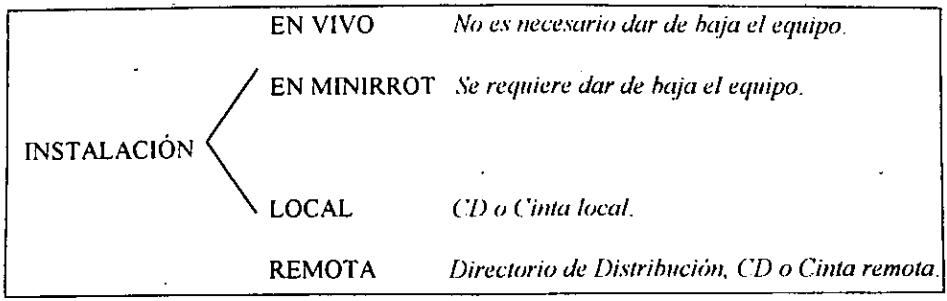


Fig. I-10 Tipos de instalaciones.

Durante la instalación remota, el equipo local se conecta con el remoto para tener acceso al software, que puede estar montado en la unidad de CD o cinta remota, o previamente grabado en un directorio de su disco. Para que todo esto se pueda llevar a cabo, se deben configurar tanto el equipo local y remoto, como los gateway que intervengan en la comunicación de los dos equipos[3].

Por otra parte, si para llevar a cabo la instalación no es necesario dar de baja el equipo, se dice que es una *instalación en vivo* y es efectuada mediante el comando ***inst*** que será descrito más adelante. En este caso el sistema operativo se encuentra funcionando y se pueden realizar otras operaciones al mismo tiempo. Este método es preferible cuando se pueda realizar, ya que no interrumpe las funciones del sistema ni de otros usuarios y requiere menos tiempo para efectuarse; pero consume más espacio de disco durante la instalación, ya que para no interrumpir ninguna función, realiza copias de los archivos que se vayan a modificar o instalar y que se estén utilizando en ese momento. Dado que para efectuar este tipo de instalación (local o remota) el SO debe estar funcionando, este método se utiliza para instalar cualquier otro software que no sea el sistema operativo.

Si para llevar a cabo la instalación es necesario dar de baja el equipo, estamos hablando de una *instalación de miniroot*. Este método se utiliza cuando ocurren problemas durante una instalación en vivo o cuando las notas que acompañan al software así lo indican. También se tiene que efectuar ésta, cuando el sistema operativo se encuentra dañado y no puede arrancar o cuando el disco del equipo es nuevo y no contiene software. Este método es fundamental, ya que durante la vida de cualquier equipo, se puede llegar a dañar el sistema y éste es el único método con que se cuenta para tratar de solucionar el problema y arrancar el equipo nuevamente.

Sea cual sea el caso, la instalación de miniroot envuelve ciertos pasos que se deben cubrir y sobre todo entender, para saber dónde se está y conocer lo que se puede hacer. La

[3] En esta tesis se tratará únicamente el método de instalación local. Para obtener información sobre los pasos para configurar los equipos y realizar una instalación remota, consultar el manual: Software Installation Administrator's Guide, Capítulo 2.

funcionalidad de este método recae sobre una serie de servicios que se encuentran grabados en un chip dentro del equipo, llamado PROM; los cuales se encargan de transferir herramientas de instalación de un medio de distribución (local o remoto), al disco del equipo y ponerlos en ejecución. A este proceso se le conoce como *el cargado de miniroot*. Por la importancia que merece, este tipo de instalación será descrito detalladamente a continuación.

I.II.I. Instalación con miniroot

Para entender el proceso de instalación de miniroot, primero daremos un vistazo al proceso en su conjunto, posteriormente se describirán algunas aplicaciones que se utilizan y finalmente se detallarán los pasos a seguir para llegar a una instalación adecuada, tomando como ejemplo, al SO.

Como ya se mencionó, los medios de distribución contienen el software que se instalará además de utilerías de instalación, entre ellas se encuentra miniroot, y durante el proceso de instalación, éste es copiado del medio de distribución al disco del equipo; en particular a la partición 1, que corresponde al área reservada para swap "intercambio". Este proceso es llevado a cabo por un programa llamado *sash* (stand alone shell "shell solitario"), que se encuentra dentro de estas utilerías también.

Para poder accesar la utilería de *sash* y ésta pueda cargar a miniroot, se utiliza el monitor PROM, que como ya se mencionó, se encuentra grabado en los circuitos de todas las máquinas. Este programa ofrece varios menús que permiten realizar varias labores sobre el sistema y es fundamental para llevar a cabo cualquier operación en el equipo; tanto instalación del sistema como el cargado del mismo.

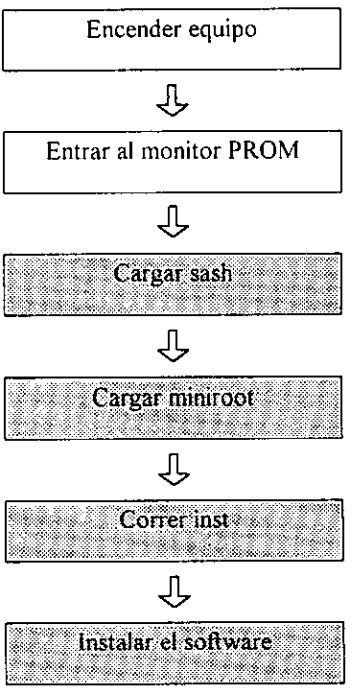


Fig. I-11 Secuencia de Instalación miniroot

Una vez cargado miniroot, éste ejecuta automáticamente un programa llamado *inst*, que es la utilería especial para realizar instalaciones de software de la compañía SGI; es decir, todos los productos de esta compañía vienen grabados en este formato. La Fig. I-11 muestra la secuencia que se sigue para poder instalar el sistema operativo.

De esta figura se puede destacar la diferencia de una instalación local a la remota; la cual depende de dónde sean leídas las utilerías y el software (procesos sombreados).

I.II.I.I. PROM

El monitor PROM[4] es un programa que reside permanentemente en la memoria programable de sólo lectura que se encuentra dentro de la máquina; es decir, que en la fábrica es grabado en un chip y éste es colocado físicamente dentro del equipo, de tal forma que no es posible borrarlo o modificarlo. Esto ocasiona que cada máquina tenga su propio PROM y éste pueda variar de equipo en equipo, mostrando diferentes interfaces u opciones de menús, ya que a medida que avanza la tecnología se crean nuevas versiones que optimizan las características del hardware.

El PROM controla el proceso de arranque y es ejecutado automáticamente al encender la computadora. Su operación normal es la siguiente: Al encender el equipo empieza a funcionar y lo primero que hace es correr una serie de pruebas para verificar el buen estado de los diversos componentes físicos; después, realiza ciertas labores para inicializar el hardware del equipo, como reconocer los dispositivos SCSI que estén conectados[5], limpiar la memoria, inicializar los dispositivos de video, etc. Si estas pruebas son pasadas sin ningún problema, llama al programa intermediario *sash*, cuya labor es la de cargar del disco a la memoria el SO y pasarle el control del equipo. Por el contrario, si se presenta un error o se interrumpe el proceso normal de encendido, el PROM despliega una serie de opciones que permiten realizar ciertas labores de mantenimiento con el

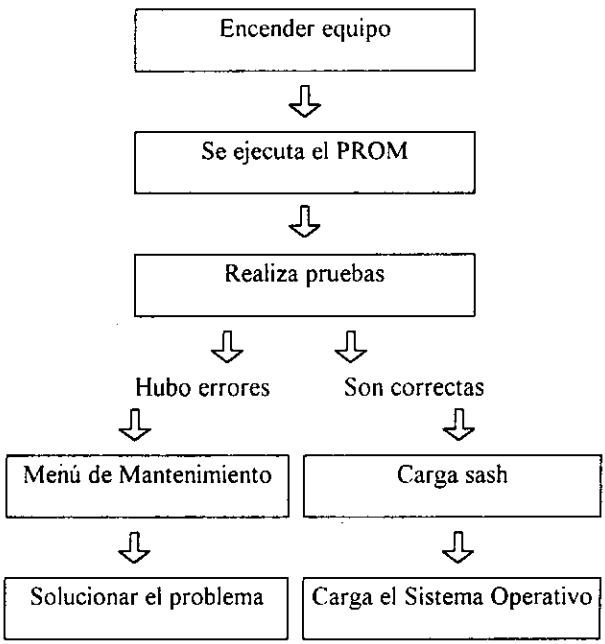


Fig. I-12 Secuencia Normal de Encendido

[4]PROM es un acrónimo de Programmed Read-Only Memory
[5]De ahí la importancia que tiene, el que primero se enciendan todos los dispositivos exteriores y al final el CPU.

propósito de corregirlo, ver Fig. I-12. A este menú se le conoce con el nombre de Menú de Mantenimiento del Sistema o Monitor PROM y es el que se muestra en la Fig. I-13.

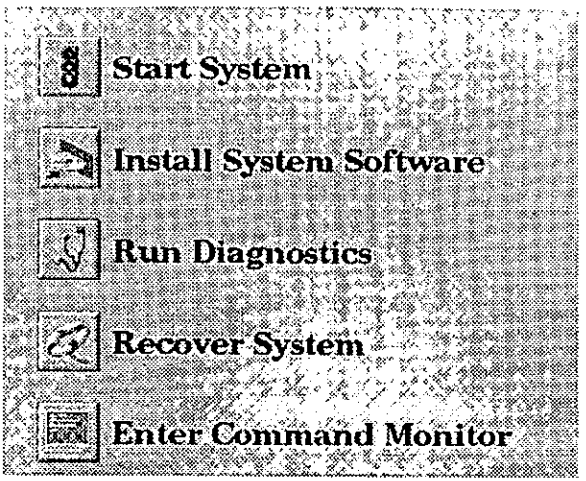


Fig. I-13 Menú de Mantenimiento del Sistema

A continuación se describe la función que desempeña cada una de estas opciones:

Tabla 1 Opciones del menú de mantenimiento del sistema.

Opción	Descripción
Start System	Permite continuar el proceso normal de encendido y se cargue el SO, Fig. I-12.
Install System Software	Permite instalar software en el disco duro del equipo, ya sea de forma local o remota.
Run Diagnostics	Ejecuta una serie de rutinas para revisar y reportar el estado de las partes principales del equipo.
Recovery System	Permite bajar de cinta un respaldo, ya sea del SO o de algún sistema de archivos. Esta opción se utiliza cuando se dañó el sistema y no puede arrancar en forma normal.
Enter Command Monitor	Pone al PROM en modo manual. De esta forma se dispone de comandos que permiten configurar, verificar y realizar labores más complejas.
Select Keyboard Layout	Permite seleccionar la configuración del teclado que estará conectado al equipo.

I.II.I.II. sash

Es un programa de los llamados ***stanalone loader*** (cargadores) cuya principal función es la de cargar el sistema operativo (o miniroot cuando se requiera). Después del PROM, el siguiente programa más inteligente es *sash*. Entiende el formato del sistema de archivos de UNIX; por lo que puede leer información almacenada en ellos, como el kernel de UNIX que es cargado del disco a memoria por esta aplicación.

Se encuentra dentro de las utilerías de instalación en los CD's y, después de instalar el sistema operativo, es colocado en la partición 8 y dentro del directorio */stand* del sistema de archivos que es creado (partición 0), de donde es leído automáticamente por el PROM cada vez que se enciende el equipo.

El nombre de *sash* viene de stand alone shell. Se dice que un programa es *stand alone*, si para ejecutarse no requiere ningún software como base. Por ejemplo, para ver un documento se requiere un procesador de texto, y para que éste funcione, puede requerir un paquete de manejo de ventanas (como Xwindows o Windows) y éste a su vez, requiere de un sistema operativo (UNIX o MS DOS); en cambio, un programa *stand alone* no requiere ningún software para operar. Por otro lado, podemos decir que un *shell* es una aplicación que entiende y en la que se pueden ejecutar comandos.

Los comandos de que dispone *sash* le permiten realizar operaciones directamente sobre el sistema de archivos de UNIX, como el listar y copiar información, e instalar y cargar el SO.

I.II.I.III. Miniroot

Podríamos decir que miniroot es un SO completo como lo es UNIX, pero en pequeño; de ahí el nombre de mini - root.

Miniroot contiene un kernel[6] de IRIX, un sistema de archivos, programas y utilerías. Por default, las herramientas que contiene permiten operaciones de instalación, como *inst*, y la interfaz de operación que utiliza es en modo terminal, ASCII; no importando si se tiene un monitor gráfico.

Durante el proceso de instalación, miniroot es copiado de un medio de distribución a la partición 1 (área de swap) del disco principal. Ahí se construye el sistema de archivos y se carga el kernel de IRIX a la memoria, el cual empieza a funcionar. Si se seleccionó la opción de INSTALL SYSTEM SOFTWARE del menú de root, automáticamente se ejecuta la utilería de *inst* que permite llevar a cabo la instalación.

[6]Es el núcleo del sistema operativo. Físicamente es un archivo que se encuentra grabado en el disco principal, y al encender el equipo, es cargado a al memoria.

Si se elige salir de *inst* aparecerá el prompt de miniroot y en él se puede ejecutar cualquiera de sus comandos para trabajar. Cabe destacar que miniroot es como un sistema UNIX pero en pequeño, y contiene únicamente comandos básicos. Se pueden utilizar los comandos *ls* y *cd* para explorar la estructura del sistema de archivos y ver las utilerías y herramientas con que cuenta. Por otra parte, si se apaga el equipo toda la estructura de este SA se pierde, y como se utiliza el área de swap para montarla, no se puede recuperar.

Una vez que se ha cargado miniroot y su sistema de archivos es creado, es montado en el directorio */root* la partición 0 del disco principal, que corresponde a root del SO IRIX propiamente dicho, quedando como se ilustra en la Fig. I-14. Esto es de utilidad cuando se presentan fallas en el sistema, ya que se puede accesar y corregir cualquier problema. Se debe tener muy en cuenta esta situación, en especial cuando se traten de bajar respaldos de cinta mediante miniroot, ya que / de SO IRIX corresponde a */root* de miniroot.

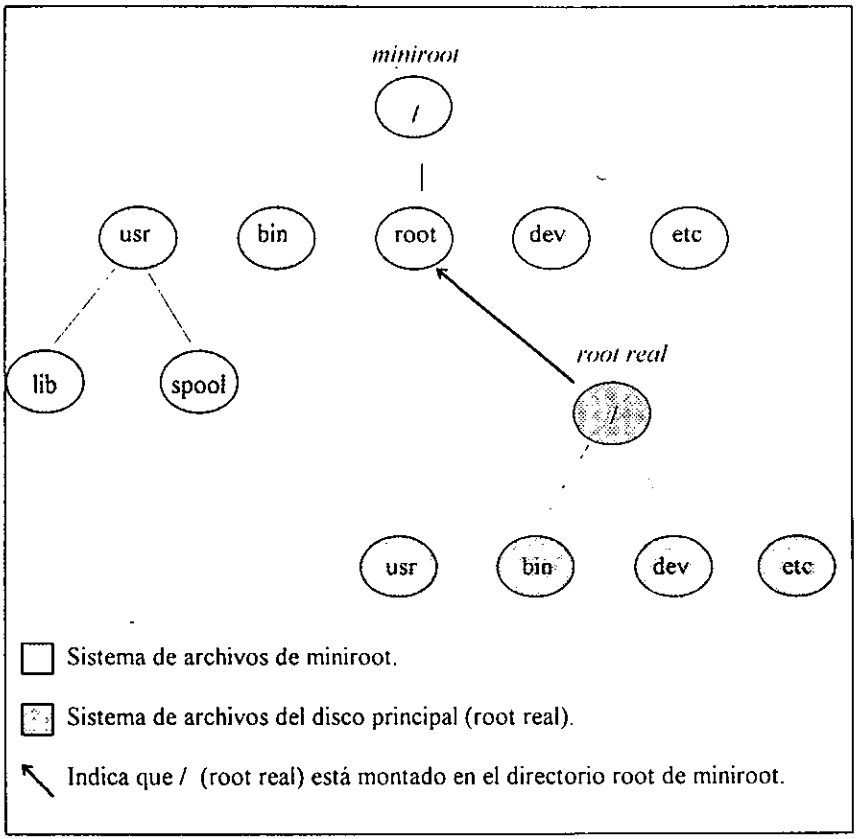


Fig. I-14 Estructura del sistema de archivos de miniroot

I.II.I.IV. inst

Como ya se mencionó, la utilería principal para instalar aplicaciones en IRIX es *inst*, que es ejecutada automáticamente después que se carga miniroot. Ésta presenta un menú que permite controlar la forma y los elementos que se instalarán; pero, para entender su funcionamiento se tiene que comprender la forma como se distribuye la información.

Un medio de distribución puede contener varios productos, que son una colección de archivos que soportan una aplicación especifica, y se distinguen mediante un nombre; como podría ser el compilador de Fortran, cuyo nombre es *fm_dev*. Cada producto puede contener varias imágenes, que son una serie de archivos que desempeñan una función dentro de él y se distinguen agregando una extensión a su nombre; por ejemplo, los programas que permiten compilar en Fortran se llaman *fm_dev.sw*, y la documentación del compilador lleva el nombre de *fm_dev.man*. Finalmente, cada imagen está compuesta por varios subsistemas, que es la unidad de información más pequeña que se puede instalar, y se identifican agregándole una extensión al nombre de la imagen; por ejemplo, *fm_dev.sw.fm* son los ejecutables del compilador y *fm_dev.sw.utils* sus utilerías.

Por otro lado, los subsistemas pueden venir marcados como default, requeridos o sin marca. Los marcados como default soportan las funciones básicas del producto y son los que el proveedor sugiere se instalen. Algunos de ellos, también son marcados como requeridos; ya que son críticos para su operación, y por esta razón, *inst* previene que se realice la instalación si alguno de ellos no es seleccionado. Finalmente los que no se encuentran marcados son aplicaciones que acompañan al producto y que pueden expandir sus capacidades, mas no son indispensables para su funcionamiento.

Al momento de cargar *inst* y una vez que realiza un análisis de los productos que se encuentran tanto en el medio de distribución (que se instalarán) como los ubicados en el disco (ya instalados) detecta cuáles son nuevos, cuáles son actualizaciones, cuáles son parches, etc. y coloca la marca **i** en aquéllos que deben instalarse (generalmente son los marcados como requeridos y default). De esta forma, se puede revisar cuáles fueron marcados para instalarse, y si se requiere, se pueden agregar o eliminar algunos.

Como recomendación, si no se sabe qué función realiza cada uno, instalar únicamente los que están marcados por default, trabajar con el paquete y después de adquirir experiencia o si el paquete o manuales lo indican, instalar los módulos adicionales.

Antes de llevar a cabo la instalacion se tiene que revisar si no existen conflictos que puedan ocurrir; para ello, *inst* cuenta con una opción la cual indica si los hay, y en tal caso, da una serie de opciones que pueden solucionarlo. Si no existen conflictos se puede realizar la instalación, la cual consta de dos fases que se efectúan al seleccionar la opción 3 (ver Fig. I-15) y que son la de bajar los archivos del medio al disco y la de efectuar una serie de tareas que permitan integrar la aplicación al sistema. Al final se despliega un mensaje indicando si se realizó correctamente. La pantalla principal de *inst* es la siguiente:

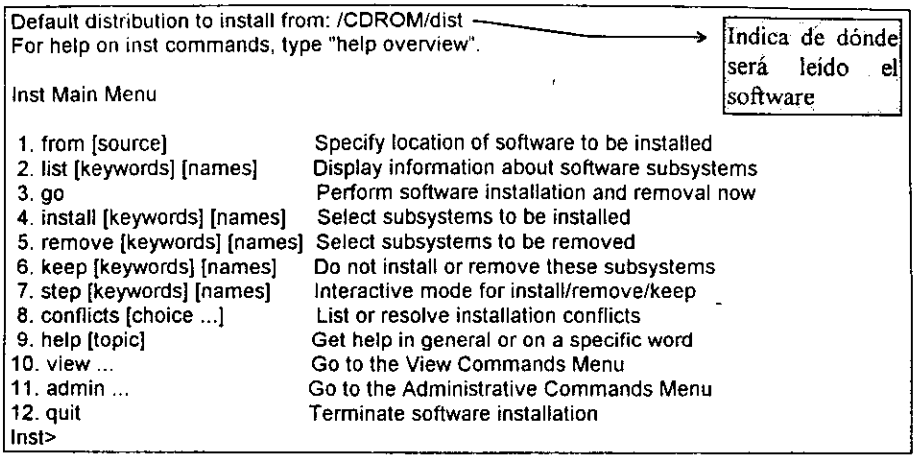


Fig. I-15 Menú principal de inst.

La opción 1 permite cambiar la selección del dispositivo que contiene el medio de distribución. En el caso de realizar la instalación mediante miniroot, éste es seleccionado automáticamente desde el PROM.

La opción 2, lista en la pantalla el contenido del medio de distribución; esto es, cada uno de los subsistemas que contenga. Además cada uno de ellos aparece con las marcas que se le asignaron, Fig. I-16.

La opción 3, es la que permite efectuar realmente la instalación. Cualquier operación que se realice antes, no es llevada a cabo hasta que se selecciona esta opción.

La opción 4, permite marcar cualquier subsistema con la **i** para indicar que es un elemento seleccionado para instalarse. La instalación no se lleva a cabo hasta que se indique mediante la opción 2, aquí únicamente es marcado.

La opción 5 elimina la marca **i** del subsistema, de tal forma que el elemento no sea instalado cuando se lleve a cabo.

La 6 opción le indica que mantenga al subsistema con las marcas actuales.

La opción 7 es muy útil, ya que muestra uno a uno los subsistemas y se puede marcar en forma individual cada uno de ellos; presionando la **i** para que sea marcado como instalable, la **r** para remover la marca y no se instale, o la **k** para que conserve sus valores y se muestre el siguiente subsistema. En cambio, las opciones 4, 5 ó 6 requieren que se especifique el nombre del subsistema correctamente.

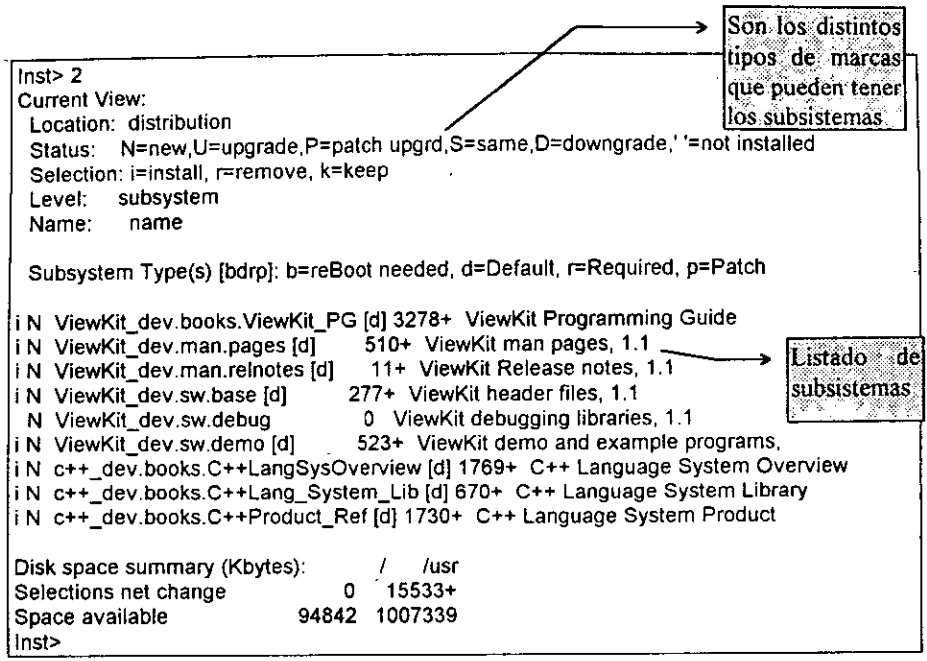


Fig. I-16 Listado de subsistemas

La opción 8 muestra si existe algún conflicto que impida instalar los subsistemas marcado así como las posibles soluciones. El conflicto más común es que sea seleccionado un módulo, el cual requiera para su funcionamiento otro que no está marcado para instalarse, o que no se encuentre en el medio de distribución. No se llevará a cabo la instalación hasta que sean solucionados todos los conflictos.

La opción 10 permite especificar qué es lo que se quiere ver (con la opción de **list**), si el contenido del medio de distribución o lo que se encuentra instalado en el disco, así como los filtros que se utilizarán para desplegar la información.

La opción 11 muestra un submenú que cuenta con opciones que permiten establecer parámetros de instalación o utilizar herramientas para poder llevarla a cabo. Desde este submenú se puede establecer el nombre y dirección IP que tendrá el equipo; crear, montar o desmontar sistema de archivos; listar las características del equipo; accesar un shell, etc.

Finalmente la opción 12 se utiliza para salir del programa y regresar a miniroot, si fue ejecutado desde el PROM, o al SO si se ejecutó desde ahí.

Es importante decir que se puede ejecutar *inst* en cualquier momento para instalar subsistemas que se vayan requiriendo, y no saturar el disco con información que no se

utilizará. Por ejemplo, el SO y algunos paquetes ofrecen alternativas para presentar información en determinados idiomas (español, inglés, etc.) y es conveniente seleccionar únicamente el que se utiliza en esa región y no todos los que hay, ya que nunca se utilizarán. Por último, la herramienta de *inst* que se utiliza en el proceso de instalación desde miniroot, funciona exactamente igual que el comando *inst* que se emplea para una instalación en vivo.

I.II.I.V. Instalación del Sistema Operativo

Este proceso consiste en colocar en el disco duro del equipo, un conjunto de programas de arranque y el mismo SO; de tal forma que al encender la computadora, pueda arrancar adecuadamente y quedar en un estado operable.

Generalmente cuando se compra una estación de trabajo, ésta viene con el software instalado; por lo menos con el SO. De tal forma que, una vez que se realizó la instalación física, se puede encender y empezar a trabajar con ella. Si por cualquier razón el SO no viene instalado, éste es el primer paso que se debe dar para poder trabajar con el equipo.

Por la importancia que merece y como ejemplo de una instalación local y desde miniroot, a continuación se describirán los pasos para efectuar la instalación del SO IRIX; el procedimiento es el mismo para cualquier otro software.

Procedimiento de instalación desde miniroot

1) Realizar la instalación física del equipo tomando en cuenta lo explicado en el punto I.I Instalación del hardware.
2) Checar las conexiones de los dispositivos al CPU y la de alimentación de energía, para asegurarse que estén correctas.
3) Encender el equipo. Ver Secuencia de Encendido, pág. I-9 y pág. I-10.
4) Cuando aparezca la Fig. I-7, dar un click donde dice *Stop for Maintenance* para entrar al menú del PROM, Fig. I-13.
5) Seleccionar la opción *Install System Software*.
6) Al aparecer la Fig. I-17 seleccionar el dispositivo local CD o Cinta y presionar el botón de *Accept*.
7) Al aparecer la ventana indicando que sea insertado el CD o cinta en la unidad, hacerlo y presionar el botón *Continue*. Cerciorarse que sea el CD que contiene al SO.

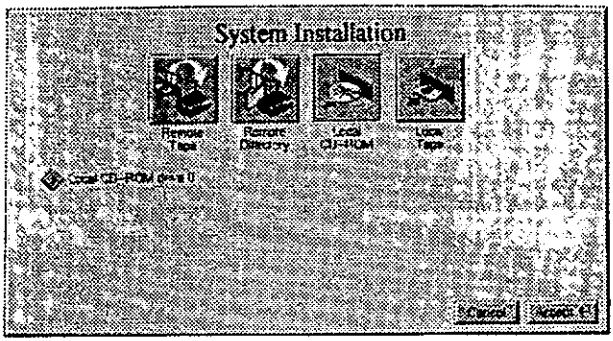


Fig. I-17 Menú de instalación de software

8) En este momento aparecerá una ventana indicando que se tratan de obtener las utilerías de instalación (Obtainin installation tools). Es aquí cuando el PROM llama a *sash* y éste a su vez carga a miniroot, que se encuentra en el medio de distribución, en el área de swap del disco. Para que este procedimiento funcione, el disco principal debe estar formateado y particionado adecuadamente. Si el disco es nuevo y sin formato, aparecerá un mensaje de error; presionar <ENTER> para regresar al menú del PROM y solucionarlo.

- El problema consiste en que el disco no tiene formato y no se puede obtener la información referente a la ubicación de la partición de swap (1), donde se cargará miniroot; la de la partición del sistema (0), donde se instalará el SO etc. que se encuentra en la partición 8 del disco principal. Para solucionarlo se tiene que dar formato y particionar el disco. Los pasos son los siguientes: cargar a *sash* y desde él, cargar la utilería para dar formato a discos, *fx*. Pero, en el mercado existen muchas versiones de CPU que han surgido a través del tiempo y el funcionamiento de cada uno puede variar; por lo que existe una versión de *sash* y de *fx* para cada tipo de CPU. Para determinar cual versión es la que requerimos:

a) Al aparecer el menú del PROM seleccionar la opción *Enter Command Monitor*.

b) Al parecer el prompt del monitor PROM (>>) ejecutar el comando siguiente para determinar los parámetros requeridos:

```
>> hinv
                System:  IP22
             Processor:  133 Mhz R4600, with FPU
Primary I-cache size:  16 Kbytes
Primary D-cache size:  16 Khytes
           Memory size:  32 Mbytes
              Graphics:  Indy 8-bit
             SCSI Disk:  scsi(0)disk(1)
             SCSI Disk:  Controller 0, ID 2, removable media
             SCSI Disk:  scsi(0)disk(3)
             SCSI Disk:  scsi(0)disk(5)
                 Audio:  Iris Audio Processor: version A2 revision 4.1.0
```

c) Buscar el tipo de CPU en la línea que dice *System*; éste empieza con las letras IP. En este caso es el *IP22*.

d) Buscar la dirección del disco principal en una de las líneas que dicen *SCSI Disk*. Como nota general, el definido de fábrica suele ser un disco interno con Identificador[7] 1 y colocado en la tarjeta controladora SCSI 0 (*scsi(0)disk(1)*). Si se tienen varios discos y se piensa utilizar otro como el disco principal, determinar cual es su ID y en qué controladora está colocado. En este ejemplo se tienen 4 dispositivos conectados a la tarjeta SCSI 0; el Primero es un disco con ID 1; el segundo tiene un ID 2 y se trata de una unidad con medio removible, puede ser una unidad de floppy óptico o normal; la tercera es un disco con ID 3 y el cuarto con ID 5, suele ser utilizado para unidades CDROM.

e) Buscar la dirección del dispositivo que contiene el medio de distribución con el software a instalar (CDROM) en las líneas que dicen *SCSI Disk*. En este caso, tiene el ID 5 y está en la controladora 0.

f) Una vez obtenidos los datos anteriores, se procederá a cargar el *sash*; para ello ejecutar el comando:

```
>>boot -f dksc(0,5,8)sash.ARCS

Standalone Shell SGI Version 5.3 ARCS Oct 18, 1994 (BE)
sash:
```

La instrucción *boot -f* indica que se desea cargar un programa. La cadena *dksc*, indica que se cargará de un dispositivo SCSI y *0,5,8*, el controlador, identificador del dispositivo y la partición respectivamente de donde se leerá el programa. Aquí se deben colocar los datos del CDROM obtenidos en los pasos anteriores. Finalmente, el nombre del programa a cargar varia dependiendo del CPU y se forma de la siguiente manera:

sash.CPU

Los CPU IP17 o anteriores son conocidos como pre-ARCS y se debe sustituir la palabra CPU por su nombre; esto es, sash.IP4, sash.IP5, sash.IP17, etc. Los procesadores mayores son conocidos como ARCS[8] y el *sash* utilizado es el mismo para todos ellos. Para ellos, se debe sustituir la palabra CPU por ARCS en la instrucción anterior, sash.ARCS.

g) Al aparecer el prompt de *sash*, ejecutar el comando para cargar la aplicación de *fx* y poder formatear el disco:

sash:**boot -f dksc(0,5,7)stand/fx.ARCS**.

[7]El Identificador (ID) es la dirección SCSI que se le da al dispositivo.
[8]Advanced RISC Computing Standard. Las máquinas producidas después de 1992, generalmente tiene un CPU de este tipo.

De forma análoga al comando anterior, *dksc* indica que se trata de un dispositivo SCSI; *0,5,7* indican el controlador, identificador y partición de donde se leerá el programa y *stand/fx.ARCS*, el nombre del programa a leer. Cabe mencionar que *fx* se encuentra en la partición 7 de los CDROM de instalación y dentro del directorio *stand*. Además, su nombre es *fx.CPU*, donde CPU se debe sustituir por el tipo de CPU utilizado.

h) A partir de este momento empieza a funcionar *fx*. Aquí se describirán únicamente los pasos necesarios para particionar el disco y poder utilizarlo; si se desea más información, se puede consultar la ayuda del comando una vez que se encuentre instalado el sistema mediante la instrucción: *$man fx*

En la primera parte de este proceso (ver pág. siguiente), *fx* pide confirmar si se desea trabajar en el modo seguro o experto. En el modo seguro no se alteran los datos, aunque está activada la opción para particionar, la cual puede destruir la información almacenada en él. El modo experto dispone de más menús que permiten dar formato al disco, así como alterar la información y parámetros del disco. Ya que no se requiere el modo experto, presionar <ENTER> .

En la segunda parte *fx* pide especificar con qué disco se desea trabajar, para ello proporcionar los datos del dispositivo (dksc), controlador (0), e identificador (0) del disco principal. En este caso se presiona <ENTER> para aceptar los parámetros default, pero si no coinciden proporcionar los correctos.

En la tercera parte *fx* trata de localizar la información del disco; pero ya que no está formateado, marca algunos errores y posteriormente lo particiona y guarda en él toda la información necesaria para poder arrancar[9]. Después de esto, aparecerá su menú.

En este punto el disco ya puede ser utilizado sin ningún problema ya que se ha particionado automáticamente; pero, la partición que genera puede que no sea la adecuada. Para evitar problemas, si no se sabe manejar este programa, se debe particionar el disco indicándole que se desea utilizar como disco principal. Para ello, entrar al menú de repartición presionando la letra ***r***, y ahí, teclear ***ro*** (rootdrive) para indicar que será un disco principal. Después de teclear ***yes*** al mensaje de advertencia, el disco es particionado.

Finalmente en el último paso se teclea .. *(dos puntos)* para regresar al menú principal y ahí ***exit*** para salir de *fx* y regresar al menú principal del PROM.

En este momento el disco se encuentra listo para poder instalar el SO en forma adecuada, para ello repetir el procedimiento a partir del punto 5.

[9] Ver capítulo 3 para más información.

1

Currently in safe read-only mode.
Do you requiere extended mode with all option available? (no). ***<ENTER>***

2

SGI Version 5.3 ARCS Oct 18, 1994
fx: "device-name" = (dksc) ***<ENTER>***
fx: ctlr# = (0) ***<ENTER>***
fx: drive# = (1) ***<ENTER>***

3

...opening dksc(0,1,)
dks0d1s10: Volume Header not valid.
...controler test... Sc0,1,0: cmd=0x1d time out after 20 sec. resetting SCSI bus.

OK
fx: Warning: invalid label or disk, ignored
scsi drive type== SEAGATE ST12400N 8650
fx:Warning: can't read sgilable on disk.
Creating new sgilabel
... creating default bootinfo
... creating default partition
... creating default volume directory.

4

----- please choose one (? for help, .. to quit this menu)-----
[exi]t [d]ebug/ [l]abel/
[b]adblock/ [exe]rcise/ [r]epartition/
fx> ***r***
fx/repartition> ***ro***

Warning: you will need to re-install all software and restore user data
from backups after changing the partition layout. Changing partitions
will cause all data on the drive to be lost. Be sure you have the drive
backed up if it contains any user data. Continue?
please enter a yes or no
Warning: you will need to re-install all software and restore user data
from backups after changing the partition layout. Changing partitions
will cause all data on the drive to be lost. Be sure you have the drive
backed up if it contains any user data. Continue? ***yes***

----- partitions-----

part	type	cyls	blocks	Megabytes (base+size)
0:	efs	3 + 2594	4752 + 4108896	2 + 2006
1:	raw	2597 + 51	4113648 + 80784	2009 + 39
8:	volhdr	0 + 3	0 + 4752	0 + 2
10:	volume	0 + 2648	0 + 4194432	0 + 2048

capacity is 4194685 blocks
----- please choose one (? for help, .. to quit this menu)-----
[ro]otdrive [u]srrootdrive [o]ptiondrive [re]size
fx/repartition> ..

5

----- please choose one (? for help, .. to quit this menu)-----
[exi]t [d]ebug/ [l]abel/
[b]adblock/ [exe]rcise/ [r]epartition/
fx> ***exit***

9. Después de obtener las utilerías de instalación aparece un mensaje indicando que se están copiando las herramientas al disco (*Copying installation tools to disk*). Posteriormente, se empieza a crear la estructura de miniroot y al final se monta el sistema de archivos del disco principal. Si el disco es nuevo y se siguió el procedimiento para formatearlo y particionarlo, no contendrá ningún sistema de archivos; por lo que miniroot marcará un error y le creará uno. Si esto sucede, responder con *yes* para aceptar la creación del sistema de archivos y con la letra *y* para confirmar. Una vez terminado este proceso, se ejecutará la utilería de instalación apareciendo su prompt, como se muestra en el siguiente listado:

```
Creating miniroot devices
Corre system date is Oct 18, 1994
Mounting file system:

/dev/dsk/dks0d1s0: Invalid argument
No valid file system found on: /dev/dsk/dks0d1s0
This is your system disk: without it we have nothing
on which to install software.

Make new file system on /dev/dsk/dks0d1s0 [yes/no/sh/help]: yes

About to remark (mkfs) file system on /dev/dsk/dks0d1s0
This will destroy all data on disk partition
        Are you sure? y
Doing mkfs /dev/dsk/dks0d1s0

Trying again to mount /dev/dsk/dks0d1s0

/dev/miniroot on /
/dev/dsk/dks0d1s0 on /root
Invoking software installation
default distribution

Inst>
```

Aparece sólo si el disco es nuevo y no tiene un sistema de archivos.

10. Al entrar a *inst*[10], automáticamente se seleccionan los principales módulos, que son los que el distribuidor recomienda se instalen, por lo que se puede ejecutar el comando para instalarlo, ***go***.

```
Inst> go
```

Al teclear este comando se empieza a instalar el SO; este proceso puede tardar varios minutos u horas, dependiendo de la velocidad de los dispositivos. Al final indicará que se realizó la instalación en forma correcta y aparecerá el prompt de *inst* nuevamente.

[10]Ver pág. I-18. Para mayor información consultar la ayuda del comando una vez instalado el SO con la instrucción $man inst.

11. Para salir de *inst* y concluir la instalación ejecutar el comando ***quit***.

    ```
    Inst> quit
    ```

    Después de ejecutarlo, se realizarán una serie de labores para configurar e integrar todo el software instalado, este proceso puede tardar varios minutos.

12. Por último, aparecerá un mensaje indicando que se restablecerá el sistema. Responder con ***yes***.

    ```
    ready to restart the system. Restart.
      [yes, no, shell, help] yes
    ```

    El equipo se restablecerá y se efectuará el proceso normal de encendido, ver "Proceso de encendido" en pág. I-10. Con esto, queda concluido el proceso.

Una vez que ha sido instalado, la mayoría del software requiere que se realicen una serie de operaciones que permitan adecuarlo a las necesidades propias; las cuales son conocidas como "labores post-instalatorias". Dependiendo del software, éstas pueden ser algo extensas, como en el caso del SO, requerir establecer únicamente variables de ambiente o ninguna acción posterior; la documentación que acompaña a cada una especifica cuales son sus requerimientos en particular. Para el caso que nos ocupa, SO, a continuación se describen las más importantes y necesarias.

Labores post-instalatorias

Las labores post-instalatorias mínimas que se deben efectuar sobre el SO son:

- Asignarle una clave secreta a root y otras cuentas.
- Asignarle un nombre al equipo.
- Si se encuentra conectado a una red, requerirá una dirección IP así como especificarle cual será su dominio, máscara y el gateway que utilizará.
- Si se utiliza un servidor de nombres por dominio, indicarle cuál es; de caso contrario, crear una lista de los equipos principales con los cuales tendrá una mayor comunicación.
- Establecer la zona horaria del lugar en la que se encuentra instalado, así como la fecha y hora actuales, entre otras.

La mayoría de estas labores pueden ser realizadas mediante la cuenta llamada EZsetup, cuya función es la de correr una aplicación gráfica que permite, de una manera sencilla, establecerlas. Al ser ejecutada por primera vez, detecta que la cuenta de root no tiene una clave secreta asignada, por lo cual solicita una; para su elección es conveniente tener en mente las recomendaciones indicadas en "**Consideraciones al elegir una clave secreta**",

pág. IV-13. Recordar que al momento de teclearla, no es desplegada en la pantalla por razones de seguridad; por ello se pide sea introducida dos veces, evitando cualquier posible error. Posteriormente aparece la pantalla de configuración, en la cual se deben seguir los siguientes pasos:

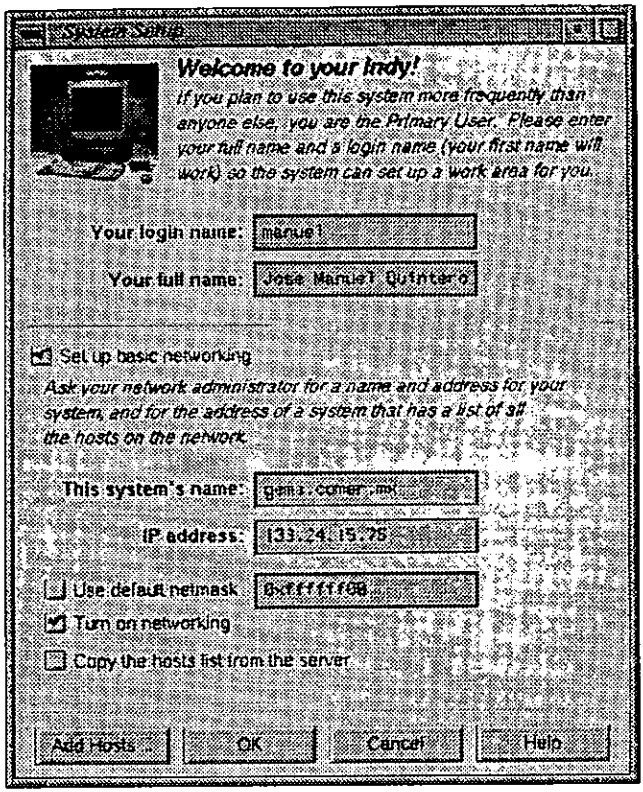
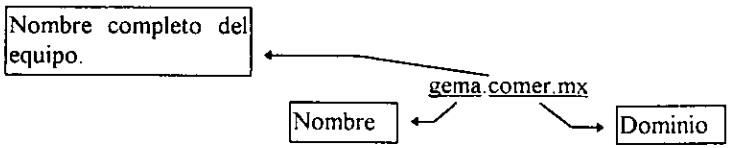


Fig. I-18 Ventana de configuración inicial.

- Por seguridad, es conveniente que cuando el administrador del equipo desee trabajar en él, lo haga con una cuenta que no sea la de root. El acceso a la cuenta de root sólo se debe efectuar cuando las labores que se efectuarán, así lo requieran; por ello, en el campo de "*Your login name*" debe ser colocado el nombre de una cuenta que será creada y que tendrá privilegios para poder ejecutar las herramientas gráficas de administración. Al dueño de esta cuenta se le denomina usuario primario y supuestamente, es la persona que utilizará el equipo con mayor frecuencia.
- En el campo de "*Your full name*" debe ser colocado el nombre real del dueño de la cuenta.

- Por default cuando es instalado el SO, le es asignado el nombre ***IRIS*** y la dirección IP de ***192.0.2.1***, lo cual no representa ningún problema de estar trabajando en forma aislada; pero si operará sobre una red, puede causar algún conflicto, ya que no pueden existir dos equipos con el mismo nombre y dirección en la red. Para poder modificar el valor inicial de estos y otros parámetros que atañen a los servicios de red, hay que seleccionar el cuadro que dice "*Setup basic networking*".
- En el campo de "*This system's name*" debe ser colocado el nombre que le será asignado al equipo. Cabe mencionar que si el equipo pertenece a un dominio dentro de la red, el nombre debe comprender el del dominio y el del equipo:



- En el campo de "*IP address*" debe ser colocada la dirección IP que le ha sido asignada.
- Si en la red se está utilizando alguna máscara que no sea la default, se debe dar un click en el cuadro "*Use default netmask*" para desactivarlo y poder especificar la adecuada. La máscara nos permite alterar la interpretación normal que se le da a una dirección IP, para poder crear subredes. Todos los equipos dentro de una subred, deben conocer cual es la máscara que se está utilizando, para poder identificarse como miembros de ella y lograr comunicarse. Su valor se expresa en notación hexadecimal, por ejemplo, para indicar una máscara de 255.255.255.0 se debe escribir en este campo el valor de: 0xffffff00.
- El cuadro de "*Trun on networking*" debe estar seleccionado para que se activen todos los servicios de red.
- El botón de "*Add Host*" permite añadir a una pequeña base de datos conocida como "tabla de hosts" representada por el archivo */etc/hosts*, el nombre y dirección IP de los equipos que se utilicen con mayor frecuencia, para poder utilizar su nombre en lugar de la dirección IP en los diversos comandos y servicios de red.
- Finalmente, al presionar el botón "*OK*" se guardan los cambios, y para que tengan efecto, el sistema se debe reinicializar; por ello se debe presionar el botón "*OK*" cuando aparezca el mensaje indicando si se desea reiniciar el equipo.

Esta herramienta establece los parámetros más importantes, pero aún existen algunos que deben ser colocados de forma manual con la ayuda de comandos de IRIX. Para ello, se debe tener acceso a una ventana de terminal, una vez que se ha ingresado al sistema.

Al seleccionar la opción "*Desktop*" del menú "*Toolchests*", y posteriormente la de "*Unix Shell*", se activa una ventana de terminal. El menú *Toolchests* es una pequeña ventana que

generalmente se encuentra ubicada en la parte superior izquierda de la pantalla y que posee un menú que permite tener acceso a las herramientas principales. Si se tienen dudas en el manejo del ambiente gráfico, es conveniente ejecutar el programa tutorial llamado "*Systemtour*", representado por un icono, que enseña el uso de sus principales elementos.

Entre los parámetros restantes de configuración se encuentran:

- Por las razones expuestas en "La cuenta o login", pág. IV-8, es necesario bloquear o asignarle una clave secreta a cualquier cuenta que carezca de una; ya que como se indica ahí, durante la instalación del SO se crean algunas que carecen de la clave secreta, originando agujeros en la seguridad del sistema. En ese mismo punto se especifica cual es la forma de determinar si alguna cuenta, registradas en el archivo */etc/passwd*, posee o no una clave secreta; ahí mismo se indica cual es el comando empleado para asignarle una o bloquearla.
- Si se emplea un servidor de nombres por dominio (DNS) en la red donde está conectado el equipo, es necesario indicarle al sistema cual es el dominio al cual pertenece, así como la dirección IP del servidor de nombres; para ello:
    - Crear el archivo */etc/resolv.conf*; debe pertenecer a root y contar con los permisos de 644[11].
    - Editarlo y colocar en él, lo siguiente:

        hostresorder local bind
        domain comer.mx
        nameserver 123.218.1.2
        nameserver 142.28.14.3
        nameserver 192.11.63.10

    - La primera línea (*hostresorder*) indica cual será el orden en que se resolverán los nombres; esto es, determinar la dirección IP a partir del nombre del equipo. Ahí se especifica que primero sea resuelto localmente (*local*) y de fallar este método, se empleen los servidores de nombre (*bind*). El método local consiste en utilizar la tabla de hosts del equipo, que de resolver el nombre, permite reducir tiempo y problemas al no tener que establecer una comunicación remota con el servidor de nombres. Este orden de resolución es el más recomendable, y para que cumpla una buena función, la tabla de hosts debe contener los nombres de los equipos con los cuales se establezca una comunicación frecuente, a mas de conservarla lo más pequeña posible.
    - La segunda línea (*domain*) especifica cual es el dominio en el que se encuentra el equipo; en este caso es *comer.mx*

[11] Ver "Permisos de archivos", pág. IV-30.

- De la tercera a la quinta se establece la dirección de tres servidores de nombres que serán utilizados en la resolución de nombres. Si el primero no está disponible, se utiliza el segundo; y si el segundo no lo está, el tercero.

- Si la red local forma parte de una red más extensa, se le debe indicar al equipo cual es la dirección y nombre del gateway que permitirá la salida de información hacia el exterior de la red. Para ello:

  - El archivo encargado de dar de alta los servicios de red es */etc/init.d/network*[12]. Básicamente está compuesto por una estructura *case* con dos alternativas: "*start*" para activar los servicios y "*stop*" para detenerlos. Ya que lo que se pretende es definir un gateway, éste debe ser dado de alta dentro del bloque de instrucciones *start*, por ello, hay que analizarlo. Este bloque está compuesto por una primera sección algo extensa que se encarga de configurar adecuadamente las diversas tarjetas de red conectadas al equipo y la tabla de ruteo, posteriormente una segunda sección encabezada por la instrucción:

    SECHO "Network daemons:\c"

    se encarga de dar de alta cada uno de los servicios de red activados en el sistema. Para que no exista ningún conflicto, la instrucción para dar de alta el gateway debe ser insertada entre las dos secciones; es decir, después que se configuran las tarjetas y tablas de ruteo, y antes de que se den de alta los servicios de red.

  - La instrucción para dar de alta el gateway es:

    ```
    if test $netstate = "ok" ; then
        /usr/etc/route add default 133.24.15.254 1
    else
        echo " Problemas al definir el Gateway default."
    fi
    ```

    El comando *route* con la opción *add*, añade como gateway default al equipo cuya dirección IP es *133.24.15.254*. Ésta es colocada dentro de una estructura *if* que la ejecuta siempre y cuando se haya dado de alta correctamente la tarjeta de red.

  - Debe ser añadida la dirección IP y nombre del gateway, tanto a la tabla de hosts (*/etc/hosts*) como al servidor de nombres, si es que se utiliza uno; ejemplo:

    133.24.15.254 gateway1.comer.mx

[12] Ver "Configuración", pág. II-16

- Para que el cambio entre en operación, se debe reiniciar el equipo o dar de baja y alta nuevamente el servicio:

  # reboot

  ó

  # /etc/init.d/network stop
  # /etc/init.d/network start

- El establecer de manera adecuada la zona horaria del lugar donde se encuentra ubicado el equipo, es indispensable para una correcta comunicación entre equipos a través de redes, sobre todo si forman parte de una red mundial; ya que de ser errónea, software que utilice y tome en cuenta retardos de tiempo, puede considerar que cierta información está llegando con una fecha anterior a la de solicitud, o que la hora de respuesta tenga un retraso enorme con respecto a la de solicitud, entre otras cosas.

  Por otra parte, todos los equipos IRIS de SGI poseen un reloj interno que mantiene la fecha. Este reloj trabaja con Tiempo Universal Coordenado (UTC) que es el número de segundos transcurridos desde el 1 de enero de 1970 a las 00:00:00 Horas del Meridiano de Greenwich (GMT). Por ello, el tiempo local es calculado en base a un número de compensación que se encuentra establecido en variables de ambiente del usuario, y que añadido al reloj del sistema, dan la hora y fecha actual. Por todo lo anterior, el establecer la zona horaria es recomendable; para ello:

  - Si no existe, crear el archivo */etc/TIMEZONE*; debe pertenecer a root y tener los permisos de 644.
  - Establecer dentro de él, la variable TZ de la siguiente forma:

    TZ="Mex6Mex"

    Se indica que existe una diferencia de 6 horas con respecto a GMT. Si se emplea el horario de verano, como recientemente se acaba de establecer en México:

    TZ="Mex6HdV5,M4.1.0/2,M10.5.0/2"

    Aquí se especifica que durante el horario normal (identificado por *Mex*) existe una diferencia de 6 horas, y durante el de verano (*HdV*) es de 5. El horario de verano iniciará el domingo de la primera semana del cuarto mes a las dos de la mañana (*4.1.0/2*) y finaliza el domingo de la quinta semana (la última) del décimo mes a las dos de la mañana (*10.5.0/2*); donde cada día de la semana se representa por un número consecutivo, empezando con el domingo que vale cero.

    Este mecanismo permite que el sistema se encargue de retroceder una hora el reloj al iniciar el horario de verano y adelantarlo otra al finalizar, de manera automática.

- Reiniciar el equipo para que tenga efecto el cambio.
- Establecer la fecha y hora actual; por ejemplo, para colocar la fecha de 15:05 hrs. del 23 de octubre de 1997 :

  # date 1023150597

Estas son la principales modificaciones que debe sufrir el sistema. A medida que se den de alta servicios y se instalen más aplicaciones, deben de ser configuradas para poder trabajar adecuadamente.

CAPÍTULO 2

ADMINISTRACIÓN BÁSICA

Conceptos y tareas básicas que permiten inicialmente, administrar eficientemente un equipo.

II. ADMINISTRACIÓN BÁSICA

La administración básica está compuesta por una serie de rutinas que todo administrador debe conocer; son esenciales. En este capítulo se presentarán las bases, los procedimientos y sugerencias que deben tomarse en cuenta para lograr el máximo rendimiento del equipo; no olvidando que esto se logra en base a conocimientos, como en la experiencia, pericia e imaginación que cada administrador posea para resolver los problemas que se presenten.

Lo que no se pretende, es dar un curso sobre los diferentes comandos de UNIX, por lo que es recomendable, si es que se tienen dudas en el manejo de cualquiera de los presentados aquí, consultar la ayuda en línea mediante el comando *man*.

II.I. Encendido y apagado del equipo

Una de las funciones básicas, es el correcto encendido y apagado del equipo; ya que de ello depende la seguridad de la información almacenada en él.

En **Secuencia de Encendido** pág. I-9 del capítulo **I**, se describen los pasos para encender físicamente el equipo, y en **Proceso de encendido** pág. I-10, las ventanas y acciones que se deben realizar para colocar al SO en un estado funcional. Cabe destacar nuevamente, que en el proceso de encendido es vital encender en primer lugar los periféricos, esperar un momento para que queden en estado operable, y finalmente el CPU; esto con el fin, de que pueda ser reconocido cada elemento del hardware sin ningún problema por el sistema. En esta sección se examinará el proceso de encendido desde el punto de vista del software.

II.I.I. Encendido

Como se examinó en el capítulo I, cuando se enciende el equipo el PROM empieza a funcionar, y lo primero que realiza, es una serie de rutinas para probar el buen funcionamiento de cada uno de los componentes principales del equipo. Si estas pruebas son pasadas, el PROM ejecuta a *sash*; que como se mencionó, es el responsable de cargar en memoria RAM al kernel de UNIX que se encuentra ubicado en el archivo */unix* del SA principal.

Finalmente, una vez que el kernel se encuentra en memoria, toma el control y empieza a ejecutar una serie de programas con el objeto de crear el medio ambiente de multiusuario y multiproceso del SO IRIX, tal y como lo conocemos. Terminada esta labor, se dice que el sistema se encuentra en operación y listo para aceptar usuarios; en este momento aparece la ventana de Bienvenida (**Fig. I-9**) pidiendo sea introducida la clave de un usuario.

Este proceso ya fue descrito en el Capítulo I, pero de aquí lo importante es entender lo que realiza el kernel de UNIX cuando es cargado a RAM, hasta el punto en que queda operable; ya que el comprenderlo permitirá resolver fallas que se presenten durante la vida del equipo.

Cuando el kernel entra en operación, corre una serie de programas que establecen la apariencia y características de IRIX; permiten el poder compartir en forma adecuada y controlada, los recursos del sistema por diversos programas y usuarios. Ellos son: *sched*, *vhand*, *bdflush*, *pdflush* e *init*.

El programa *sched* es el que controla y decide quién es el que puede utilizar en determinado momento el CPU; es decir, ya que IRIX es un sistema multiproceso, puede aparentar que están corriendo varios programas en la memoria en un momento dado. Para ello se mantiene una lista de los programas cargados en RAM, y *sched* controla quién puede usar el CPU y quién debe esperar; claro está, que todo esto sucede tan rápido que el usuario no se percata de ello, y para él, todos están corriendo al mismo tiempo (ver Prioridades, pág. II-29). Al programa *sched* suele conocérsele con el nombre de *scheduler*.

El manejador de la memoria virtual es *vhand* (virtual memory handler); controla y permite que los diversos procesos puedan compartir la memoria disponible del equipo. *bdflush* (buffer to disk flush) se encarga de vaciar los datos de RAM a disco cada 2 segundos, y *pdflush* (page to disk plush), de vaciar páginas de programas mapeados en RAM a disco. Todos estos procesos de vaciado de RAM a disco y de disco a RAM suceden continuamente y es lo que permite ejecutar aplicaciones con un tamaño mayor a la memoria física instalada en el equipo.

Finalmente *init*, tiene un identificador de proceso, PID, de 1; es decir, es el primero que ejecuta el SO. De él descienden todos los demás que posteriormente se corren en el sistema, y por ello es considerado como el padre de todos los procesos.

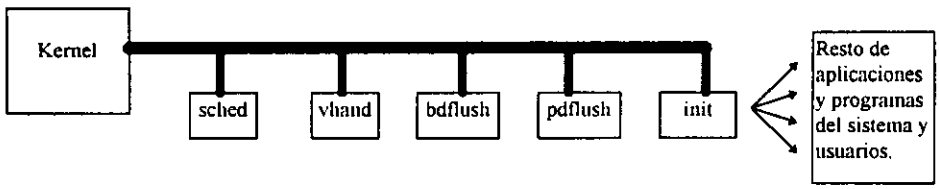


Fig. II-1 Secuencia de programas ejecutados durante el arranque del sistema.

En términos generales, *init* es un productor o generador de procesos, cuya función principal, es la de generar los procesos necesarios para colocar al SO en diversos modos de operación, como son: el de usuario simple, multiusuario, mantenimiento, etc.; por lo que termina la configuración del sistema. Para llevar a cabo esta labor, utiliza un archivo de configuración que le indica qué es lo que debe hacer. Este archivo es el */etc/inittab* y a continuación se muestra un listado de él. Dependiendo de las caracteristicas y hardware de cada equipo, éste puede variar, pero las acciones que realiza son esencialmente las mismas:

```
is:2:initdefault:
fs::sysinit:/etc/bcheckrc </dev/console >/dev/console 2>&1
mt::sysinit:/etc/brc </dev/console >/dev/console 2>&1
link::wait:/etc/lnsyscon > /dev/console 2>&1 < /dev/null
s0:06s:wait:/etc/rc0 >/dev/console 2>&1 </dev/console
s1:1:wait:/etc/shutdown -y -iS -g0 >/dev/console 2>&1 </dev/console
s2:23:wait:/etc/rc2 >/dev/console 2>&1 </dev/console
s3:3:wait:/etc/rc3 >/dev/console 2>&1 </dev/console
or:06:wait:/etc/umount -ak -b /proc,/debug > /dev/console 2>&1
of:0:wait:/etc/uadmin 2 0 >/dev/console 2>&1 </dev/console
RB:6:wait:/etc/init.d/announce restart
rb:6:wait:/etc/uadmin 2 1 >/dev/console 2>&1 </dev/console
```

Listado del archivo /etc/inittab

Al ser ejecutado *init*, busca este archivo; si no lo encuentra, ya sea por cualquier problema, arranca el SO en modo de sólo un usuario para que pueda ser solucionado el conflicto. Si lo halla, busca entradas en él de tipo *boot* y *bootwait* que son ejecutadas únicamente cuando arranca el sistema. Posteriormente busca una entrada que defina cuál será el estado inicial en que será colocado el sistema. En este caso, la línea 1 establece que sea colocado en el estado 2 correspondiente al de multiusario; por lo que todas las entradas que carezcan de nivel (que se ejecutan para cualquier estado) y aquéllas que indiquen el nivel 2, serán ejecutadas. Es por ello que según el listado anterior, se ejecutan las lineas 2,3,4 y 7.

La segunda línea (*fs*) corre el programa */etc/bcheckrc*, que se encarga de ejecutar los comandos necesarios para checar el estado de los SA, y que puedan ser montados posteriormente; de generar ligas a ciertos archivos de dispositivo necesarias para el correcto funcionamiento del SO; inicializar los volúmenes lógicos, y en general, cualquier acción necesaria que se deba ejecutar antes de montar los SA del equipo.

La tercera (*mt*) ejecuta el comando */etc/brc*, que se encarga de inicializar la tabla de SA montados, (archivo */etc/mtab*), montar los SA virtuales como */proc* y otras acciones relacionadas.

La cuarta inicializa la consola, y la séptima, ejecuta el programa */etc/rc2* que se encarga de establecer el medio ambiente de trabajo multiusuario. Para lograrlo, ejecuta cada uno de los archivos almacenados en el directorio */etc/rc2.d* que empiecen con una *S*. La Tabla 2 muestra una breve descripción de cada uno[13] de los programas contenidos en él.

Tabla 2 Contenido del directorio /etc/rc2.d

Archivo	Función
S00announce	Despliega mensajes de que el sistema está arrancado o siendo dado de baja, que son guardados en la bitácora del sistema (*SYSLOG*).
S21perf	Si se encuentra activado *sar*, Ejecuta el software de monitoreo y reporte de la actividad del sistema.
S49swap	Añade las particiones de swap definidas en */etc/fstab*. Si se encuentra activada la memoria virtual de swap, la crea y la añade (ver pág. V-27).
S64dynaweb	Si se encuentra activado, ejecuta el servidor de *dynaweb*; que permite que los usuarios puedan consultar los manuales disponibles en *Insight* a través del WEB.
S97mediad	Activa el software de *mediad*, que monitorea los dispositivos que utilizan medios removibles (discos flexibles, CDROM, etc.) y cuando es introducido uno, lo monta automáticamente.
S04usr	Monta el directorio *usr*, si se encuentra en una partición diferente a la de root.
S22acct	Si se encuentra activado, ejecuta el software de *acct* (contabilidad de recursos).
S75cron	Activa y configura el *cron*.
S12filesystems	Monta los SA aún no montados de */etc/fstab*.
S23autoconfig	Cuando ha sufrido alguna modificación el kernel, lo reconfigura.
S50mail	Configura el software de *sendmail* para correo electrónico.
S88configmsg	Imprime mensajes si durante la última instalación de software quedaron productos con conflictos que requieren ser resueltos. También regenera los documentos del comando *man* que han sido añadidos.
S14quotas	Si se encuentra activado, inicializa y arranca el sistema de *quotas*, que permite controlar la cantidad de espacio que puede utilizar cada usuario dentro de un SA para almacenar su información.
S30network	Activa los servicios de red.
S58rmtmpfiles	Si se encuentra activado, limpia el contenido del directorio */tmp* y configura */tmp* y */usr/tmp* como directorios sticky.
S90chkdev	Checa por la existencia de nuevos dispositivos, o por si ha sido removido alguno, para crear los dispositivos de acceso adecuados.
S98xdm	Activa el manejador de ventanas XDM.
S16postinst	Checa y ejecuta cualquier proceso de instalación que haya quedado inconcluso.
S60lp	Activa el sistema de impresión.
S20sysetup	Realiza las configuraciones iniciales del sistema (nombre del equipo, activa *SYSLOG*, etc.).
S48savecore	Salva el archivo *core* generado por el kernel si se está arrancado después de que ocurrió una caída del sistema.

[13] Dependiendo de la configuración, el contenido de este directorio puede variar.

Por lo general, cada uno de estos programas, revisan la configuración del sistema (ver Configuración, pág. II-16) para determinar si deben ser ejecutados. Es decir, que el programa *S14quotas* es ejecutado por encontrarse en este directorio y empezar con S; pero antes de proceder, determina si el sistema de *quotas* se encuentra activado. Si lo está, continúa su ejecución; en caso contrario, termina inmediatamente.

A medida que son ejecutadas estas aplicaciones, se va generando el medio ambiente de trabajo, y cuando se han terminado de procesar, se dice que el sistema se encuentra en el nivel 2 de multiusuario y listo para trabajar. Es cuando aparece la ventana de bienvenida solicitando una clave en la consola y cualquier usuario puede accesar a través de terminales o la red.

Es importante mencionar que todos los programas de configuración del sistema, independientemente del nivel de que se trate, se encuentran en el directorio */etc/init.d*, y que dentro del directorio */etc/rc2*, se encuentran únicamente ligas apuntando a los archivos fuentes adecuados para el nivel 2, de ese directorio.

Como conclusión del análisis del proceso de arranque, podemos decir que el Kernel es el corazón del SO IRIX, y se encuentra ubicado en la raíz del directorio principal de la partición 0 del disco de arranque y lleva por nombre */unix*. De los programas anteriores, es el más inteligente; controla los diversos dispositivos del equipo (discos, RAM, terminales, impresoras, etc.) y permite el acceso a usuarios así como la ejecución de programas dentro de él.

II.I.II. Apagado

En lo tocante al proceso de apagado, el SO UNIX no es como otros, en los cuales se puede apagar físicamente el equipo y ya. Por la forma de trabajar, es indispensable que antes de ser apagado se lleven a cabo ciertas labores que lo coloquen en una posición estable.

Durante su diseño, se buscó crear un sistema que fuera lo más ágil, poderoso y rápido posible, para poder brindar un mejor rendimiento a los usuarios finales. Para ello, se encontró que una de las labores que consumen mayor tiempo, es la rutina mediante la cual se lee o graba información al disco; por lo que había que mejorarla. Este hecho fue tomado en cuanta durante su creación.

En primer lugar, la disposición del sistema de archivos (ver pág. III-14, Sistema de archivos EFS), consiste básicamente de un bloque de datos que contiene toda la información de su estructura (superbloque); inodos, que contienen la información que especifica las características y ubicación de cada archivo, y finalmente, el espacio para almacenar los

datos. Por otro lado, para agilizar el proceso de lectura escritura, al arrancar el sistema se leen del disco estas tablas y se mantiene una copia de ellas en la memoria RAM; de tal forma que cuando se accesa un archivo, esta información es obtenida de RAM y no del disco; lo cual reduce enormemente el tiempo de espera. De igual manera, cuando se escribe se actualizan en la RAM, agilizando el proceso. Esto es automático y transparente al usuario.

Para mantener un control, el sistema cuenta con rutinas que, cada determinado tiempo, o cuando se llena el espacio reservado para esta labor, realizan un vaciado de las modificaciones hechas en RAM al disco. Este vaciado lo llevan a cabo en forma consecutiva, por lo que el proceso de escritura al disco se realiza de una forma óptima. Cuando esto sucede, la información del disco y la memoria RAM es idéntica, y podríamos decir que el sistema se encuentra estable. Posteriormente con el uso del equipo, el estado de la información en RAM y disco llega a ser diferente nuevamente.

Ésta es la principal razón por la que todo SO UNIX, o cualquiera que se encuentre basado en él, como lo es IRIX, deben ser apagados siguiendo un procedimiento. Si la energía llega a interrumpirse o se apaga el equipo sin realizar este proceso, las tablas del superbloque, inodos libres y otra información contenida en RAM, se perderá. Por su parte, las que se encuentran en el disco pudieran no estar actualizadas; por lo que no reflejan el estado actual de los datos almacenados. En este momento se dice que el Sistema de Archivos se encuentra corrupto, y cuando el disco se encuentra en este estado, no puede ser utilizado. Para ser resuelto el problema, se debe ejecutar un proceso (ver pág. V-31) que limpie y actualice la información y tablas del disco, durante el cual, se puede dar el caso de que se pierda un archivo, varios, todo el SA o ninguno, dependiendo de la gravedad del daño.

El proceso mediante el cual se verifica el estado de un SA, que no este corrompido, y se lee la información de él a memoria RAM, para que el SO lo reconozca y pueda trabajar con él, se llama montaje y es llevado a cabo mediante el comando *mount* (ver Fig. III-12); por ello, hasta que no se monta un disco, no puede ser utilizado. De forma similar, si ya no se va a trabajar con un disco (SA) se puede desmontar con el comando *umount*, y en ese momento, se actualiza la información de RAM al disco, para que pueda ser removido lógicamente del SA principal. Estos procesos pueden ser llevados a cabo sin necesidad de apagar el equipo.

Las causas por las cuales se puede desear apagar un equipo son muy variadas: por mantenimiento físico; instalación de nuevas partes o traslado del equipo; por mantenimiento de software, ya sea desde el PROM, *sash* o en modo de un usuario; realización de respaldos completos, instalación o remoción de software; revisión del sistema; para liberarlo, si es que se bloquea, etc. Por todo esto y lo anterior, se crearon y se pueden utilizar ciertos comandos para realizar el apagado del equipo correctamente, dependiendo de lo que se desee hacer. Ellos son: *shutdown*, *reboot*, *halt*, botón de apagado e *init*.

II.I.II.I. Init

Entre los estados en que puede colocar al sistema, se encuentra el 0; que lo deja en un modo adecuado para poder apagar el equipo, conocido como "Cerrado del Sistema". Por lo tanto, cuando se desee apagar el equipo por cualquier razón, desde la clave de root se puede dar el siguiente comando para ello:

 # init 0

Ésta es la principal herramienta para realizar el proceso; las demás se basan en ella, como se examinará posteriormente. Básicamente se encarga de ejecutar todos los comandos definidos para el estado 0, en el archivo */etc/inittab*. Como se puede observar del listado anterior (pág. II-5), procesa los comandos de */etc/rc0* (s0), *umount* (or) y *uadmin* (of) cuya función en conjunto es:

- Ejecutar todos los programas que empiecen con una *K* del directorio */etc/rc0.d*; que se encargan de terminar de forma adecuada, los procesos que componen el sistema.
- Ejecuta los programas que empiezan con una *S* del directorio */etc/rc0.d*, para efectuar labores de limpieza finales antes de poder apagar el sistema; siempre y cuando sean necesarias. Estas labores no consumen mucho tiempo, por lo que después de un pequeño lapso de espera, se matan todos los procesos.
- Desmontar todos los SA que no lo están aún.
- Determinar si se reconfiguró el kernel, cuyo proceso genera el archivo */unix.install*. Si se encuentra este archivo, lo instala como */unix* para que sea utilizada esta nueva versión, la próxima vez que se encienda el equipo.
- Mata cualquier proceso de montaje que se haya quedado pendiente.
- Sincroniza los discos, para que la información de RAM sea grabada a disco y se encuentren actualizados y estables los SA.
- Cuando termina su función, el CPU es apagado, o en la pantalla debe aparecer el mensaje de que ya puede ser apagado el sistema; dependiendo de las opciones dadas. Funciones administrativas básicas como éstas, pueden ser efectuadas con el comando *uadmin*; pero debido a sus características, si se requiere debe ser empleado con sumo cuidado.

Se recomienda utilizarla cuando se está trabajando en modo de un sólo usuario o de administración, donde se supone no existen más personas conectadas al equipo. Si se está operando en cualquiera de los modos de multiusuario, es conveniente utilizar otras herramientas o tener en cuenta las siguientes recomendaciones:

- En primer lugar, como su labor la empieza a ejecutar inmediatamente, es conveniente terminar todas las aplicaciones que se estén procesando en el sistema, antes de ejecutarlo;

como el cierre de bases de datos, terminar de forma correcta cada una de las aplicaciones, etc.

- En segundo lugar, como IRIX es un sistema multiusuario, probablemente se encuentren usuarios conectados a él de forma remota; por lo que es conveniente enviarles mensajes con cierto tiempo de anticipación, de que el sistema será apagado. De tal forma que cada uno pueda cerrar sus aplicaciones y salirse de sesión correcta y oportunamente.
- En tercer lugar, es conveniente implementar un mecanismo para que una vez emitido el primer anuncio de que se dará de baja el sistema, no se permita el acceso a más usuarios; ya que éstos no sabrán que se apagará el equipo, y pueden empezar a trabajar con información que puede ser dañada al darse de baja el sistema.

II.I.II.II. shutdown

Esta herramienta intenta suplir las deficiencias mencionadas en el comando *init*, enviando un mensaje de advertencia y uno final indicando que se dará de baja el equipo, antes de proceder. Por ello, es la herramienta recomendada cuando se está trabajando en alguno de los modos de multiusuario.

El comando *shutdown* permite cambiar el estado en que se encuentra el SO al de simple usuario (1, s ó S), el estado de reboot (6) y el de cerrado de sistema (0); por default supone que se apagará el sistema. Para ejecutarlo se debe estar en la cuenta de root y en la raíz del SA (*/*). Las labores que desempeñan son:

1. Antes de proceder, checa que la cuenta que lo ejecutó sea root y verifica los parámetros que se le dieron. Entre éstos, se cerciora de que el estado al cual se desee cambiar, indicado con la opción *-i*, sea 1, s, S, 6 ó 0 únicamente. Si no es uno de ellos, emite el siguiente mensaje y termina:

   Initstate 2 is not for system shutdown

2. Después despliega la fecha y el siguiente mensaje:

   Shutdown started. 20 Jul 97

3. Sincroniza la información de los discos.

4. Despliega el siguiente mensaje de advertencia en las terminales de todos los usuarios conectados al equipo:

   The system will be shut down in 300 seconds.
   Please log off now

5. El mensaje indica que el sistema será dado de baja en 300 segundos y que se deben salir de sesión; después de lo cual, espera por 300 segundos (5 min.) antes de continuar. Por default asigna un período de gracia de 60 segundos (1 minuto), el cual puede ser modificado con la opción *-g*.

6. Después del período de gracia, envía el siguiente mensaje final indicando que el sistema se está dando de baja y, espera nuevamente el tiempo de gracia definido (300 segundos en este ejemplo) antes de proceder realmente con el proceso:

   THE SYSTEM IS BEING SHUT DOWN! Log off now.

7. Una vez transcurrido este segundo periodo de gracia, despliega el siguiente mensaje en la terminal desde donde se ejecutó el comando, preguntando si se desea continuar con el proceso de cerrado del sistema:

   Do you want to continue with the shutdown ?

8. Si se responde ***n*** (no), despliega el siguiente mensaje en todas las terminales indicando que fue una falsa alarma y que no se dará de baja el sistema; posteriormente aborta el proceso y todo permanece intacto:

   False Alarm: The system will not be brought down
   Shut down aborted

9. Si se responde con ***y*** (yes), y si se especificó la opción *-p* (power off = apagado), se establecen los parámetros para que una vez que se cierre el sistema, se apague fisicamente el CPU. Es importante mencionar que no todos los equipos poseen los mecanismos para implementar esta opción. El equipo INDY sí lo soporta. A continuación, se procede a cerrar el sistema; para lo cual se ejecuta el comando:

   init $initstate

10. *$initstate* es uno de los estados válidos a los cuales se puede cambiar este comando. Por default se asume que se dará de baja el equipo y ejecuta con el valor de 0.

Con esto se ejecutan todas las labores examinadas en el punto anterior referente a *init*, cerrando de forma adecuada los procesos y el sistema. Por todo lo anterior, si se desea dar de baja el sistema dando un tiempo de gracia de 5 minutos y que se apague el equipo fisicamente, se debe ejecutar el comando

 # cd /
 # shutdown -y -g300 -p

La opción *-y* indica que se asuma un ***y*** (yes) como respuesta a cualquier pregunta; por lo que el comando procederá sin ninguna intervención posterior y no realizará la pregunta indicada en el punto 7. Esto permite que pueda ser programada para apagar automáticamente el equipo a determinadas horas, ya sea mediante el comando *at*, si se trata de una ocasión, o mediante el sistema de *cron* para que se realice en forma periódica determinados días del año.

Éste es el comando adecuado cuando se desee pasar al modo de un sólo usuario para realizar labores de mantenimiento o apagar el equipo; ya que antes de proceder despliega mensajes de advertencia; siempre y cuando no se dé un tiempo de gracia de 0 seg.

II.I.II.III. halt

Es el comando empleado para detener el sistema. Básicamente, *halt* llama a *shutdown*, quien como ya se vio, llama a *init*. Cuando es ejecutado *halt*, determina si el comando fue dado desde una terminal remota, en cuyo caso pregunta si realmente se desea dar de baja el equipo. Si se responde con ***n*** (no), se aborta el proceso. Si se responde con ***y*** (yes), se ejecuta el siguiente comando que da de baja el equipo, como ya se describió:

/etc/shutdown -y -g0 -i0 $poweroff

Si al ejecutar el comando *halt* se da la opción *-p*, *$poweroff* le indica a *shutdown* que debe ser apagado fisicamente el equipo una vez terminada su labor; en caso contrario, se queda en el PROM para poder realizar labores de mantenimiento o apagarlo manualmente.

Ya que se establece un tiempo de gracia de 0 segundos, el sistema es dado de baja inmediatamente; por lo que *halt* suele ser utilizado cuando se trabaja en modo de un sólo usuario y se desea apagar el equipo (si se da la opción *-p*) o entrar al PROM para realizar labores de mantenimiento desde ahí. Por esto mismo, no es recomendable ejecutarlo desde cualquiera de los modos de multiusuario.

II.I.II.IV. reboot

Al igual que *halt*, *reboot* determina si el comando fue dado desde un equipo remoto; si es así, presunta si realmente se desea proceder. Si se responde con ***n*** (no) se aborta el proceso; si es con ***y*** (yes) se ejecuta el comando *shutdown*:

/etc/shutdown -y -g0 -i6

Como ya se observó, la opción *-i6* indica que se ejecute el comando *init 6*, correspondiente al modo de reboot que da de baja el equipo y posteriormente vuelve a arrancar.

Este comando suele ser dado cuando se está trabajando en modo de un usuario, ya que al dar la opción *-g0* no se establece ningún tiempo de gracia y el sistema es dado de baja inmediatamente. También, es utilizado cuando se han realizado modificaciones al sistema y se desea apagar y encender nuevamente para ponerlas en función.

II.I.II.V. Botón de apagado

Ésta debe ser la última de las alternativas empleadas para dar de baja el sistema; ya que aunque efectúa procesos para dar de baja el equipo en forma adecuada, elimina de una manera más drástica los procesos del sistema y, no envía ningún mensaje de advertencia. Este método se recomienda cuando el sistema se encuentra bloqueado y no puede ser ejecutado alguno de los métodos estudiados anteriormente. Cabe mencionar, que SGI afirma que la información no sufre daños al utilizar este método; pero también sugiere primero utilizar otra alternativa.

Este método consiste en presionar el botón de encendido, el cual primero da de baja el equipo y posteriormente quita la corriente al CPU.

II.I.II.VI. Recomendaciones

Las recomendaciones para apagar un equipo son las siguientes:

- De ser posible, planear con tiempo las labores que impliquen el apagado o suspensión de servicio, y anunciar a los diferentes usuarios, las causas, hora y tiempo de suspensión. Para esta labor se pueden emplear comandos como el de *news*, el mensaje del dia (*/etc/motd*), enviando mensajes a través del correo, etc.

- Antes de apagar el equipo, realizar revisiones de último momento y verificar que no existan usuarios en sesión, ni procesos corriendo que puedan ser afectados; recordar que pueden existir personas utilizando diversos servicios desde diferentes puntos: *ftp*, *telnet*, *www*, *rlogin*, *rexec*, etc. Para ello, utilizar las herramientas de que dispone el sistema: *w*, *who*, *whodo*, *ps*, etc.

- Si es necesario, enviar mensajes a intervalos consecutivos avisando de la próxima suspensión del servicio mediante el comando *wall*.

- Ejecutar el comando para dar de baja el equipo. Utilizar alguno de los repasados anteriormente; de preferencia utilizar el comando *shutdown* con un periodo de gracia.

 # shutdown -y -i0 -g300

- Una vez terminado el proceso de baja, apagar el equipo y llevar a cabo las labores planeadas.

Si el equipo se debe apagar todos los días a la misma hora, el proceso anterior puede ser programado para que se realice automáticamente mediante el uso del *cron*. Para ello:

- Añadir la siguiente línea al archivo */usr/spool/cron/crontab/root*.

*45 20 * * 1-5 /etc/shutdown -y -g300 -p*

- Ejecuta el siguiente comando para actualizar la tabla del *cron* en RAM.

 # crontab */usr/spool/cron/crontab/root*

Con ello, de lunes a viernes a las 8:45 p.m. el sistema de *cron* lanzará el comando *shutdown* con un periodo de gracia de 5 minutos.

Si el sistema se encuentra bloqueado, ya sea por alguna falla, un proceso estancado, u otra razón:

- Si la consola no se encuentra bloqueada, ingresar al sistema para solucionar el problema: ya sea buscando y eliminando el proceso que causa el conflicto, sacando de sesión a cualquier usuario atorado que esté afectando el sistema, reiniciando el equipo con los comandos adecuados, etc..

- Si la consola se encuentra bloqueada, tratar de ingresar desde una terminal conectada al equipo.

- Si no se tienen terminales o se encuentran bloqueadas éstas, y el equipo se encuentra conectado a una red, tratar de ingresar remotamente desde cualquier otro equipo mediante un *telnet*.

- Si en definitiva, no se puede ingresar al equipo, el último recurso es el de apagarlo; para lo cual:

  ◊ Como otra alternativa, es conveniente esperar unos minutos para que si *bdflush* no se encuentra bloqueado, actualice las tablas de RAM a disco automáticamente.

sincronizándose la información de éstos y que los daños que pudiera sufrir el SA, sean mínimos o nulos.

◊ Después de lo anterior, para el caso de la INDY se tiene otra alternativa, y consiste en presionar el botón de apagado; para que ejecute la interrupción que tiene incorporada y dé de baja el SO antes de quitar la energía eléctrica.

◊ En otros modelos apagar el equipo mediante el interruptor apropiado.

◊ Encender nuevamente el equipo e inspeccionar y solucionar cualquier problema que se haya presentado.

II.I.III. Consideraciones

- Estos equipos están diseñados para trabajar las 24 hrs. del día y el encenderlos y apagarlos continuamente, puede causar un mayor desgaste de los componentes; sin tomar en cuenta que el riesgo de presentarse fallas se incrementa.

- La mayoría de los equipos cuentan con un interruptor que permite encender y apagar el equipo físicamente. Si éste es oprimido mientras el equipo está funcionando, se suspende la corriente y la información de RAM se pierde, pudiendo quedar dañados los SA igualmente. En el caso del equipo INDY (ver Fig. I-5), si este botón es presionado, realiza un procedimiento mediante el cual termina todos los procesos, actualiza la información de RAM a disco (lo sincroniza) y una vez efectuadas estas labores, apaga el equipo.

  Aunque SGI asegura el correcto funcionamiento del botón para dar de baja el equipo, éste deber ser utilizado en el último de los casos, ya que no concede período de gracia para que los usuarios terminen sus labores y salgan del sistema antes de apagarlo; sino que simplemente mata los procesos actuales y lo apaga. Podríamos ver este botón como una mejoría y una buena alternativa al estilo de los demás equipos; los cuales simplemente eliminan la corriente y la información se pierde y dañan los discos.

- No mover el equipo mientras se encuentre encendido; ya que puede afectar el funcionamiento de los discos y ocasionar, posiblemente con cualquier movimiento brusco, que las cabezas se estrellen contra la superficie de los platos del disco causando daños irreparables.

- De igual manera, antes de mover el equipo y después de apagarlo, es conveniente esperar unos minutos para que cese el movimiento de los platos en los discos.

II.II. Configuración

De lo expuesto anteriormente se sabe que al encender el equipo, se carga el kernel, quien empieza su configuración. Para tal efecto, ejecuta ciertos procesos entre los que se encuentra *init*, cuya función es la de checar el estado de los discos, montar las particiones adecuadas, definir la consola y configurar el ambiente de trabajo para el estado en que va a operar, generalmente es el 2. Para lograr esto último, ejecuta el comando */etc/rc2* que se encarga de procesar todos los archivos dentro del directorio */etc/rc2.d* que empiecen con una *S*. Cada uno de ellos activa un subsistema que permite ir adquiriendo las condiciones necesarias para el correcto funcionamiento del equipo (ver Tabla 2, pág. II-6); de tal forma que cada módulo se va añadiendo al kernel para formar un todo, que es conocido propiamente como el SO.

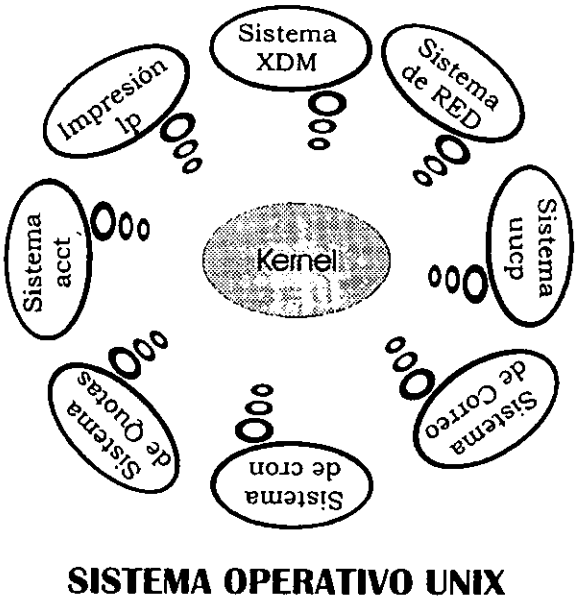


Fig. II-2 Componentes del sistema operativo.

Existe un mecanismo de configuración que permite controlar qué módulos deben funcionar y cuáles no; es decir, aunque todos los archivos del directorio */etc/rc2.d* se ejecutan, no todos se activan. Este mecanismo es controlado por el comando *chkconfig*, mediante la utilización de banderas. Su forma de operar es la siguiente:

En primer lugar, este sistema de configuración se encuentra diseñado e implementado en la mayoría del software original (demonios y subsistemas) del SO IRIX; por lo que si se instala un software de terceros, éste puede o no estar preparado para funcionar del mismo modo, por lo que se debe leer la documentación que lo acompaña para determinar el correcto método de activarlo o eliminarlo del sistema.

Cuando se instala cualquier aplicación mediante *inst*, se copian los programas y archivos en los directorios adecuados dentro del disco duro; entre ellos, queda instalado el archivo de configuración apropiado en el directorio */etc/init.d* y la liga al directorio correspondiente al estado de ejecución, */etc/rc2.d* para al estado de multiusuario o algún otro. Con esto, el producto queda listo para ser utilizado.

Para activarlo, generalmente se utiliza el comando *chkconfig*, al cual se le debe dar como parámetro, el nombre apropiado del producto y la palabra *on*.

 # chkconfig acct on

que activa la bandera indicando que el proceso de contabilidad de recurso, *acct*, debe ser ejecutado. Esencialmente la bandera consiste en crear un archivo dentro del directorio */var/config*, cuyo nombre es el mismo que el del módulo (*acct*), y conteniendo la palabra *on*.

Este cambio en la configuración no entra en función hasta que se reinicia el equipo; por lo que la próxima vez que se encienda, el programa adecuado de *init* ejecuta todos los procesos para el estado 2, multiusuario, ejecutándose los archivos de configuración del directorio */etc/rc2.d*, entre ellos el de *acct* que detecta la bandera y activa el módulo de contabilidad de recursos. Si se desea que el cambio entre en función inmediatamente, se puede ejecutar el programa que lo activa con la opción START:

 # /etc/rc2.d/acct START

Por todo esto, la mayoría de los archivos de configuración colocados en este directorio, utilizan rutinas para determinar el estado de su bandera y contemplar parámetros como el *START*, para activar el proceso, o el de *STOP*, para detenerlo. En lo tocante a su implementación, para los parámetros suele utilizarse una estructura de *CASE*, y para determinar el estado de la bandera, el siguiente segmento de programa:

```
if /etc/chkconfig acct; then :
else
        exit
fi
```

La instrucción *chkconfig acct* determina el estado actual de la bandera del módulo *acct*. Para realizar esta función, checa por la existencia de un archivo llamado */var/config/acct*, en este caso, y que contenga la palabra *on*. Si lo encuentra, entrega un valor verdadero; por lo que la estructura de *if* continúa con las demás secuencias de instrucciones almacenadas en el archivo, cargando a RAM el módulo. Si no existe o contiene cualquier otro valor, *off* por ejemplo, entrega un valor falso, indicando que no se encuentra activado; por ello, se ejecuta la instrucción *exit* que termina el proceso, absteniéndose de ejecutar las restantes, y por ende, de cargar el módulo. Por todo lo anterior, si se desea detener algún subsistema en especial, se pueden utilizar los comandos:

 # /etc/rc2.d/acct STOP
 # chkconfig acct off

La primera instrucción detiene el proceso de contabilidad inmediatamente; la segunda se utiliza para desactivar la bandera, de tal forma que la próxima vez que se arranque el sistema, éste ya no sea cargado. Si se desea conocer cuál es el estado y cuáles son las banderas definidas actualmente, se puede utilizar el comando:

% chkconfig

Flag	*State*
====	=====
accesoremoto	*on*
acct	*off*
autoconfig_ipaddress	*off*
desktop	*on*
directoryserver	*off*

Un resumen de las más importantes se encuentran en la Tabla 3; si se tiene dudas con respecto a su función, se puede ver la ayuda de cada una de ellas o de los demonios que activan mediante el comando *man*.

Es conveniente mencionar que pueden no estar definidas algunas de ellas. Esto se puede deber a que el módulo que controlan no se encuentre instalado en el sistema, o a que aún no se ha empleado; por lo tanto, creada su bandera. Para el primer caso, se puede utilizar el comando *versions* y determinar los módulos actualmente instalados; si no se encuentra, se deberá emplear el comando *inst* para instalarlo. Si se encuentra instalado, se puede crear su bandera mediante el comando:

 # chkconfig -f acct on

Tabla 3 Banderas de configuración

Bandera	Descripción
acct	Sistema de contabilidad de procesos
desktop	Activa la mayoría de las características de la interfaz de usuario Indigo Magic; entre ellas se encuentra el ambiente de trabajo característico de los Sistemas IRIX y el *toolchest*[14].
directoryserver	Activa el servidor de directorios de *Cadmin*.
dynaweb	Activa el servicio de *dynaweb*.
gated	Activa el demonio de ruteo Cornell; en lugar del de BSD.
mediad	Demonio de los medios removibles; se encarga de montar automáticamente, cualquier SA contenido en un medio removible (CDROM, diskettes, etc.) cuando es introducido.
mrouted	Activa el demonio de ruteo de paquetes multicast IP. (sólo se debe de activar si el equipo funciona como un gateway).
named	Servidor de Nombres por Dominio de Internet (DNS)
network	Permite la entrada y salida de información a través de la red.
noiconlogin	Cuando se activa, no muestra el icono de los usuarios de cada cuenta del equipo en la pantalla de bienvenida de *clogin*.
objectserver	Activa el demonio de servidor de objetos de *Cadmin*.
rarpd	Demonio de *rarp* (Reverse ARP).
routed	Demonio de ruteo RIP 4.3BSD.
rwhod	Demonio *rwho* 4.3 de BSD
sar	Activa el Sistema de Reporte de Actividad.
soundscheme	Activa el demonio de señal de audio de Indigo Magic.
timed	Activa el demonio de sincronización de tiempo de 4.3BSD.
timeslave	Activa el demonio de sincronización de tiempo de SGI.
verbose	Cuando se activa, se imprime el nombre de los demonios a medida que se ejecutan al arrancar el SO.
visuallogin	Activa el sistema de autentificación *clogin*, en la consola.
vswap	Añade área de swap virtual al sistema. Por default agrega 80000 bloques. Este valor puede ser modificado en el archivo */var/config/vswap.options*
windowsystem	Activa el Ambiente Gráfico de Ventanas X
xdm	Activa el Administrador de Ventanas X
yp	Activa el servicio de NIS

[14] Es un programa gráfico que implementa un menú, por medio del cual, se pueden ejecutar diversas aplicaciones.

La opción *-f* (forzar) le indica que si su archivo de bandera no existe, */var/config/acct*, lo cree. Otro ejemplo, para deshabilitar la autentificación de *clogin*, y que sea utilizada la default de *xdm*, ejecutar el comando:

 # chkconfig visuallogin off

La próxima vez que se arranque el equipo, *clogin* ya no controlará el acceso a la consola.

Este simple mecanismo permite de una manera sencilla, el poder habilitar o deshabilitar un subsistema sin mayores complicaciones; además, es empleado por demonios y aplicaciones para determinar si deben o no realizar alguna labor. Es conveniente analizarlo con cuidado, y en dado caso, implantar esta misma técnica para cualquier proceso o módulo adicional que se instale o cree, fomentando así, una práctica simple y congruente en la manera de configurar el SO. Para implementar esta técnica en un módulo que se desee añadir, realizar lo siguiente:

- Instalar los archivos del programa dentro del lugar adecuado del SA:

- Determinar cuál será la bandera que lo activará. Elegir un nombre diferente a los utilizados por el Sistema y que vaya de acuerdo a la función que desempeña del módulo instalado. Para el ejemplo, suponer que será *modulouno*

- Crear un archivo que se encargue de activarlo. Este debe incluir al inicio, las instrucciones que permitan detectar si su bandera se encuentra activada o no:

Caso uno: mediante su bandera

```
if /etc/chkconfig modulouno
then
    echo Activando servicio
else
    exit
fi
```

Caso dos: con la utilización de opciones.

```
case "S1" in
    'start')
      echo activar funcion.
    ;;
    'stop')
      echo detener el proceso.
    ;;
esac
```

- Colocarlo con los permisos adecuados en el directorio */etc/init.d*.

 # cp modulouno /etc/init.d/modulouno

- Crear la liga adecuada al estado en el cual se desea activar.

  # cd /etc/rc2.d
  # ln ../init.d/modulouno S14modulouno

  La primera instrucción es para cambiarse al directorio *rc2.d* correspondiente al estado de multiusuario. Desde ahí, se hace una liga al programa *modulouno*, que llevará por nombre *S14modulouno*: la *S* indica que debe ser ejecutado al arrancar el sistema. El *14* sirve para establecer el lugar que ocupará al ejecutarse; ya que se procesan conforme a su nombre. Y *modulouno* es el nombre del servicio que se activará.

- Crear su bandera

  # chkconfig -f modulouno on

- Documentar la nueva función. Es conveniente crear una nueva entrada en el *man*, que especifique cuál es la función y opciones de esta nueva aplicación; así como el procedimiento para activarla o desactivarla. En este sentido, se puede crear una nueva entrada en el catálogo del comando *man*, a fin de que pueda ser consultada por cualquier usuario. Esta labor puede ser realizada con herramientas especiales que permiten crear documentos formateados, como lo es el Documenter's Workbench. Si no se cuenta con alguna, se puede utilizar el siguiente método:

  - Crear un archivo con cualquier editor de textos que explique el funcionamiento del comando. Es conveniente seguir el estilo definido para las páginas de referencia del comando *man*; es decir, que entre otras incluya las secciones de:

Name	Nombre del comando.
Synopsis	Descripción de las diversas formas de ejecutarlo.
Description	Descripción detallada de su funcionamiento.
Options	Posibles opciones a utilizar durante su ejecución.
Files	Archivos que se utilizan durante su funcionamiento.
See Also	Referencias a otros comandos relacionados.

  - Dar a este archivo un nombre que tenga referencia con el comando o servicio que explica y, añadirle una extensión de *.1* para indicar que es un documento de referencia local.

  - Colocarlo dentro del directorio de búsqueda del comando *man*; es decir, dentro del directorio */usr/man*. Es conveniente agrupar todos los documentos que expliquen comandos y funciones locales, en un directorio para distinguirlos; por lo que es una buena práctica, ubicarlos en el directorio */usr/man/man1*, nuevamente la *1* significa que son documentos locales. Si no existe, puede ser creado:

 # mkdir /usr/man/man1
 # mv modulouno.1 /usr/man/man1/modulouno.1

- Con esto, cualquier usuario puede utilizar el comando *man* para leer su documentación:

$ man modulouno

Un punto adicional, si el archivo es demasiado grande, se puede utilizar el comando *pack* para compactar su información y ahorrar espacio en disco; el cual produce archivos con el mismo nombre, pero con la extensión *.z*:

#pack modulouno.1
 # mv modulouno.1.z /usr/man/man1/modulouno.1.z

La forma de utilizar el comando *man* para leer la ayuda de este tipo de archivos es idéntica. Al ejecutarlo, se encarga de descompactar el archivo y mostrar su información en la pantalla; por lo que para el usuario este proceso resulta transparente.

II.III. Procesos

Un proceso es una tarea que el SO lleva a cabo para mantenerse corriendo adecuadamente, o para completar un comando dado explícitamente; en términos generales, se dice que un proceso es un programa en ejecución.

Podemos clasificar a los procesos en interactivos, discontinuos y demonios. Los primeros son los que se ejecutan desde un shell, por lo que se puede interactuar con ellos, y sobre todo, que están asociados a una cuenta, que pertenece al usuario que lo ejecutó, a una terminal, desde donde se encuentra conectado el usuario, o a una ventana, si se trabaja en ambiente gráfico.

El segundo tipo, discontinuos o por lotes, son procesos que pueden no tener una cuenta asociada, pero que pertenecen a una cola de procesos pendientes de ejecución. Un caso típico de ellos, son los lanzados desde el sistema de *cron* o mediante los comandos de *batch* o *at* por el usuario.

El tercer caso, demonios, son procesos que generalmente son iniciados al arrancar el sistema y que generalmente permanecen dormidos, esperando a que otro proceso solicite sus servicios, y entonces, se activan. Cuando cumplen su labor, vuelven a su estado de letargo, en espera de otro proceso que los active nuevamente. A este tipo de procesos (discontinuos

y demonios) también suele conocérseles como *background* por estar ejecutándose en un segundo plano, posterior o en el fondo, donde permanecen durmiendo o efectuando su labor, sin la intervención de una persona para funcionar; en cambio, para los procesos normales que generalmente se ejecutan (interactivos), suele decirse que permanecen en el primer plano, visibles y en ejecución, por lo que también suele conocérseles como *foreground*. Esto brinda otra forma de clasificarlos: *background* o *foreground*.

Finalmente, existe casos especiales que se describirán más adelante; una vez revisado los conceptos necesarios para su comprensión. Ellos son los procesos *Zombis* y los *Huérfanos*.

Ahora, ya que el funcionamiento de todo el sistema está basado en el concepto de los procesos, al fin y al cabo el propio SO está formado por una serie de procesos, es muy importante entender cuál es su mecanismo de creación y funcionamiento, a fin de poder comprender sus diferentes estados y mantener un cierto control de ellos.

II.III.I. Ciclo de vida de un Proceso

Entenderemos por ciclo de vida, a todos los estados por los que pasa o puede pasar un proceso, desde su inicio hasta su final. La primera etapa es la de creación, por la que todo proceso debe pasar (Intermediate state of creation); dentro de ella, le es asignado un número que lo identifica, conocido como Número de IDentificación de Proceso (PID). Así también, le es asignada una entrada en la tabla de procesos que mantiene el kernel en memoria RAM, en la cual, se mantiene información relevante del proceso: como su PID, la utilización de recursos, datos de su ejecución, etc.

Dentro de este contexto, existe una llama de sistema utilizada para crear un nuevo proceso: *fork*, que genera uno nuevo conocido como hijo y es una copia exacta del proceso que realizó la llamada, conocido como padre; por tal motivo, el hijo hereda ciertos atributos del padre. Esta relación se puede apreciar mejor cuando se estudia lo que sucede al ejecutar un comando desde un shell; para ello, suponer que se ejecuta el siguiente:

```
% date
Tue Oct 21 11:53:59 HdV 1997
%
```

El shell *csh* con PID 350, que en este caso recibe la instrucción de ejecutar el comando *date*, efectúa una llamada de *fork*, por lo que es creado un proceso hijo *csh* con PID 370, quien es el encargado de consumar realmente la labor. Para esto, el proceso hijo realiza una llamada de *exec* que se encarga de ejecutar el comando *date*. Ya que el comando *date* fue creado a través de una llamada *exec*, éste tiene el mismo PID que el del proceso hijo que la realizó (379). Cuando *date* entra en operación, despliega la fecha del sistema, ya que esa es su

labor; posteriormente se ejecuta una llamada de *exit* que se encarga de terminar el proceso hijo. Durante todo este tiempo, el proceso padre permanece dormido esperando a que se realice la tarea, por lo que al ejecutarse la llamada de *exit* y finalizar el proceso hijo, éste se despierta y continúa su operación; en este momento en la pantalla aparece el prompt % nuevamente. La siguiente figura ilustra este ciclo.

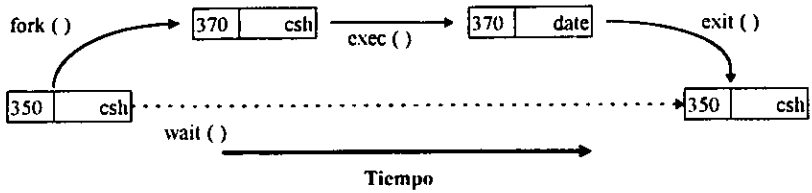


Fig. II-3 Ciclo de vida de la llamada de sistema "fork()"

De este hecho resalta la importancia que tiene el exportar las variables que vayan a ser utilizadas en la ejecución de los procesos hijos que se generen; esto es, que cuando se trabaja sobre un shell y se definen variables de ambiente en él, éstas son reconocidas únicamente por ese shell, pero no por los shell hijos que se generan durante la operación de los comandos. Analizándolo de otra forma, cuando se ejecuta un comando sobre un shell, éste crea a otro shell, el cual puede en determinado momento, desconocer variables de ambiente definidas en el shell padre; por lo que no podrá utilizarlas. Si se desea que todos los procesos que sean ejecutados desde un shell reconozcan y puedan utilizar alguna variable de ambiente, ésta debe ser exportada. Cuando esto sucede, es colocada en un segmento de memoria en donde todos los procesos hijos pueden localizarla y utilizarla.

Volviendo al ciclo de vida, una vez que el proceso ha sido creado, éste empieza a funcionar; por lo que se encuentra en el estado de ejecución (**R**unning), hasta que termina. En ocasiones, durante su ejecución puede solicitar algún recurso que no esté disponible en ese momento o encontrarse en espera de que suceda algún evento, por lo que puede pasar al estado de dormido (**S**leeping) hasta que se pueda tener acceso al recurso solicitado o suceda el evento que lo active. Si ese recurso es memoria, entra en un estado de espera (**X** waiting for memory) mientras se le puede asignar.

Finalmente, existe otro estado al cual ingresa cuando es interrumpida su labor por otro proceso o un usuario; es decir, alguno de éstos le puede enviar una señal para que detenga su ejecución, por lo que pasa al estado de detenido (s**T**opped), hasta que se le envíe una señal indicando que puede reanudar su función.

Recordando el punto anterior y entendido el presente, se puede decir que existen dos tipos de procesos más, o estados en los cuales se puede encontrar, que representan casos

especiales. El primero es conocido como Huérfano, y es aquel proceso que sigue activo después de que su proceso padre (el que lo generó) ha terminado. Cuando ocurre esto, el proceso *init* lo hereda, por lo que el Identificador de Proceso del Padre (PPID) se le asigna el valor de 1, correspondiente el PID de *init*.

El segundo son los procesos Zombis (**Z**), que es un proceso que ha terminado de ejecutarse, pero que su entrada en la tabla de procesos aún no ha sido limpiada; por lo que sigue ocupando un espacio. Esto llega a suceder cuando el proceso que lo ejecutó inicialmente (su padre) no ejecuta una llamada de *wait* para esperarlo (ver Fig. II-3). Este tipo de procesos son eliminados cuando su padre termina o cuando son heredados por *init*; pero llega a ocurrir circunstancias en las cuales, el proceso zombi sigue funcionando y por lo tanto, ocupando su entrada en la tabla de procesos. Si la cantidad de zombis existentes es demasiado grande, pueden interferir con la creación de nuevos procesos; y ya que no pueden ser eliminados, en ocasiones se tiene que reiniciar el sistema para borrarlos definitivamente. Si esta situación se llega a presentar continuamente, es esencial el investigar detalladamente cuál es el proceso que está generándolos, a fin de encontrar la causa y solucionar el problema. Cuando esto se presenta, muy frecuentemente se detecta que la causa es una falla en procesos de tipo *background*.

II.III.II. Monitoreo

El monitoreo de los procesos debe ser una labor rutinaria; brinda bastante información del comportamiento del sistema y es posible determinar problemas antes de que se puedan agravar. Para esta labor, el administrador cuenta con varias herramientas, como son: El comando *top* y *gr_top* que muestra una lista que se actualiza periódicamente, indicando cuáles son los procesos que consumen la mayor cantidad de tiempo del CPU. La primera se utiliza en terminales tipo carácter (tty) y la segunda cuando se trabaja en ambiente gráfico. Con esta herramienta se puede determinar si un proceso está saturando el sistema.

La herramienta *osview* y *gr_osview* muestran en una terminal o una ventana gráfica, la actividad de los diversos recursos del sistema: CPU, memoria, tiempos de espera de recursos, acceso a disco, etc. Con esto se puede detectar si el lanzar una aplicación ocasiona saturación en alguno de ellos, a fin de poder reprogramarla y darle la mejor solución.

El comando *ps*, es uno de los que brinda la mayor información en lo referente a los procesos. Con él se puede ver, por cada proceso: su estado, la fecha en que fue iniciado, el tiempo de ejecución consumido, su PID y PPID, el comando que se utilizó para ejecutarlo, el usuario que lo ejecutó, etc.. Estos datos son una fuente invaluable para el administrador. De ellos se puede deducir si un usuario está ejecutando procesos no autorizados; si un proceso se encuentra bloqueado (la fecha en que fue iniciado es muy antigua), se encuentra en un ciclo repetitivo sin salir (tiene demasiado tiempo de ejecución acumulado) o existen

procesos zombis; mediante el PID y el PPID, rastrear cuál es el proceso padre que los ha activado; cuántos procesos en total se encuentran en ejecución, etc... Para obtener la mayor información posible se suele utilizar las siguientes opciones:

 # ps -ef
 # ps -el

En el punto anterior se describió cada uno de los estados en los cuales puede encontrarse un proceso; en la definición de cada uno se remarcó un letra, que corresponde a la que puede aparecer bajo la columna de *S* (STATE, estado) al utilizar la opción *-l* (long, largo) en el comando *ps*, para indicar cuál es el estado de cada proceso en el momento en que fue tomada la información de la memoria. Este un punto importante, cuando es ejecutado el comando *ps*, se obtiene una copia instantánea de las características y el estado de cada proceso en ejecución en ese preciso momento; por lo que al ser desplegada esta información en la pantalla, un tiempo después, ésta ya no corresponde con la situación actual. Lo anterior quiere decir, que el tiempo de vida de los procesos es efímero, así como fugaz el tiempo que puede permanecer en un estado y características determinadas, y lo único que se puede obtener, son copias del estado en el que se encontraban en el instante en que se tomó la muestra. Un caso típico: podemos ejecutar el comando *ps -ef* y en la lista que aparecerá en la pantalla se encontrará un proceso en ejecución llamado *ps -ef*, ya que en el instante que *ps* estaba tomando la información de la memoria RAM, éste se encontraba corriendo; pero al momento que la despliega en la pantalla termina su función, por lo que fue removido de RAM. Por ello, cuando examinamos la información desplegada por él, ésta ya no es tan real, ya que el comando *ps -ef* que aparece en ella, ya no se encuentra corriendo.

Finalmente, dentro de todos los beneficios que puede brindar este comando, se encuentra el poder determinar cuál es el PID de un proceso en especial; que es necesario cuando se desea utilizar el comando *kill* para eliminarlo, como se describirá más adelante. En el capítulo de Mantenimiento, y en especial en el tema de Monitoreo, se describirán otros métodos y consideraciones a este respecto.

II.III.III. Señales

Habitualmente un proceso es creado y entra en función cuando el sistema lo genera para realizar una labor específica, o cuando un usuario lanza un comando. Normalmente permanece en este estado (ejecución) hasta completar su labor, en ese momento, termina y es removido de la tabla de procesos, desapareciendo. Ésta sería la función ordinaria de todo proceso, pero en la práctica, existen algunos que por problemas presentados, pueden permanecer atorados sin poder concluir; por otro lado, existen algunos que se encuentran ligados a determinados eventos, y sólo cuando éstos ocurren, se activan. Por todo ello, los

programas suelen estar diseñados y contener rutinas para efectuar ciertas labores al recibir determinadas señales.

Se dice que una señal es una notificación asíncrona de que un evento ha ocurrido. Las señales son generadas por eventos anormales, como: cuando un usuario presiona la tecla detener (stop), de interrupción (interrupt) o de terminar (quit) de una terminal; cuando la terminal se encuentra lista para que el proceso lea la información; cuando ocurre un error en un proceso en ejecución, indicando que debe ser removido de la memoria; cuando un programa solicita la atención de otro; cuando un proceso es detenido, o reiniciado, en espera de algún evento; cuando ocurre una falla de hardware; cuando determinado tiempo ha transcurrido, etc.. En general, una señal es enviada a un proceso cuando el evento para el cual fue diseñado ocurre.

De todas las formas de generar una señal, la que nos permite tener un control de los procesos y que se describirá a continuación, es el comando *kill*. Éste permite enviar de forma directa, una señal a un proceso; por default envía la señal número 15, correspondiente a TERMINAR, que causa que el proceso termine sus funciones de una manera limpia y ordenada. Éste suele ser el uso más frecuente que suele dársele a *kill*; ya sea porque el proceso ha sido ejecutado por error, se encuentra bloqueado, se desea abortar, o cualquier otra causa. La forma de ejecutar el comando es la siguiente:

$ kill 15 740

El primer parámetro es la señal que será enviada y el segundo es el PID del proceso al cual se desea enviar la señal. Por ello, el comando anterior envía la señal de TERMINAR al proceso cuyo PID es el 740. Si este proceso está diseñado para recibirla, tomará las medidas necesarias para terminar sus funciones inmediatamente; de una forma limpia; es decir, actualizando datos aún no procesados, cerrando archivos, etc.

Se puede dar el caso que dicho programa no este preparado para recibir esta señal o que se encuentre bloqueado y no pueda efectuarla; por lo que la única solución, será enviar la señal 9 correspondiente a matar:

$ kill 9 740

En este caso el proceso es matado inmediatamente; esto es, no efectúa ningún tipo de limpieza como en el caso de la señal 15, que le permita terminar adecuadamente. Por regla general, cuando se desea eliminar una aplicación se debe enviar la señal 15, default, y si por alguna causa el proceso no termina, debe emplearse el método rudo enviando la señal 9, la cual mata todo proceso; una excepción a este caso pueden ser los procesos Zombis.

La Tabla 4 muestra las señales actualmente definidas en IRIX. La primer columna contiene el nombre simbólico de la señal, el cual puede ser utilizado en el comando *kill*, eliminado la

frase *SIG* de él. La segunda, muestra el número de la señal, que suele ser el más empleado al ejecutar el comando. La cuarta el evento que anuncia u ocasiona dicha señal, y la tercera indica la acción que se tomará: Exit, Core, Stop e Ignore. La acción de Exit indica que el proceso debe ser terminado, para lo cual se siguen las medidas tomadas con la llamada de sistema de *exit*, que termina un proceso de forma adecuada. La de Core generalmente indica que ha ocurrido un error, y la acción que toma es idéntica a la de Exit, pero además, se genera un archivo en el directorio actual que es una copia imagen del proceso en memoria RAM al momento de recibir la señal; es decir, se genera un archivo *core*. Este archivo puede ser analizado para determinar la causa que ocasionó el error. La de Stop provoca que el proceso se detenga y la de Ignore suele utilizarse como un aviso y generalmente el proceso que la recibe la ignora.

Tabla 4 Señales actualmente definidas

Nombre	Valor	Acción Default	Evento
SIGHUP	1	Exit	Colgar (Hangup)
SIGINT	2	Exit	Interrupción (Interrupt)
SIGQUIT	3	Core	Abandonar (Quit)
SIGILL	4	Core	Instrucción Ilegal
SIGTRAP	5	Core	Rastreo (Trace/Breakpoint Trap)
SIGABRT	6	Core	Abortar
SIGEMT	7	Core	Trampa de Emulación
SIGFPE	8	Core	Exclusión Aritmética
SIGKILL	9	Exit	Matado (Killed)
SIGBUS	10	Core	Error de Bus
SIGSEGV	11	Core	Falla de Segmentación (Segmentation Fault)
SIGSYS	12	Core	Llamada de Sistema Errónea
SIGPIPE	13	Exit	Pipe roto (Broken Pipe)
SIGALRM	14	Exit	Reloj de alarma
SIGTERM	15	Exit	Terminado (Terminated)
SIGUSR1	16	Exit	Señal de Usuario 1
SIGUSR2	17	Exit	Señal de Usuario 2
SIGCHLD	18	Ignore	Cambio de Estado en un proceso hijo
SIGPWR	19	Ignore	Falla de energía/Restaurar
SIGWINCH	20	Ignore	Cambio de tamaño de una ventana
SIGURG	21	Ignore	Condición Urgente de Socket
SIGPOLL	22	Ignore	Evento Votable (Pollable Event)
SIGSTOP	23	Stop	Detenido (señal)
SIGTSTP	24	Stop	Detenido (usuario)
SIGCONT	25	Ignore	Continuar
SIGTTIN	26	Stop	Detenido (Entrada de una tty)
SIGTTOU	27	Stop	Detenido (Salida de una tty)
SIGVTALRM	28	Exit	Tiempo Virtual Terminado
SIGPROF	29	Exit	Tiempo Profiling Terminado
SIGXCPU	30	Core	Límite de tiempo de CPU excedido
SIGXFSZ	31	Core	Límite de tamaño de archivo excedido
SIGRTMIN	49	Exit	Posix.4 SIGRTMIN
SIGRTMAX	64	Exit	Posix.4 SIGRTMAX

Por ejemplo, la señal 1 HUP suele ser enviada a los procesos que se han iniciado desde una terminal remota cuando se rompe la comunicación (se cuelga la línea), lo que ocasiona una acción de Exit por lo que el proceso termina limpiamente.

II.III.IV. Prioridades

Una de las características principales del SO IRIX, es que es multiusuario y multitareas. El primer término implica que el sistema cuenta con los mecanismos adecuados para permitir que varias personas puedan conectarse y trabajar al mismo tiempo en el equipo. El segundo establece que pueden estar ejecutándose varios procesos al mismo tiempo; todo esto con sólo un CPU.

Aunque lo anterior sugiere que se encuentran en ejecución varias aplicaciones al mismo tiempo, esto no sucede; ya que el CPU sólo puede procesar una a la vez. Lo que ocurre en realidad es lo siguiente: Cuando una aplicación es ejecutada, entra por un pequeño lapso en el estado I (Intermedio de creación), dentro de la cual, es colocada en una lista donde se encuentran los procesos en ejecución. Por su parte, el CPU toma un proceso de esta lista, lo carga y lo empieza a ejecutar por un lapso; al termino del cual, lo vuelve a colocar en esta lista. Esto mismo lo realiza con cada uno de los procesos colocados ahí, y la actividad la realiza tan rápido, que pareciera que estuviesen corriendo todos ellos al mismo tiempo; aunque esto no sea así.

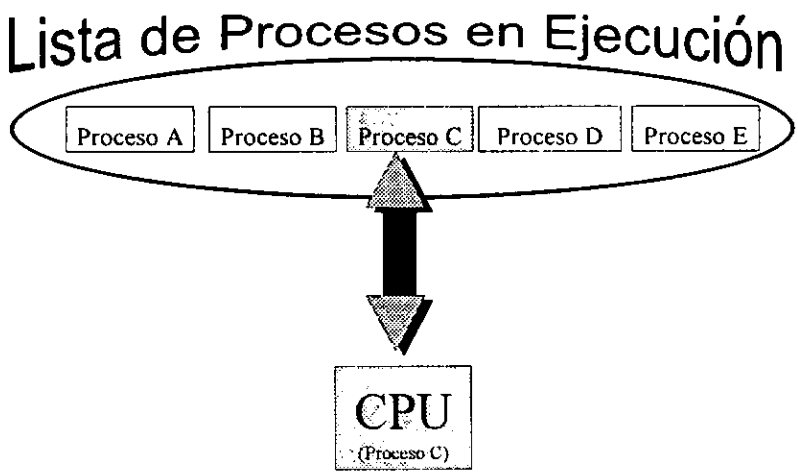


Fig. II-4 Diagrama del funcionamiento del CPU en la ejecución de Procesos.

Por otra parte el CPU toma ciertos criterios para elegir el orden en que se ejecutarán los procesos colocados en esta lista, qué tan seguido un proceso puede utilizar el CPU, así como cuánto tiempo permanecerán en ejecución dentro del CPU una aplicación determinada. Para llevar a cabo esta labor, a cada aplicación se le establecen ciertos atributos que permiten controlar estos aspectos, y para ello, se utilizan comandos como *nice*, *npri*, *renice*, *slice* y *deadline*; pero antes de tocar cada uno de ellos, es conveniente entender ciertos términos utilizados, como el de Propiedades no degradables, envejecimiento de prioridades y el esquema de periodos de procesamiento a plazos máximos.

Cuando es creada un aplicación, el sistema le asigna un número de prioridad; si es que el programador no especifica otro. Cuando es ejecutada y colocada en la lista de procesos, el CPU toma en cuenta este número para determinar cuál será el proceso al que le dará mayor prioridad en su ejecución; de tal forma, que una aplicación que tenga el doble de prioridad que otra, será cargada y procesada en el CPU un número mayor de veces, es decir, consumirá mayor tiempo de CPU en un cierto período. Por ello, a cada proceso se le debe asignar una prioridad, la cual debe estar relacionada con su importancia.

Por otra parte, cuando es cargada la aplicación en RAM, le es asignada la prioridad que le fue otorgada; pero con el transcurso del tiempo, ésta empieza a bajar. A esto se le llama 'Envejecimiento de la Prioridad' y es la forma de trabajar del SO IRIX. Con ello, se garantiza que procesos de larga duración y que consumen demasiado tiempo de CPU, se vayan degradando, disminuyendo su prioridad, a fin de que den paso a los de corta duración, para que sean procesados rápidamente y a su máxima prioridad establecida, dando agilidad al rendimiento del equipo. Dado que existen aplicaciones que por su importancia o requerimientos no es conveniente ingresar dentro de este tipo de esquema, se estableció un nuevo mecanismo, llamado 'Prioridades no degradables'. Cuando a un proceso se le asigna una de este tipo, ésta no disminuye con el transcurso del tiempo; por lo que siempre es ejecutada con la prioridad que se le asignó.

El rango de prioridades válido se encuentra entre 30 y 255. A los procesos interactivos normales, les son asignadas prioridades entre 40 y 127; por lo que se consideran como procesos privilegiados, aquéllos que les son establecidas prioridades en el rango de 30 a 39, procesándose con mayor rapidez. Por el contrario, se consideran como los peores, en términos de prioridad, a los que se encuentran entre 128 y 255; una aplicación en este rango, correrá muy frecuentemente sólo cuando no existan procesos con prioridad normal ejecutándose. Los rangos menores a 30 son reservados para el sistema.

El término de Tick (se refiere al tiempo que tarda un reloj en hacer tic-tac) se utiliza para referirse a la frecuencia básica del reloj del kernel, que es de 10 milisegundos. Este término se utiliza para establecer cuál será el período que permanecerá en ejecución dentro del CPU, una aplicación. Es decir, que si a un proceso se le establece un período de 3, permanecerá 30 milisegundos en ejecución, cada que sea cargado en el CPU.

Existe otro esquema para establecer la periodicidad con que se ejecutará un proceso, llamado periodos de procesamiento a plazos máximos (*periodic deadline*), en el cual, se establece y garantiza que por cada determinado lapso que transcurra, la aplicación será procesada un período establecido. Para habilitar este esquema se requieren dos parámetros: el primero que define el tamaño del período, y el segundo, que establece cuanto de este tiempo de CPU será dedicado a esta aplicación.

Como se mencionó, estas y otras características se le puede asignar a una aplicación al momento de ser creada, pero también pueden ser modificadas en tiempo real, al momento de estar ejecutándose. Para ello se emplean aplicaciones como *npri*, *nice*, *renice*, etc., que nos permiten controlarlos, y dar privilegios o negarlos a ciertas aplicaciones en específico. Claro está, que los usuarios únicamente pueden alterar, y en un rango definido específicamente para ellos, sus aplicaciones únicamente. El superusuario root puede modificar cualquiera, pero es conveniente que actúe con prudencia y no asigne altos grados de prioridades indiscriminadamente; ya que pudiera desbalancear las cargas de trabajo, ocasionando una degradación del rendimiento general del equipo, que puede concluir en una caída del sistema. Esto sucede cuando se asignan prioridades altas y fuera de los rangos recomendables, ocasionando que procesos de usuarios tengan mayores privilegios que los procesos vitales del sistema.

Es importante mencionar que el tiempo total de CPU consumido por una aplicación, no se ve afectado al bajar o incrementar su prioridad; es decir, el consumo del CPU sigue siendo el mismo. Lo que se afecta es el tiempo que se tarda en realizarla; ya que si se baja la prioridad de un proceso, éste será ejecutado con menor frecuencia, y por lo tanto, tardará más tiempo en terminar su labor.

II.III.V. Control

Por todo lo anterior, es importante mantener un control sobre los procesos que se encuentren ejecutándose en el sistema; por ello es conveniente estar monitoreándolos, y de ser necesario, actuar para solucionar cualquier problema. A este respecto, la columna de *PRI* que se obtiene al ejecutar el comando *ps -l* o *top*, muestra la prioridad de cada proceso; al igual que la de *%CPU* que indica el porcentaje de uso de CPU de cada aplicación. Con estos datos se puede determinar si existen aplicaciones con prioridades altas que estén consumiendo mucho del recurso del CPU, ocasionando lentitud en el sistema. Y con las herramientas como las mencionadas anteriormente y tratadas a continuación[15], aunadas al comando *kill*, se pueden modificar sus valores a fin de conservar en buen estado su rendimiento.

[15] Es recomendable referirse a la ayuda proporcionada por el comando *man*, para obtener una explicación más detallada de las diversas opciones que admiten cada uno.

nice

Ejecuta un comando a un prioridad baja. Éste debe ser empleado por usuarios, que conscientes de la importancia que tiene el recurso de CPU, deciden por ellos mismos ejecutar sus aplicaciones que no son importantes, a un baja prioridad, a fin de que aquéllas que sí lo sean, se procesen más rápidamente.

Este comando acepta un parámetro llamado *incremento*, que puede variar de 0 a 20; mientras más grande sea, es menor la prioridad a la que correrá la aplicación. Por default se establece el valor de 10. Ejemplo:

$ nice cc programa.c

Ejecuta el compilador de *c* a baja prioridad.

renice

Permite alterar la prioridad de un proceso en ejecución. Éste puede ser empleado para alterar uno en específico, un grupo de ellos, o todos los pertenecientes a un determinado usuario. El rango que se puede establecer va de 0 a 20, donde un proceso al que se le asignó el valor de 20 será ejecutado sólo cuando no exista otra aplicación que requiera ser procesada.

npri

Este comando permite alterar los parámetros de ejecución de una aplicación; o crear una nueva con ciertos valores establecidos. Como se mencionó, existen limites para los diversos parámetros, tanto para los usuarios normales como para root. Dentro de sus opciones más importantes destacan el de poder modificar o establecer la prioridad no degradable (*-h*), el período que permanecerá en ejecución dentro del CPU (*-t*) o el esquema periodos de procesamiento -deadlinne- (*-d*). Ejemplo:

 # npri -h 38 -t 3 -p 5682

Estableee para el proceso cuyo PID sea el 5682, una prioridad de 38 y un tiempo de permanencia en el CPU de 3 tick (30 milisegundos), cada vez que sea cargado en el CPU.

Multiprocesadores

En equipos que poseen más de un CPU, existen comandos como *runon* y *mpadmin* que permiten controlar su uso. Por ejemplo, el comando *runon* permite especificar en cuál de los diversos CPU será ejecutada una aplicación:

 # runon 2 programa1

Aquí se establece que la aplicación *programa1* será ejecutada en el CPU número 2. Por otro lado, el comando *mpadmin* permite restringir el uso de algún CPU en particular:

 # mpadmin -r1

Impide que aplicaciones ejecutadas con el comando *runon*, puedan ser corridas en el CPU 1.

II.IV. Soporte a usuarios

En este punto se presentarán labores tendientes a mantener las cuentas, que permiten a los usuarios ingresar al sistema; mantener un control de la información almacenada en el disco; separar los trabajos creados por una persona, del resto de los usuarios; mantener segura e íntegra la información; conocer qué es lo que está haciendo cada uno; controlar quién puede utilizar algún recurso, etc. Existen herramientas gráficas como la de *cpeople* que facilitan este proceso, haciéndolo consistente; pero es conveniente el conocer cuáles son los pasos básicos que deben seguirse para realizarlo de forma manual, ya que en las situaciones críticas, generalmente no se tiene acceso a esta herramienta, por ejemplo: cuando se accesa remotamente desde una terminal, cuando se reparan problemas desde miniroot, SO que no tiene activado el ambiente gráfico, etc.

II.IV.I. Creación de cuentas de acceso

Si una persona requiere accesar al equipo para realizar cualquier labor, debe tener una cuenta. Por las razones expresadas en el capítulo de Seguridad, es conveniente que las cuentas sean individuales, y evitar al máximo la existencias de cuentas compartidas; ya que pueden ser un punto de conflicto. Los pasos para crear una nueva cuenta son:

1. Recolectar los datos y características de trabajo que requiere la persona a la cual le será asignada.

2. Con los datos recolectados, añadir una nueva entrada en el archivo */etc/passwd*, para que sea reconocida por el SO.
3. Modificar el archivo */etc/group*, si es necesario, para especificar a qué grupos pertenecerá el usuario.
4. Establecer una clave secreta inicial para la cuenta.
5. Crear el directorio hogar del usuario.
6. Acondicionar su ambiente de trabajo.
7. Establecer los permisos, propiedades y grupo, del directorio y los archivos iniciales que pertenecerán al usuarios.
8. Dar a conocer al usuario, las políticas de uso establecidas en el equipo.

Paso 1

Los datos recolectados del primer punto, deben ser los necesarios para poder completar el resto del procedimiento; por lo que deben encontrarse entre ellos, dos grandes grupos: Datos personales, como el nombre, dirección y ubicación de la oficina, etc.; Características de la cuenta, como el proyecto y grupo de trabajo al que estará asignado, shell que se utilizará, herramientas a las que tendrá acceso, etc. Por ello, es favorable crear una forma de registro que deba ser llenada por el solicitante, a fin de mantener un control y el poder recabar esta información de manera más eficiente.

Paso 2

El SO lleva el control de las cuentas declaradas en el sistema mediante el archivo */etc/passwd*; por lo que para poder dar de alta una nueva, se debe agregar una entrada en este archivo como se indica en el punto dos. Su estructura es sencilla: se debe utilizar una línea por cada cuenta activa dentro del sistema; cada entrada se encuentra dividida en 7 campos separados por el carácter de dos puntos (:), como se muestra a continuación:

cuenta:clave_secreta:id_usuario:id_grupo:nombre_real:dir_hogar:shell

La *cuenta* es la que permite tener acceso al sistema; por lo que debe ser única. Generalmente se estila el poner el nombre del usuario o alguna combinación de su nombre o apellidos, pero se puede utilizar cualquier texto. Para que pueda ser portable a través de diversos SO, se recomienda el utilizar 8 caracteres alfanuméricos como máximo; donde el primero debe ser una letra. Es ampliamente recomendado utilizar letras minúsculas únicamente.

En el campo de *clave_secreta*, debe ser colocada ya cifrada[16], la clave secreta asignada a la cuenta. Por ello, es recomendable que cuando se cree una entrada, se deje este campo vacío, y posteriormente se utilice el comando *passwd* para asignarle una; lo que se ve reflejado en el punto cuatro.

En el campo de *id_usuario*, debe ser colocado un número que identifica al usuario ante el SO; es decir, el SO utiliza este identificador para reconocer al usuario durante todos los procesos que realice en el equipo. Este identificador está asociado a la cuenta, que permite de una manera más simple, que los usuarios se identifiquen mutuamente. En el capítulo de seguridad nuevamente, se tratarán a detalle características que deben tomarse en cuenta al elegir este parámetro; por el momento, es conveniente recordar que los identificadores válidos para asignar a los usuarios, son del 100 al 65535; como se puede observar de la Tabla 5. Además, es importante que no existan dos cuentas con el mismo identificador.

El *id_grupo* tiene una función similar; se utiliza para identificar los diversos grupos creados en el equipo. Un conjunto de cuentas puede pertenecer a un *grupo* y compartir ciertas características. De esta forma, se pueden controlar y delimitar las acciones de un cierto grupo de usuarios, de un solo tajo, en vez de hacerlo individualmente. Por ejemplo, se puede dar acceso de lectura, escritura, o ejecución de un archivo a un usuario, en base al grupo al cual pertenezca. Una cuenta puede ser miembro de varios grupos y poseer las ventajas de cada uno; pero cuando entra a sesión, por default es reconocida como miembro del grupo definido en este campo. Posteriormente puede cambiarse a otro mediante el comando *newgroup* o *multigrps*.

De manera análoga al identificador de usuario, al de grupo se le puede asociar un nombre para que pueda ser manejado de manera más sencilla por las personas; pero el sistema trabaja con el identificador de grupo numérico. Por ello, es importante que cada identificador de grupo tenga asociado un único nombre; el control de esto se lleva en el archivo */etc/group* (ver paso 3). Cuando se trabaja mediante la herramienta gráfica *cpeople*, asigna por default el valor 20, correspondiente al grupo *user* (ver Tabla 5).

El campo de *nombre_real*, se utiliza para colocar en él, el nombre completo del usuario. Tradicionalmente, este campo es conocido como GECOS (General Electric Company Operation System) y es utilizado para almacenar además del nombre completo del usuario, la dirección de su oficina, número telefónico de la oficina y de su hogar, separados por una coma. Esta información es leída y desplegada por el comando *finger*.

El campo *dir-hogar* especifica cuál será el directorio hogar de la cuenta; es decir, cuando entre a sesión éste será el directorio que se le asigne y dónde será colocado inicialmente. Por tal motivo, es conocido como directorio hogar, quedando registrado en la variable de ambiente $HOME, y se debe designar a la clave como dueño de dicho directorio.

[16] Ver La clave secreta o password, pág. IV-13

Finalmente el campo de shell se utiliza para designar cuál será el intérprete de comandos que atenderá a la cuenta cuando se entre a sesión. Por default se asigna al shell *Bourne* (*/bin/sh*), pero se puede especificar la ruta completa de cualquier otro.

Tabla 5 Identificadores reservados.

ID de Usuario	Usuario	ID de Grupo	Grupo
0	super usuario (root)	0	root
-2 (60001)	nobody-NFS	1 - 10	Demonios y servicios del sistema.
60002	noaccess	20	user (grupo default)
1 - 10	pseudo usuarios y demonios del sistema	995 -998	Grupos de usuarios especiales. (other, demos y guest)
11 - 99	reservados por el sistema para uucp y cuentas especiales	90 - 65535	Disponibles para la generación de nuevos grupos.
100 - 2147483647	A excepción de 60001 y 60002, son utilizados para usuarios normales. Ya que otros sistemas se encuentran limitados a 65535, es aconsejable no asignar identificadores superiores a este número.		

Cuando se desee crear una nueva clave se deben recabar estos datos y crear la entrada en el archivo */etc/passwd*. Esta acción puede ser realizada de dos formas: manualmente o con el comando *passmgmt*

Manualmente el archivo puede ser modificado con cualquier editor de texto, como el *vi* o *jot*, para añadir una nueva cuenta, eliminar alguna o modificar las características de las ya definidas. Se debe realizar con sumo cuidado para no afectar otras ya declaradas en él. En el caso de que se encuentre activada la seguridad de *shadow*[17], el archivo */etc/shadow* que contiene la clave secreta así como el control de vigencia de la cuenta, también debe ser modificado. Para ello, primero se actualiza el archivo */etc/passwd* y después se debe correr el comando *pwconv* que regenera y actualiza el archivo */etc/shadow*.

El comando *passmgmt* es muy seguro y los datos del usuarios le son proporcionados como parámetros. Éste permite realizar básicamente tres acciones: añadir usuarios mediante la

[17] Para mayores detalles, ver Archivo de seguridad shadow, pág. IV-26.

opción *-a*, modificar los datos de alguno con la opción *-m* y eliminar cuentas con la de *-d*. Por default asume:

- Un valor de **LK** para la clave secreta; por lo que se encuentra bloqueada y debe ser asignada posteriormente con el comando *passwd*.
- Como Identificador de Usuario, utiliza el número más próximo disponible a partir de 99; para lo cual, examina todo el archivo */etc/passwd*, garantizando que no exista duplicidad.
- Es asignado a la cuenta, el Identificador de Grupo 1; que pertenece al grupo *daemon*. Por ello, es importante especificar por lo menos este parámetro, y evitar que se asigne el default.
- Un campo de GECOS o *nombre_real* nulo.
- Es asignado el directorio */usr/people/cuenta*, como directorio hogar de la cuenta.
- Y finalmente asigna el Bourne shell (*/bin/sh*), como el intérprete de comandos.

Por lo tanto, el comando:

 # passmgmt -a luis

crea la entrada:

*luis:*LK*:107:1::/usr/people/luis:/bin/sh*

Se puede usar cualquiera de las opciones disponibles para anular los valores default y establecer los apropiados. Este comando trabaja modificando las entradas de los campos, tanto del archivo */etc/passwd*, como del */etc/shadow* si es que existe; por lo que no es necesario ejecutar posteriormente el comando *pwconv*. Cuando realiza modificaciones en ellos, genera los archivo */etc/opasswd* y */etc/oshadow*, que contienen las versiones pasadas de cada uno respectivamente.

Paso 3

El siguiente punto consiste en modificar el archivo */etc/group*, que define los grupos y los usuarios que corresponden a cada uno, para especificar a cuáles pertenecerá la cuenta que se está creando; si es necesario, se puede generar uno nuevo para incluir al usuario. Este archivo puede ser modificado con cualquier editor de textos, y el formato de éste será examinado con mayor detalle en 'Grupos de usuarios', pág. IV-28. Si el usuario pertenecerá únicamente al grupo definido en el archivo */etc/passwd*, no es necesario agregarlo al campo de dicho grupo en el archivo */etc/group*, ya que se supone por default, que pertenece a él. Este archivo sólo debe ser modificado cuando el usuario vaya a formar parte de varios grupos, y se debe añadir la cuenta en el campo de la membresia de los grupos adicionales únicamente.

Paso 4

Una vez creada la cuenta, es aconsejable asignarle una clave secreta temporal a fin de darle seguridad; posteriormente su dueño puede modificarla a su gusto[18], mediante el comando *passwd*. Si no se asigna una, se recomienda bloquearla hasta que sea otorgada y utilizada por el dueño; cuya primera acción, deberá ser el colocarle la clave secreta. Con esto se deduce, que nunca se debe dejar sin clave secreta una cuenta, ya que puede representar un punto vulnerable en la seguridad.

Paso 5

En este paso se debe crear el directorio hogar mediante el comando *mkdir*; ya que en el paso dos, únicamente se definió cual sería, pero no se creó. En IRIX se estila que los directorios de los usuarios sean colocados bajo el directorio */etc/people*, por lo que es conveniente seguir con esta política para mantener una consistencia a través de los diversos equipos con este sistema; aunque se pueden ubicar en cualquier otro sitio. En otros sistemas, como el SVR4, se estila colocar el directorio hogar de los usuarios bajo */home* o */d*, y en otros más, bajo */users*. No importa donde sea, pero se debe tener congruencia y elegir un punto, para facilitar labores de mantenimiento como el respaldo de la información, monitores y control del espacio consumido por los usuarios, etc.

Paso 6

Éste se refiere a acondicionar el ambiente de trabajo, que será tratado más adelante, y que básicamente consiste en colocar una copia modificada adecuadamente de los archivos */etc/stdlogin*, */etc/stdcshrc* y */etc/stdprofile* en el directorio hogar de la cuenta, con el nombre apropiado.

 # cp /etc/stdprofile $HOME/.profile
 # cp /etc/stdlogin $HOME/.login
 # cp /etc/stdcshrc $HOME/.cshrc

Paso 7

Consiste en asignar los permisos adecuados a los archivos que inicialmente le pertenecerán al usuario. Este problema se suele presentar, ya que los directorios y archivos colocados dentro de ellos para configurar la cuenta del usuario, suelen ser creados desde root, y por tanto, pertenecen a root y no al usuario. Por ello, se debe utilizar el comando *chown* y *chgrp* con la opción *-R* para designar como dueño de todos ellos, a la cuenta que se está creando y

[18] Recordar que la clave secreta debe cumplir ciertas condiciones para evitar que pueda ser descifrada. Ver pág. IV-13.

el grupo adecuado respectivamente. El comando *chmod* debe ser utilizado para asignar los permisos de lectura, escritura y ejecución adecuados, a cada archivo; generalmente suelen ser *766*.

Paso 8

El último paso consiste en dar a conocer al nuevo usuario, su cuenta y clave secreta inicial; para que pueda tener acceso al sistema. Es sumamente importante, el informarle de las políticas de uso vigentes del equipo, para que tome las medidas pertinentes y no incurra en violaciones que puedan perjudicar, tanto la seguridad como información del sistema y los demás usuarios.

II.IV.II. Eliminación de cuentas

El eliminar una cuenta es un proceso sencillo, pero que requiere atención para evitar que en el futuro se presenten problemas. Los pasos para ello son:

1. Respaldar la información del usuario.
2. Decidir si se desea bloquear o suspender la cuenta.
3. Si se eligió suspenderla, proceder a ello.
4. Si se eligió eliminarla:
    - Eliminar la entrada del archivo */etc/passwd* y */etc/shadow*.
    - Eliminar la cuenta de listas, como la de correo, de grupo, etc.
    - Eliminar los archivos pertenecientes a la cuenta.

Paso 1

Como ya se ha mencionado, antes de realizar cualquier labor que pueda ser destructiva es conveniente realizar un respaldo; por ello, ya sea que se decida eliminar definitivamente la cuenta, o que se vaya a suspender durante un tiempo, es conveniente efectuar un respaldo que garantice que dicha información podrá ser recuperada posteriormente. No se deben emplear las cintas normales de respaldo, ya que la información se perderá al ser reutilizadas en ocasiones posteriores (ver Estrategias de respaldo, pág. V-53); por ello, es aconsejable utilizar una cinta para esta labor, que contenga los archivos de las cuentas que han sido eliminadas del sistema únicamente. Las posibles herramientas utilizadas para esta labor, se describirán en 'Herramientas de respaldo', pág. V-57.

Paso 2

Es importante analizar la conveniencia de suspender o eliminar la cuenta. Si el dueño interrumpirá sus labores por algún tiempo, pero volverá a requerir de los servicios, es más práctico y menos problemático el bloquearla únicamente; dejando su información intacta. De esta forma, cuando regrese sólo se tendrá que desbloquear, pudiendo hacer uso de ella inmediatamente y sin ningún problema. Por el contrario, si abandonará su trabajo definitivamente, se puede eliminar, liberando espacio en disco y entradas dentro en los archivos */etc/passwd*; haciendo más rápida la búsqueda de cuentas durante el proceso de entrada. Si se elige este método, es conveniente considerar los aspectos tratados en el capítulo de seguridad, que indican que un Identificador de Usuario ya asignado a una cuenta, no debe ser reasignado; claro está, que todo va en función de los requerimientos y políticas establecidas.

En ocasiones, es aconsejable establecer un punto intermedio en esta decisión; sobre todo si no se sabe con certeza si se volverá a requerir dicha cuenta o no. En este caso se puede proceder a bloquearla, y si pasado algún tiempo razonable no fue solicitada nuevamente, se puede proceder a eliminarla definitivamente; sin olvidarse de realizar los respaldos correspondientes.

Paso 3

Para bloquear una cuenta se debe editar el archivo */etc/passwd* y colocar un asterisco (*) en el campo de la clave secreta; también, se puede colocar una frase entre asteriscos que indique cuál es su situación como:

Dada de Baja

Esto mismo se puede hacer en el archivo */etc/shadow*. Como medida adicional de seguridad, se puede proceder a comentar la entrada; es decir, colocar el signo de # al inicio de la línea que la define en estos archivos. Con ello se garantiza que nadie podrá ingresar al sistema utilizándola. Esta función puede ser llevada a cabo mediante el comando:

 # passwd -l cuenta

Si el usuario vuelve a requerir de dicha cuenta, bastará con descomentar la línea y asignarle una nueva clave secreta para activarla.

Paso 4

En el caso que se desee eliminarla, se puede: ya sea, mantenerla bloqueada indefinidamente, siguiendo el procedimiento descrito en el punto anterior, o borrarla físicamente.

Para borrarla, como primer punto se tiene que remover la línea que la define con la ayuda de algún editor de textos, tanto en el archivo */etc/passwd*, como en */ect/shadow* en el caso de que exista. Otra alternativa es el empleo del siguiente comando:

 # passmgmt -d cuenta

El segundo punto es un poco laborioso, y frecuentemente suele olvidarse. Dentro del equipo suelen existir sistemas que, para su funcionamiento, requieren de listas de usuarios; tal es el caso del mecanismo de grupos empleados por el SO, y el de correo. En el primero, un usuario puede ser miembro de varios grupos, para lo cual, su cuenta debe ser incluida en la definición de cada grupo, dentro del archivo */etc/group*. Por ello, cuando es eliminada la cuenta, ésta debe ser removida de ese lugar; evitando posibles problemas posteriores. Por ejemplo: una lista de grupo tiene cierto limite, y si se colocan más miembros de los permitidos, puede causar conflictos; por ello, todas las cuentas eliminadas del sistema deben ser removidas de los grupos de este archivo, evitando así, su saturación.

En el mismo caso se encuentra el sistema de correo, que emplean listas de distribución llamadas *alias*; de tal forma que cuando se desea enviar un mensaje a un grupo de usuarios, se envía al *alias* que se les asignó, recibiéndolo todos los miembros. Esto es más simple y menos tedioso que el enviarle el mensaje a cada uno. Las listas de correo suelen ser mantenidas en un archivo llamado *aliases*, que en IRIX, se encuentra ubicado dentro del directorio */etc*. Por ello, cuando una cuenta es removida, debe ser eliminada su entrada de todo *alias* al que pertenezca; evitando que le sean enviados más mensajes en el futuro, lo que ocasionaría errores en el sistema de correo al tratar de distribuir los mensajes en cuentas que ya no existen. Para modificar y eliminar las cuentas en cualquiera de los dos casos anteriores, se puede utilizar cualquier editor de texto; con la recomendación de actuar con cuidado para evitar alterar por error otras.

De forma análoga a los posibles usos que se le da al nombre de las cuentas, se debe tener precaución con cualquier dato relacionado con ella; como suele ser la exportación de directorios, implementada mediante el mecanismo conocido como NFS (Network File System). En este sistema se definen qué directorios, que suelen ser los de los usuarios, se podrán exportar, y por tanto, importar desde otro equipo para ser utilizados de forma remota. Este sistema emplea el archivo */etc/export* para indicar qué directorios pueden ser exportados, por lo que si la cuenta que se eliminó se encontraba definida dentro de éste, debe ser eliminada para evitar posteriores problemas al tratar de montar un directorio que no existe.

Esto mismo se debe tener en cuenta para toda aplicación que requiera el ingreso de datos referentes a cuentas para su funcionamiento; por ello, es muy importante documentar, tanto el proceso de creación, para añadir la cuenta en los diversos archivos de los sistemas implementados, como en el de eliminación, para removerla de ellos.

El último punto consiste en borrar del disco cualquier archivo o directorio que haya pertenecido al usuario. En este aspecto, se suele cometer el error de pensar que todos los archivos de un usuario suelen estar bajo su directorio hogar, lo que realmente sucede en ocasiones, pero no siempre es así. Dependiendo de las herramientas que haya utilizado el usuario durante su estancia, pueden haberse generado diversos archivos que les pertenezcan y que se encuentran distribuidos a lo largo de todo el sistema de archivos; como en el caso de el correo, que crea un archivo con el nombre de la cuenta en el directorio */usr/mail* para almacenar los mensajes que le son enviados al usuario. Otro ejemplo son algunas herramientas como *vi*, que suelen emplear archivos temporales durante su funcionamiento, creándolos en directorios destinados para ello, como el */tmp* y */usr/tmp*. Aunque se supone que la información almacenada en estos directorios no es vital y puede ser eliminada, si esto no se hace los archivos que el usuario haya creado permanecerán ahí[19], ocupando espacio. Por ello, es conveniente efectuar una búsqueda sobre todo el SA, para localizar cualquiera que le haya pertenecido a la cuenta y eliminarlos. Esta búsqueda también debe ser efectuada al momento de realizar el respaldo de la información de la cuenta, y se puede emplear una estructura que envuelva el comando *find*, como la siguiente:

```
# find / -user juan -print -exec rm {} \;
```

Ya que el comando *find* anterior realiza un uso intensivo de acceso a disco, es aconsejable efectuarlo en horarios no pico, y finalmente, a fin de no omitir algún punto, es importante el documentar este proceso.

II.IV.III. Control del ambiente de trabajo

Cuando un usuario entra a sesión, le es asignado un shell que interpretará las ordenes que él dé y se las enviará al CPU. De esta forma, el usuario podrá lanzar nuevos procesos para realizar sus tareas. En el merado existen gran variedad de programas que realizan estas funciones, pero en particular, mencionaremos tres, por ser ampliamente difundidos y encontrarse en todos los sistemas operativos UNIX: el Bourne, el C y el Korn. Cada uno tiene ciertas características, que el usuario debe evaluar para decidir con cuál trabajará. Dentro de IRIX, éstos se encuentran en el directorio */bin*, y llevan por nombre *sh*, *csh* y *ksh* respectivamente.

Independientemente cuál de ellos se elija, se debe configurar el ambiente de trabajo que permita efectuar las labores con mayor facilidad. Para ello, se pueden definir y establecer variables que serán utilizadas por los diversos programas ejecutados durante la sesión, y que

[19] Por default el sistema se encuentra configurado para eliminar todos los archivos contenidos en el directorio */tmp* cada vez que se inicie el equipo; lo cual no sucede con */usr/tmp*.

sin ellas, pueden no funcionar, o hacerlo con deficiencia. Dentro de este caso se encuentran variables como:

HOME — Que establece, cuál es el directorio hogar de la cuenta. Con ella, al ejecutar el comando *cd*, éste ya sabrá cuál es el directorio hogar y podrá cambiarse a él.

PATH — Que contiene una lista de directorios, los cuales, serán revisados consecutivamente, en busca de los archivos y comandos que el usuario trate de ejecutar. Si un programa no se encuentra dentro de esta ruta de búsqueda, se debe especificar en forma completa su ubicación; es decir algo como : */usr/bin /X/xterm*, para poder ejecutarlo.

TERM — Que establece el tipo de terminal que se está empleando; con la información que guarda esta variable, aplicaciones como *vi* pueden buscar en una base de datos de terminales, sus caracteristicas en especifico, y así poder desplegar y trabajar adecuadamente en la edición de archivos. Si no se establece la variable, resulta imposible trabajar con vi; ya que no sabrá cómo controlar la pantalla que se está empleando. Esto mismo puede suceder con aplicaciones similares que requieren organizar lo que se despliega en la pantalla.

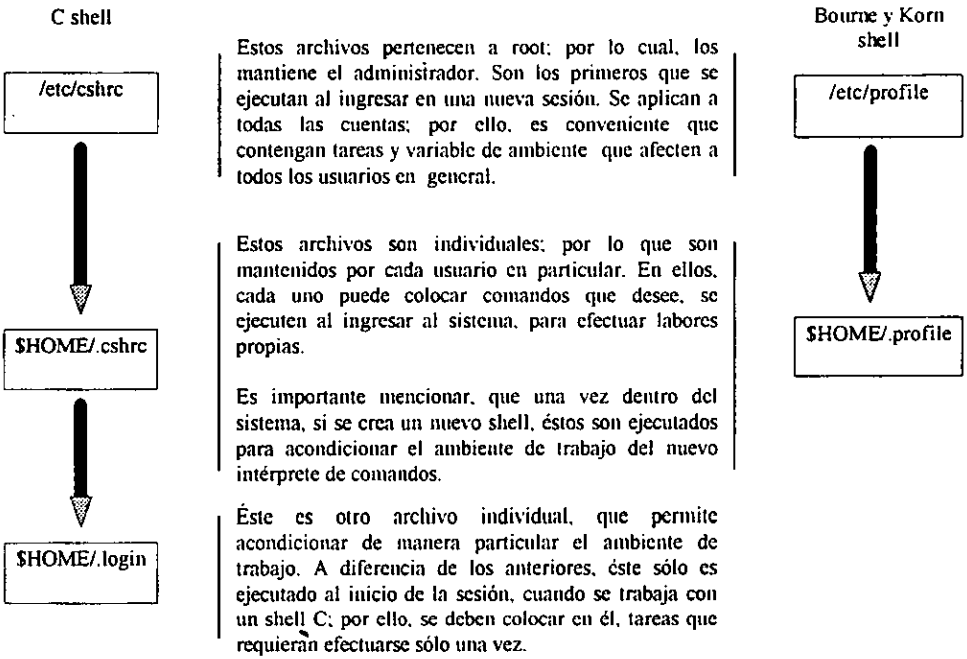
DISPLAY — Que define cuál es la dirección de la pantalla gráfica donde se está trabajando. Cuando se ejecuta una aplicación gráfica, checa esta variable para determinar en qué pantalla será desplegada. Sin ella, no se podrá ejecutar ninguna aplicación de este tipo.

De igual manera, existen labores rutinarias que deben ejecutarse al ingresar al sistema como:

cat /etc/motd	Checar y desplegar en la pantalla, el mensaje del día; si es que existe.
umask 022	El definir cuáles serán los permisos que se otorguen automáticamente a los archivos y directorios que se creen. Esto evitará la tarea de ejecutar el comando *chmod* para hacerlo manualmente.
set filec	Establece un mecanismo mediante el cual, después de escribir una secuencia de caracteres y presionar la tecla *esc*, el C shell busca en el directorio actual un archivo que empiece con las mismas letras, y si lo encuentra, completa su nombre. Esto evita el tener que teclear nombres largos; bastará con escribir unos cuantos caracteres y posteriormente presionar la tecla *esc* para que sea completado.
If /bin/mail -e then echo "existe correo" fi	Checar si existe correo en el buzón, y de caso afirmativo, enviar un mensaje indicándolo.

Así como éstas, existen otras funciones que deben ser efectuadas cada vez que se entra a sesión; algunas son necesarias, otras son por conveniencia. En general, cada vez que se adquiere e instala una nueva aplicación, ésta viene acompañada con un instructivo que indica cuáles son las medidas que deben tomarse para que funcione correctamente; como el establecer variables de ambiente particulares, realizar ciertas labores antes de ejecutarla, etc. En resumen, al conjunto de labores que se efectúan al ingresar al sistema para acondicionar la sesión, se le llama "configuración del ambiente de trabajo".

Los diversos programas shell mencionados anteriormente, cuentan con mecanismos que permiten llevar a cabo estas labores automáticamente, y se encuentran implementados mediante una serie de archivos, o programas shell, que se ejecutan dependiendo del shell que se utilice y en el siguiente orden.



La información contenida en los archivos *cshrc* y *profile*, es mantenida por el administrador y puede ser modificada en cualquier momento, entrando en vigor inmediatamente. En cambio, la mantenida en los archivos *.cshrc*, *.profile* y *.login*, inicialmente es una copia hecha por el administrador, de los archivos */etc/stdcshrc*, */etc/stdprofile* y */etc/stdlogin*

respectivamente, al directorio del usuario, para que éste pueda acondicionar de forma individual su ambiente de trabajo. Por ello, cuando el administrador desee efectuar un cambio en éstos, debe actualizar primero los archivos colocados en el directorio */etc*, a fin de que las nuevas cuentas ya los incluyan, y a continuación, cada una de las copias colocadas en los directorios de cada una de las cuentas existentes; a fin de que el cambio también las involucre.

Así como estos casos, existen diversas herramientas que pueden generar archivos de configuración individuales, y que generalmente son colocados dentro del directorio de cada usuario que utiliza dicha aplicación. Se suele estandarizar que los archivos de configuración empiecen con un punto; ya que este tipo de archivos se consideran como ocultos y no son desplegados al ejecutar el comando *ls*. Para verlos se debe añadir la opción *-a* (*all*), para que todos los archivos sean desplegados; incluso los ocultos. Por esto, es conveniente leer cuidadosamente la documentación que acompaña a cada aplicación, especialmente lo relacionado a la configuración, a fin de tomar las medidas pertinentes para acondicionar las cuentas a las necesidades propias de cada usuario, y lograr así, que pueda operar sin ningún problema.

II.IV.IV. Comunicación con usuarios

Es importante el mantener una estrecha comunicación con los usuarios, para brindar un mejor servicio; manteniéndolos al tanto de fechas y horarios de suspensión de servicios, eventos importantes, nuevas aplicaciones instaladas que faciliten sus trabajos, problemas existentes que deben ser evitados, etc.. Para todo ello existen una cantidad de comandos y herramientas que se pueden disponer, las cuales serán tratadas a continuación.

En primer término se encuentran los comandos como *wall*, que permite enviar un mensaje a la terminal de todas las cuentas que se encuentren en sesión. Esta herramienta es muy útil para enviar mensajes de último momento; como cuando se desea dar de baja el sistema por cualquier razón. Se puede mandar uno a los usuarios conectados al equipo, indicándoles que debe cerrar sus aplicaciones y salir de sesión; para ello, se puede teclear lo siguiente:

 # wall

El sistema será dado de baja dentro de 15 minutos por labores de mantenimiento.
El servicio se reanudará a partir de las 12:30 hrs. nuevamente.

Atte.
Administrador
^d

En este renglón, es conveniente que una vez que se ha lanzado este mensaje, se active un mecanismo que impida que otros usuarios ingresen al sistema y empiecen a hacer uso de él, sin saber que será dado de baja. Para esto, se puede emplear el archivo */etc/nologin*. Durante el proceso normal de conexión que se sigue, varios servicios como el de *telnet* y *ftp*, verifican la existencia de este archivo y si es localizado, niegan sus servicios y despliegan en la pantalla el texto que contenga. Por ello, si se crea y coloca dentro de él un mensaje, lo siguiente sucederá al tratar de conectarse desde cualquier sitio mediante el comando *telnet*:

$ telnet servidor
Trying 132.348.15.55...
Connected to servidor.
Escape character is '^]'.

IRIX System V.4 (servidor.elr.com.mx)

El sistema se encuentra suspendido por labores de mantenimiento.
Se reanudará a partir de las 12:30 hrs. Nuevamente.

Mensaje incluido dentro del archivo */etc/nologin*.

Atte.
Administrador

Connection closed by foreign host.
$

Otro mecanismo de información valioso, es el del 'mensaje del día', que es implementado por el archivo */etc/motd*. Cualquier anuncio que se desee dar al usuario, puede ser colocado dentro de este archivo; cuya información es desplegada en la pantalla del usuarios cuando ingrese al sistema. Esta labor es desempeñada a través de los archivos */etc/profile* o */etc/cshrc* dependiendo del shell que se esté utilizando. Es conveniente colocar la información de una forma breve y sencilla; ya que si ésta es basta, cuando es desplegada en la pantalla pasa rápidamente y se perderá gran parte de ella sin que el usuario la pueda leer. Por ello, el mensaje máximo recomendado es de una pantalla de largo. También, no se aconseja sobre utilizar este servicio y dar mantenimiento a la información contenida en él; ya que es desplegada cada vez que el usuario entra a sesión, y se ha demostrado que cuando no se respeta esto, el usuario tiende a ignorarla; no cumpliéndose el objetivo para el cual fue creado el mecanismo, dar información al usuario de cualquier evento que vaya a ocurrir en el transcurso del día.

Siguiendo esta tendencia, se puede crear el archivo */etc/issue*, el cual despliega un mensaje cuando un usuario se conecta de forma remota y antes de que aparezca la palabra de *login*, solicitando su clave para permitirle el acceso. Este archivo suele ser utilizado para colocar el nombre y un breve resumen de las políticas de uso más importantes del equipo. También,

este servicio no debe ser sobre saturado con datos irrelevantes y que degraden los beneficios que pueda brindar.

Los archivos anteriores funcionan en su conjunto de la siguiente forma:

- Cuando un usuario ejecuta el comando *telnet* lanza una petición al equipo remoto; el cual lanza un demonio que se encarga de atender la llamada (*telnetd*). Éste busca el archivo */etc/nologin*; si lo encuentra despliega su contenido y cancela la conexión.

- Si el archivo no existe, empieza el proceso de conexión, dentro del cual, se busca el archivo */etc/issue* cuyo contenido es desplegado en la pantalla. Posteriormente aparece el mensaje de *login* y *passwd* solicitando la cuenta y clave secreta para poder ingresar al sistema.

- Si el usuario da estos datos correctamente, ingresa al sistema, por lo cual se empieza a configurar su cuenta y se corren los programas de */etc/profile* y *$HOME/.profile* o */etc/cshrc*, *$HOME/.cshrc* y *$HOME/.login* dependiendo del shell que se utilice.

- Dentro de las labores que desempeñan estos programas, se encuentra el buscar el archivo */etc/motd*; si se localiza, su contenido es desplegado en la pantalla.

- Posteriormente, aparece el *prompt* del sistema y el usuario puede empezar a trabajar.

Otro mecanismo de información, puede ser el envío de mensaje a través del correo; para ello, se puede utilizar el comando *mail*, o cualquier otro.

Una herramienta más, diseñada para la comunicación de mensajes, es la de *news*; que permite implementar de forma simple, lo que suele llamarse un tablero de boletines electrónico. Para ello, por cada mensaje que se desee anunciar en él, se debe generar un archivo. El nombre de éste debe ser congruente con el mensaje y ser colocado dentro del directorio */usr/news*; con ello quedará publicado. Por su parte, cada usuario puede ejecutar el comando *news* para listar el número de mensajes publicados; listar el contenido de este directorio, lo que nos indicará los títulos de los mensajes; listar cada artículo; eliminar los que no sean de interés o todos los anuncios.

Para determinar qué mensajes son nuevos, el comando *news* utiliza el archivo *$HOME/.news_time*. Se considera que un anuncio es nuevo cuando la fecha de creación del archivo que contiene el artículo, es más reciente que la fecha de creación del archivo *.new_time*. De esta forma cuando se utiliza el comando *news* y se leen los mensajes, se crea este archivo y se podrá separar fácilmente los nuevos anuncios que incluya el administrador. Cabe destacar que los anuncios permanecen publicados hasta que sean removidos del directorio, por lo que debe dársele mantenimiento constante.

Finalmente, ¿qué herramienta elegir para comunicarse con los usuarios?, es decisión del administrador; aunque por la diversidad en las características de los usuarios, puede ser necesario el implementar todas ellas; claro está, que con cierta medida. Por ejemplo, no todos los usuarios se conectan en forma remota al equipo mediante un *telnet*; unos lo pueden hacer directamente desde la consola, otros desde terminales conectadas al equipo y otros más a través del ambiente gráfico XDM, que no utilizan todos los mecanismos mencionados anteriormente. No todos los usuarios conocen o acostumbran leer sus mensajes de correo. No todos ellos tienen el tiempo suficiente como para detenerse a leer mensajes largos o cada uno de los publicados dentro del servicio *news*, etc. Por ello, la diversidad de herramientas que pueden satisfacer la gran variedad de demandas; pero no se debe abusar en ningún caso.

CAPÍTULO 3

ADMINISTRACIÓN AVANZADA

Aspectos internos respecto a la estructura que guardan los discos, así como temas avanzados en el manejo del SO.

III. ADMINISTRACIÓN AVANZADA

En este capítulo se tratarán aspectos más profundos acerca del sistema operativo IRIX; aspectos que generalmente no se utilizan, pero que son indispensables que un administrador conozca, ya que se pueden presentar ocasiones en las que se dañe el SO o se tengan que realizar actualizaciones o expansiones de dispositivos, y la comprensión de estos conceptos permitirá realizar las tareas con mayor seguridad, y por ende, más rápido y confiable.

Los aspectos tratados aquí comprenden el estudio de lo relacionado al almacenamiento de la información, como la estructura que guarda el disco; procedimientos para recuperarse después de una caída del sistema; la afinación del SO; así como la configuración de diversos servicios.

III.I. Unidades de almacenamiento

Los discos duros representan el principal medio de almacenamiento de información, y todo equipo debe contar por lo menos con uno; el cual contiene el software necesario para arrancar el equipo, así como el SO y la invaluable información de los usuarios.

El mantener un disco en óptimas condiciones es vital para el equipo y los problemas más serios ocurren cuando la información almacenada en ellos, se daña. Es por esto que el instalarlos, configurarlos y mantenerlos, es una de las funciones más cruciales del administrador. Lo expuesto en esta sección, permitirá comprender la importancia de estos dispositivos, así como la forma en que están organizados.

En el mercado existen una gran variedad de discos, pero para que cada uno funcione, se requiere un dispositivo llamado controlador; que es quien conoce todas las características de un cierto tipo de discos y se encarga de controlarlos y organizar los datos que se almacenan en ellos. Algunos pueden ser tarjetas individuales que se instalan en las ranuras de expansión ubicadas en la parte posterior del CPU, o venir ya integrados a la tarjeta principal. La estación de trabajo Indy cuenta con 2 ranuras de expansión GIO[20] únicamente, pero el SO IRIX puede soportar 5 tipos de controladores: SMD, ESDI SCSI, JAG e IPI, de los cuales, el SCSI es el elegido como default; ya que viene interconstruido en la tarjeta principal del equipo.

Cada controlador tiene ciertas características que lo definen, y por ende, a los discos que pueden conectarse a él.

[20] Graphics Input/Output. Son tarjetas que se conectan a las ranuras de expansión sobre la tarjeta gráfica del equipo.

Tabla 6 Tipos de tarjetas controladoras.

Controlador	Discos por Controladora	Promedio de Búsqueda* (Average Seek)	Velocidad de transferencia (Transfer Rate)**
SCSI	7	16 ms.	0.5 - 2.5 MB/seg.
ESDI	4	16 ms.	0.5 - 2.5 MB/seg.
SMD	4	16 ms.	0.9 - 1.7 MB/seg.
JAG	14	16 ms.	2.0 - 2.1 MB/seg.
IPI	16	16 ms.	3.6 MB/seg.

* Puede variar dependiendo del disco.

** Indica una transferencia de datos sostenida del sistema de archivos para lecturas secuenciales.

Sea cual sea el tipo de disco que se utilice, existen ciertos puntos de interés que son comunes a todos ellos, ya que los establece el SO IRIX, y que se deben entender:

- La estructura que guarda el disco.
- La estructura del sistema de archivos.
- La forma de accesar la información.

Por este motivo, se detallarán estos aspectos, y al final, se describirán los procedimientos más importantes que se efectúan sobre los discos; como el añadir uno, incrementar el área de swap, reparticionarlos, y su mantenimiento en general.

III.I.I. Estructura de los discos en IRIX

Físicamente un disco duro está compuesto por varios platos que giran sobre un eje - el número de platos puede variar de disco en disco. La información es grabada en las dos superficies de cada uno de los platos; para ello se utilizan las cabezas de lectura y escritura, que como se puede observar de la Fig. III-1 , se desplazan hacia el centro o el extremo de los platos, y de esta forma, cubren toda el área disponible con información.

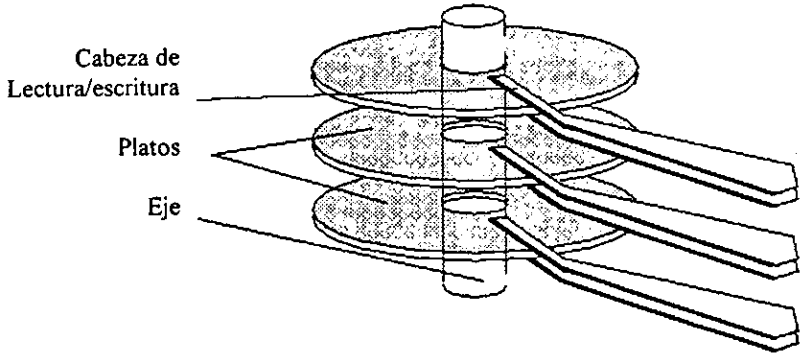


Fig. III-1 Estructura interna de un Disco Duro.

Ahora, para cada plato, la ruta que genera la cabeza sobre la superficie, al girar el disco y mantenerse estable la cabeza, es denominada una pista. Cada pista se encuentra dividida en segmentos llamados sectores, que son las unidades básicas de almacenamiento. Cada sector, también llamado bloque básico, está compuesto por 512 bytes de largo.

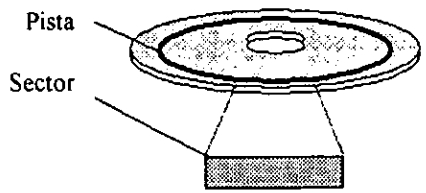


Fig. III-2 Estructura de un plato

Al conjunto de todas las pista que generan las cabezas sobre todas las superficies de los platos, se le llama cilindro; y a un conjunto de cilindros contiguos, se le llama grupo de cilindros.

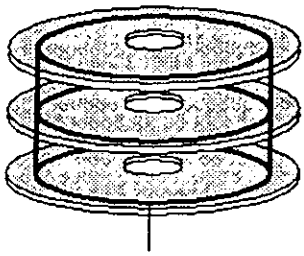


Fig. III-3 Estructura de un cilindro.

Estos conceptos son importantes (sector, cilindro y grupo de cilindros) ya que son las unidades de medida que se utilizarán más adelante.

Por otro lado, el espacio del disco puede ser utilizado para diferentes propósitos, por lo que resulta práctico y deseable, el poder dividir el disco en unidades que puedan funcionar de forma diferente e independiente. A estas unidades se les llama particiones, y el SO las puede reconocer y las trata de forma diferente. Una partición es un conjunto de grupos de cilindros contiguos e IRIX, divide los discos en particiones para operar con ellos.

IRIX puede dividir un disco en un máximo de 16 particiones, numerándolas del 0 a la 15. También, utiliza tres esquemas fundamentales para dividir el disco, llamados:

rootdrive	estilo actual para un disco principal.
usrrootdrive	estilo antiguo para un disco principal.
optiondrive	estilo utilizado para discos opcionales.

Cada uno de estos estilos divide el disco de una forma predeterminada, como se verá adelante, y son las que se utilizan generalmente en estos equipos; además de que el sistema viene configurado para reconocerlas sin mayores complicaciones. Si se desea, también se puede dividir el disco de cualquier forma, lo único que se tiene que hacer, es crear los archivos especiales de dispositivos[21] para cada partición y montarlos manualmente. Independientemente de ello, las particiones en IRIX pueden contener los siguientes tipos de datos:

- **EFS** (Extent FileSystem), Sistema de archivo extendido.
- **lvol**, Datos del volumen lógico.
- **raw** (swap), utilizado como memoria de intercambio.
- **volhdr** (Volume Header), Encabezado del volumen del disco.

[21] Los archivos de dispositivos (device files) son el medio que utiliza el SO UNIX para comunicarse con cualquier dispositivo periférico del equipo. Se describirán más adelante.

Las particiones tipo EFS son utilizadas para almacenar los archivos que forman el SO, así como los datos, programas, textos, directorios, etc. que generan los usuarios. En la pág. III-14 se detallará la estructura que guarda la información en este tipo de particiones.

Las particiones tipo lvol son utilizadas para crear volúmenes lógicos; esto es lo contrario a dividir un disco. Se pueden definir varias particiones ubicadas en uno o varios discos como si fueran una sola, a lo que se llama volumen lógico. De esta forma, al grabar la información el sistema se encarga de distribuirla en los distintos discos o particiones que forman al volumen lógico de manera automática.

Las particiones tipo raw son utilizadas por el sistema como área de intercambio o mejor conocido como swap. Esta área la utiliza el sistema como memoria virtual y coloca en ella, información que se encuentra en memoria RAM y que no se ocupa en ese momento; ya sea porque espera que ocurra un evento, o porque requiere que se le proporcionen datos para continuar su proceso. Posteriormente, cuando las condiciones se logran y el proceso requiere continuar, es leída la información de esta área, nuevamente a la memoria RAM. Esto permite, entre otras cosas, que se puedan ejecutar sin mayores contratiempos, programas que son más grandes que la memoria RAM disponible.

Finalmente, la partición tipo volhdr es utilizada para almacenar información del disco, así como los programas que permiten arrancar el sistema y algunos de mantenimiento que funcionan en modo standalone. Es una partición pequeña que generalmente ocupa 1 Mb. y contiene, como ya se mencionó, dos cosas: una etiqueta del disco y archivos de programas ejecutables. La etiqueta está compuesta por la siguiente información:

- Un número que la identifica como etiqueta del disco.
- Si se trata de un disco principal, contiene un campo que indica cuál partición deberá ser usada como raíz del sistema de archivos, root (generalmente indica la partición 0).
- Un campo que identifica qué partición debe ser usada como área de swap (generalmente es la 1).
- Un campo el cual contiene el nombre del archivo que será cargado al inicio; el del sistema operativo (generalmente contiene */unix*).
- Una sección la cual contiene información relativa al disco; como número de cilindros, cabezas, platos, etc. Esta información es leída al inicio para configurar el controlador del disco.
- Un directorio, el cual contiene el nombre y ubicación de los archivos o programas almacenados en el volhdr.
- Y finalmente, información que indica la forma en que se encuentra dividido el disco; es decir, el sector en el que empieza cada partición así como la longitud y tipo de cada una.

Además de la etiqueta, el volhdr contiene programas que se utilizan para recuperarse cuando han ocurrido problemas en el disco y no se puede arrancar en forma normal. Estos programas son copiados a esta área al momento de instalar el SO o cuando se da formato al disco con el comando *fx*. También se cuenta con el comando *dvhtool* que permite copiar cualquier archivo a esta área; debe recordarse que los programas que se deseen colocar ahí, deben haber sido diseñados para trabajar en forma independiente, es decir, en modo standalone. Así también, ya que el espacio reservado a el volhdr es pequeño, debe optimizarse y colocar únicamente las aplicaciones que puedan ayudar a solucionar los posibles problemas que se presenten; si no han sido colocados ya.

Entre las herramientas que se colocan por default en esta área, está la de *sash*, ya que nos permite montar el SO o a miniroot; pero también es buena idea colocar a *fx* en ella, ya que es el que permite dar formato y particionar un disco, como se verá más adelante. Finalmente, antes de describir los tipos de particiones definidas por SGI, se describirá la utilería *fx*.

III.I.I.I. fx

Es *fx*, la herramienta utilizada para dar formato y particionar los discos. Existen 2 versiones; una es utilizada directamente sobre el SO y la segunda, ubicada en */stand/fx*, es una versión standalone que puede arrancar desde el PROM. Cuando se adquiere un disco, puede ser que ya haya sido particionado por el distribuidor y se encuentre listo para ser instalado; pero también, puede ser que se encuentre sin formato, y en tal caso, se debe utilizar esta herramienta para prepararlo y poder utilizarlo.

En este sentido, existen dos términos que se utilizan frecuentemente: formatear (dar formato o inicializar un disco) y particionar (dividirlo). El proceso de formatear consiste, en realizar una serie de pruebas (lectura y escritura) sobre cada uno de los sectores del disco para determinar cuáles están dañados y marcarlos, de forma que no se utilicen para almacenar datos. Cuando se adquiere un disco, éste viene formateado de fábrica, y los sectores que puedan tener daño, ya se encuentran identificados y colocados en una tabla de sectores dañados; de tal forma que no son utilizados. Todos los discos tienen esta tabla y se pueden anexar o eliminar sectores de ella manualmente con opciones del comando *fx*.

Al arrancar el programa *fx*, solicita que se le indique en qué modo se desea trabajar; si el experto o el seguro. Para poder realizar la operación de dar formato se tiene que ingresar al modo experto, y en él, seleccionar la opción de formatear. Si se ingresa al modo seguro, esta opción no aparece, y por lo tanto, únicamente se puede particionar y ver la información del disco.

Por otro lado, el particionar un disco consiste en dividir el espacio total del disco en áreas de diferente tamaño llamadas particiones, como se describió anteriormente. En cualquiera de los dos modos (experto o seguro), *fx* puede particionar discos.

Si un disco es nuevo, con particionarlo es más que suficiente; pero si se sospecha que pueda tener daños, o durante la vida útil el SO empieza a marcar errores indicando que se encuentran sectores dañados, es conveniente darle formato.

Como último punto, el proceso de formatear borra toda la información del disco, y el de particionar o reparticionar, afecta únicamente las áreas que sean modificadas. Independientemente de la labor que se realice, es conveniente e imperativo, que se haga un respaldo de toda la información que contenga el disco antes de efectuarlo.

Al ejecutar este comando, pide que se le indique en qué modo va a trabajar (experto o seguro); después, que se le proporcione la controladora y el identificador del disco con el cuál se trabajará, y finalmente muestra un menú con las opciones:

exit	Permite salir del comando *fx*.
badblock	Pasa a un submenú que permite trabajar (añadir, borrar, mostrar, etc.) con la tabla de sectores dañados.
debug	Pasa al submenú que permite depurar el disco (lectura y escritura de sectores, etc.).
exercise	Entra a un menú que permite realizar pruebas sobre la superficie del disco, para detectar sectores defectuosos y poder anexarlos a la tabla de sectores dañados.
label	Pasa a un submenú que permite ver y cambiar (en modo experto) la información almacenada en la etiqueta del disco (volhdr), la cual se agrupa en distintos submenús según su función.
repartition	Permite particionar el disco en los esquemas definidos por Silicon Graphics (que se detallan más adelante); reparticionar cambiando el tamaño actual de las particiones o particionar el disco como el usuario lo desee.
auto	Aparece sólo en modo experto e inicializa un disco nuevo: le da formato, le crea la etiqueta del volumen y pone en la tabla de sectores dañados, aquéllos que se detectaron al efectuar las pruebas sobre su superficie.
format	Aparece únicamente en modo experto y se utiliza para dar formato a un disco.

En el punto "Instalación del Sistema Operativo", pág. I-21, y en algunos de éste, se describen más características de él, así como ejemplos de su funcionamiento. Para mayores detalles, consultar su ayuda mediante el comando: *S man fx*.

III.1.1.II. Rootdrive

Este tipo de partición es la que se utiliza actualmente para los discos principales; discos de arranque. Cuando un disco es particionado utilizando este esquema, se divide en 3 particiones fisicas y 1 lógica.

Tabla 7 Esquema de división de discos 'rootdrive'.

Partición	Tipo	Descripción
0	efs	fisica
1	raw	fisica
8	volhdr	fisica
10	volume	lógica

La partición 0 es de tipo efs, por lo que es utilizada para almacenar la información del sistema (archivos, directorios, etc.) y es a la que se le asigna la mayor cantidad de espacio disponible.

La partición 1 es de tipo raw, por lo que es utilizada por el sistema como área de swap, y el espacio que se le asigna, varía dependiendo de la configuración del equipo.

La 8 la ocupa el encabezado de volumen voldhr, para almacenar, como ya se dijo, la información del disco y programas especiales.

Finalmente, la 10 es una partición lógica que hace referencia al disco completo, y ésta es utilizada por algunas aplicaciones que tratan al disco como una unidad entera.

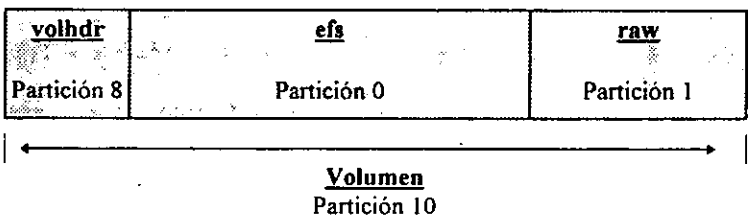


Fig. III-4 Particiones en estilo rootdrive

La Fig. III-4 muestra cómo están distribuidas las particiones. Se puede ver claramente que la partición 10 es lógica y cubre o representa al disco completo. Un ejemplo del uso de esta partición ocurre con los equipos en los cuales se encuentra instalada la base de datos SYBASE. Ésta tiene su propio manejador de sistemas de archivos, y al utilizarlo, se tiene acceso al disco completo para almacenar la base de datos. Se suelen utilizar discos secundarios[22] para este propósito.

Por otro lado, la Fig. III-5 muestra una vista, tanto lateral como superior, de los platos que forman al disco duro. En ella se muestra la ubicación física de las particiones sobre la superficie de los platos. Cabe destacar que la partición volhdr, que es una partición especial y crítica para el funcionamiento del disco, ocupa los primeros cilindros del disco, los exteriores. En cambio, la raw ocupa los últimos, los más internos, quedando el resto para la partición efs.

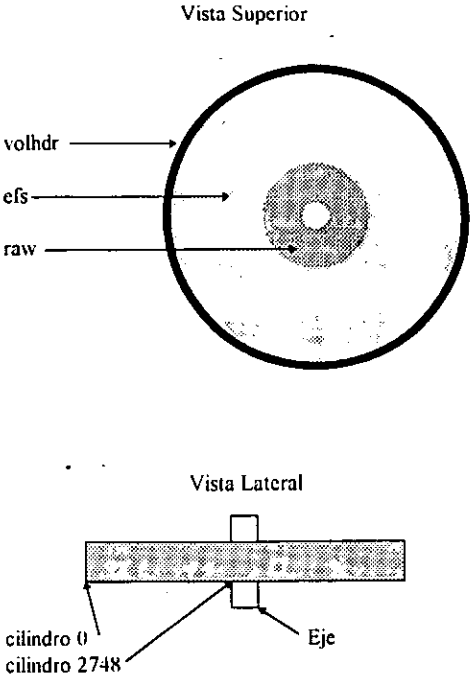


Fig. III-5 Estructura de las particiones en un disco principal SCSI en IRIX.

III.I.I.III. Usrrootdrive

Éste es el estilo de partición que se utilizaba anteriormente. En él se crean 4 particiones físicas y una lógica.

Tabla 8 Esquema de división de discos 'usrrootdrive'.

Partición	Tipo	Descripción
0	efs (/)	física
1	raw	física
6	efs (usr)	física
8	volhdr	física
10	volume	lógica

[22] Al disco de arranque, de donde se lee el SO, suele llamarse disco primario; y a los demás, secundarios.

La partición 0 y la 6 se utilizan para almacenar el SO y la información de los usuarios. La 0 se utilizaba como la raiz (/) del sistema de archivos, y la 6 para almacenar la información del directorio */usr*; por lo que la partición 6 suele montarse en el directorio */usr* de la partición 0; por este hecho, es que se llama a este estilo usr-root-drive.

Al igual que en el modelo de rootdrive, la 1 sirve como área de swap, la 8 como encabezado de volumen y la 10, que es una partición lógica, como referencia al disco entero.

III.I.I.IV. Optiondrive

En este esquema el disco es dividido en dos particiones fisicas y una lógica.

Tabla 9 Esquema de división de discos 'optiondrive'.

Partición	Tipo	Descripción
7	efs	fisica
8	volhdr	fisica
10	volume	lógica

Al igual que en los estilos anteriores, la partición lógica 10 se utiliza para referirse al disco entero y la 8 como encabezado del volumen.

A la partición 7 es a la que se le asigna la totalidad del espacio de disco (a excepción del pequeño espacio requerido por la 8), y es utilizada por el SO para almacenar cualquier tipo de información, archivos, directorios, etc.

Este esquema de particionamiento es utilizado generalmente en los CD, pero cualquier disco duro puede utilizarlo.

III.I.II. Estructura del sistema de archivos

Un sistema de archivos es la estructura y forma en la cual se organizan los datos dentro de una unidad de almacenamiento, de tal forma que puedan ser accesados por el SO que lo utiliza. Actualmente existe una gran cantidad de sistemas de archivos que son utilizados por una diversidad de sistema operativos, y cada uno organiza y distribuye la información en el medio de forma diferente. Es por esto, que el sistema operativo para poder leer información de un disco, debe saber cómo se encuentra organizada dentro de él; es decir, cuál es el sistema de archivos que se utilizó.

IRIX 5.3 puede detectar, y por lo tanto leer o guardar información, en un cierto número de sistema de archivos:

efs — Sistema de Archivos Extendido (Extent File System). Éste es el sistema de archivo de alto rendimiento estándar en IRIX 5.3. Es utilizado para guardar la información del sistema en los disco duros, así como en los CD's de distribución que contienen los productos a instalarse. Por ser el utilizado en IRIX 5.3, será tratado en forma amplia, más adelante.

nfs — Sistema de Archivos de Red (Network File System). Permite ver el SA o un directorio de una máquina remota, como si estuviera fisicamente en la máquina local. De tal forma que se puede trabajar con los archivos (leerlos, editarlos, ejecutarlos, etc.) como si estuvieran en el equipo; aunque realmente se encuentren en otro situado a miles de kilómetros. El soporte para entenderlo es proveído por un paquete opcional que se puede adquirir e instalar por separado; por lo que no se encuentra en el CD de instalación del SO.

kfs — Es un sistema de archivos que permite accesar discos locales o remotos, mediante el protocolo AppleShare; que es usado generalmente por computadoras Macintosh. Es muy parecido al nfs, y para que el SO lo pueda entender, se debe instalar un paquete opcional que se adquiere por separado.

dbg — Sistema de archivos de depuración (Debug File System). Éste, provee una interfaz a una imagen de cada proceso que se encuentra corriendo en el equipo, y que es utilizada generalmente por programas de depuración como *dbx*, y comandos como *ps* y *top*. Este sistema de archivos era montado anteriormente en el directorio */debug*; pero actualmente se encuentra en */proc* y se crea una liga a */debug*. Cada proceso que se encuentra corriendo en el equipo, es representado en este directorio por un archivo que lleva por nombre, el número del identificador de proceso[23] asignado a ese proceso precisamente. Este sistema de archivos es creado, montado y manejado por el SO desde el momento en que toma control del equipo, y no consume espacio en el disco.

fd — Sistema de archivos usado para accesar los descriptores de archivos de procesos. Es montado por el SO en el directorio */dev/fd*; por lo que los descriptores de archivos se encuentran en */dev/fd* y llevan por nombre 0, 1, 2, etc. Estos son manejados por el sistema y permiten al usuario accesar los procesos.

fat — Tabla de Alojamiento de Archivos (File Alocation Table). Sistema de archivos utilizado en los equipos PC que trabajan con el SO MS-DOS, por lo que se conoce también como formato de DOS, refiriéndose a MS-DOS. IRIX puede soportar discos flexibles y ópticos de 720 KB y 1.44 MB que se encuentren en este formato.

[23] Un Identificador de proceso es un número que se le asigna a cada proceso al momento de ejecutarse y que lo identifica. Este número se incrementa consecutivamente por cada proceso que se ejecuta, de tal forma que nunca pueden existir dos procesos que tengan el mismo número.

hfs Éste es el sistema de archivos utilizado en los equipos Macintosh, por lo que se conoce como formato Mac. IRIX puede soportar discos flexibles de 1.44 MB y CDROM que se encuentren en este formato.

ISO9660 El estándar definido para CD-ROM 9660 de ISO. Ya que se trata de un estándar definido por ISO, se utiliza para exportar información que pueda ser leída por todos los sistemas (es hacia lo que se enfoca un estándar). Para que IRIX pueda entender este formato, debe estar instalado el paquete *eoe2.sw.cdrom* que se encuentra en el CD-ROM de instalación del sistema.

CD de Música Este es un estándar-industrial que permite grabar música en CD y no se trata de un sistema de archivos en sí. IRIX puede leer y reproducir la música o cualquier sonido almacenado en ellos; para ello se vale de una aplicación llamada *cdman* y *cdplayer*.

El mantener en buen estado el sistema de archivos es una de las labores primordiales del administrador, ya que cualquier daño que sufra se ve reflejado en la pérdida de información; ya sea un archivo, varios o toda la información contenida en él. Por este motivo se describirá también la herramienta que permite llevar a cabo esta labor.

III.I.II.I. Sistema de archivos EFS

Como se mencionó, éste es el sistema de archivos utilizado en la versión 5.3 de IRIX. Presenta una serie de extensiones sobre el sistema de archivos estándar de UNIX, por tal motivo se llama Sistema de Archivos Extendido. En la actualidad se dispone de uno nuevo que es utilizado en la versión 6 y se conoce con el nombre de XFS, el cual incluye nuevas mejoras importantes.

Un sistema de archivos puede ocupar una partición o varias. Cuando se crea, éste ocupa en su totalidad la partición donde reside; de esta forma se pueden crear varios, ocupando cada uno de ellos, una partición. Por otro lado, una partición puede ocupar como máximo el tamaño del disco, y por lo tanto, éste será el tamaño máximo del sistema de archivos. Si se requiere uno que sea de un tamaño mayor, lo que se hace es expandirlo, de tal forma que abarque varios; es decir, varios sistema de archivos se unen formando uno solo, cuyo tamaño es igual a la suma del tamaño de cada uno de ellos. A este tipo de sistema de archivos se le conoce como volumen lógico *lvol*. Cabe destacar que el tamaño máximo que puede abarcar un Sistema de Archivos EFS es de 16777214 bloques, que equivale a 8 Gigabytes. Por el momento se describirá la estructura que guarda un SA EFS normal, dejando el análisis de los volúmenes lógicos para después.

Cuando se crea un sistema de archivos, el espacio de la partición que ocupa, es dividido en bloques de 512 bytes y cilindros de la siguiente forma:

- El primer bloque (0) no es utilizado. Servía para almacenar información que permitiera arrancar el sistema. En la actualidad, IRIX utiliza la partición volhdr para guardarla.
- El siguiente bloque (1), es utilizado para almacenar la información que describe la estructura del sistema de archivos. A este bloque se le conoce como super bloque.
- A continuación del super bloque (a partir del bloque 2) empieza el mapa de bits de los bloques de datos. El mapa de bits, como su nombre lo indica, es un mapa que contiene únicamente bits; en donde cada bit en este mapa; indica un bloque de datos libre. Por su parte, los bloques de datos son bloques que se utilizan para guardar la información dentro del sistema de archivos.
- A continuación se encuentran una serie de grupos de cilindros que abarcan en su totalidad el espacio del sistema de archivos. Es en estos grupos de cilindros donde se almacena la información. Entre el mapa de bits y el primer grupo de cilindros se encuentra un espacio que no se utiliza, y cuya única función, es la de alinear el inicio del primer grupo de cilindros en los linderos de un cilindro. Esto se realiza con el fin de agilizar la lectura y escritura de datos en los grupos de cilindros.
- Finalmente, el último bloque del sistema de archivo es utilizado para almacenar una copia del super bloque. Existe un espacio ubicado entre el último cilindro y este último bloque, que no se utiliza por razones de alineamiento, al igual que él que se encuentra antes del inicio del grupo de cilindros.

En la figura siguiente se muestra gráficamente esta estructura.

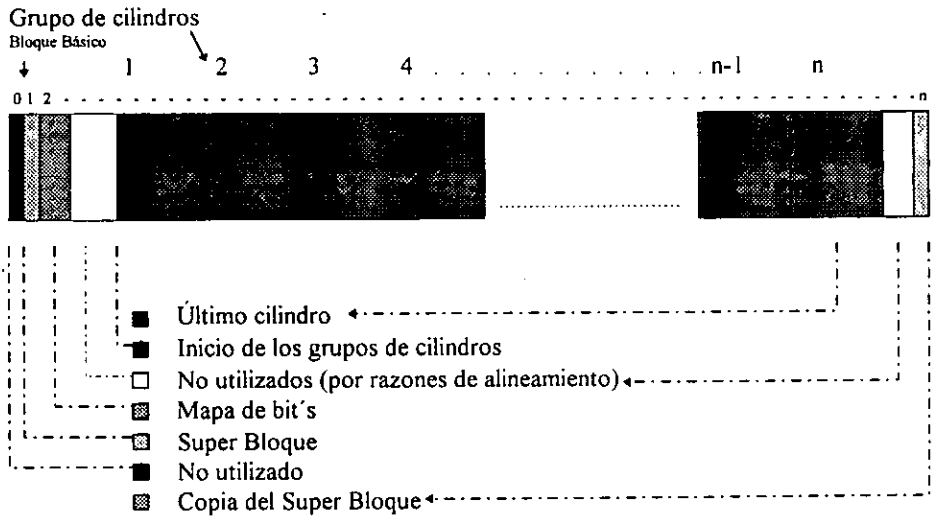


Fig. III-6 Estructura de un Sistema de Archivos EFS.

De todo esto se puede observar, que básicamente un sistema de archivos está compuesto por: un super bloque, que define su estructura; el mapa de bit, que define los bloques libres y una serie de grupos de cilindros, que almacenan la información.

III.I.II.I.I. Super Bloque

El super bloque es el más importante dentro del SA; ya que describe la estructura de éste. Si se llega a dañar, se puede perder toda la información de esta partición. Es por esta razón, que se mantiene una copia de él, al final de la estructura de este sistema; para de alguna manera, tratar de recuperarlo en caso de que sufra algún daño.

Tabla 10 Contenido del super bloque.

Tamaño del Sistema de Archivos, indicado en bloques.
Distancia que hay al primer grupo de cilindros; es decir, su ubicación.
Tamaño de los grupos de cilindros, indicado en bloques.
El número de bloques que serán utilizados para almacenar los *inodos*[24] dentro de cada grupo de cilindros.
El número de sectores por pista. *
El número de cabezas por cilindro. *
El número de grupos de cilindros que hay en el Sistema de Archivos.
Una bandera que indica si el Sistema de Archivos necesita ser revisado. Esta bandera indica si el equipo no fue dado de baja correctamente; por lo que pueden haber inconsistencias.
Fecha de la última actualización del super bloque.
Un número que identifica que se trata de un Sistema de Archivos EFS (número mágico).
El nombre del Sistema de Archivos; es decir, donde fue montado anteriormente.
Un nombre que indica a qué volumen pertenece. Si es que forma parte de uno.
Tamaño del *mapa de bit* de los bloques de datos, medido en bytes.
El número total de bloques libres para almacenar datos.
El número total de inodos libres.
La localización del mapa de bits de los bloques de datos.
Localización del lugar donde se encuentra una copia del super bloque.
Un espacio disponible para expansión. Este campo no es utilizado en la actualidad.
Un número que sirve para determinar la integridad de los datos almacenados en el super bloque. A este número se le conoce como checksum y es generado a partir de los mismos datos:

* Estos campos no son utilizados; la información se encuentra grabada en la partición volhdr.

[24] inodo es la abreviación de Index nodo, ver pág. III-17

El super bloque contiene más información, que es utilizada exclusivamente por el sistema operativo. De todo esto, se puede ver la importancia que guarda este bloque.

III.I.II.I.II. Estructura de los Grupos de Cilindros.

Como ya se mencionó anteriormente, un conjunto de cilindros contiguos forman un grupos de cilindros, cuyo propósito principal es el de almacenar la información del sistema. Por otro lado, cada grupo de cilindros está formado por dos componente: inodos y bloques de datos.

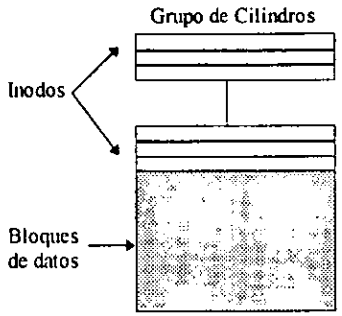


Fig. III-7 Estructura del grupo de cilindros.

Los bloques de datos, como su nombre lo indica, son bloques que contienen la información de los archivos que se almacenan en el disco, y un inodo, es la estructura que guarda toda la información referente a un archivo; a excepción de su nombre, el cual se encuentra almacenado en el directorio. La Tabla 11 muestra la información que contiene un inodo.

El tipo de archivo (ver Tabla 11) indica si se trata de un archivo normal, un directorio, un archivo de dispositivo o una liga. Los permisos indican los derechos de escritura, lectura y ejecución que tiene el dueño del archivo, los miembros del grupo al cual pertenece y el público en general. El número de ligas indica la cantidad de ligas que hacen referencia a este archivo; este término se describirá más adelante. El identificador de usuario y grupo, definen quién es el dueño del archivo y a qué grupo pertenece, y poder así, aplicar los permisos indicados anteriormente. El tamaño del archivo indica el número de bytes que ocupa. A continuación siguen tres campos que indican la hora, medida en segundos a partir de 1970 GMT[25], del último acceso y modificación al contenido del archivo y modificación al inodo.

[25] Tiempo del Meridiano de Greenwich.

Tabla 11 Contenido del inodo

Tipo del Archivo.
Permisos de acceso.
Número de ligas.
Número de identificación de usuario del dueño del archivo.
Número de identificación de grupo del dueño del archivo.
Tamaño del archivo.
Hora del último acceso al contenido del archivo.
Hora de la última modificación al contenido del archivo.
Hora de la última modificación al inodo.
Número de generación del inodo
Número de extensiones ocupadas por el archivo.
Versión del inodo
No utilizado
Descriptor de extensión 1
:
Descriptor de extensión 12

El número de generación del inodo, se coloca a cada inodo en forma consecutiva al ser creado el Sistema de Archivos, para identificar a cada uno de ellos. La versión del inodo sirve para identificarlo y poder determinar la estructura que tiene; ya que nuevas versiones pueden ser creadas en las cuales, el número, distribución y tamaño de los campos que lo componen, varíen.

Finalmente, la parte fundamental de comprender son las extensiones. Las extensiones son bloques de datos que forman al archivo y son de longitud variable, de 1 a 248 bloques contiguos; es decir, que una extensión puede ser como máximo de 248 bloques y almacenar un total de 126 976 bytes. Un descriptor de extensión es una estructura que define la ubicación de la extensión o su dirección. Para entender este concepto, primero se explicará la forma en la que se almacena un archivo en el bloque de datos.

Como se mencionó, en el bloque de datos se almacena el contenido de los archivos; para lo cual, al momento de ser grabado un archivo es dividido y almacenado en extensiones, como se ilustra en la Fig. III-8. Las extensiones que forman al archivo pueden quedar en forma consecutiva o dispersas en el área de almacenamiento, por lo que para identificar a un archivo se requiere conocer por cada extensión que lo forma: la dirección del primer bloque donde empieza la extensión; la longitud que tiene la extensión, ya que su tamaño puede variar; y el lugar que ocupa la información de la extensión dentro del archivo original, para

poder reconstruirlo adecuadamente. A estos tres datos se les conoce como descriptor de extensiones y podríamos decir que son apuntadores que nos indican el lugar donde se encuentran almacenadas las extensiones que contienen la información del archivo.

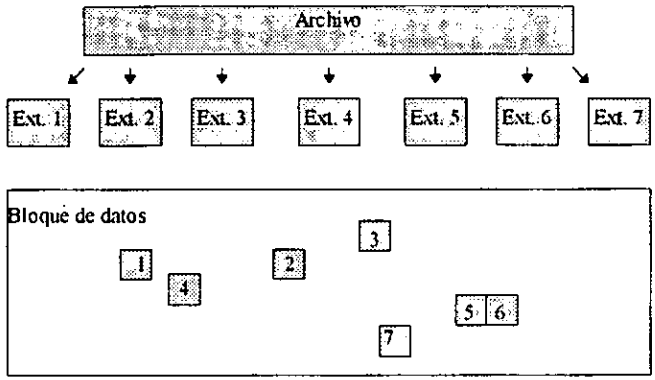


Fig. III-8 Forma en la que se almacenan los archivos.

Un inodo contiene como máximo 12 descriptores de extensiones, y dado que una extensión puede almacenar como máximo 126 976 bytes, se tiene que un inodo puede almacenar archivos que contengan 1,523,712 bytes (1.45 MB) en forma directa.

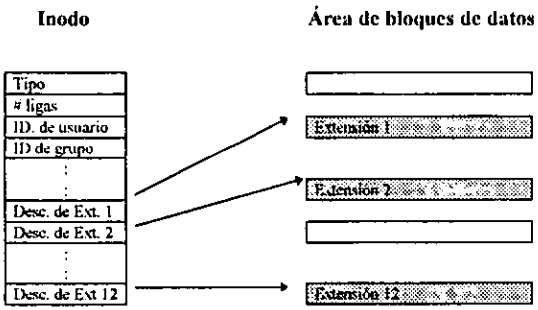


Fig. III-9 Direccionamiento directo de las extensiones

Si el tamaño de un archivo es mayor que 1,523,712 bytes se utiliza una técnica de direccionamiento indirecto. En este método, cada descriptor de extensión ubicado en el inodo apunta a una extensión, cada una de las cuales, puede contener como máximo la dirección de 248 extensiones. Esto permite que el tamaño máximo de un archivo sea de 2GB.

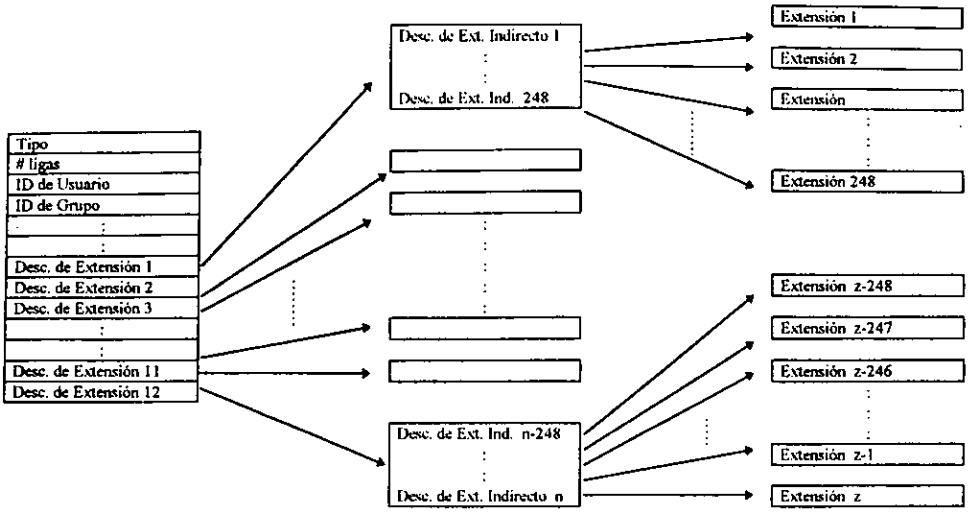


Fig. III-10 Direccionamiento indirecto de las extensiones.

III.I.II.I.III. Organización de los directorios

Un archivo es un arreglo unidimencional sin ninguna otra estructura implementada dentro de él, que permite almacenar la información. En cambio, un directorio es un tipo especial de archivo que a los usuarios les está permitido usar; pero no escribir directamente sobre él. El SO tiene la responsabilidad de escribir sobre él y de mantenerlo en perfecto estado.

La estructura de un directorio es simple; está compuesto por registros que contienen dos campos: el nombre de un archivo y el número de inodo que le pertenece.

Toda la estructura trabaja de la siguiente forma: Cuando se crea un archivo, el nombre de éste es almacenado en el primer campo de un registro libre del directorio donde residirá. Se localiza un inodo libre para asignárselo, y el número de este inodo es colocado en el segundo campo del registro. La información del archivo es grabada en bloques de datos libres dentro del grupo de cilindros, y todos los detalles referentes al archivo, así como la ubicación de cada uno de los bloques de datos que lo componen, es almacenada en los campos del inodo que le fue asignado, ver Fig. III-11.

De lo anterior se puede observar que si la cantidad de inodos creados en un SA es pequeña, se pueden agotar; ya que por cada archivo que se cree, se utiliza uno. Si esto llega a suceder, ya no se podrán crear archivos; aunque exista espacio disponible para almacenarlos. Para la

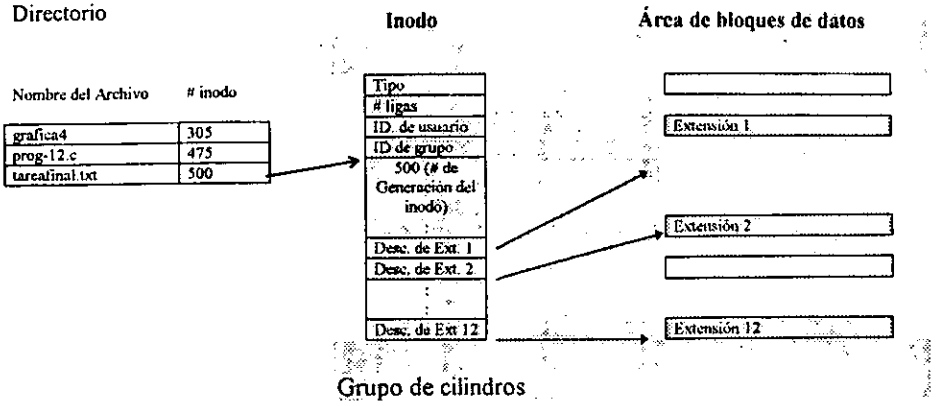


Fig. III-11 Estructura del directorio

mayoría de las situaciones y ambientes en los cuales puede estar trabajando un SO IRIX, la cantidad de inodos que se crean suele ser suficiente; pero si debido a circunstancias especiales, como el que la mayoría de los archivos creados sea de tamaño pequeño y en gran cantidad, será necesario el regenerar el SA e indicarle que se creen una mayor cantidad de inodos; los suficientes para satisfacer las necesidades futuras. Esto se realiza con el comando *mkfs*, que permite crear los SA.

De lo anterior se puede observar que si la cantidad de inodos creados en un SA es pequeña, se pueden agotar; ya que por cada archivo que se cree, se utiliza uno. Si esto llega a suceder, ya no se podrán crear archivos; aunque exista espacio disponible para almacenarlos. Para la mayoría de las situaciones y ambientes en los cuales puede estar trabajando un SO IRIX, la cantidad de inodos que se crean por default, suele ser suficiente; pero si debido a circunstancias especiales, como el que la mayoría de los archivos creados sea de tamaño pequeño y en gran cantidad, será necesario el regenerar el SA e indicarle que se creen una mayor cantidad de inodos; los suficientes para satisfacer las necesidades futuras. Esto se realiza con el comando *mkfs*, que permite crear los SA.

Cuando se crea un SA se divide y organiza el espacio perteneciente a la partición, de la forma descrita en los temas anteriores; posteriormente se crea un directorio que sirve como punto de partida, o raíz, para organizar los archivos y directorios que se crearán dentro de él. A este primer directorio no se le asigna ningún nombre; pero si se trata del SA principal, que se encuentra en el disco de arranque, se denomina *root* y se representa mediante un /. A partir de él, se crean los demás directorios y archivos, formando una estructura de árbol como se ilustra en la Fig. III-12.

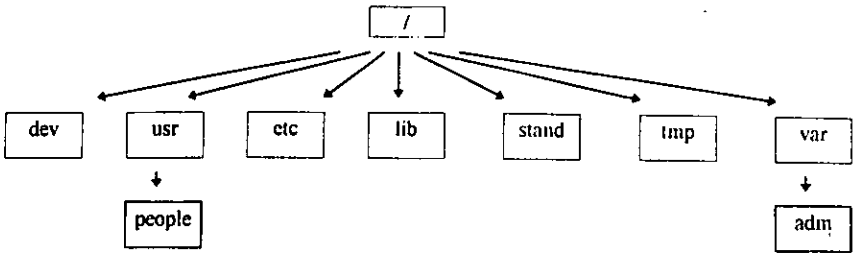


Fig. III-12 Estructura de árbol del Sistema de Archivos.

Todos los sistemas UNIX cuentan con una serie de directorios estándares que contienen información definida dentro de ellos. La organización de esta estructura es heredada de sus primeras versiones. Una descripción de los más importantes se encuentra en el APENDICE B. Por otro lado, los diferentes tipos de archivos que pueden ser creados, se encuentran listados en la Tabla 12. El carácter es un símbolo que se utiliza para reconocer el tipo de archivo; puede ser observado en el primer campo, al listar los archivos de un directorio mediante el comando *$ ls -l*.

Tabla 12 Tipos de archivos que pueden ser creados en IRIX.

Carácter	*Tipo*	*Descripción*
-	Regular	Son archivos normales.
d	Directorio	Los archivos de este tipo son directorios.
l	Liga	Es una liga; es decir, una entrada dentro de un directorio cuyo número de inodo hace referencia a un archivo que ya existe.
c	Dispositivo en modo carácter	Es un archivo de dispositivo creado mediante el comando *mknod* que accesa la información en modo carácter.
b	Dispositivo en modo bloque	Es un archivo de dispositivo creado mediante el comando *mknod* que accesa la información en modo de bloque.
p	FIFO[26]	Permiten la comunicación entre dos procesos en el mismo equipo. Es creado con el comando *mknod*.
s	sockets	Son archivos que permiten la comunicación entre procesos a través de una red.

Esta estructura de árbol es la que reconocen y utilizan todos los usuarios; para los cuales, permanece oculto la estructura interna que maneja el SO. Pero para un administrador, es importante el comprenderla, ya que le facilitará su tarea y le permitirá deducir y solucionar problemas rápidamente.

[26] First Input, First Output; Primero en Entrar, Primero en Salir.

III.I.II.I.IV. Archivos de dispositivos

Son un tipo especiales de archivos que suelen ser creados mediante el comando *mknod*, y que permiten al SO, comunicarse con los diferentes periféricos con que cuenta el equipo. Se requiere uno por cada elemento con el cual se desee comunicar; por lo que existe uno para la memoria */dev/men*, uno para la consola */dev/console*, para la memoria de intercambio */dev/swap*, para el teclado */dev/keybd*, ratón */dev/mouse*, etc. En especial, para comunicarse con un disco se utilizan dos tipos diferentes de estos archivos: uno mediante el cual, se accesa la información de manera secuencial, y cuyos archivos se encuentran en el directorio */dev/rdsk*; y otros que lo hacen en bloques, y cuyos archivos se localizan en */dev/dsk*.

Cada partición de un disco es considerada como una entidad separada; por lo que se utiliza un archivo de dispositivo para cada una. Por convención, un archivo de dispositivo que hace referencia a una partición de un disco, lleva por nombre:

/dev/dsk/tipo-controlador-disco-partición

donde:	*/dev/dsk*	Indica la ubicación del archivo; por lo que se espera que accese la información en bloques.
	tipo	Especifica el tipo de disco utilizado: Si es SCSI = *dsk* ESDI = *ips* SMD = *xyl* IPI = *ipi* VME SCSI (disco Jaguar) = *jag*
	controlador	La tarjeta controladora utilizada; ya que un equipo puede tener varias tarjetas del mismo tipo. Cero (*0*) para la primera.
	disco	El disco en dicha controladora; ya que una tarjeta controladora puede manejar varios discos a la vez. Para el primero es *d0*.
	Partición	La partición dentro del disco, a la que se va a ser referencia mediante este archivo de dispositivo. Para la primera es *s0*.

De tal forma que para acceder en forma de bloques a la partición 0, que contiene un SA EFS, de un disco SCSI principal colocado en la primer tarjeta controladora (0), se utiliza el archivo de dispositivo */dev/dsk/dks0d0s0*, y para acceder a él en forma secuencial, el archivo */dev/rdsk/dks0d0s0*.

Cabe mencionar que esto es sólo una convención de nombres, y que se puede crear cualquier archivo de dispositivo con cualquier nombre, para acceder a la información almacenada en los discos; pero para mantener un estándar y hacer fácil la administración de los equipos, es conveniente seguir con este acuerdo.

III.I.II.I.V. Montaje de un SA

Recordando que: Al directorio principal del disco de arranque a partir del cual se desprende la estructura del SA, que tiene una forma de árbol, se le conoce como root o raíz, ver Fig. III-12, el cual es reconocido y montado automáticamente cada vez que se enciende el equipo.

Y teniendo en cuenta que: Todo SA cuenta con un directorio principal que no tiene nombre y que se representa mediante un /.

Si se dispone de otra partición o disco que contengan un SA con el cual se desee trabajar, éste debe ser reconocido por el SO y añadido al SA principal. A este proceso se le llama montaje y es llevado a cabo mediante el comando *mount*.

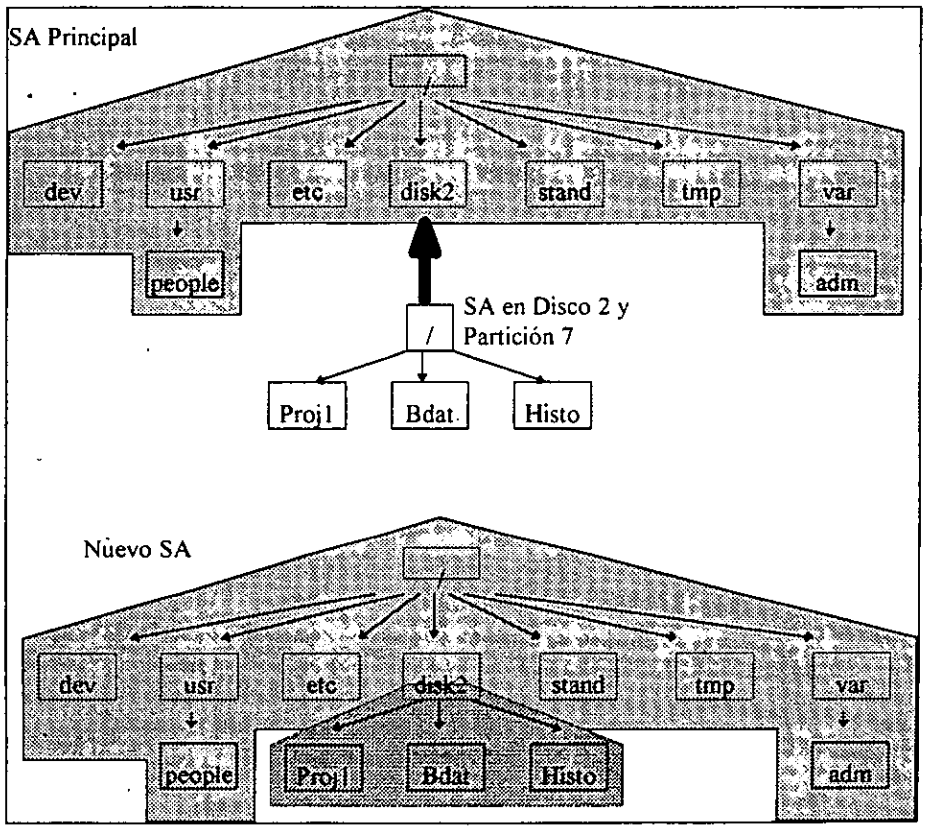


Fig. III-13 Montaje de un SA

De la Fig. I-13 se puede observar que la raíz del SA de la partición 7 del segundo disco (*/*), es reconocida como */disk2* una vez que se ha montado en el SA principal; por lo que el SO ya podrá usar la información almacenada en cualquiera de sus directorios.

Cualquier SA puede ser montado para trabajar con él, o desmontado mediante el comando *umount* para dejar de utilizarlo; a excepción del SA principal, que como ya se dijo, es montado automáticamente al arrancar el equipo, y no podrá ser desmontado sino hasta que se da de baja el SO. Si se tienen varios SA que deben ser montados cada vez que se arranque el equipo, se puede automatizar esta labor al agregar una entrada por cada uno de ellos, en el archivo */etc/fstab*; para ello se puede utilizar cualquier editor de textos.

Como punto adicional, cuando se monta un SA, toda la información que se encontraba en el directorio que sirve como punto de montaje (*/disk2* en el ejemplo anterior), se oculta y no podrá ser utilizada hasta que se desmonte y quede libre el directorio de montaje.

III.I.II.I.VI. Creación de un SA

La herramienta utilizada para crear un SA, es el comando *mkfs*. Éste puede ser ejecutado de dos formas: En la primera y más sencilla, únicamente se especifica el archivo de dispositivo de acceso secuencial de la partición donde residirá el SA. Se encargará *mkfs* de determinar los parámetros adecuados para su construcción. En la segunda, se deben proporcionar los datos necesarios al momento de ejecutar el comando, como: su tamaño, el número de inodos que se generarán, el tamaño de los grupos de cilindros, el número de cabezas y sectores del disco, etc. Por lo que el siguiente comando creará un SA EFS sobre la partición cero del disco cero de la tarjeta controladora principal:

 # mkfs /dev/rdsk/dks0d0s0

Una vez construido el SA de acuerdo a lo descrito anteriormente, y antes de terminar, crea dos directorios: el principal (*/*), y dentro de él, al directorio */lost+found*.

El programa de mantenimiento *fsck* que checa y repara SA dañados, utiliza el directorio */lost+found* para reconectar los archivos recuperados. Para que pueda desempeñar sus labores en forma adecuada, requiere que existan registros libres dentro del directorio; por lo que *mkfs* llena con ceros, 10 bloques de disco pertenecientes al directorio, generando de esta forma, los registros libres. Esto se debe a lo siguiente: cuando se crea un directorio, no se generan registros dentro de él; por lo que cuando se le indica al SO que almacene un archivo, éste añade un registro al directorio y almacena su nombre y número de inodo dentro de él. Cuando se le indica que borre un archivo, simplemente limpia los datos del registro, dejándolo vacío. Si se crea otro archivo, utiliza el registro vacío o crea uno nuevo para almacenar sus datos. En cambio, *fsck* no puede crear los registros, ya que esta función es exclusiva del SO; por lo que ya deben estar creados, para que simplemente escriba en ellos.

III.II. Reconfiguración del kernel

El corazón del SO IRIX es un programa conocido como kernel y que generalmente se encuentra localizado en la raíz del SA principal, con el nombre de */unix*. Es un programa muy inteligente cuya función es la de llevar el control del los recursos de la computadora. Al encender el equipo, éste es cargado a RAM, y a partir de ese momento toma el control; lo primero que hace, es terminar su configuración añadiendo módulos que le permitan brindar los servicios instalados (ver 'Encendido y apagado del equipo', pág. II-3).

Por otro lado, el kernel no es más que un programa generalmente escrito en lenguaje C, que contiene tablas, variables, etc., definidas y alojadas en RAM. Como toda aplicación, únicamente realiza las instrucciones que tiene programadas; así mismo, cuando se genera se definen tamaños máximos para distintos arreglos y tablas que permiten mantener el control de los procesos: como el número máximo de Volúmenes Lógicos que se pueden crear en el sistema, el número máximo de archivos abiertos que pueden existir en un determinado momento, la cantidad máxima de procesos corriendo en un instante, etc. Si es necesario incrementar el número de elementos que pueda alojar una tabla en específico, es necesario localizar los archivos fuente, modificar las variables adecuadas, recompilarlo para generar el nuevo programa ejecutable o kernel y activarlo colocándolo en el archivo */unix*, para que sea ejecutado la próxima vez que se arranque el equipo. A este proceso se le conoce con el término de 'Reconfiguración del Kernel'.

De aquí surgen varios aspectos que tendremos que analizar: dónde se encuentran los archivos fuentes del kernel, cuáles son las variables que se pueden modificar y cuál es su función, cuál es el proceso para reconfigurar el kernel, y cuáles pueden ser los motivos que conducen a su regeneración. Éstos y otros puntos serán tratados a continuación.

III.II.I. Motivos para reconfigurar el kernel

Los motivos pueden ser varios, pero entre los más importantes destacan:

Mejorar el rendimiento. Cuando se instala el SO se genera un kernel que va de acuerdo a los distintos dispositivos del sistema y a sus capacidades, de tal forma que éste suele ser la mejor opción para la mayoría de los ambientes de trabajo; pero, si esto no llega a ser suficiente, y se tienen los conocimientos suficientes para poder determinar las consecuencias, benéficas o dañinas, que pueda tener el modificar ciertos valores de parámetros en el kernel, se puede incursionar en ello. El procedimiento es conocido como afinación del sistema y envuelve pasos simples y cíclicos:

- Análisis del estado actual: que permite determinar dónde se tienen deficiencias, con lo que se puede establecer qué parámetros requieren un cambio.
- Modificar los parámetros adecuados: que implica asignar nuevos valores tendientes a eliminar el problema y mejorar su rendimiento.
- Generación y puesta en marcha del nuevo kernel.
- Análisis del nuevo rendimiento obtenido: para determinar los beneficios obtenidos, o deficiencias generadas con los cambios.
- Comparación de los resultados: para determinar si es necesario modificarlos nuevamente, en cuyo caso se reinicia el ciclo, o se ha alcanzado la meta deseada.

Demanda de recursos. Si durante la operación diaria del equipo, se empieza a encontrar conflictos en los que se indique un error que tenga que ver con la configuración del kernel, y éstos llegan a ser considerables, será necesario reconfigurarlo a fin de solucionarlos. Los casos más comunes son cuando se han agotado las entradas de las tablas; como la de procesos por ejemplo (ver Parámetros configurables, pág. III-29).

Incrementar controladores de nuevos dispositivos. Incorporados en el kernel, se encuentran controladores para los dispositivos más comúnmente utilizados y aquéllos que son detectados cuando se instala el SO y se genera el kernel. Cuando se añade un nuevo hardware, el sistema requiere de un programa controlador que le indique cómo funciona; éste puede ser anexado al sistema mediante la ejecución de una aplicación, un vez que ha arrancado el SO, o requerir la generación de un nuevo kernel que incluya el controlador especifico para el dispositivo. Cuál método emplear, se determinará al momento de adquirirlo y leer la documentación que lo acompañe.

Instalación de nuevo software mediante *inst*. Al igual que en el hardware, puede ser necesario reconfigurar el kernel cuando sea instalado un nuevo programa. En especial, si se utiliza el comando *inst*, éste determina si es necesario reconfigurarlo, y si es así, realiza los cambios necesarios a los parámetros y añade los módulos requeridos durante el proceso de generación, que inicia automáticamente; posteriormente reinicia el equipo para que sea utilizado el nuevo SO. De forma análoga, si se instala cualquier software por medio de otro procedimiento, puede ser necesario la reconfiguración del kernel para cubrir las necesidades o incrementar nuevos controladores que le permitan al SO, utilizar y soportar las capacidades de dicho producto.

Así como éstos, existen otros factores que intervienen en la decisión de si reconfigurarlo o no. Algunas veces se efectúa de manera automática, y en otras, se requiere la intervención del administrador; pero en general, se recomienda dejar la configuración inicial. Si se llegan a presentar errores o circunstancias por las cuales sea necesario modificar los valores, se debe determinar qué parámetros son los que requieren el cambio, así como su nuevo valor, antes de efectuarlo; de caso contrario abstenerse, ya que los valores default suelen ser los óptimos, y si se eligen y cambian indiscriminadamente, pueden traer resultados inesperados.

III.II.II. Archivos del kernel

Toda la información referente al kernel, archivos fuente, objeto, de compilación, librerías, etc. se encuentra localizada bajo el directorio */var/sysgen*. Dentro de los archivos ubicados en este directorio, destacan por su importancia los siguientes:

root — Liga a */usr/cpu/sysgen/root*. Este directorio actúa como un directorio raíz virtual, que contiene entre otras cosas, las herramientas para efectuar la compilación y generación del nuevo kernel: *root/usr/bin/cc* es el compilador y *root/usr/bin/ld* es el cargador o ligador que permite unir los diferentes módulos a fin de generar el programa ejecutable.

boot — Liga a */usr/cpu/sysgen/IP22boot*. Contiene archivos objeto que son cargados y ligados al kernel durante su construcción, si es especificado en los archivos de configuración del directorio *system*.

master.d — Directorio conocido como "Base de datos de configuración maestra". Estos archivos son utilizados por *lboot* durante la generación del kernel. Cada uno contiene información que permite configurar los diferentes módulos y controladores de dispositivos que serán colocados dentro del kernel. Cada archivo lleva el nombre del módulo que define.

stune — Archivo que contiene los cambios locales efectuados a los parámetros del kernel en el equipo; es decir, los valores asignados en este archivo, reemplazan a los default.

mtune — Directorio de parámetros afinables del sistema. Contiene archivos describiendo todos los parámetros que pueden ser modificados, a fin de mejorar el funcionamiento del equipo. Los distintos parámetros están agrupados en módulos de acuerdo a la función que realizan. Cada módulo lleva un nombre que generalmente concuerda con el del archivo donde se encuentran agrupados.

system — Directorio de información y configuración del sistema. Contiene archivos que permiten configurar el sistema, determinando de qué hardware dispone el equipo, así como qué módulos y controladores del directorio *master.d*, serán colocados dentro del kernel por las herramientas como *lboot* o *autoconfig*, durante la creación del SO.

edt.list y *master.c* — Archivos creados durante el proceso de generación del kernel.

III.II.III. Parámetros configurables

El directorio *mtune* contiene una serie de archivos, dentro de los cuales, se describen cada uno de los parámetros que pueden ser modificados a fin de mejorar el rendimiento del kernel. En cada definición se indica su valor por defecto y los posibles rangos que pueden adquirir, así como si el cambio puede ser efectuado mientras está corriendo el SO, o si es necesario regenerar y cargar el nuevo kernel, a fin de que surta efecto. Aunque se puede modificar directamente cualquiera de estos archivos, es recomendable que no se haga, y que si se desea realizar un cambio en ellos, se efectúe en el archivo */var/sysgen/stune*; ya que como se mencionó, estos valores reemplazan a los establecidos en el directorio *mtune*. De forma análoga, aunque el cambio se puede efectuar manualmente utilizando un editor de textos, es ampliamente recomendado utilizar herramientas como *systune*, que a demás de efectuarlo, se cerciora de que el nuevo valor se encuentre dentro del rango válido para dicho parámetro y que no cause conflictos antes de realizarlo.

Los parámetros, que definen la estructura que tendrá el kernel y mantienen el control de procesos, archivos, y actividad del sistema, se encuentran agrupados de acuerdo a la función que desempeñan. Cada grupo se encuentra concentrado dentro de un archivo, cuyo nombre concuerda con el del grupo. Algunos de los más importantes se encuentran en la Tabla 13.

Tabla 13 Grupos de parámetros configurables en el kernel.

GRUPO	Descripción
Parámetros generales	Especifican el tamaño de la mayoría de las estructuras mantenidas por el sistema. Suelen ser las que generalmente requieren ser modificadas por el administrador. Entre ellas se encuentra la de *nproc*.
Parámetros de límites del sistema	Establecen los límites de ciertos recursos del sistema, como: *maxup*, *ngroups_max*, etc. Pueden modificarse de acuerdo a las necesidades.
Parámetros de límites de recursos	Establece los límites por cada proceso individual ejecutado en el sistema, p. ej: *rlimit_vmem_cur* y *ncargs* entre otros.
Parámetros de paginación.	Determinan el tiempo y condiciones en las cuales, procesos como *vhand* y *bdflush*, efectuarán su labor de paginación[27] para liberar memoria RAM. Se encuentran parámetros como *bdflushr* y *maxfc*. Generalmente los valores establecidos son los más adecuados.
Parámetros IPC	Definen el tamaño de las estructuras para mantener la comunicación entre los procesos (Inter Process Communication). Dentro de este gran conjunto, se pueden destacar variables que definen: parámetros, semáforos, mensajes y memoria compartida.

[27] Es un término empleado para indicar qué páginas de información serán intercambiadas entre la memoria RAM y área de swap en el disco duro.

Éstos son sólo algunos de los grupos existentes, que contiene parámetros configurables como los descritos en la Tabla 14 (una descripción completa puede ser consultada en el Apéndice "A" del manual "IRIX Advanced Site and Server Administration Guide", o al examinar cada uno de los archivos localizados en el directorio *mtune*).

Tabla 14 Parámetros configurables del kernel.

Parámetro	Descripción
bdflushr	Especifica que tan seguido el demonio *bdflush* será ejecutado. Su valor debe estar dentro del rango de 1 a 31536000. El valor default de 5 segundos es el más adecuado; incrementar su tiempo puede causar que la cantidad de datos del disco que se pierda, en el supuesto caso de que ocurra una caída de sistema, se incremente. Disminuirlo, intensificaría la carga de trabajo del sistema.
maxfc	Establece el número de páginas que *vhand* puede liberar en una simple operación. El valor default de 100, es el más adecuado; por lo que no se recomienda modificarlo. Su rango válido es de 50 a 100.
maxup	Es el número de procesos que cada usuario, o cuenta en particular, puede ejecutar en un momento dado. El valor default es de 150; por lo que cada usuario puede ejecutar 150 procesos simultáneamente, que satisface la mayoría de los requerimientos individuales. Si se desea, se puede incrementar su valor en el rango de 5 a 10000; pero nunca debe ser mayor que *nproc -5*. Esto se debe a que como mínimo deben existir 5 procesos del sistema que siempre están corriendo (*init*, *sched*, *vhand*, *bdflush* y *pdflush*). Si se quiere limitar el número de aplicaciones que pueda ejecutar cada usuario en un medio ambiente de trabajo muy cargado, se puede reducir este valor.
ncargs	Especifica el tamaño máximo de argumentos, en bytes, que pueden ser pasados a un proceso mediante una llamada de sistema *exec*. El valor default es de 20480, pero puede variar en el rango de 5120 a 262144. Este valor suele ser el adecuado; pero si por los requerimientos de trabajo se presentan errores continuos como "*E2BIG arg list too long*", se puede incrementar para solucionarlos.
ngrups_max	Determina el número máximo de grupos a los cuales puede pertenecer simultáneamente un usuario o clave. El valor default es 16, pero puede variar de 0 a 32. Sólo debe ser modificado si existen usuarios que requieren pertenecer a más de 16 grupos de trabajo simultáneamente en el equipo.
nproc	Especifica el número de procesos que pueden ejecutarse simultáneamente en el sistema; es decir, especifica el número de entradas que tendrá la tabla de procesos mantenida en RAM por el sistema. Este

Parámetro	Descripción
	parámetro puede variar en el rango de 30 a 10000. Si se especifica el valor de 0, el default, se establece que su valor sea determinado automáticamente en base a la cantidad de memoria RAM de que disponga el equipo. Si debido a la cantidad de usuarios y carga de trabajo del equipo, se presentan mensajes continuos como "no more processes", "too many processes" o "No process slots", indicando que no se pueden generar más procesos, se debe determinar si la causa es a que el usuario alcanzó su límite máximo de procesos individuales (*maxup*) o se ha saturado la tabla de procesos del kernel, en cuyo caso se debe incrementar el valor de esta variable. Por el contrario, si se establece un número grande, se está desperdiciando memoria RAM para mantener entradas para toda la cantidad de procesos que pueden correr, pero que probablemente nunca se ejecuten simultáneamente.
Rlimit_vmem_cur	Determina el límite máximo de memoria virtual que puede ser utilizada cuando se ejecuta un proceso. El rango válido para este parámetro es de 0 a 2 GB. El valor asignado suele ser el más adecuado, por lo que no es conveniente modificar este parámetro.

Éstos son sólo algunos ejemplos, pero de ellos se destaca que pueden existir dependencias entre los diversos parámetros; por lo que al modificar uno, se pueden alterar varios. Por tal motivo, cuando se afine el sistema es recomendable modificar uno a la vez y determinar si se obtuvo el resultado deseado o no. Cuando se ha encontrado el valor deseado de uno, se puede proceder con otro, y así hasta terminar con el proceso.

III.II.IV. Proceso de reconfiguración

La generación de un nuevo kernel puede efectuarse utilizando varias herramientas como *lboot* y *systune*, o de forma automática mediante el empleo de rutinas como la de */etc/autoconfig*, que serán descritas más adelante; pero en cualquiera de los casos es conveniente entender cuál es el proceso de generación, antes de llevarlo a cabo. Desde un punto de vista general, éste es muy simple, como se ilustra en la Fig. III-14.

En base a los archivos maestros de configuración y los que especifican los valores de los diversos parámetros, se deben tomar los archivos fuente para ser compilados y generar archivos objeto, que serán ligados junto con archivos objeto y librerías propias del sistema, para formar el nuevo programa ejecutable.

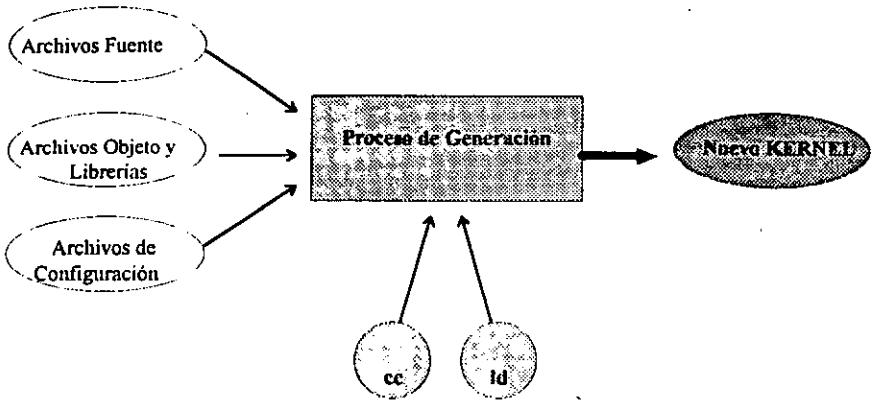


Fig. III-14 Proceso de generación del kernel.

En otro aspecto, no debe olvidarse que es sumamente importante el realizar un respaldo completo del sistema antes de efectuar cualquier cambio, a fin de poder regresar al estado original, en caso de cualquier problema. También, es recomendable realizar una copia de seguridad del kernel original y del archivo *stune*; ya que son alterados durante el proceso de generación, y esto nos permitirá restablecer el kernel original sin la necesidad de restaurar todo el respaldo del sistema:

 # cp /unix /unix.back
 # cp /var/sysgen/stune /var/sysgen/stune.back

De igual forma, independientemente del método o herramienta empleada, existe un paso que debe efectuarse y que nos permitirá determinar si es necesario el reconfigurar o no, el kernel: Rastrear y encontrar los parámetros a modificar. En este sentido, se pueden utilizar herramientas como *sar*, *ps*, *osview*, etc. y las examinadas en "Monitoreo del sistema", pág. V-5, para determinar el rendimiento del equipo, y en base a ello, localizar los parámetros que requieren de algún cambio. También, si se presentan mensajes en la consola indicando problemas con ciertos procesos, se debe investigar para determinar si es debido al kernel, o a alguna aplicación en específico; para ello, es aconsejable repasar lo visto en el inciso anterior y estudiar las características y cada uno de los parámetros configurables. En cualquier caso, es importante determinar qué variable es la que requiere el cambio y cuál debe ser su nuevo valor. Para este último punto, es conveniente obtener antes, un listado de los parámetros del sistema y valores actuales; para lo cual, se puede examinar el contenido del archivo */var/sysgen/stune*, del directorio */var/sysgen/mtune* o utilizar el comando:

 # systune

Una vez conocido y determinado el valor que se desea asignar a cualquier parámetro, se debe emplear algunas de las siguientes técnicas para efectuar la generación del nuevo kernel.

III.II.IV.I. lboot

Ésta es la principal herramienta utilizada para generar un nuevo kernel. Las demás ofrecen mejoras y rutinas que permiten prever cualquier tipo de error, por lo que representan una mejor alternativa; pero para realizar su labor, utilizan a *lboot*.

En cuanto al proceso de generación, una vez localizados los parámetros a modificar, el proceso es relativamente simple: efectuar los cambios, generar el nuevo kernel y ponerlo a funcionar.

Cuando se utiliza este método, se debe modificar manualmente el contenido del archivo */var/sysgen/stune*, a fin de colocar en él, los nuevos valores que se deseen establecer a los parámetros del sistema. Recordar que no es conveniente alterar los archivos fuente localizados en el directorio *mtune*, y que es recomendable modificar un parámetro a la vez, para no confundirse. Este paso puede ser hecho con la ayuda de cualquier editor de textos. Se debe tener cuidado de conservar el formato, que consiste en utilizar una linea por parámetro, especificando su nombre, un signo de igual y el valor asignado:

mproc = 1000
maxup = 100

Una vez realizado lo anterior, se debe utilizar el comando *lboot*, que conoce e interpreta la información contenida en el directorio */var/sysgen*, a fin de generar un nuevo kernel. Básicamente su función es la siguiente: lee la información contenida en los archivos del sistema, ubicados en el directorio */var/sysgen/system*, para determinar qué módulos contenidos en el directorio */var/sysgen/master.d* serán incluidos dentro del kernel. Por cada módulo incluido, se examina el directorio */var/sysgen/boot* a fin de localizar un archivo objeto cuyo nombre sea idéntico al del módulo; si es encontrado, es incluido dentro del kernel al momento de ligar todos los módulos y generar el archivo ejecutable. El directorio */var/sysgen/mtune* es examinado para determinar los valores default de los diversos parámetros. El archivo */var/sysgen/stune* es leído para modificar y restablecer los valores definidos por el usuarios.

Las definiciones dentro de los archivos el sistema, especifican la forma de actuar durante la generación del nuevo kernel; es decir, especifican qué acciones se deben seguir para determinar si un dispositivo existe en el hardware del equipo, en cuyo caso, el controlador adecuado es cargado; qué módulo debe ser excluido, etc. Para compilar los archivos fuente y generar archivos objeto, se utiliza el compilador *cc* ubicado dentro del directorio */var/sysgen/root*; y para ligar todos los archivos objeto y formar el ejecutable, se emplea el programa *ld* del mismo directorio. Por tal motivo, una vez establecido y actualizado el nuevo valor del parámetro en el archivo *stune*, se debe ejecutar el comando:

 # lboot

Por default, coloca el nuevo kernel en el archivo */unix.new*, por lo que se debe utilizar este archivo para reemplazar al kernel original a fin de que la próxima vez que se arranque el sistema, sea cargado y los cambios entren en operación. Por ello, se debe ejecutar el comando:

 # cp /unix.new /unix
 # reboot

La última instrucción reinicia el equipo para que entre en función el nuevo kernel. En este tipo de método se pueden presentar problemas que impidan arrancar el SO, por lo que se debe tener cuidado al establecer el valor de los parámetros, que no salgan del rango válido. Si se llega a presentar cualquier problema, se deben restaurar los archivos */unix.back* y */var/sysgen/stune.back* con sus nombres originales y arrancar nuevamente el equipo. Esta labor puede ser efectuada desde miniroot, si no puede arrancar el nuevo kernel.

III.II.IV.II. Generación automática

Existen varias circunstancias durante las cuales, se puede efectuar una regeneración automática del Kernel. Si se añade un nuevo dispositivo al equipo, más memoria RAM, otro CPU, una nueva tarjeta controladora, etc., se puede utilizar el comando */etc/autoconfig* para generar un nuevo kernel que contemple los controladores adecuados para los nuevos dispositivos o el incremento en la memoria RAM, que como ya se describió, interviene en el número de procesos que se pueden ejecutar dentro del equipo.

Es importante indicar que si se añade un nuevo disco que es soportado por la tarjeta controladora ya instalada, no es necesario regenerar el kernel; ya que el software controlador de la tarjeta se encuentra aún definido en el kernel, y éste puede manejar uno o varios discos. Por el contrario, si se añadió un disco diferente que requiere una tarjeta controladora diferente a la instalada, si es necesario reconfigurarlo para incluir el módulo adecuado para dicha tarjeta. Para generar el nuevo kernel simplemente se ejecuta el comando:

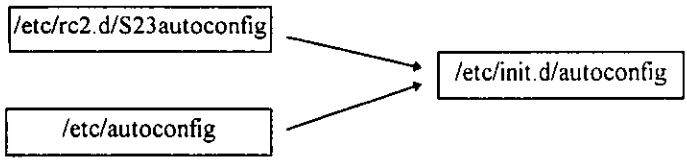
 # /etc/autoconfig -f
 # reboot

Ya que los archivos de sistema (*/var/sysgen/system*) tienen entre sus funciones el determinar de qué hardware dispone el equipo para cargar los módulos adecuados. El primer comando crea un nuevo kernel y lo coloca en el archivo */unix.install*. La opción *-f* le indica que lo genere, aunque parezca que no ha sido añadido ningún hardware o software al equipo; por lo que este comando también puede ser empleado cuando se añade un programa especial, como se verá más adelante. El segundo reinicia el equipo, para lo cual, se ejecutan los procesos colocados en el directorio */etc/rc0.d*, dentro de los cuales se encuentra uno que si

localiza el archivo */unix.install*, lo renombra a */unix*. Con ello entra en función el nuevo kernel la próxima vez que se arranque el equipo, y el nuevo SO podrá accesar el dispositivo añadido.

Por otro lado, el proceso de autoconfiguración también se efectúa automáticamente cuando se instala algún software. Si se utiliza el comando *inst* para instalar aplicaciones, nuevos módulos del sistema, etc., éste determina si es necesario generar un nuevo kernel para que entren en operación. Si es así, efectúa esta labor y posteriormente reinicia el equipo, a fin de que tengan efecto los cambios.

Finalmente, dentro del directorio */etc/rc2.d* se encuentra el comando *S23autoconfig* que se encarga de determinar si han sufrido cambios los archivos originales del sistema, y si el kernel se encuentra actualizado. Para esta función se basa en las fechas de creación de los archivos. Si existe una discrepancia indicando que la versión del kernel no se encuentra actualizada con la de los archivos fuente, corre el proceso de generación para que entre en función inmediatamente la versión actual. Por este motivo, cada vez que se inicia el equipo se checa, y en caso de ser necesario, se genera un nuevo kernel. Es importante mencionar que este programa, que se ejecuta con la opción START al arrancar el equipo, al igual que */etc/autoconfig*, son una liga al archivo */etc/init.d/autoconfig*, por lo que finalmente estos tres son el mismo archivo.



Es por esto que cualquier cambio, tanto de software como de hardware, ocasiona que se genere de forma automática un nuevo kernel, y en muchas ocasiones, sin que se entere la persona que instala, enciende o apaga el equipo.

III.II.IV.III. systune

La herramienta idónea para realizar cambios en los valores de los parámetros establecidos en el kernel, es la de *systune*. Permite examinarlos y ajustarlos en tiempo real, sin apagar el equipo; si alguno lo requiere, puede reiniciar el sistema para que entren en función. Si se van a ajustar parámetros, ya sea por problemas que se presenten o para afinar el sistema, ésta es la herramienta que se recomienda ampliamente; ya que antes de efectuar cualquier cambio,

se cerciora de que el valor especificado se encuentre dentro del rango válido para él, además de prevenir situaciones en las cuales, se pueda generar un kernel que en determinado momento no pueda ser cargado y arrancar, debido a los cambios que ha sufrido.

Básicamente tiene dos modos de operación. El primero y más simple, consiste en ejecutarlo sin ninguna opción, en cuyo caso despliega un listado de los valores asignados actualmente a los parámetros. Es conveniente antes de efectuar cualquier cambio y antes de empezar el proceso de regeneración y afinación del kernel, obtener un listado de los valores actuales. Para ello:

 # systune

El segundo es un modo interactivo, en el cual, se puede desplegar el valor de algún parámetro, grupo o categoría, así como cambiarlos. *systune* determina si es necesario reiniciar el equipo para efectuar el cambio, o puede ser modificado directamente en el kernel que se encuentra corriendo. Dependiendo de ello y de las opciones dadas al ejecutar el comando, los cambios realizados se verí reflejados en:

- El archivo */var/sysgen/stune*, que contiene los parámetros definidos por el usuario.
- El archivo */unix* o */unix.install*, que contiene el nuevo kernel cuando es generado.
- Dentro del kernel que se encuentra corriendo, si el parámetro lo permite.

Para entrar en el modo interactivo se utiliza la opción *-i* como se muestra a continuación:

 # systune -i

En este modo se disponen de comandos para desplegar y modificar los valores asignados a los parámetros así como el comando *help*, que muestra una descripción de cada uno de los comandos. Por ejemplo, para cambiar el valor de una variable, simplemente se escribe su nombre y su nuevo valor:

*systune -> **nproc 800***

Finalmente, al ejecutar el comando *quit*, y dependiendo de las opciones indicadas, se determina si es necesario generar un nuevo kernel, el cual es colocado en la raíz del SA con el nombre de *unix.install*. Posteriormente se reinicia el equipo; recordar que el comando *init 0* que permite apagar el equipo, ejecuta dentro de sus labores el copiado del archivo */unix.install* a */unix* a fin de que entre en función esta nueva versión del kernel la próxima vez que se arranque el equipo.

III.II.IV.IV. Consideraciones finales

- El reconfigurar el kernel puede mejorar el rendimiento del equipo, pero tiene un límite que es fijado por el rendimiento de los periféricos instalados. Es decir que si después de regenerar y afinar el kernel, los SA, el área de swap, etc., se ha llegado al máximo rendimiento del equipo y aun así se presentan problemas, la alternativa es incrementar o actualizar los dispositivos para solucionarlo; ya sea añadiendo un nuevo disco, más memoria RAM, otro CPU, o incluso, hasta adquirir un equipo superior que dé soporte a la cantidad y demanda de las aplicaciones que se requieren ejecutar.

- Debe guardarse una copia del kernel y los valores asignados originalmente, a fin de poder restaurarlos si los cambios efectuados no son lo suficientemente satisfactorios, después de realizar su análisis. Ello implica que se debe guardar un registro de todos los cambios y procesos efectuados.

- Cuando se desee modificar algún parámetro, de todos los comandos y métodos vistos, *systune* es el recomendado; por lo que debe ser estudiado minuciosamente.

CAPÍTULO 4

SEGURIDAD

Consejos, puntos de interés, referencias y herramientas que permiten incrementar el nivel de seguridad establecido dentro del equipo.

TABLAS

FIGURAS

IV. SEGURIDAD

La seguridad es la principal preocupación de todo administrador. Un sistema UNIX puede ser una gran herramienta, por todo el potencial que ofrece al permitir la comunicación con otros sistemas y compartir información de una manera simple y sencilla, tanto local como en forma remota; en modo terminal o ambiente gráfico. Pero, debido a ello, también ofrece un mundo de puntos vulnerables para violar su seguridad.

Según Ritchie, Dennis M. en su libro " On the Security of UNIX " Mayo 1975: los sistemas operativos UNIX, aunque hoy en día son muy utilizados en medio ambiente que requieren seguridad, no fueron diseñados teniéndola en mente. Todo esto parte del hecho de que el sistema UNIX fue creado por programadores en los laboratorios Bell, donde el propósito era el de compartir, y permitir que la información viajara sin tantas restricciones de seguridad. Por otro lado, debido a sus características pronto fue empleado por universidades, el gobierno y empresas, hasta llegar a los días actuales en los que es utilizado en una gran variedad de sectores y tareas, donde la seguridad es vital; ya que en los equipos donde corre se manejan transacciones de dinero, como en bancos y nóminas empresariales; trabajos de investigaciones y desarrollo, que es información secreta de las empresas y gobierno; etc.. Por todo esto, se han implementado herramientas, métodos y dispositivos que permita dar seguridad tanto al equipo como al sistema.

La seguridad puede ser violada ya sea por personas o programas (virus) y las técnicas y herramientas utilizadas para prevenir, monitorear y solucionar esta situación son variadas. Podemos dividir el estudio de la seguridad en los siguientes puntos:

a) seguridad del equipo.
b) seguridad en el acceso al sistema.
c) seguridad dentro del sistema
d) seguridad en la red
e) software malicioso
f) medidas de recuperación.

De esto podemos decir en resumen, que para impedir que una persona pueda violar nuestra seguridad: primero se debe de impedir el acceso al equipo, ya sea en forma directa o remota (a) y (d); si logra tener el acceso, impedir la entrada al sistema (b); si logra entrar al sistema, tener medidas de seguridad que impidan su libre desplazamiento para que el daño sea mínimo, así como monitorear la actividad del sistema para detectar cualquier anomalía rápidamente (c) y si llega a causar daño, tener implementadas medidas que nos permitan recuperarnos, determinar la causa y erradicar la falla en la seguridad que ocasionó el problema, para que no vuelva a ocurrir en el futuro (f). Finalmente se tiene que contemplar el tema del software malicioso, dentro del cual se encuentran los tan famosos virus, para conocer cuál es la forma en que actúan, cómo atacan y cómo protegernos de ellos.

Por último, en algunos documentos y en el argot computacional, a las personas que intentan violar la seguridad de un equipo suelen llamárseles cracker. No confundir este término con el de hacker, que según se define en The Hacker's Dictionary 1988, es una persona que se deleita aprendiendo los detalles de la programación de sistemas y cómo extender sus capacidades - a contrario de la mayoría de los usuarios que prefieren aprender sólo lo necesariamente mínimo.

IV.I. Seguridad del equipo

Este punto tiene que ver con la seguridad que se tenga, tanto para el acceso al equipo, como la implementada dentro de él para impedir ser violado; por ello se describirán tanto las causas de por qué implantar políticas de uso, como las medidas de seguridad que tienen incorporadas los equipos.

IV.I.I. Políticas de uso y otras medidas

Este punto pretende señalar que la seguridad no sólo depende de programas y comandos del sistema operativo; sino de establecer medidas que permitan controlar las acciones de los usuarios, como lo es el educarlos para que tomen medidas que incrementen la seguridad, el capacitarlos en el correcto uso del equipo, el establecer políticas de uso y difundirlas entre los usuarios, etc..

Al igual que el mantener un equipo en un ambiente adecuado (ver pág. I-3) permite prolongar su vida útil, el mantener controlado el acceso a él, permite minimizar los riesgos de posibles intentos de violación, así como fallas. Esto se debe implementar con una serie de políticas de uso, al igual que educando y capacitando a los usuarios; ya que la seguridad es responsabilidad de todas las personas que lo utilizan, y no sólo del administrador del sistema.

Las políticas de uso deben definir y contener los elementos que permitan controlar y mantener seguro tanto el equipo, como la información dentro de él. Deben incluir los requisitos a cumplir por parte de los usuarios: quién puede tener acceso, qué condiciones debe satisfacer, en qué horario debe hacerlo, etc.; así como los derechos, responsabilidades, sanciones y el uso adecuado que debe dársele al equipo. Con estas simples medidas de control, se pueden detectar personar ajenas y no autorizadas que traten de tener acceso a él, así como minimizar el riesgo de intentos de violación. Si no se lleva alguna medida similar a ésta, y si se permite el acceso a cualquier persona sin ningún control, el detectar los

intentos, como las causas de posibles fallas que pueda tener el equipo debido al mal uso que se le da, resulta muy difícil de solucionar y erradicar .

Por otra parte, el educar a los usuarios en las técnicas y medidas de seguridad que deben tomarse (que se describirán a lo largo de este capítulo), aumenta la seguridad del sistema. Como se describirá más adelante, la mayoría de las violaciones se dan por los malos hábitos y el mal uso que hacen los usuarios del equipo: el asignar claves secretas vulnerables a sus cuentas, el no mantener confidencial y segura tanto su cuenta como su clave secreta, el dejar encendidos los equipos sin salirse de sesión, etc. Por esto y otras causas, es de vital importancia el educarlos en las medidas que deben tomarse a fin de prevenir situaciones como éstas.

El capacitar a los usuarios y mantenerlos actualizados en el correcto uso del equipo y del sistema operativo, permite mantener también la seguridad, reducir las fallas, al igual que agilizar las labores y minimizar la carga de trabajo del equipo. Por el lado de la seguridad: si conocen el sistema en forma correcta, podrán utilizar los comandos adecuados para proteger su información y dar acceso únicamente a las personas autorizadas para ello, incrementándose la protección dentro del sistema y, reduciendo y complicando las labores de posibles intrusos. Por el lado del rendimiento: si las personas que hacen uso del equipo, saben cuáles son las herramientas y comandos más adecuados para realizar sus labores, al igual que cómo utilizarlos, las efectuarán de manera eficiente, no sobrecargando el sistema con procesos inútiles y requerirán de un tiempo menor para efectuarlas, incrementando por tanto, su rendimiento. Finalmente, por el lado de las fallas se tiene que, si a los usuarios se les educa y se les indica cuál es la manera de encender el equipo, de apagarlo, el de utilizar en forma correcta los distintos dispositivos a los que tienen acceso (impresoras, unidades de CDROM, cámaras de video, micrófonos, el ratón, el teclado, plotters, etc.), así como en el cuidado que se les tiene que dar, permite que las fallas que se presenten en éstos, sean mínimas, incrementándose su vida útil.

Como puede observarse, el llevar estas simples medidas y políticas de uso, además de incrementar la seguridad y mejorar el rendimiento del equipo y los usuarios, permite el ahorro de dinero; al presentarse menos fallas del equipo e incrementar el desempeño del personal, que realizan las labores en un menor tiempo y con mejores resultados.

IV.I.II. Seguridad incorporada al hardware

Generalmente, todos los equipos cuentan con métodos que permitan instalar el sistema operativo a un disco de un equipo nuevo, que permitan arrancarlo para trabajar con él, al igual que con herramientas para darle mantenimiento en caso de fallas del sistema, etc. Estas herramientas se encuentran grabadas en un circuito del equipo y se denominan PROM o monitor PROM (ver pág. I-14).

Por ejemplo, si no se tiene un dispositivo para poder colocar, ya sea, un CD o cinta e instalar el SO, se puede utilizar el PROM del equipo e indicarle que lo cargue de un equipo remoto que se encuentra conectado a una red; si durante el uso del equipo se llegan a dañar algunos archivos y el SO no puede arrancar en forma adecuada, se utiliza el PROM para arrancar de otro SO y corregir el problema; Si se llega a olvidar la clave secreta del usuario root, se puede mediante el PROM, utilizar el programa *sash* o miniroot para quitársela o cambiársela; Si se requiere, desde el PROM se puede dar formato a cualquier disco, borrando toda su información, para instalar un SO nuevo, etc.

Por éstas y otras razones, se debe conocer bien los alcances y limitaciones que tiene el PROM de cada equipo, ya que como se ve, puede ser un punto vulnerable en la seguridad. De igual forma, de esto se puede ver la importancia de qué personas son las que tienen derecho a utilizar el equipo en forma directa, y en especial, en los momentos cuando se encuentre apagado; ya que utilizando el PROM se puede violar la seguridad del sistema, que en esos momentos aún no se encuentra funcionado, y alterar, borrar o modificar la información almacenada en él. Si una persona tiene acceso al equipo, puede simplemente apagarlo, encenderlo nuevamente e interrumpir la secuencia normal de encendido (ver pág. I-10) para tener acceso al PROM, desde donde puede alterar la información del sistema.

Existen equipos que no tienen implementado ningún tipo de seguridad en el PROM y resulta muy sencillo arrancar el sistema y obtener los privilegios de root sin ningún contratiempo o dificultad, en unos cuantos segundos o minutos. A este tipo de equipos se les debe dar una mayor seguridad durante el tiempo que se encuentran trabajando, y en especial, cuando se encuentran fuera de operación, restringiendo el acceso a ellos de personal no autorizado.

En particular, la estación de trabajo INDY, cuenta con la posibilidad de poner una clave secreta que impida el acceso en forma libre al PROM. De esta forma, si durante el arranque del equipo alguna persona intenta tener acceso al PROM, le será solicitado que la introduzca.

La clave secreta del PROM puede ser colocada desde el PROM directamente, o desde el SO una vez que éste ha arrancado. Para colocarla desde el SO ejecutar el comando:

 # nvram passwd_key "**XXXXXXXX**"

donde: **XXXXXXXX** es la clave secreta que se desea colocar.

Para ejecutar este comando se debe estar en la cuenta de root. Para colocarla desde el PROM:

- Encender el equipo e interrumpir el proceso normal de arranque para tener acceso al PROM; pasos del 1 a 4 de "Instalación del Sistema Operativo", pág. I-21.
- Seleccionar la opción 5 'Command Monitor' para ingresar al monitor PROM.

- Ejecutar el comando:

>>passwd

- Aparecerá un mensaje indicando que se teclee el password[28], después del cual se pedirá que se teclee nuevamente para confirmarlo:

Enter new password: **XXXXXXXX**
Confirm new password: **XXXXXXXX**

- Si el password tecleado en ambas ocasiones es idéntico, éste será la nueva clave secreta que pedirá el equipo al momento de ingresar al PROM.

Cabe destacar que ésta es únicamente una medida de seguridad más, pero a pesar de ello, una persona que tenga acceso al equipo lo puede abrir y tomar el disco para instalarlo en otro equipo y obtener así la información guardada en él; o simplemente puede desactivar la protección de la clave secreta del PROM, al modificar interruptores que se encuentran en la tarjeta madre y que inhiben esta protección. Por estas razones, en la parte posterior del equipo se localiza un orificio que permite colocar un candado e impedir, o dificultar de alguna manera, cualquier intento de abrir o mover el CPU.

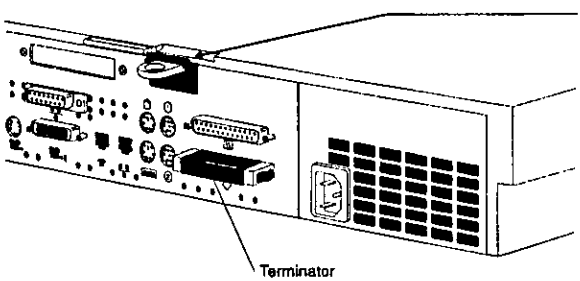


Fig. IV-1 Ubicación del orificio para cadena de seguridad.

Como puede observarse, la seguridad resulta ser un asunto muy complicado, no tan sólo cuando el SO se encuentra funcionando y su capacidad de protección se encuentra activada; sino también en los momentos en que está fuera de servicio y se encuentra vulnerable. Lo que se debe hacer es reducir los riesgos, al implantar y tomar las medidas pertinentes para ello.

[28] El password es la clave secreta, que sólo el usuario interesado conoce.

IV.II. Seguridad en el acceso al sistema

Si un persona logra tener acceso al equipo, la mayoría de los sistemas cuentan con medidas que permiten impedir que el intruso entre en él. En este renglón, los sistemas UNIX cuentan únicamente con el sistema de seguridad para las cuentas de acceso, conocidas como login.

Para que el sistema deje entrar a una persona, y ésta pueda trabajar dentro de él, debe identificarse como un usuario legítimo. De esta forma, el sistema puede asignarle privilegios y delimitar las acciones y comandos que pueda ejecutar, sobre los archivos e información almacenada dentro del equipo, así como el tiempo que le está permitido trabajar. Este mecanismo de identificación es a través de una cuenta o login y un password o clave secreta.

A cada usuario se le asigna una cuenta y una clave secreta, y es su responsabilidad el mantenerla confidencial y segura, así como reportar cualquier anomalía que note durante su uso (debe estar especificado dentro de las políticas de uso). Cuando una persona tiene acceso y se conecta al equipo, lo primero que se le pide que introduzca es la cuenta que se le asignó, después se le pide su clave secreta; si son válidos, se le dan los privilegios que le fueron asignados y se le permite el acceso; en caso contrario se rechaza la petición. De estos dos, el más vulnerable es la cuenta, ya que al teclearla aparece en la pantalla y cualquier persona que esté cerca la puede observar; en cambio, la clave secreta es una palabra confidencial que únicamente el dueño la conoce, por lo que para mantenerla segura no aparece en la pantalla cuando se escribe. Por esta razón, escoger una clave secreta adecuada es importante, y las reglas que se deben seguir se explicarán más adelante.

Todos los sistemas cuentan básicamente con este mecanismo de protección, pero las mejoras que se van incorporando a través del tiempo, permiten incrementar y mejorar su funcionamiento; por lo que es importante conocer todas y cada una de las características que pueda tener su sistema en particular. En esta sección se describirán las facilidades con que cuenta en particular, el sistema operativo IRIX; para ello, primero se describirán las características y consideraciones que deben tomarse en la asignación y uso de la cuenta y de la clave secreta y después, se describirá la secuencia de entrada al sistema así como la seguridad que se puede implementar en éste.

IV.II.I. La cuenta o login

Como se mencionó, la cuenta es el mecanismo para identificar a un usuario. Una cuenta es un nombre que consta de 8 caracteres alfanuméricos como máximo. Si se pretende que ésta sea portable a través de múltiples versiones y SO UNIX, debe comenzar con una letra, seguida de letras o números; además, sólo deben utilizarse letras minúsculas.

Cada cuenta puede y debe tener una clave secreta para protegerla, ya que como se mencionó, ésta resulta ser la parte más vulnerable. Es por ello que se debe mantener un estricto control sobre este medio: bloqueando las cuentas inactivas, conociendo el propósito, funcionamiento y alcances de cada cuenta del sistema, llevando un control sobre todas las cuentas existentes, etc. Estos puntos serán tratados a continuación.

IV.II.I.I. Bloqueo de cuentas

Si una cuenta no será utilizada por un período, resulta más práctico y seguro el bloquearla. Cuando una cuenta se bloquea, se elimina la clave secreta que tenía y en su lugar se coloca una señal que le indica al sistema que se encuentra deshabilitada, impidiendo que se pueda accesar al sistema si alguien la utiliza.

Para bloquear una cuenta se puede utilizar el comando *passwd* con la opción *-l*:

 # passwd -l juan

Cuando se ejecute este comando, se bloqueará la cuenta *juan*; para ello, se sustituye la clave secreta que tenía por *LK* en el archivo */etc/passwd*, indicándole con ello al sistema, que la cuenta está deshabilitada.

Por otro lado, esta operación puede ser efectuada en forma directa, editando el archivo y realizando la sustitución manualmente. También, se puede utilizar cualquier frase que desee y que se encuentre entre comillas " ", ya que éstas no son utilizadas por el sistema de cifrado, y por lo tanto, indican que la clave está bloqueada. Por último, éste y otros sistemas suelen utilizar como marca de bloqueo al asterisco *.

Dentro de este mismo tema, es importante señalar una medida más de protección recomendada por algunos expertos: Si una cuenta ya no va a ser utilizada por un usuario, se puede obtener un grado más de seguridad si se bloquea en lugar de eliminarla.

Esto se debe al hecho de que, para que un usuario ingrese al sistema, se le asigna una cuenta a la cual le corresponde un número conocido como identificador de usuario (user ID). Para la mayoría de los efectos, se utiliza la cuenta para referenciarse al usuario; pero para el sistema operativo, y en especial para el sistema de archivos, el identificador de usuario es el utilizado. Por ello, cuando una cuenta es borrada y se crea una nueva a la cual se le asigna el identificador del usuario borrado, esta nueva cuenta tendrá los derechos (privilegios, permisos y restricciones) que tenía el usuario anterior; por lo que será reconocido como dueño de los archivos que le pertenecían al anterior usuario, aunque las cuentas tengan nombres distintos.

Este problema se puede ver aumentado, si por alguna causa se llega a dañar el sistema y se baja un respaldo de cinta. Pueden ocurrir problemas con los archivos, en especial los creados por los usuarios, al no coincidir los identificadores o determinar que algunos usuarios no existen, ya que los identificadores no son encontrados, aunque si exista una cuenta con el nombre original.

Por otro lado, el bloquear una cuenta en lugar de borrarla, puede resultar muy complicado en aquellos equipos en los cuales se realizan estas operaciones continuamente; ya que el tamaño del archivo */etc/passwd* puede crecer rápidamente, retardando la búsqueda de las cuentas, y por ende, algunos procesos que lo utilizan. Ahora, por el lado de la administración resulta más difícil llevar un control, y determinar en un momento dado, cuáles claves son válidas y cuáles no.

Por esto, la opción de si borrar las cuentas o simplemente bloquearlas queda en manos del administrador. No importando cuál sea la decisión, es importante llevar un control de las cuentas generadas y los identificadores utilizados para cada una; para evitar posibles conflictos, y aún más, detectar cualquier anomalía, como se describirá más adelante.

IV.II.I.II. Cuentas especiales

Cuando el sistema es instalado por vez primera, una serie de cuentas es generada automáticamente, las cuales tienen una función especial dentro del sistema operativo. Estas cuentas no deben ser borradas, ya que afectaría los procesos para los cuales fueron creadas. Las cuentas son:

root — Ésta es la cuenta principal del sistema. El administrador es el único que debe utilizarla, ya que el sistema no pone ninguna restricción a ésta. Por la trascendencia que tiene, y ya que todos los equipos UNIX poseen una, la clave secreta es lo único que la protege; de ahí la importancia de seleccionar una adecuada. Al ser instalado el sistema, no se le asigna ninguna clave secreta a esta cuenta; por lo que el primer paso después de la instalación, es ponerle una.

bin — Esta cuenta tiene los privilegios de un usuario normal y es dueña de una gran variedad de archivos que se encuentran distribuidos a través del sistema de archivos completo. Es conveniente bloquearla para que nadie pueda utilizarla.

adm — Ésta es una cuenta más del sistema y posee, generalmente, los archivos del directorio */var/adm* que son utilizados para la administración del sistema. También debe ser bloqueada para evitar que alguien pueda utilizarla.

lp — Esta cuenta posee el sistema de spool, que permite la impresión de documentos por medio de un sistema de colas de espera, que se encarga de controlar. Los archivos que le pertenecen se encuentran generalmente en el directorio */var/spool/lp*. A ésta no se le asigna ninguna clave secreta inicialmente, por lo que es conveniente bloquearla o ponerle una después de la instalación.

sys — Ésta es otra cuenta del sistema y los archivos que posee, se encuentran generalmente en el directorio */usr/src*.

uucp — Esta cuenta posee los directorios de */usr/lib/uucp* y */etc/uucp*.

nuucp — Esta cuenta es utilizada por estaciones remotas cuando ingresan al sistema para realizar operaciones de transferencia de archivos mediante */usr/lib/uucp/uucico*. A esta cuenta tampoco se le asigna una clave secreta, ni se encuentra bloqueada inicialmente, por lo que es importante el hacerlo.

daemon — Esta cuenta es la que controla los procesos que se encuentran corriendo en el fondo del sistema (background) llamados demonios.

Así mismo, algunas aplicaciones al ser instaladas crean y mantienen cuentas especiales que requieren para su funcionamiento. Se debe conocer y proteger estas cuentas para evitar que sean utilizadas con otro propósito para el cual fueron creadas. Al ser instalado el sistema, y dependiendo de los módulos que se instalen, se pueden generar las siguientes cuentas:

EZsetup — Ésta es utilizada para establecer los primeros parámetros del sistema (nombre, dirección IP, subred, clave secreta de la cuenta root). Se debe utilizar una vez que se ha realizado una instalación y posteriormente bloquearla o ponerle una clave secreta para protegerla.

tutor — Al ingresar a esta cuenta se tiene acceso a un programa tutor que explica el funcionamiento del equipo y permite al usuario, empezar a familiarizarse y trabajar con el ambiente gráfico; ya que el tema principal, son las distintas herramientas y forma de trabajar en este ambiente. A esta cuenta tampoco se le asigna una clave secreta inicialmente, y corresponde al administrador, la decisión de colocársela, o no. La sugerencia es que lo haga y dé el acceso a las personas que lo soliciten en forma controlada. Esta cuenta posee un problema de seguridad potencial, que permite a un usuario adquirir privilegios de root. Ver Apéndice A.

OutOfBox — Ésta es una cuenta que permite experimentar con las capacidades de video, sonido, potencia, etc. del equipo. Al igual que la cuenta anterior, carece de clave secreta y tiene una deficiencia en la seguridad. Ver Apéndice A.

demos — Es una cuenta más para el sistema; cuando se entra a ella, aparecen una serie de aplicaciones de demostración. Estas aplicaciones fueron diseñadas para trabajar en ambiente gráfico, y su propósito es el demostrar a los usuarios las capacidades del sistema. A esta cuenta no se le asigna ninguna clave secreta; por lo que cualquier usuario puede entrar a ella. Es responsabilidad del administrador el dejarla así, para que la puedan utilizar los usuarios, o el ponerle una, y permitir la entrada sólo cuando se solicite explícitamente o se desee realizar alguna demostración.

Guest — Es la cuenta anfitrión o para invitados. Es utilizada por algunas aplicaciones del sistema, para tener acceso y poder compartir información y recursos, como unidades de cinta y CDROM, entre los distintos equipos (con sistema operativo IRIX) a través de la red. Para que algunas de estas operaciones puedan efectuarse, la cuenta no debe tener ninguna clave secreta. La recomendación en este sentido, es que se bloquee o se le ponga alguna, y sólo cuando se requiera, se le quite, se utilice, y se vuelva a bloquear; ya que puede ser un punto vulnerable en la seguridad.

Como último punto, es importante mencionar que se deben revisar las cuentas existentes, y vigilar que no exista ninguna libre; por lo que es conveniente bloquearlas o asignarles una clave secreta, tomando especial atención a las mencionadas en este inciso, las cuales se suponen existen en todos los sistemas IRIX.

IV.II.I.III. Control de las cuentas

Como se puede observar de los puntos anteriores y de las razones expuestas en los siguientes, se debe llevar un buen control de las cuentas existentes, las cuales se encuentran inscritas en el archivo */etc/passwd*.

Otra razón más para llevar un control, y en especial un registro de las cuentas del sistema, es que cuando una persona ha logrado infiltrarse en el equipo, ésta suele crearse una para no llamar mucho la atención. Si no se lleva un control de las cuentas reales del sistema, será difícil saber cuáles son las válidas y cuáles fueron creadas sin autorización.

Además, se pueden utilizar herramientas y comandos que verifican y pueden determinar posibles problemas o fallas en el archivo */etc/passwd*; como lo es el comando *pwck*. Este comando checa cualquier inconsistencia, y puede validar cada uno de los campos de que consta una entrada en este archivo. Si se a generado un archivo de seguridad shadow[29], éste

[29] Ver pág. IV-26, Archivo de seguridad shadow

también es verificado. Al igual que este comando, existe otro que verifica la integridad de los grupos del sistema[30]; los cuales se encuentran en el archivo */etc/group*. Este comando es el *grpck*.

IV.II.II. La clave secreta o password

Como se mencionó, la clave secreta es la parte fundamental en el sistema de seguridad de las cuentas de acceso al sistema. Es ampliamente reconocido que la mayoría de las transgresiones al sistema, se dan al haberse encontrado las claves secretas asignadas a las cuentas. En este ámbito, se han hecho estudios en los cuales se a detectado una serie de hábitos en los usuarios al momento de asignar la clave secreta a su cuenta; así como otros en los que se han tratado de violar las cuentas, al probar con una gran variedad de claves hasta lograr entrar al sistema. Todos estos datos recabados han llegado a determinar cuáles son las claves que no deben asignarse a una cuenta; por ser la primera instancia que intenta una persona al tratar de violarla. Por ello se deben tomar las siguientes:

Consideraciones al elegir una clave secreta

- No elegir palabras cortas; En el reporte No. CSC-STD-002-85 "Department of Defense Password Management Guideline" del National Computer Security Center se definió una fórmula para determinar la probabilidad de violar una clave secreta; de esta misma se determinó que, la longitud adecuada para que una clave no sea descifrada durante un mes de uso, cuando se intenta violar a una velocidad de 1000 intentos por segundo, es de 8 caracteres. En donde éstos pueden ser letras mayúsculas o minúsculas y dígitos.
- No utilizar: la cuenta, o la cuenta en sentido inverso o en ninguna otra forma; alguno de sus apellidos o alguna variación y concatenación de ellos; sus iniciales ni el nombre de familiares ni amigos; la fecha de nacimiento, ni ningún dato personal, etc.. Ya que según estudios, esto es lo que eligen los usuarios, y por ende, lo primero que intenta un cracker (ver pág. IV-4).
- En general, no utilizar una palabra que pueda aparecer en algún diccionario; no importando el idioma, ya que los cracker suelen utilizar programas que intentan violar esta seguridad, y para ello, emplean diccionarios de donde toman las palabras para realizar los intentos. Estos programas trabajan a una gran velocidad, realizando una gran cantidad de intentos por minuto, y si no se tienen medidas para controlarlos, pueden llegar a encontrar la clave de alguna cuenta. Por este motivo, los nuevos sistemas disponen de medidas que impiden esta labor y frustran estos intentos.

[30] Ver pág. IV-28, Grupos de usuarios.

IV.II.II.I. Asignación de la clave secreta

Como se vio en el capítulo II, cuando se crea una cuenta nueva se genera una entrada en el archivo */etc/passwd*, la cual contiene entre otras cosas, un campo para colocar la clave secreta de la cuenta.

Juan:**RudSOaEg.2m.E**:100:1:Juan Gonzalez P.:/usr/people/invest/juan:/bin/csh

Campo para la clave secreta, ya cifrada

Si este campo se encuentra vacío, entonces no le será pedida ninguna clave secreta al usuario al momento de entrar a sesión; y ya que este archivo existe en todos los sistemas UNIX, y entre sus peculiaridades se encuentra el que debe tener los permisos necesarios para que lo puedan leer todos los usuarios, y por tanto se pueda efectuar el proceso de entrada al sistema, cualquiera lo puede examinar y detectar rápidamente cuáles cuentas no están protegidas, creado una gran vulnerabilidad en la seguridad del sistema. Es por eso que la clave secreta se cifra antes de ser colocada en este campo, evitando que la puedan leer los usuarios. El comando encargado de generar el cifrado y colocar el resultado en este campo, es el *passwd*.

Este comando, en algunos sistemas, no pone ninguna restricción en la clave secreta que se desee elegir, por lo que gran cantidad de usuarios suelen incurrir en alguna de las faltas indicadas en el punto anterior. Si éste es el caso en el sistema operativo que se esté utilizando, se pueden obtener programas de dominio público[31] que realicen esta función, asegurándose además, de que la clave elegida por la persona no sea fácil de adivinar.

En particular, el comando *passwd* del sistema IRIX, verifica que la clave secreta tecleada por el usuario cumpla con las siguientes condiciones:

- El tamaño mínimo es de 6 caracteres. Con esto se evita el uso de claves cortas que resultan rápidas de descifrar.
- Sólo los primeros 8 caracteres son significativos. Esto quiere decir que para el sistema las claves secretas *the5Piso4room*, *the5Piso6cuarto* y *the5Piso* son iguales.
- Debe contener por lo menos 2 caracteres alfabéticos y uno numérico o especial.
- Debe ser diferente del nombre de la cuenta, la cuenta al revés, o el cambio de posición de los caracteres que la componen en forma circular. Para esta comparación, mayúsculas y minúsculas son iguales. Por ejemplo, para la cuenta josegp no son válidos: josegp, pgesoj, osegpj, segpjo, egpjos, etc.

- La nueva clave secreta, debe ser diferente de la anterior en tres caracteres por lo menos. Para motivos de comparación, mayúsculas y minúsculas son iguales.

Cabe destacar que a root le está permitido asignar cualquier clave a una cuenta; por lo que las restricciones indicadas arriba, no se aplican con este usuario.

Cuando un usuario asigna o cambia la clave secreta a su cuenta, contrario a lo que la mayoría piensa, la clave no es cifrada; sino que se utiliza como llave para cifrar un bloque de bytes definidos en la aplicación. El resultado de esta operación (trece caracteres), es colocado en el campo reservado para la clave secreta de dicha cuenta en el archivo */etc/passwd*. La secuencia seguida para ello, es todo un tema de estudio, pero baste con sabernos que el resultado de este proceso no se puede descifrar. Esto se debe a que durante la secuencia de cifrado se obtienen una serie de bits, los cuales son convertidos en caracteres; para ello, se usa una técnica en la cual, diferentes arreglos de bits pueden ser representados por el mismo carácter. Este hecho es importante para la seguridad, ya que de esta forma se asegura que nadie, ni root, pueda obtener la clave asignada a una cuenta. En particular, el método empleado en el Sistema Operativo IRIX, es el algoritmo de cifrado conocido como hashing en un sólo sentido, con algunas variaciones para frustrar el uso de implementaciones de búsqueda de llaves que puedan intentar descifrar la clave.

Hasta el momento, no se sabe que alguien haya encontrado un método para descifrar una clave secreta cifrada con esta técnica. Ello nos hace realizarnos una pregunta:

¿Cuál es el método utilizado para validar una clave cuando se quiere entrar a sesión?

La respuesta es sencilla: Cuando un usuario teclea su cuenta y su clave secreta, esta última es utilizada como llave para cifrar el mismo bloque de bytes, siguiendo la misma técnica. Al final, se compara el resultado de esta operación con el conjunto de caracteres almacenados en el campo reservado para la clave secreta de la cuenta que se está validando. Si ambos son iguales, se da por hecho que la clave tecleada es la correcta y se permite el acceso al sistema; si son diferentes, se niega la entrada.

Por último, recabando lo anterior, volvemos al mismo punto: hay que crear una conciencia en los usuarios de que elijan una clave secreta adecuada, para poder mantener e incrementar la seguridad del sistema. Por otra parte se pueden utilizar aplicaciones (creadas, de dominio público o compradas) que los obliguen a cumplir con estas normas.

[31] Son aplicaciones creadas por personas que permiten su distribución libremente, sin necesidad de pagar un costo por éste. En Internet, existen un gran número de sitios de dónde obtener software para todos los sistemas y todos los temas.

IV.II.II.II. Vigencia de la clave secreta

Por las razones expuestas en "**Consideraciones al elegir una clave secreta**", pág. IV-13, y en especial en la primera, las posibilidades de que la clave de una cuenta sea descifrada se incrementan si se mantiene la misma durante mucho tiempo. Es por esto que se debe inculcar a los usuarios, el hábito de cambiarla continuamente, según las necesidades y características de la seguridad implantada en el equipo: cada quincena, mes, trimestre o anual.

Muchas personas se mantienen renuentes a cambiarla, y prefieren mantenerse con la misma aun y cuando se les den las razones por las cuales se debe hacer. Es por ello que la mayoría de los sistemas actuales cuentan con un medio para forzar a los usuarios a que cambien su clave periódicamente, llamado vigencia de la clave.

Este mecanismo permite definir cuál es el tiempo máximo en que una clave será vigente, al término del cual, se fuerza a que el usuario la cambie; también se puede establecer cuál es el tiempo mínimo y durante el cual, la clave actual no puede ser cambiada. Esto impide que una vez cambiada, el usuario vuelva a utilizar el comando para cambiarla nuevamente a la que tenía. En esencia, este mecanismo obliga a que un usuario tenga por lo menos dos claves que utilizará alternativamente durante su estancia como usuario en el sistema.

Una forma sencilla de establecer la vigencia de la clave de una cuenta, es mediante el comando *passwd*. La opción *-x* establece el número máximo de días en que la clave será válida, después de los cuales, el usuario será forzado a cambiarla, si aún no lo ha hecho; la *-n*, establece el mínimo de días que deben transcurrir antes de que la clave pueda ser cambiada nuevamente. El siguiente comando establece para la clave de la cuenta *gonzzad*, una vigencia de 30 días y un mínimo de 7 antes de que se pueda cambiar nuevamente:

passwd -x 30 -n 7 -w 4 gonzzad

La opción *-w* permite establecer un número de días (4) antes de que se venza la vigencia de la clave, durante los cuales, el usuario recibirá mensajes de advertencia indicando esta situación y pueda cambiarla sin ningún contratiempo. Existen casos especiales:

- Si la opción *-x* es igual a *-n* y ambas son igual cero (0), el usuarios será forzado a cambiar su clave secreta la próxima vez que entre a sesión; después de esto, se desactivará el mecanismo de vigencia de claves para esta cuenta. Esto también puede ser implementado mediante la opción *-f* del comando *passwd*.
- Si la opción *-n* es mayor que *-x*, sólo root puede cambiar la clave de esta cuenta. Ésta es una forma de impedir que los usuarios cambien la clave secreta de una cuenta específica.
- Si la opción *-x* es igual a menos uno (-1), el mecanismo de vigencia es desactivado.

Particularmente, este mecanismo es implementado al agregar una serie de caracteres en el campo de la clave secreta del archivo */etc/passwd*. La clave y estos caracteres son separados mediante una coma (,). El primero de éstos indica el número de semanas que será vigente la clave; el segundo, el número mínimo de semanas que deben transcurrir antes de que se pueda cambiar. Los caracteres que se utilizan para ello son: ./,0-9, A-Z, a-z. Donde el primero (.) indica una semana; el segundo (/), dos; el tercero (0), tres y así sucesivamente. Si se está utilizando el archivo de seguridad de cuentas, shadow (ver más adelante), el control de la clave secreta como el mecanismo de vigencia es llevado en este último archivo.

IV.II.II.III. La clave secreta secundaria

Esta segunda clave permite agregar un grado mayor de seguridad en el mecanismo de entrada al sistema. Ya que suele emplearse para conexiones hechas mediante un modem a través del estilo conocido como dial-up, el cual permite conectarse via telefónica al equipo. Por esta razón, suele llamársele clave secreta diaul-up.

Este mecanismo establece la seguridad en base al shell[32] que se utiliza, y dependiendo de la línea a través de la cual, se realiza la conexión; por ello, no se utiliza cuando se lleva a cabo la conexión desde la consola o en ambiente gráfico (XDM[33]).

Para activarlo, primeramente se debe crear un archivo llamado */etc/dialups*; el cual debe contener una lista de los puertos (líneas) sobre los cuales se establecerá esta protección, uno por renglón. Ejemplo:

/dev/ttyd1
/dev/ttyd2
/dev/ttyd3

Y por último, crear el archivo */etc/d_passwd* y colocar en él, la clave secreta correspondiente a cada shell; uno por renglón como se indica:

*shell:**clave-secreta**:*

ejemplo:

/bin/sh:e8C9Ghfd2BUgH:
/bin/csh:ontJyinBALdE6:

Cabe destacar que la clave colocada, debe estar cifrada. Por ello, se recomienda utilizar el comando *passwd* para cambiarle la clave secreta a una cuenta conocida; después, copiarla

[32] Un shell es un programa utilizado como intérprete de comandos. UNIX posee varios shell que pueden ser utilizados por los usuarios; cada uno tiene sus propias características y funciones.

[33] X Display Manager. Es un manejador de pantallas para el ambiente gráfico X windows. Este sistema es utilizado generalmente en Estaciones de Trabajo y en Terminales Gráficas.

del campo de esta cuenta, a este archivo; y finalmente, restablecer la clave secreta que tenía originalmente la cuenta.

Del ejemplo anterior, cuando un usuario se conecte desde una de las líneas *ttyd1*, *ttyd2* y *ttyd3* que utilicen el Bourne shell (*sh*) o el C shell (*csh*), les será solicitada, además de su cuenta y clave secreta particular, la clave secreta secundaria establecida para el shell que esté utilizando.

Si el equipo que se está administrando cuenta con este tipo de servicio (conexiones dial-up), cabe, como medida de protección, realizar pruebas para cerciorarse de que se encuentra configurado adecuadamente; aunque los sistemas actuales ya toman estas consideraciones y medidas. El sistema debe dar de baja la sesión en caso de que la conexión remota se rompa o el usuario cuelgue el teléfono sin salirse de sesión. Además, si el usuario sale de sesión en forma normal, el sistema debe indicar a su modem que cuelgue para poder recibir nuevas llamadas. Deben realizarse estas pruebas para cerciorarse que no se queden ni líneas, ni sesiones abiertas que puedan ser utilizadas para romper la seguridad.

IV.II.III. Sistema de autentificación

Una vez explicadas las consideraciones tanto de la cuenta como de la clave secreta, así como sus características, el siguiente punto es el entender cuál es el proceso de entrada al sistema o autentificación, que permite dar o negar el acceso a una persona basado en estos dos recursos.

Cada vez que un usuario se conecta con el equipo, este último ejecuta un programa encargado de autentificarlo y darle o negarle el acceso al sistema. Se pueden utilizar para este propósito, una gran diversidad de programas; los cuales llevan a cabo la autentificación basados en diversos aspectos, como podría ser: una tarjeta con código de barras o cinta magnética; un dispositivo reconocedor de retinas, de huellas digitales o de voz; un dispositivo reconocedor de firmas o cualquier otro medio.

Generalmente los sistemas vienen diseñados y configurados para realizar esta labor mediante la clave secreta que se asigna a cada una de las cuentas. El nombre del programa que realiza esta función puede variar entre los diversos sistemas operativos existentes; pero la tarea que desempeña es la misma: preguntar por la cuenta y la clave secreta correspondiente, verificar que sean correctos y dar o negar el acceso en base al resultado, positivo o negativo.

En la mayoría de las estaciones de trabajo, que están diseñadas para laborar en ambiente gráfico, se encuentra activado el sistema XDM; que permite trabajar con el sistema de ventanas X windows y que terminales gráficas se conecten utilizando el protocolo

XDMCP[34], además de la tradicional conexión a través de terminales basadas en caracteres o tty. Por lo cual, dependiendo desde dónde y cuál sea el mecanismo que se utiliza para tratar de ingresar al sistema, es el programa utilizado para autentificar al usuario. En particular en el SO IRIX, esta labor la realizan tres programas:

- Si el usuario trata de ingresar directamente desde la consola gráfica, trabajando en ambiente gráfico, se utiliza el programa llamado *clogin*. En las versiones anteriores 4.0.X se utiliza el programa llamado *pandora* que es el antecesor de *clogin*.
- Si el usuario trata de ingresar desde una terminal gráfica se utiliza el mecanismo de autentificación que tiene incorporado el sistema XDM.
- Si el usuario se conecta desde una terminal o desde la consola sin utilizar el ambiente gráfico, se utiliza el programa llamado *login*[35].

Esta gran variedad hace que la labor de controlar, configurar y monitorear los distintos intentos de accesos al sistema sea más complicada; por lo cual, pueden tenerse más puntos vulnerables si no se conoce perfectamente cada uno de estos medios.

Si una estación de trabajo no tiene activado el sistema XDM, el único medio para ingresar al sistema, desde la consola o en forma remota, es a través del proceso de autentificación implementado por el programa *login*; lo cual facilita la labor de administración. Por ser éste el medio más utilizado, es el que se expondrá con mayor detalle; tocándose los restantes en forma superficial, ya que el estudio de la forma como trabajan es todo un tema a tratar.

IV.II.III.I. Autentificación mediante login

En el capítulo II se describió la secuencia que se sigue cuando un usuario entra a sesión, indicándose cuáles archivos y programas se ejecutan antes y después del proceso autentificación; permitiendo así, configurar el ambiente de trabajo del usuario.

Lo expuesto en este punto, será la manera de configurar y controlar el programa *login*; que es el que controla el acceso al sistema. Si a un usuario le es permitido entrar a sesión, entonces se ejecutan los archivos descritos en el capítulo II, permitiendo establecer medidas que permitan controlar su desplazamiento y limitando los alcances de las acciones que pueda realizar dentro del sistema.

El programa *login* dispone de una serie de opciones que permiten configurar la manera de funcionar del proceso de entrada al sistema, a través de esta aplicación. En particular se

[34] X Display Manager Control Protocol: Protocolo de control de XDM.

[35] No confundir el programa login, que realiza la autentificación del usuario, con el término de login, utilizado para indicar la cuenta que se asigna a cada usuario y permite autentificarlo ante el SO.

utiliza el archivo */etc/default/login*, en el cual se colocan las opciones, una por renglón. En versiones anteriores el archivo utilizado es */etc/config/login.options*, que permite establecer básicamente 5 variables: *passwdreq*, *maxtries*, *disabletime*, *lastlog* y *syslog*. La versión actual permite configurar estas variables y muchas más, permitiendo tener un mayor control sobre este recurso.

El primer grupo de opciones que examinaremos, permiten configurar la seguridad del proceso de autentificación: la primera que analizaremos permite definir el número máximo de intentos fallidos que se pueden realizar antes de que el sistema intervenga. Si una persona trata de adivinar la clave secreta de una cuenta intentando, ya sea manualmente o mediante un programa, el acceso al sistema y falla en el proceso el número de veces indicada por esta opción, éste dará por terminado el programa *login*, rompiendo la conexión y posteriormente suspenderá la línea (tty) que ocasionó el problema un tiempo determinado, desalentando y frustrando el intento. Esta opción es la siguiente:

MAXTRIES=4

En este ejemplo se define un número máximo de intentos de 4; por default el sistema establece un máximo de 5 intentos, si es que no se especifica nada. Esta opción soluciona una deficiencia de los primeros SO; los cuales sufrían ataques de programas que trataban de accesar al sistema, intentando distintas claves tomadas de un diccionario en forma consecutiva y a una gran velocidad, hasta conseguir su propósito y sin que se pudiera hacer algo al respecto. El único síntoma que se podía observar para determinar que se sufría un ataque de este tipo, era que el sistema no respondía cuando se trataba de accesar a él; debido a la velocidad del programa atacante, que consumía los recursos del equipo. Los sistemas actuales que cuentan con un mecanismo semejante a esta opción, impiden este tipo de agresiones, o por lo menos, las reducen; ya que como se vio, si se detecta una falla consecutiva al tratar de ingresar la cuenta o la clave secreta (un usuario legítimo no puede fallar tantas veces al ingresar estos datos; ya que se supone, él conoce), la línea se desactiva interrumpiendo cualquier intento; de ahí la importancia de mantener esta variable con un número bajo.

La segunda es la opción *LOGFAILURES*, permite especificar un número máximo de intentos fallidos que le está permitido al usuario; después de los cuales, son registrados en el archivo */var/adm/loginlog* todos los intentos. Si esta variable está definida en tres, y el usuario comete uno o dos errores, éstos no son registrados; pero, si comete tres, se registran los tres intentos en este archivo. Esto permite llevar un control y no registrar los casos comunes; sino intentos múltiples que uno considere como posibles violaciones.

Por cada intento fallido se registra el nombre de la cuenta, la terminal (tty) y la hora en que se realizó el intento. Para que este registro se pueda llevar, el archivo */var/adm/loginlog* debe existir. Por default este archivo no existe y por lo tanto, no se lleva el registro; aunque la variable esté definida. Si se desea activar el mecanismo, se debe crear este archivo con los

premisos de lectura y escritura activados para el dueño únicamente; que debe ser root y el grupo de sys.

La siguiente opción permite establecer el tiempo durante el cual, será suspendida la línea al alcanzar el máximo de intentos fallidos de cualquiera de las variables *MAXTRYS* o *LOGFAILURES*. Por default el sistema la suspende por 20 segundos. Si se desea establecer el tiempo de suspensión en un minuto colocar la opción:

DISABLETIME=**60**

La variable *SLEEPTIME* permite especificar el número de segundos en que permanecerá inactiva la línea donde se ha tenido un intento fallido, antes de desplegar el mensaje de *"login incorrect"* y volver a brindar, si aún no se ha alcanzado el máximo de las variables *MAXTRYS* o *LOGFAILURES*, la oportunidad para teclear la cuenta y clave secreta nuevamente.

Si al examinar el archivo */var/adm/loginlog* se detecta que por una línea se han realizado varios intentos de violar una cuenta, se puede reducir la opción *MAXTRIES* e incrementar el tiempo en que se desactiva la línea, *DISABLETIME*, para frustrar aún más el intento; o por el contrario, si se es experto en el uso del sistema, se puede dar a la tarea de rastrear a la persona que lo lleva a cabo.

Finalmente, dentro de este grupo de opciones afines se encuentra la de *SYSLOG*; que permite establecer el tipo de registro que se llevará sobre los intentos para ingresar al sistema. Esta opción tiene dos alternativas:

SYSLOG=ALL
SYSLOG=FAIL

La primera lleva un registro de todos los intentos de entrada al sistema, fallidos o aceptados. La segunda especifica que sólo se registren aquellos intentos que no tuvieron éxito. El registro de toda esta actividad es llevado en el archivo */var/adm/SYSLOG*; por lo que no se envía ningún registro al archivo */var/adm/loginlog*. Cabe destacar que en este archivo también se registran otras actividades y errores del sistema y es recomendable, si es que se utiliza esta opción, revisarlo continuamente para no perder el control del recurso y determinar si alguien ha intentado violar alguna clave; así como rotarlo[36], ya que puede crecer rápidamente dependiendo de la actividad del sistema.

Con el siguiente grupo de opciones se puede controlar la forma de funcionar del proceso login, además de contemplar nuevamente el aspecto de seguridad. La variable *TIMEOUT* permite establecer el número máximo de segundos de inactividad permitidos para una

[36] Es el proceso que se sigue en el cual se renombra el archivo original (conservando el registro previo) y se crea uno nuevo vacío que permita registrar la nueva información.

conexión. Si un usuario deja inactiva la terminal durante el proceso de autentificación por esta cantidad de segundos, el proceso es terminado y se desconecta la línea.

Con la siguiente opción se puede especificar el lugar desde donde se puede tener acceso a el equipo con la cuenta de *root*. Ejemplo:

CONSOLE=/dev/console

Ésta especifica que la cuenta de *root* sólo se puede teclear y tener acceso desde la consola del equipo. Es importante definirla de esta forma, para que nadie pueda tratar de violarla remotamente; ya que aunque se teclee su clave secreta correctamente, el acceso le será negado si no se utiliza la consola. Es importante mencionar que esta opción funciona durante el proceso de entrada; por lo que el administrador puede ingresar al sistema con su cuenta particular desde cualquier equipo, y una vez dentro de él, utilizar el comando *su* para adoptar la personalidad de *root* y realizar labores de administración. Si esta variable es definida como */dev/syscon* o */dev/systty*, la cuenta de root podrá ser utilizado desde cualquier terminal.

Tanto la variable *PASSREQ* y *MANDPASS* pueden ser establecidas en *NO* y *YES*. La primera sirve para especificar el requerimiento de la clave secreta en todas las cuentas; por lo que si es establecida en *YES*, y un usuario cuya cuenta no tiene clave asignada trata de ingresar, se le pedirá que le ponga una antes de entrar a sesión. El único inconveniente de esta función, es que si la cuenta es usada por alguna persona que no sea el dueño, le podrá establecer una clave secreta y el verdadero dueño no podrá entrar a sesión cuando trate de utilizarla; por lo que aunque esta función esté activada, es aconsejable para mantener la seguridad el no crear ni dejar cuentas sin clave secreta por ningún motivo.

Por el contrario, la segunda de las opciones anteriores impide el acceso al equipo a cualquier usuario que introduzca una cuenta que no tenga clave; por este motivo es recomendable utilizar, de estas dos alternativas, la de *MANDPASS*.

La opción *LOCKOUT* permite especificar el número consecutivo de intentos fallidos permitidos para tratar de ingresar con una cuenta; después de alcanzar esta cantidad, se utiliza el comando *passwd -l* para bloquear la cuenta.

Finalmente, la variable *IDLEWEEKS* permite dar un tiempo de gracia a las cuentas en las que ya ha expirado el tiempo de validez de su clave secreta, para que la cambien. Pasado este tiempo, la cuenta es bloqueada.

Además de estos grupos de opciones, existen otras que permiten configurar variables de ambiente durante la sección del usuario o como la variable *SITECHECK*, que permite definir cuál será el programa encargado de llevar a cabo la autentificación del usuario.

La opción *LASTLOG* que se menciona al inició de este punto, se utilizaba en las versiones anteriores del programa *login*, y cuando era colocada, le indica al sistema que desplegará en la pantalla, al momento de entrar a sesión, la fecha, hora y terminal desde donde se entro la última vez más reciente. Esta información le permitía al usuario saber si alguien había ingresado con su cuenta y, por lo tanto, reportar esta situación inmediatamente al administrador del equipo. Este control aún es llevado a cabo en esta versión del sistema; pero es efectuado por el comando *last*, el cual puede ser colocado dentro de alguno de los archivos de inicialización si se desea (como el archivo */etc/profile*), para que despliegue esta información cuando entre el usuario.

IV.II.III.II. Autentificación a través de XDM

XDM es un servicio que permite manejar un conjunto de pantallas X, que pueden estar ubicadas y trabajando en el equipo local o remotamente. Para lograr esta función, XDM cuenta con una serie de servicios, entre los cuales se encuentran aquéllos que brindan el acceso y autentificación del usuario. En el capítulo II se describió el concepto de XDM así como la forma de configurar el ambiente de trabajo en este sistema; por ello, en esta sección se describirá únicamente, el sistema de autentificación utilizado en él.

Podríamos decir que el sistema empleado en XDM para verificar y autorizar el ingreso al sistema, es el tradicional empleado en los sistemas UNIX: solicitar una cuenta y una clave secreta. La diferencia consiste en que este método es realizado en ambiente gráfico, a través de ventanas. Para ello se despliega en una pantalla gráfica, una ventana en la que se le solicita al usuario introduzca su cuenta y clave secreta; después de verificar estos datos, se brinda o niega el acceso a el equipo en base al resultado.

El proceso de autentificación puede ser configurado para incrementar la seguridad, y para ello, se utilizan las mismas variables y archivo que se emplean en el proceso de *login*; es decir, que las opciones colocadas en el archivo */etc/default/login* permiten configurar tanto el proceso de *login*, como el de XDM. En el punto anterior, Autentificación mediante login, se describieron las opciones más importantes que suelen ser colocadas en este archivo, así como el efecto que tienen sobre el proceso. Por ello, lo expuesto en ese punto se aplica también al proceso de autentificación empleado en XDM; referirse a él para mayor información.

Un caso particular es el de la opción *CONSOLE*. Si ésta es definida en */dev/console*, los accesos de la cuenta root estarán restringidos únicamente a la primer pantalla local definida en el archivo */var/X11/xdm/Xservers*, que generalmente es la consola. Por otro lado, si es establecida en cualquier otro valor, el acceso de root es deshabilitado a través de XDM, por lo que no se podrá ingresar directamente con esta clave. Y por último, si no es definido ningún valor, root puede entrar desde cualquier pantalla.

Finalmente, como cualquier aplicación gráfica, la ventana de autentificación puede ser configurada para mostrar los mensajes, colores, tamaños y características preferidas; para ello se cuenta con una serie de recursos[37] (*Xlogin** :) que pueden ser definidos en un archivo. Este archivo generalmente es */var/X11/xdm/Xresources*, el cual puede ser editado manualmente para adecuarlo a las características que mejor convengan. Cabe mencionar que el recurso *DisplayManager.DISPLAY.resources* ubicado en el archivo */var/X11/xdm/xdm-config* define cuál es el archivo de configuración que se utilizará; por default es */var/X11/xdm/Xresources*, como ya se mencionó.

IV.II.III.III. Autentificación mediante clogin

Cuando se instala el SO, queda configurado para que *clogin* realice el proceso de autentificación cuando se trate de ingresar a través de la consola; pero puede ser modificado para atender a otras pantallas locales o remotas, lo cual no es muy aconsejable por lo que examinaremos a continuación.

Ya que se supone, *clogin* será utilizado desde la consola únicamente, en la cual el acceso está restringido a las personas autorizadas exclusivamente, no tiene mayores medidas de control y seguridad que la de solicitar la cuenta y clave secreta; en cambio, ofrece un ambiente y menú atractivo de uso. Por ello, de las opciones que se pueden establecer en el archivo */etc/default/login*, únicamente tienen significado en este método de autentificación, la de *UMASK* y *SVR4_SIGNALS*, que no se examinaron en los puntos anteriores.

La opción *UMASK* , permite definir la protección que será dada a los archivos creados por el usuario; que es la función que desempeña el comando *umask* de UNIX. Por otro lado, la opción *SVR4_SIGNALS* permite definir cuál será el significado que posean las señales *SIGXCPU* y *SIGXFSZ*; si el normal definido en IRIX, o el definido en la versión SVR4 de UNIX (System V, Release 4). Si es establecida en *YES*, el significado de SVR4 es preservado; por lo que todos los procesos ignorarán estas señales. Si es establecida en *NO*, estas señales retendrán su significado; el cual es que los procesos que reciban la señal efectúen un core dump. Esta opción generalmente no debe ser cambiada.

Durante el proceso normal de entrada, se despliega una ventana gráfica (ver Fig. I-9) en la que aparecen iconos[38] representando a los distintos usuarios del sistema. Abajo de cada icono, que puede ser una fotografía digitalizada del usuario, aparece la cuenta que le corresponde. Abajo de esta ventana aparece un campo en el cual se puede teclear la cuenta,

[37] Los recursos especifican cuáles deberán ser las características que tengan las aplicaciones al momento de aparecer en las ventanas. Éstos deben estar en formato de recursos X; es decir, en formato de recursos del sistema de ventanas X11.

[38] Un icono es una pequeña figura gráfica que representa a un proceso corriendo en el equipo. Éstos suelen emplearse cuando se trabaja con el sistema de ventanas gráficas X.

y posteriormente la clave secreta del usuario. Para ingresar al sistema, se puede posicionar en este campo y proporcionar estos datos, o dar un doble click en el icono del usuario deseado y posteriormente escribir su clave secreta únicamente.

Como se puede observar, este método es elegante y atractivo; pero en cuanto a seguridad tiene un defecto: facilita de antemano el nombre de la cuenta de cada usuario en el sistema. Por ello, si se utiliza, es recomendable hacerlo únicamente en la consola; que es para lo que está destinado, y donde se supone existe seguridad en el acceso.

Para configurar tanto la apariencia de esta ventana , así como los iconos, o fotografías de los usuarios que se deseen aparezcan, se utiliza el programa *configClogin*. Con esta aplicación, se pueden definir los parámetros que se utilizarán; los cuales son colocados en el archivo */usr/Cadmin/clogin.conf*, que puede ser editado para realizar la configuración manualmente, si se tiene experiencia.

Por otra parte, existe una bandera de configuración del sistema que permite cambiar la apariencia de esta ventana, *noiconlogin*:

a) *$ chkconfig noiconlogin on*
b) *$ chkconfig noiconlogin off*

Cuando esta bandera es colocada en *on* (a), no se despliegan los iconos de los usuarios y en su lugar aparece la imagen colocada en el archivo */usr/Cadmin/images/cloginlogo.rgb*, que es el símbolo de la compañía Silicon Graphics. Cuando es establecida en *off* (b), aparecen los iconos de los usuarios; para ello el sistema busca en los siguientes lugares:

- El archivo *login.icon*, que contendrá la imagen personalizada del usuario, en el subdirectorio *.icons* de cada usuario.
- En el directorio */usr/local/lib/faces*, buscará un archivo que tenga el nombre de la cuenta del usuario. Generalmente son colocadas en este directorio las fotografías digitalizadas de cada usuario.
- En el directorio */usr/lib/faces*, donde también buscará archivos cuyo nombre sea el de la cuenta de cada usuario.
- Si para un usuario en particular, no encuentra su archivo de imagen, aparecerá en la ventana de *clogin* un icono genérico que se utiliza en estos casos.

IV.II.III.IV. Selección del método

Por lo expuesto en los puntos anteriores, los métodos para ingresar al sistema son variados y se debe conocer cuáles de ellos están activados, para poder llevar un control y establecer medidas de seguridad sobre cada uno. Los métodos expuestos aquí, son los utilizados en el sistema operativo IRIX; de los cuales, el de *login* es el que se utiliza generalmente en todos los sistemas UNIX, aunque el nombre de la aplicación que la desempeña puede variar, así como las opciones disponibles para configurarla.

Si en el sistema operativo IRIX que se esté utilizando, el sistema de XDM no esta activado, entonces el método de autentificación utilizado para todos los propósitos es el de *login*; a menos que se especifique otro mediante la opción *SITECHECK* del archivo */etc/default/login*. Para habilitar o deshabilitar a XDM se utiliza la siguiente bandera de sistema:

$ chkconfig windowsystem on ←——— Lo habilita
$ chkconfig windowsystem off ←——— Lo deshabilita

Si se encuentra activado XDM, entonces se utilizará el proceso de *login* para conexiones remotas en modo terminal (tty) y el proceso de autentificación de XDM para conexiones remotas que utilicen sesiones gráficas. En particular, en este caso, el acceso desde la consola es controlado por la bandera de sistema *visuallogin*.

a) *$ chkconfig visuallogin on*
b) *$ chkconfig visuallogin off*

Si ésta es establecida en *on* (a), se utilizará la autentificación mediante el programa de *clogin*; si es establecida en *off* (b), se empleará el método estándar del XDM.

IV.II.III.V. Archivo de seguridad shadow

Aunque el método utilizado para cifrar las claves secretas es un método seguro, se deben tomar todas las medidas para evitar que se pueda descifrar. El archivo */etc/passwd* debe tener permisos para que pueda ser leído por todos; por lo cual, cualquier usuario puede listar su contenido y revisarlo, tratando de encontrar una cuenta sin clave. Por otro lado, un usuario conociendo su clave secreta y obteniendo el código cifrado del campo correspondiente en este archivo, puede tratar de descifrar el método utilizado y así romper la seguridad, descifrando las claves restantes. Para evitar este tipo de intenciones se puede generar el archivo de seguridad shadow.

Este método consiste en sustraer el campo de la clave secreta de todas las cuentas del archivo */etc/passwd* y colocarlos en un archivo llamado */etc/shadow*. El archivo *shadow* contiene una entrada por cada usuario y cada una contiene los siguientes campos:

Cuenta	El nombre de la cuenta o login.
Clave secreta	El password o clave secreta cifrada de la cuenta. Si la cuenta estaba bloqueada, aquí se coloca la palabra o carácter que así lo indica. Si no tenía clave secreta, este campo quedará vacío.
Último cambio	El número de días entre el 1 de Enero de 1970 y la fecha de la última modificación de la clave secreta.
Mínimo	Es el mínimo de días necesarios de transcurrir, antes de que la clave pueda ser cambiada nuevamente.
Máximo	Es el máximo número de días en que la clave es válida.
Advertencia	Es el número de días antes de que la clave vigente expire, en que el usuario será advertido de esta situación, para que la cambie.
Inactividad	Es el número de días de inactividad, que le es permitido a la cuenta.
Expiración	Es la fecha absoluta a partir de la cual, la cuenta ya no será usada; por lo que le es negado el acceso al sistema.
Banderas	Este campo actualmente no tiene ningún uso; es reservado para futuras implementaciones.

A este archivo le son colocados los permisos necesarios para que ninguna persona pueda tener acceso a él. Cuando se utiliza este mecanismo, generalmente no se ve afectada ninguna operación del sistema, ya que resulta transparente para todas ellas; a excepción de los programas antiguos que utilizan las llamadas de librería *getpwent* y *getpwnam*, como los programas protectores de pantalla en el sistema IRIX.

En el archivo */etc/passwd* se coloca una "x" en el campo correspondiente a la clave para indicar que se encuentra activada esta protección. Para activarla se utiliza el comando *pwconv*. Una vez ejecutado, todas las herramientas que utilizan la clave secreta, funcionan transparentemente.

IV.III. Seguridad dentro del sistema

El siguiente nivel de seguridad implantado dentro de un equipo, es el establecido dentro del sistema de archivos; el cual limita las acciones y daños que pueda causar un usuario a la información almacenada dentro de él, mediante una serie de permisos que son asignados a cada uno de los recursos de que se disponen (archivos, programas, directorios, bases de datos, etc.).

Si una persona que logra violar la seguridad del acceso al sistema o un usuario legitimo[39] tratan de alterar, borrar, copiar o causar algún daño tanto a la información almacenada, como al propio sistema operativo, se encontrarán con una serie de medidas que impiden su libre desplazamiento y acciones que puedan efectuar.

En este renglón, todos los sistemas UNIX cuentan con elementos para controlar y limitar las acciones de los usuarios dentro del sistema, que los colocan dentro de un nivel de seguridad denominado "Protección de Seguridad Discrecional" o nivel de seguridad "C1" según el Criterio de Evaluación de Sistemas de Cómputo Confiables del Departamento de Defensa de los Estados Unidos, conocido como el ORANGE BOOK[40] por el color de su portada.

Estas medidas permiten identificar[41] y reunir a los usuarios por grupos, así como asignar un conjunto de permisos a los archivos del sistema, que permite controlar, quién y qué es lo que puede hacer sobre cada uno de ellos. Por otro lado, se cuenta con herramientas para monitorear la actividad del sistema, y poder detectar anomalías que estén ocurriendo, o que hayan ocurrido durante la operación diaria. Esto es lo que se expondrá dentro de este punto.

IV.III.I. Grupos de usuarios

Cuando se define un nuevo usuario, se le asigna entre otras cosas, una cuenta y una clave secreta que lo identifican ante el sistema, así como la posibilidad de pertenecer a un grupo de usuarios en particular, los cuales tienen ciertas afinidades en común que los hacen requerir ciertos privilegios o tener acceso a cierta información en particular.

La administración de los recursos, así como la seguridad de ellos, se puede controlar de una mejor manera si se pueden asignar permisos en base a los usuarios o a un grupo de usuarios en particular; por ejemplo: Un grupo de personas encargadas de efectuar captura de datos, no deben tener acceso a herramientas de diseño, a bases de datos en forma directa, a

[39] Que tiene asignado una cuenta y una clave secreta dentro del sistema.
[40] LIBRO NARANJA. Consultar página IV-60, ORANGE BOOK.
[41] Un usuario se identifica ante el sistema, mediante su cuenta y clave. Ver Seguridad en el acceso al sistema, pág. IV-8.

comandos que puedan borrar archivos, etc.; sólo se les debe dar acceso a los comandos y programas de captura que les permitan realizar su labor. En cambio, personal de un departamento de desarrollo, debe tener acceso a las herramientas que les permitan diseñar y crear nuevas aplicaciones; como bases de datos, programas de captura, programas para analizar la información, etc. Este tipo de personas requieren tener un mayor grado de libertad para realizar su labor; pero siempre controlado y restringido a sus necesidades.

De lo anterior se desprende, que del universo de usuarios de un sistema, existen grupos de ellos que tienen exigencias similares, dando como resultado, la necesidad de poder agruparlos y asignarles privilegios en conjunto; además de los que se les han asignado en forma particular a cada uno de ellos. Por otro lado, la existencia de los grupos facilita que un usuario tenga acceso a la información perteneciente a los demás miembros de su grupo; permitiendo así, compartirla sin mayores problemas. Claro está, que este acceso es controlado, y puede ser restringido si algún miembro desea conservar cierta información privada o liberarla en cierta medida a los demás miembros del grupo.

Anteriormente y aún en la actualidad, administradores de equipos siguen una mala costumbre de crear cuentas que son compartidas por varios usuarios que trabajan sobre un proyecto en particular. Este mal hábito ocasiona que no se tenga un control y no se puedan asignar responsabilidades; dando oportunidad a que alguno de ellos puede tratar de violar la información, ya sea involuntariamente o intencionalmente, y ocultar sus actos en el hecho de que son muchas las personas que utilizan dicha cuenta. Es por ello que este tipo de cuentas debe ser evitada, y en su lugar, utilizar la habilidad del sistema para crear grupos de trabajo; los cuales, como ya se mencionó, son un conjunto de usuarios con cuentas individuales, pero que por pertenecer al mismo grupo, tiene una serie de privilegios que les permiten compartir información entre ellos sin ningún problema.

En el archivo */etc/group*, se encuentran enlistados los grupos de trabajo creados en el sistema. En cada definición de un grupo en este archivo, se especifica, separado por dos puntos(:):

- El nombre del grupo, el cual es un medio para identificarlo ante las personas.
- Un identificador de grupo; que es un número que sirve para identificar al grupo ante el sistema. Es decir, el sistema reconoce al grupo por su identificador y no por su nombre; lo cual es análogo a lo que sucede con las cuentas. Ver las consideraciones al respecto en Bloqueo de cuentas, pág. IV-9.
- Una clave secreta; que aumenta la seguridad cuando un usuario trate de cambiarse de grupo de trabajo. Cabe destacar que un usuario sólo se puede cambiar a los grupos a los cuales él pertenece. La clave secreta debe estar cifrada; por lo tanto, el único método de asignar una a un grupo, es el copiarla de una cuenta conocida en el archivo */etc/passwd* a este campo.
- Una lista de todas las cuentas, separadas por una coma (,), que pertenecen al grupo.

Cuando se crea una cuenta (ver 'Creación de cuentas de acceso', pág. II-33), se define a qué grupo pertenecerá el usuario por default. Si se desea, se puede editar el archivo */etc/group* y agregar la cuenta de cualquier usuario a la lista de uno o varios grupos; para que el sistema lo reconozca como miembro de todos ellos.

Por otra parte, al momento que el usuario entra a sesión, se considera como miembro del grupo al cual fue asignado al ser creada su cuenta, que está definida en el cuarto campo de la entrada de la cuenta, en el archivo */etc/passwd*. Posteriormente si el usuario desea trabajar, y por lo tanto, tener los privilegios de otro grupo al cual él pertenece, puede utilizar el comando *newgrp* o *multgrps* para cambiarse de grupo o trabajar con varios a la vez.

Cuando se utiliza cualquiera de estos comandos para cambiar la membresía al grupo con el cual se desea trabajar, el sistema genera un nuevo shell, dentro del cual, el usuario pertenece a ese nuevo grupo. Por ello, para abandonar ese grupo y regresar al original, basta con salirse del shell hijo mediante el comando *exit*, para regresar al shell original (padre).

Por último, es conveniente revisar la integridad del archivo de grupos */etc/group*, así como el mantener una copia de seguridad actual de él; ya que cualquier inconsistencia, redunda en un mal funcionamiento de la seguridad y del sistema en general. Para verificar este archivo se tiene el comando *grpck*, que examina cada una de sus entradas; detectando anomalías en la cantidad y consistencia de campos, el que realmente existan todas las cuentas definidas en los grupos, etc.

IV.III.II. Permisos de archivos

Cada entidad[42] dentro del sistema de archivos, posee un conjunto de atributos que permiten controlar el acceso, por parte de los usuarios, a este recurso. Esencialmente son 5 los atributos que controlan su seguridad, y son:

El dueño

Cada entidad dentro del sistema de archivos tiene un dueño; que inicialmente es el usuario (cuenta) que lo creó. El dueño es la única persona que puede modificar todos los atributos de la entidad. Si el dueño modifica este atributo indicando que el dueño sea otra cuenta, entonces pierde todos sus derechos como dueño y los adquiere la cuenta que se asignó, siendo ahora, la única que puede modificar los atributos de esta entidad.

[42] Una entidad es cualquier recurso que pueda existir en el sistema de archivos. Ejemplo: Un archivo de texto o programa, un directorio, un archivo de dispositivo, un archivo tipo fifo, una liga, etc.

mayoría de los sistemas son básicamente permisos de lectura, escritura y ejecución representados mediante una *r*,*w*, y *x* respectivamente. Añadiéndose a este conjunto de permisos, esta otra clase que no todos los sistemas tienen, pero que son igual de importantes a los anteriores: los de set-user-id, set-group-id, mandatory locking y sticky bit[43].

IV.III.II.I. Permisos estándar

Éstos son los permisos de lectura, escritura y ejecución que son utilizados en todos los sistemas UNIX y que pueden ser aplicados a el propietario de la entidad, al grupo al cual está asignada dicha entidad, o a el resto de los usuarios del sistema. Ahora, como se mencionó anteriormente, el significado de cada uno de estos permisos varía dependiendo de si es asignado a un archivo o a un directorio, como se indica a continuación:

Permiso de lectura (r)

Cuando un archivo tiene activado el permiso de lectura, se da el acceso para que pueda ser consultada la información que contiene. En cambio, cuando este permiso es dado a un directorio, se da el acceso para poder leer su contenido; es decir, ver los nombres de los archivos almacenados dentro de él.

Permiso de escritura (w)

Cuando es asignado este permiso a un archivo, permite que su contenido pueda ser modificado por las personas a las que les fue dada la autorización. En cambio, cuando se trata de un directorio implica la habilidad de modificar su contenido; es decir, el poder cambiar de nombre, crear o borrar cualquier archivo dentro de él.

Permiso de ejecución (x)

Si este permiso es dado a un archivo, autoriza que su contenido pueda ser ejecutado; siempre y cuando se trate de un programa ya compilado y en código binario. Si se trata de un archivo que contiene un programa escrito en shell, se requiere además de este permiso, el de lectura; ya que se requiere leer su contenido para poder ser ejecutado.

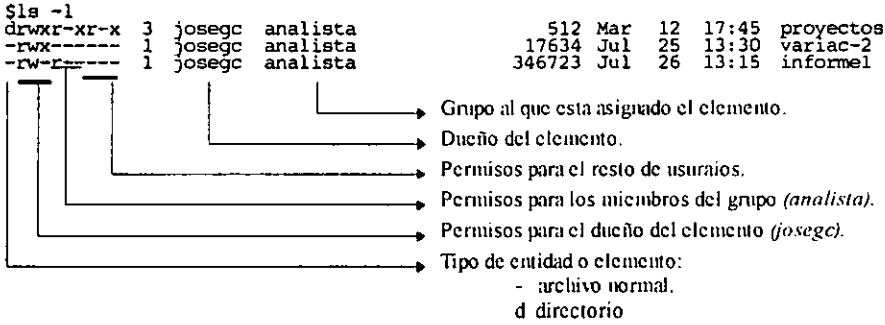
Si el permiso es dado a un directorio, da la autorización para realizar búsquedas dentro de él; es decir, permiso para accesar sus archivos. Cabe destacar que el permiso de lectura en un directorio permite que los archivos dentro de él puedan ser vistos; pero no da la autorización para accesarlos. En algunos sistemas este permiso da la autorización para poderse cambiar a este directorio con el comando *cd*.

[43] Éstos son muy reconocidos con sus nombres en inglés, y resultaría un poco problemático el traducirlos e interpretarlos en español

Un grupo	Cada entidad pertenece o tiene asignado un grupo; que inicialmente es el grupo al que pertenece el dueño que creo el archivo.
Permisos para el dueño	Éste es un conjunto de tres permisos que delimitan las acciones del dueño sobre esta entidad.
Permisos para el grupo	Es otro conjunto de tres permisos que delimitan las acciones que pueden realizar todos los usuarios (cuentas) que pertenecen al grupo al cual está asignada esta entidad.
Permisos para el resto	Éste es el conjunto de tres permisos asignados a las cuentas que no caen dentro de las dos categorías anteriores.

Estos 5 atributos asignados a las entidades del sistema de archivos, son esencialmente el sistema de seguridad que se tiene en la mayoría de los sistemas UNIX, y son junto con otros elementos de protección, los que definen un nivel de seguridad C1.

El comando *ls -l* permite ver el contenido del sistema de archivos, mostrando cada uno de los elementos almacenados dentro de él y exhibiendo los atributos asignados a cada uno de ellos. Ejemplo:



El propietario del archivo puede utilizar el comando *chmod* para cambiar los permisos asignados a él (el dueño), a el grupo o a los demás usuarios, para cualquier entidad que le pertenezca. Por otro lado, puede utilizar el comando *chgrp* para cambiar el grupo al cual está asignada la entidad, o el comando *chown* para indicarle que otra cuenta sea el nuevo dueño.

Fundamentalmente las entidades se pueden clasificar en dos elementos: archivos, que pueden contener programas, textos, imágenes digitalizadas, bases de datos, etc.; y directorios, que sirven para organizar los archivos dentro del sistema. En base a estos dos elementos, varía el significado que se da al conjunto de permisos mencionados anteriormente, y que en la

IV.III.II.II. Sticky bit (t)

Cuando era colocado este permiso en un archivo ejecutable, le indicaba al sistema que realizara una acción especial sobre él. Este uso ya ha sido abandonado por algunos sistemas operativos, dándole otro significado (el que se describirá a continuación) cuando es colocado a un directorio.

En particular, si se asigna este permiso a un directorio en el sistema operativo IRIX, y éste tiene activado el permiso de escritura (w), un proceso puede borrar o renombrar archivos que estén almacenados dentro de él, si:

- El identificador de usuario real del proceso es el mismo que el del dueño del archivo.
- El identificador de usuario real del proceso, es el mismo que el identificador del dueño del directorio.
- El proceso tiene permiso de escritura sobre el archivo.
- El proceso pertenece a *root*.

Es decir que un usuario puede borrar o renombrar archivos dentro de este directorio, sólo si es el dueño del archivo o del directorio; o visto de otra forma, el usuario no puede borrar o renombrar archivos que estén guardados en ese directorio y que pertenezcan a otro usuario. Este permiso es de suma importancia en directorios que son de dominio público como */tmp*, donde todos los usuarios del sistema pueden grabar archivos temporales, pero sólo el dueño de ellos puede borrarlos o cambiarles el nombre. Si este permiso no estuviera establecido en este directorio, un usuario podría guardar un archivo y otro lo podría borrar sin ninguna dificultad.

De lo anterior se deduce la importancia de este permiso sobre este tipo de directorios; ya que permite mantener cierto grado de seguridad en los archivos almacenados en directorios públicos, entre otras cosas.

Por otro lado, si este permiso es dado a un archivo que contiene un cargador dinámico para programas ejecutables ELF, entonces cuando el programa termina, el espacio de direcciones de sólo lectura que pertenecía al proceso, estará disponible para el cargador dinámico en el nuevo proceso. Este mecanismo permite mejorar considerablemente el tiempo de arranque del programa.

Este permiso sólo tiene significado cuando es utilizado dentro del conjunto de permisos que pertenecen a el dueño, y por otro lado, si es dado a cualquier otro archivo, no tiene ningún efecto. Para colocarlo a un directorio llamado *programas*, se puede utilizar el comando *chmod* en cualquiera de sus dos modalidades:

$ chmod u+t programas
$ chmod 1000 programas

Por otro lado, el sistema cuenta con una bandera que permite activar o desactivar este mecanismo sobre el directorio */tmp*:

$ chkconfig -f nostickytmp off
$ chkconfig -f nostickytmp on

La primera vez que sea utilizado este comando, debe llevar la opción *-f* para crear el archivo de bandera; ya que por default no existe. El primer ejemplo activa la protección, por lo que un usuario no podrá borrar archivos del directorio */tmp*, a menos de que tenga los permisos para hacerlo, él sea el dueño o tenga privilegios de super usuario.

El segundo ejemplo desactiva este mecanismo y cualquier usuario podrá borrar cualquier archivo almacenado dentro del directorio */tmp*. Esto puede ocasionar algunos problemas en el funcionamiento y la seguridad, pero puede ser de utilidad cuando se tiene poco espacio en disco; ya que los usuarios y aplicaciones del sistema suelen crear archivos temporales en este directorio, y al terminar su labor, algunos archivos no son removidos, ocupando el poco espacio existente. En este sentido, el sistema cuenta con otra bandera, la cual le indica al sistema que borre el contenido del directorio */tmp* cada vez que se arranca el sistema (*off*) o que conserve su contenido, que no lo borre (*on*).

$ chkconfig -f nocleantmp off
$ chkconfig -f nocleantmp on

Un punto importante al respecto, es que algunos programas así como herramientas de instalación, crean un archivo de mensajes en el cual son registrados todos los errores, mensajes de advertencia y algunos comentarios importantes que ocurren durante la instalación, en este directorio. Y por otro lado, algunos de ellos dentro de sus tareas post-instalatorias, requieren y generan un reboot del sistema. Si esto ocurre cuando la bandera *nocleantmp* está en *on*, toda la posible información que podría servir para determinar fallas durante la instalación, o importantes mensajes que permitan configurar adecuadamente la aplicación instalada, se perderán. Por ello, se debe tomar en cuenta esta bandera y colocarla en *off* durante el periodo de pruebas e instalación de aplicaciones.

IV.III.II.III. Set-user-ID ó Set-UID (s)

Este permiso es utilizado en archivos ejecutables exclusivamente, y debe ser colocado dentro de los permisos pertenecientes a el dueño; si es que se quiere que tenga este significado. Cuando es utilizado en otro tipo de archivos no tiene ningún sentido.

Cuando un usuario ejecuta un programa sobre el cual tiene permiso de ejecución y que no necesariamente le pertenece, le son asignados a este programa los permisos del usuario que lo ejecutó y con ellos opera hasta que finaliza su labor. En cambio, cuando es dado el

permiso de set-UID a un programa y éste es ejecutado, adquiere los permisos del dueño del archivo en adición a los permisos del usuario que lo ejecutó.

Esto es requerido por ciertos programas para poder funcionar. Por ejemplo, el comando *ps* despliega un resumen de los procesos que se encuentran corriendo en el sistema; para ello, requiere leer información de la memoria del equipo, la cual está protegida para que sólo root la pueda leer. Si el programa *ps* no tuviera este permiso activado, cuando un usuario lo ejecutara, el comando *ps* adquiriría los permisos del usuario y le sería negado el acceso de lectura a la memoria; en cambio, si lo tiene activado y cualquier usuario lo ejecuta, entonces el programa adquiere los permisos del dueño del archivo, que es root, y puede leer la información de la memoria del equipo, dando el resumen de los procesos que se encuentran en ejecución dentro del sistema.

De lo anterior se deduce la importancia de este tipo de permisos, ya que le dan la posibilidad a un usuario de adquirir los privilegios de otro. Este hecho suele ser utilizado para tratar de violar la seguridad del sistema, creando caballos de troya u otro tipo de artificios. Cuando un intruso ha logrado entrar a la cuenta de root, suelen colocar archivos con este tipo de permisos, que le permitan, utilizando una cuenta de usuario normal, adquirir los privilegios de root. Es por ello que se debe tener un estricto control sobre la cantidad y el propósito de los programas que tengan este tipo de permiso, y en especial, los que pertenezcan a root. Por otro lado, los sistemas operativos suelen utilizar un cierto número de programas, que para su funcionamiento, requieren de este permiso. Es por ello que se debe guardar un registro de los programas del sistema que originalmente lo tienen activado, y revisar continuamente, tomando en cuenta esta lista, para detectar posibles intentos de violación. Para generar un archivo llamado *listado-set-uid* que contenga una lista de los programas perteneciente a root con este permiso, ejecutar el siguiente comando:

```
$ find / -user root -perm -4000 -print > listado-set-uid
```

Posteriormente se pueden generar otros listados y compararlos para encontrar diferencia; esto es, archivos que fueron creados y contienen este permiso. Se puede utilizar el comando *diff* para obtener la diferencia entre los dos listados. Si se desea una lista de todos los programas con este tipo de permiso, y no sólo los de root:

```
$ find / -type f -a -perm -4000 -print  ó
$ ls -l `/etc/ncheck  -s  /dev/root | cut  -f2 | grep -v dev `
```

Para colocar este permiso sobre el archivo llamado *prog1*, se utiliza el comando *chmod* en cualquier de sus dos modalidades:

```
$ chmod  u+s  prog1
$ chmod 4000 prog1
```

IV.III.II.IV. Set-group-ID Ó SET-GID (s)

Este permiso se representa mediante una *s*, pero es aplicado al conjunto de permisos pertenecientes al grupo; a diferencia del set-uid, que es aplicado al conjunto perteneciente al usuario o dueño.

Este tipo de permiso tiene tres distintos significados; dependiendo de si se trata de un archivo ejecutable, un archivo normal o un directorio, el elemento sobre el cual se active.

Si el archivo tiene asignado el permiso de ejecución (x) sobre el conjunto de permisos perteneciente a el grupo (es ejecutable para el grupo), y se asigna el permiso set-gid (s), tendrá un funcionamiento similar al del permiso set-uid, con la salvedad de que los permisos que se asignan al programa al ser utilizado, son los permisos que tiene el archivo asignados al grupo. Para asignar este permiso ejecutar el comando:

$ chmod g+s prog1 ó
$ chmod 2711 prog1

El primer ejemplo agrega el permiso de set-gid al programa *prog1*, el cual, ya debe tener el permiso de ejecución activado para el grupo. El segundo ejemplo asigna a el dueño los premiso de *rwx*; al grupo y al resto de usuarios, el permiso de ejecución; y activa el permiso de set-gid.

Si el archivo al cual se le asigna el permiso set-gid no tiene el permiso de ejecución activado para el grupo, el significado que se le da a este permiso es el de Bloqueo Obligatorio representado por una *l*, que se refiere a la habilidad del archivo para mantener bloqueado sus permisos de lectura y escritura mientras un proceso lo utiliza. Para activar este permiso sobre el archivo llamado *datos*, utilizar el comando:

$ chmod +l datos

Finalmente, si este permiso es dado a un directorio, el grupo al cual pertenecerá cada archivo creado dentro de él, será el grupo al cual está asignado el directorio. Este mecanismo es el empleado en la versión Berkeley de UNIX. Por otro lado, si un directorio no tiene asignado este permiso, el grupo al cual pertenecerá cada archivo creado dentro de él, será el grupo al cual pertenece el usuario o programa que lo creo; el cual es el mecanismo empleado en las versiones System V de UNIX. Para activar este mecanismo sobre el directorio *aplicaciones*, ejecutar el comando:

$ chmod g+s aplicaciones

IV.III.II.V. umask

Cuando un usuario genera un elemento dentro del sistema de archivos, le son asignados por default un conjunto de permisos que previamente se han establecido. El comando *umask*, que generalmente se encuentra dentro de uno de los archivos de configuración que se ejecutan automáticamente al entrar a sesión, es el encargado de definir cuáles serán estos permisos que se asignarán a los archivos que cree el usuario.

La forma en que son interpretados los dígitos que se dan como parámetros al comando *umask* es la contraria al significado que se le dan en el comando *chmod*. En ambos se utilizan tres dígitos, donde cada uno representa los permisos para el usuario o dueño, el grupo y los demás usuarios; pero en *chmod*, éstos indican los permisos que se van otorgando, en cambio, en *umask* representan los permisos que se van restando de un máximo de 666 para un acceso completo a los archivos y 777 para los directorios y programas ejecutables. Es decir, que si se dan los dígitos 000, se otorgan el máximo de permisos a el dueño, grupo y los demás; en cambio, el asignar un *umask* de 037, elimina los permisos de escritura y ejecución para el grupo y todos los permisos para los demás usuarios.

El establecer un *umask* no afecta a los archivos ya creados; sino exclusivamente a los nuevos que genere el usuario. Si no se define ningún valor con este comando, el sistema utiliza el default, que son los valores de 022; es decir, los permisos de 644 para archivos ó 755 para directorios y programas ejecutables que normalmente se asignan con el comando *chmod*.

Es importante entender su funcionamiento y el utilizarlo para definir cuáles son los permisos que deseamos se asignen a los archivos que generemos; y de esta forma, automatizar su protección.

IV.III.II.VI. Otras consideraciones

El administrador del sistema debe tener un control estricto sobre los permisos que son asignados a los distintos archivos y directorios que forman el sistema. Durante la instalación le son asignados permisos a cada uno de estos elementos, algunos de los cuales, requieren para su funcionamiento que los usuarios los puedan accesar o modificar. Es por ello que se debe generar un lista de estos elementos e implementar una rutina constante de monitoreo que permita determinar si han sufrido alteraciones que puedan considerarse como un riesgo a la seguridad. Para ello, cada administrador debe inspeccionar la estructura de su sistema de archivo en busca de los elementos que sean accesibles por todos los usuarios. A continuación se enlistan algunos de los elementos que pueden, y están diseñados, para que sean accesibles a todos los usuarios:

/tmp
/usr/demos/.xsession
/usr/Insight.tmp
/usr/Insight/tmp/ebtpriv
/usr/Insight/tmp/ebtpub
/usr/Insight/tmp/install.insight.log
/usr/lib/emacs/maclib
/usr/lib/showcase/fonts
/usr/lib/showcase/images
usr/lib/showcase/models
usr/lib/showcase/templates
var/spool/locks
var/spool/uucppublic
usr/tmp.O
var/tmp

De igual forma, es aconsejable realizar periódicamente labores de limpieza sobre el sistema de archivos; en especial en los directorios cuya función es la de servir de almacén de archivos temporales como: */tmp*, */usr/tmp*, etc. Durante estas labores debe incluirse la búsqueda de archivos que carezcan de un dueño o tengan permisos inadecuados; para ello se puede emplear el comando:

$ find / -nouser -print

Con él, se obtiene una lista de todos los archivos en el sistema que no tienen un dueño asignado; esto suele ocurrir cuando se elimina una cuenta de usuario y no se borran todos los archivos que le pertenecían. También se puede dar el caso, que por alguna causa se restaure información de un respaldo, dentro de la cual, se encuentran archivos que pertenecían a un usuario vigente al momento de realizarlo; pero que ya no existe. En algunos sistemas cuando se presenta esta condición, es asignado a root como dueño del archivo; pero en otros, quedan sin dueño.

Un punto más, resulta importante el mencionar que las medidas de seguridad deben comprender en especial, a los distintos archivos de dispositivos que se encuentran localizados dentro del directorio */dev*. Estos archivos dan acceso directo a los recursos cuyo nombre señalan; es decir, el archivo */dev/mem*, da acceso a la memoria del equipo, donde se encuentran corriendo las aplicaciones; el archivo */dev/tape*, da acceso a la unidad de cinta; los archivos dentro del directorio */dev/dsk* dan acceso a las distintas particiones de los discos duros; etc. Por ello, se debe mantener un estricto control sobre estos dispositivos y sobre todo, los permisos que tengan; para que sólo puedan ser accesados por el sistema y un usuario normal, no pueda alterar su información.

De los puntos anteriores, se deduce la importancia de conocer cada uno de los mecanismos de protección, al igual que comprender adecuadamente su funcionamiento. Por ello, es conveniente el capacitar a los usuarios en su correcto uso, permitiéndoles de esta manera, que contribuyan en el mantenimiento de la seguridad del sistema de archivos.

IV.III.III. Sistema de intervención y rastreo de cuentas

El sistema de intervención y rastreo de cuentas conocido como audit trail, es un mecanismo mediante el cual, se puede obtener un registro de toda la actividad del sistema operativo. Si se sospecha de algún intento de violación de la seguridad por parte de un usuario, mediante este sistema se puede rastrear cada uno de sus pasos; es decir, los procesos que ejecuta o trata de ejecutar.

Al mecanismo de seguridad empleado por la mayoría de los sistemas UNIX basado en los permisos de archivos, se le conoce como Control de Acceso Discrecional o DAC por sus siglas en inglés. Con el uso del subsistema de intervención y rastreo de cuentas, el sistema IRIX 5.X cumple, aunque no está certificado, con el nivel de seguridad C2, descrito en el Orange Book (ver pág. IV-60); el cual especifica que se debe establecer un control de acceso discrecional o DAC entre otras cosas. Si se cuenta con la versión TRUSTED IRIX/B, se dispone de un sistema que implementa una mayor seguridad mediante el uso de la Listas de Control de Acceso Obligatorio o Mandatore Control Acces List (MAC), certificándose dentro de la categoría de seguridad B de este mismo documento.

El sistema de intervención y rastreo de cuentas se encuentra dentro del medio de distribución del Sistema, pero durante el proceso de instalación, no es cargado por default dentro del equipo. Si éste fuera el caso, se debe utilizar la herramienta *inst* para instalar el modulo *eoe2.sw.audit* que lo contiene. Una vez hecho esto, se puede utilizar la bandera del sistema *audit* para activarlo:

S chkconfig audit on ←——— Lo activa
S chkconfig audit off ←——— Lo desactiva

Una vez activado, este sistema empieza a rastrear cada uno de los eventos que se le han especificado. Por default se encuentran establecidos algunos eventos, 35 aproximadamente, dentro del archivo *etc/init.d/audit*. Si se desea se puede utilizar las herramientas *sat_select* o *satconfig* para indicar cuáles, de la lista completa de eventos, se deben rastrear. La primera de estas herramientas está diseñada para ser utilizada a través de terminales; y la segunda, en pantallas gráficas que brindan la facilidad de utilizar una ventana como interfaz, en la que con un simple click de un botón, se puede activar o desactivar el rastreo de algún evento.

El demonio *satd* (system audit trail demon) guarda la información referente al rastreo de los eventos en archivos que pueden crecer rápidamente dependiendo de la actividad del sistema. Estos archivos llevan por nombre:

sat_fecha

Donde: ***fecha*** Es la fecha cuando se empezó a llevar el rastreo de cuentas.

Al momento de ser ejecutado este demonio, se debe utilizar la opción *-f* para indicar la ruta donde serán creados y almacenados los archivos, bases de datos, que contendrán el registro de la actividad del sistema. Tiene establecido *satd*, un máximo para el tamaño de estos archivos; cuando este máximo es alcanzado, lo cierra y abre uno nuevo. Si el espacio del sistema se está terminando, *satd* manda un mensaje indicando que los archivos de registro de actividad sean respaldados y borrados para liberar espacio. Si este mensaje no es atendido y el espacio está por agotarse, *satd* cambia el estado del sistema a single-user o modo de sólo un usuario, dando un período de gracia muy corto antes de realizar el cambio, 10 segundo.

Este subsistema cuenta además con otras tres herramientas que permiten interpretar la información almacenada durante la intervención y rastreo de las cuentas: *sat_reduce* se encarga de eliminar la información superflua; *sat_interpret* le da formato y, *sat_sumarize* despliega un resumen de ella; por lo que con estos comandos, se puede generar un paquete compacto de información. Por ejemplo:

Para examinar los registros de un usuario en particular:

*sat_reduce -P**archivo** | sat_sumarize -u**usuario***

Donde	*-P*	Indica que sólo se listen los intentos de violación.
	archivo	Es el nombre del archivo que contiene los registros.
	-u	Especifica que sólo los registros del ***usuario*** sean tomados.

ó

*sat_reduce -u**usuario** < **archivo** > **usuario**.log*
*sat_interpret **usuario**.log | more*

Para examinar la actividad generada sobre un archivo:

*sat_reduce -e sat_open -e sat_open_ro **satfile** | sat_interpret | grep **archivo***

ó

*sat_reduce -n**archivo** -e sat_open -e sat_open_ro **satfile** sat_interpret*

Donde	*-e*	Indica que se desean listar los eventos *sat_open* y *sat_open_ro*; los cuales registran los accesos a los archivos.
	archivo	Es el nombre del archivo del cual interesa obtener su actividad.
	satfile	Es el nombre de la base de datos que contiene el registro de la actividad.
	-n	Especifica cuál es el archivo, del cual, se desea obtener su actividad.

Finalmente, dentro de la información que se almacena en los registros, se encuentra entre otra: el tipo de evento; la fecha en que se ejecutó; las llamadas de sistema utilizadas; el identificador de usuario que lo ejecutó; el estado, que indica si se llevó a cabo el evento o fallo, etc. Por lo que si se desea llevar un control estricto sobre la actividad de un usuario, los posibles intentos por modificar o borrar un archivo en particular, los intentos de ingresar al sistema, etc., se debe utilizar este subsistema que vigila toda esta actividad.

IV.III.IV. Monitoreo del sistema

Dentro de la mayoría de los sistemas existen una serie de comandos básicos que permiten realizar búsquedas, rastreos y monitoreos en el seguimiento de actividades que puedan ser sospechosas. Algunos de los comandos descritos a continuación, son comunes a la mayoría de los sistemas; pero otros son particulares a IRIX, aunque pueden existir aplicaciones que realicen la misma función. Éstos son algunos de ellos:

ps El cual permite obtener una lista de los procesos que se encuentran corriendo dentro del sistema al momento de ejecutarse el comando. En el Capítulo II se describe su uso y funcionamiento.

w Muestra un resumen de la actividad de los usuarios en el sistema.

who Con este comando se puede obtener una lista de los usuarios que se encuentran actualmente en sesión, el lugar y la hora desde donde se efectuó la conexión.

whodo Igual al comando *who*; pero además indica los comandos que está ejecutando cada uno de los usuarios.

last Despliega las fechas de las más recientes ocasiones que un usuario ha entrado a sesión. Generalmente se coloca dentro de los archivos de inicialización y configuración de ambiente de los shell, para que muestre la fecha de la última conexión; permitiendo así, determinar fácilmente si la cuenta fue utilizada sin autorización.

top Permite desplegar a intervalos continuos, una lista ordenada de los procesos que consumen una mayor cantidad de recursos en el sistema. Si el rendimiento del equipo baja drásticamente, se puede deber a la carga de trabajo existente en ese momento, a aplicaciones que entraron en ciclos infinitos y que hay que eliminar, o por aplicaciones que consumen muchos recursos. Esta herramienta nos permite precisar y monitorear la actividad del sistema, comprobando si son válidas y acreditadas las aplicaciones causantes del conflicto.

gr_top Es una herramienta similar a *top*; pero utilizada en ambiente gráfico.

osview Permite ver en una terminal, el rendimiento del sistema en general.

gr_osview Similar a *osview*; pero para ambiente gráfico.

sar Reportero de la Actividad del Sistema. Este comando se encarga de desplegar en la pantalla un reporte del uso de diversos recursos del sistema (memoria, cpu, terminales, etc.). Este reporte lo puede realizar cada determinado número de segundos durante un determinado número de veces que se dan como parámetros al ejecutar el comando.

Por otra parte, es conveniente revisar periódicamente los diversos archivos donde son registrados los mensajes de error, advertencias, violaciones de seguridad, etc., por diversos programas y procesos del sistema, durante su funcionamiento.

/var/cron/log Aquí se encuentra un registro de los errores y mensajes emitidos por aplicaciones que son ejecutadas mediante el comando *cron*. Éste, al igual que el comando *at*, pueden ser utilizados en programas maliciosos conocidos como bombas de tiempo (ver, pág. IV-58).

/var/adm/SYSLOG Aquí se registran los mensajes de la mayoría de las aplicaciones y demonios del sistema que utilizan el mecanismo de 'Registro de errores del sistema' implementado por *syslogd*.

/var/adm/sulog Contiene el registro de la utilización del comando *su*; el cual permite que un usuario adquiera la identidad de otro. Si el usuario logró adquirir la identidad, aparecerá un signo más (+) en el cuarto campo del registro; si no lo logró, aparecerá un signo menos (-). Por la importancia de este comando, se debe inspeccionar continuamente el archivo; si un usuario intenta adquirir la identidad de otro en forma continua mediante este comando, aquí se guardará el registro de ello.

Además de estos archivos, se deben revisar los mencionados a lo largo de este capítulo.

IV.III.V. Otras medidas de seguridad

En este punto se describirán una serie de comandos que permiten establecer un grado más de certeza, de que la información guardada en el sistema se encuentra segura.

IV.III.V.I. Cifrado de la información

El cifrado de la información consiste, como ya se mencionó, en revolverla y alterarla de tal forma que no pueda ser interpretada; a menos que se dé una llave correcta que permita descifrarla. En el mercado existe una gran cantidad de aplicaciones que llevan a cabo esta labor, implementando diversas técnicas que impiden, con mayor o menor grado, los posibles intentos por descifrarla. En particular, en los sistemas UNIX se cuenta con la aplicación llamada *crypt*.

Este comando implementa un mecanismo conocido como 'máquina de un rotor' junto con la técnica empleada por los alemanes durante la segunda guerra mundial, líneas de la máquina de Enigma o Enigma Alemán. El rotor permite la utilización de 256 contactos para el cifrado de 8 bits. Se conocen métodos de ataque para estas técnicas de cifrado, pero no son ampliamente difundidos y además, se requiere mucho tiempo y trabajo para lograrlo.

La llave o clave requerida para llevar a cabo esta labor, permite establecer las características únicas que tendrán las máquinas. Esta puede ser dada al momento de ejecutar el comando, pero este método tiene un inconveniente: si un usuario ejecuta el comando *ps* mientras se está procesando el programa *crypt*, éste aparecerá en el listado desplegado por *ps* junto con sus parámetros, entre los cuales se encuentra la llave; permitiendo así, obtenerla para efectuar el descifrado del archivo.

Si la llave no es dada, *crypt* la solicitará y el usuario la tendrá que teclear manualmente. Durante este proceso los caracteres tecleados no son desplegados en la pantalla, añadiendo así, una seguridad mayor. Éste es el método que se debe usar cuando sea posible para cifrar o descifrar un archivo.

IV.III.V.II. Software de dominio público

Existe una gran variedad de software de dominio público que puede ser obtenido a través de la red, el cual permite revisar e incrementar la seguridad del sistema. Si se desea, se pueden bajar estas aplicaciones; pero recordar que todo el software gratuito obtenido de lugares que no gocen de reputación, pueden ser susceptibles de contener dentro de él, algún tipo de mecanismo que viole la seguridad del sistema donde es instalado. Por eso se debe ser muy minucioso al seleccionar el lugar de donde se obtendrán, y de preferencia, se deben localizar las aplicaciones en código fuente; para poderlas analizar y efectuar las pruebas necesarias que permitan garantizar su fiabilidad.

Uno de estos paquetes es el llamado COPS, el cual revisa minuciosamente, numerosos problemas de seguridad encontrados muy frecuentemente en los sistemas UNIX, entregando al final, un reporte de las fallas encontradas. Entre los aspectos verificados por éste, se encuentran:

- Claves secretas vulnerables.
- Inconsistencias en el archivo */etc/passwd*.
- Inconsistencias en el archivo */etc/groups*.
- Cambios en el permiso de set-uid de los archivos dentro del sistema.
- Checa las rutas de búsqueda establecidas para root (path).
- Checa en general los directorios de los usuarios, y muy en especial el de root, por cualquier problema que redunde en una falla de seguridad.

Un sitio confiable para obtener este paquete es el directorio *pub/cops* del equipo *cert.sei.cmu.edu* que pertenece al Computer Emergency Response Team o Equipo de Respuesta a la Emergencia en Computadoras. El CERT fue creado por DARPA[44] después de la aparición del gusano de Internet en Noviembre de 1988, cuyo Centro de Coordinación (CERT/CC) está localizado en el Instituto de Ingeniería de Software (SEI, por su siglas en Ingles) de la Universidad Carnegie Mellon (CMU). " El CERT es un grupo proyectado para facilitar a la comunidad, respuesta a los acontecimiento relacionados con la seguridad en computadoras que involucran a los equipos conectados a Internet". Este grupo está formado por cientos de voluntarios altamente calificados en la comunidad de cómputo, como dentro del CERT/CC y otros grupos de respuesta emergentes. Este equipo emite documentos en los cuales se describen vulnerabilidades, violaciones, o demás problemas relacionados con la seguridad, encontrados y reportados; así como recomendaciones que permiten prevenirlos o solucionarlos. Este equipo cuenta con un buzón de correo electrónico mediante el cual, se puede establecer una comunicación con él: *cert(a)cert.sei.cmu.edu*.

Así como este ejemplo, existen una gran cantidad de aplicaciones diseñadas para incrementar y mantener la seguridad de los sistemas que vale la pena contemplar.

IV.IV. Seguridad en la red

La red es una de las herramientas que incrementan en gran medida el poder de un equipo: la habilidad para compartir información, recursos, dispositivos, poder de cómputo, etc. Pero, es el medio más utilizado para intentar su corrupción, así como el más difícil y complicado de controlar, por la gran cantidad de aplicaciones que trabajan sobre él.

El poder de una red se ve reflejado en el enorme desarrollo de programas y paquetes diseñados para trabajar sobre este medio; desde el mismo sistema operativo, gran cantidad de sus comandos, aplicaciones comercialmente utilizadas y ampliamente difundidas, programas especializados y de dominio público, etc. Cada uno de éstos funcionan y se configuran de forma independiente; cada uno cuenta con sus propios mecanismos de defensa

[44] Defense Advanced Research Projects Agency

y seguridad. Generalmente cuando se instalan, las opciones seleccionadas automáticamente son las más adecuadas para mantener seguro su funcionamiento; pero en ocasiones no es así o pueden ser violadas fácilmente. Es por ello que se debe conocer y analizar su estructura completamente, para poder brindar a los usuarios la confiabilidad de que su información estará resguardada eficientemente.

Por esto y otros motivos, el asegurar un equipo conectado a una red es una labor difícil, si no es que imposible. Pero, siguiendo una serie de políticas de uso, conociendo y configurando cada una de las aplicaciones que utilizan este medio, controlando los distintos medios de conexión, etc., se puede tener un grado bastante confiable y aceptable de seguridad.

La seguridad total es muy difícil de conseguir, porque durante el ciclo de vida (diseño, creación, implementación, uso, mantenimiento y generación de nuevas versiones) de las aplicaciones y sistemas operativos, se descubren y reparan nuevos agujeros en ella. Por eso es importante el mantenerse actualizado en cuanto a medidas de seguridad, así como en contacto con fuentes de información que difunden fallas y soluciones encontradas a este respecto.

A continuación se describirán las medidas más importante que deben tomarse en cuenta cundo se utilicen servicios de red en un equipo.

IV.IV.I. Uso de la opción "Equipos y cuentas de confianza"

El concepto de equipos y cuentas de confianza permite, que un usuario o un equipo que se encuentre dentro de una lista, sea considerado como confiable; y por lo tanto, no le será pedida su clave secreta al momento de realizar la conexión. Esta característica puede permitir y generar un punto vulnerable en la seguridad, por lo que se debe evitar al máximo el uso de él.

Este técnica es utilizada por muchos comandos implementados dentro del sistema; como por ejemplo: *rlogin*, *rcp*, *rdist*, *rsh*, etc. Cuando uno de éstos es utilizado para realizar una conexión a un equipo remoto, se revisa en dicho equipo si el usuario o equipo que está tratando de conectarse para realizar la operación, se encuentra dentro de esta lista; si es así, se le permite el acceso sin preguntarle por su cuenta o clave secreta.

Existen dos formas de implementar este mecanismo: el primero es utilizado por el administrador del sistema mediante la creación del archivo *.etc hosts.equiv*, para designar los equipos y usuarios que serán tomados como confiables; el segundo, es utilizado por cada usuario en forma particular mediante la creación del archivo *.rhost* en su directorio

particular, en el cual puede especificar cuáles cuentas en otros equipos remotos, se tomarán como confiables cuando se conecte al equipo mediante su cuenta.

Cada uno de estos archivos debe tener los permisos de 644[45] y los debe crear el usuario dentro de su directorio. En el caso de root, él debe crear el archivo */etc/host.equiv* y puede tener en forma particular el archivo */.rhosts* como cualquier usuario. Dentro de éstos se puede especificar en cada línea, cualquier opción de las siguientes:

- -El nombre del equipo que será considerado como confiable (*gama.proy.mx*); en cuyo caso, todos los usuarios que se conecten desde este equipo, estarán dentro de este concepto.
- El nombre del equipo seguido de la cuenta de un usuario (*beta.proy.mx arturo*); en cuyo caso, sólo el usuario *arturo* de este equipo, será considerado como confiable.
- Un signo menos (-) antepuesto a cualquiera de los dos casos anteriores, es utilizado para negar el derecho; por ejemplo: *-gama.proy.mx* niega el uso de este mecanismo a cualquier usuario proveniente de este equipo, y *beta.proy.mx -arturo* niega el concepto al usuario *arturo* del equipo *beta.proy.mx* únicamente.
- Un signo más (+); que lo habilita para todos los usuarios de cualquier equipo.

Como nota importante, es conveniente indicar que se deben evitar al máximo el uso de estos mecanismos, ya que si se llega a violar alguna cuenta en un equipo que tenga este mecanismo activado, se tendrá acceso rápidamente a cualquiera de los equipos contemplados dentro de él. Un punto que los cracker y software malicioso explotan en primera instancia, es la existencia de este mecanismo; por ello, se debe evitar.

IV.IV.II. El "super-servidor" de Internet

Super-servidor es el nombre con el que se le conoce al demonio *inetd* que se encarga de controlar los servicios de Internet. Básicamente, éste es un demonio que se encuentra corriendo dentro del equipo, monitoreando los puertos de comunicación de la red; cuando detecta que algún servicio externo trata de comunicarse por alguno de estos puertos, ejecuta a la aplicación adecuada que lo atenderá. De esta forma, en lugar de tener corriendo en el equipo todos los demonios de los distintos servicios, únicamente se encuentra corriendo *inetd*, que a su vez, se encarga de correr al servicio adecuado cuando llegue un paquete para él; reduciendo la carga del equipo.

El archivo */etc/inetd.conf* es el que se utiliza para configurar este demonio; en él se coloca el nombre del servicio y el programa que se ejecutará cuando le llegue alguna petición. Junto

[45] Si se tiene dudas sobre el tipo de permisos, consultar la ayuda en línea mediante el comando: *$ man chmod.*

con este archivo, se utiliza el *'etc/services*; el cual contiene una lista de los servicios y los puertos que atenderá cada uno. Con estos archivos, *inetd* sabe qué aplicación correr cuando llegue algún mensaje por cualquiera de los puertos. Estos archivos deben pertenecer a root y tener los permisos de 644; es decir que únicamente los puede modificar root.

Como medida de seguridad, se debe tener una lista de todos los comandos permitidos, así como de los activados, y revisarla continuamente. Si se llega a detectar cualquier anomalía, se debe investigar. Gran cantidad de software de dominio público, requiere modificar e ingresar un servicio a la lista para funcionar, por lo que si alguien logra activar un servicio sin el consentimiento de root, aquí se puede detectar fácilmente[46].

Básicamente, el super demonio se encarga de controlar tres tipos de servicios: Los servicios estándar, que son aquéllos que tienen un puerto definido y autorizado (well-know port) y que corresponden a alguna implementación de uno de los protocolos estándar de Internet, generalmente se encuentran enlistados en el archivo */etc services*; Los servicios RPC que son implementaciones que utilizan como transporte las llamadas RPC de SUN, generalmente se encuentran enlistados en el archivo */etc/rpc*; y los servicios tcpmux, que son servicios no estándar que no utilizan puertos definidos. Tcpmux tiene un puerto definido y cuando alguna aplicación se comunica por él, debe especificar el servicio que desea utilizar.

Al igual que *inetd*, los servicios que lo utilizan y que están diseñados para ello, reportan mensaje de error mediante *syslogd*[47], que registra cada evento en el archivo */var/adm/SYSLOG*. Si a algunos de los servicios indicados en *etc/inetd.conf* se antepone un signo de interrogación (*?*) en el campo para la ruta del comando, es deshabilitada esta opción; por este motivo se debe llevar un control sobre qué servicios deberán registrar sus errores y cuáles no.

Algunos comandos requieren alguna opción para habilitar el reporte de errores; como *ftpd*, al cual se le debe incluir la opción *-l* en la línea del comando a ejecutar dentro del archivo */etc/inetd.conf*, para que registre en el archivo *var/adm SYSLOG*, cada conexión exitosa o fallida que se lleve a cabo con este servicio. Si esta opción es especificada doble vez (*-ll*), se registrarán también los subcomandos *get*, *put*, *append*, *delete*, *make directory*, *remove directory* y *rename* utilizados en las sesiones, junto con los parámetros que se utilicen. Si es especificada tres veces, se registrará además, el número de bytes transferidos por los subcomandos *get* y *put*. Existen otras opciones para este servicio; como la *-d* que permite registrar información adicional utilizada en labores de depuración; la *-t*, que permite especificar el tiempo máximo de inactividad permitido para una conexión, como default es de 15 minutos, etc..

[46] Cabe destacar que no todos los servicios utilizan el super demonio para funcionar

[47] *Syslog* es el sistema de reporte de errores implementado dentro de IRIX. Recolecta los mensajes generados por varios procesos y los coloca en su destino, según se especifique en el archivo de configuración *etc syslog.conf*

Si se desea suspender el servicio de *ftp* momentáneamente, se puede crear el archivo *etc/nologin*[48]; ya que cuando existe, no permite ninguna conexión y presenta en la pantalla el contenido de este archivo, que suele contener un mensaje indicando cuál es la causa por la que el servicio se suspendió, así como la hora y fecha en que será reanudado (ver 'Comunicación con usuarios', pág. II-45).

Otro servicio que merece atención es el *tftp*[49], que da la posibilidad a los usuarios de enviar o traer archivos de equipos remotos. La opción *-s* de este comando, se utiliza para especificar la ruta de algún directorio a la cual pueden tener únicamente acceso, los usuarios que utilicen este servicio. Por default se especifica */usr/local/boot*, que suele ser utilizada para arrancar equipos sin unidad de disco en forma remota. Si es necesario cambiarla o colocar otra ruta, se debe utilizar esta opción; pero no es aconsejable el eliminarla, ya que daría acceso al sistema completo sin restricciones.

Otro demonio importante es el *rexd*, que pertenece a los servicios RPC. Se encuentra deshabilitado por default, y por medidas de seguridad, sólo debe ser utilizado cuando se requiera, o estar activado en forma permanente cuando se esté trabajando en una red aislada, que contiene equipos conocidos o confiables.

El analizar y configurar cada uno de los servicios de red disponibles, es una tarea muy larga que el administrador debe realizar minuciosamente, al igual que se hizo con estos pocos comandos descritos, para garantizar la seguridad del equipo. Por último, el archivo */etc/config/inetd.options* es utilizado para colocar opciones[50] que permiten controlar el funcionamiento del demonio *inetd*.

IV.IV.III. ftp

Este servicio permite que usuarios remotos puedan transferir archivos entre los dos equipos involucrados. Debido a lo crítico que puede resultar el que un usuario pueda robar información mediante este servicio, éste cuenta con varias mediadas de seguridad.

En primer lugar, para que un usuario pueda conectarse mediante este servicio, debe identificarse mediante su cuenta y clave secreta. Ésta es la principal medida de seguridad y funciona como se indicó en el punto de 'Seguridad en el acceso al sistema' de este capítulo. Como medida adicional, el acceso le es negado a cualquier cuenta que carezca de una clave asignada. También se puede consultar el punto anterior para ver las medidas de seguridad

[48] Cuando existe este archivo, también se suspende el servicio de *telnet*.
[49] Trivial File Transfer Protocol.
[50] Para mayor información sobre las posibles opciones de este demonio, consultar la ayuda en línea mediante el comando: *$ man inetd*

que se pueden implementar en el demonio de este servicio, mediante diversas opciones dadas al comando dentro del archivo *etc inetd.conf*.

Para incrementar la seguridad, el administrador puede crear el archivo *'etc/ftpusers* y colocar en él, la lista de las cuentas de los usuarios a los cuales no les será permitido utilizar este servicio; de tal forma que cuando *ftp* recibe una petición, primero es inspeccionado este archivo y si la cuenta utilizada para tratar de ingresar se encuentra en él, el servicio le será negado.

El archivo */etc/ftpusers* permite trabajar con otra medida de seguridad denominada "Cuentas restingidas". Para indicar que una cuenta será restringida, se debe agregar a continuación de la cuenta listada en este archivo, la palabra *restricted*. Cuando una cuenta es restringida, se efectúa un *chroot* cuando el usuario entra al sistema mediante este servicio. Esto permite restringir el acceso del usuario al directorio que le fue asignado. Mediante el proceso que efectúa *chroot*, el directorio del usuario aparenta ser la raíz del sistema de archivos (/); éste es el mismo proceso efectuado con la cuenta de *anonymous*[51] implementada también, dentro del servicio *ftp*. Por este motivo, la cuenta de los usuarios restringidos debe contener ciertos comandos y directorios para funcionar.

El servicio *ftp* dispone de una medida en la cual, un usuario puede especificar un número de cuentas, a las que automáticamente le será permitido el acceso; es decir, que no le será pedido introducir ni la cuenta ni la clave secreta. Para ello, cada usuario que lo desee activar, debe crear el archivo *.netrc* en su directorio y colocar en él, el nombre del equipo, cuenta y clave secreta sin cifrar, de cada cuenta que desee utilizar este mecanismo. Como se puede observar, ya que en este archivo se debe colocar información valiosa y sobre todo, no cifrada, una persona que logre ingresar a esta cuenta, podrá tener acceso inmediato a las distinta cuentas listadas en este archivo; por ello, debe ser evitado cuando sea posible. Si se decide permitir la utilización de esta medida, se deben tomar consideraciones especiales, como el dar los permisos de 700 al archivo para evitar que otros usuarios del mismo equipo lo puedan leer.

IV.IV.IV. Seguridad en el sistema de ventanas X

El sistemas de ventanas X es un ambiente que permite ejecutar aplicaciones remotas en forma transparente para el usuario; facilitando el aprendizaje y manejo de los sistema UNIX. En este sistemas se manejan básicamente dos términos: un servidor X y los clientes. Los clientes son programas que se encuentran corriendo en un equipo UNIX. Los servidores X

[51] Es una cuenta pública que no tiene clave secreta y que es utilizada para distribuir archivos a la comunidad. Si se utiliza este sistema, se debe llevar una estricta medida de seguridad sobre cada una de los archivos que permiten configurarla e implementarla.

son programas que se encuentran corriendo en cualquier equipo conectado a la red y que utilice el sistema de ventanas X; controlan el acceso tanto a la pantalla como al teclado y ratón. Por ello, cuando un programa cliente corriendo en un servidor UNIX requiere desplegar o leer información del teclado o ratón de un equipo, realiza la transacción a través del servidor X que atiende a dicho equipo.

Esta característica da ocasión a que puedan existir ataques que traten de explotar las deficiencias de este sistema; por ejemplo: La mayoría de la aplicaciones diseñadas para este medio, tienen la posibilidad de ser ejecutadas e indicarles mediante parámetros, que sean desplegadas o atendidas por servidores X remotos. Dentro de estas aplicaciones existen algunas que permiten capturar y guardar en un archivo, una copia de la información desplegada en la pantalla. Por lo que si no se dispone de una medida que permita restringir quién puede hacer uso del servidor X, cualquier persona utilizando estas herramientas puede tomar una copia de la información que esté desplegada en la pantalla de otro usuario.

Debido a la deficiencia de seguridad anterior, así como otras, se ha ido perfeccionando este sistema al incorporar nuevas y mejores medidas de control que permiten restringir quién y qué aplicaciones pueden ser ejecutadas remotamente. A este respecto, el sistema de ventanas X versión 6 implantado en el sistema operativo IRIX, cuenta con 5 mecanismos de protección o sistemas de control de acceso, que permiten mantener la seguridad en este medio. Cada uno de ellos, supone un grado mayor de protección y son:

a) Host Access
b) MIT-Magic-cookie-1
c) XDM-Authorization-1
d) SUN-DES-1
e) MIT-Kerberos-5

El primero, Host Access, es el más ampliamente difundido y utilizado. En este sistema , el control se lleva en base a equipos; es decir, el acceso se da o niega a equipos en particular. Este hecho hace que sea utilizado en ambientes donde cada persona utiliza su propio equipo; de tal forma que al restringir a un equipo se restringe a una persona. Para ello, se lleva una base de datos inicial en la cual se especifican los equipo que están autorizados a realizar conexiones. Esta lista puede ser modificada mediante el comando *xhost* en forma particular por cada usuario.

Los métodos siguientes comparten el hecho de que el control se lleva en base a usuarios; es decir que se puede dar o restringir el acceso a usuarios en particular. Además, todos ellos utilizan el archivo *.Xauthority* para almacenar una copia de una llave que permite verificar la integridad de las personas y saber si están autorizadas a realizar la conexión. Al ingresar al sistema en ambiente gráfico, XDM crea en el directorio particular de cada usuario, este archivo.

El segundo, MIT-Magic-cookie-1, es un método implementado por el Instituto de Tecnología de Massachusetts que consiste en la creación de una galleta mágica (llave). Cuando un usuario entra a sesión remotamente mediante XDM, este último se encarga de generar un galleta que es asignada a la conexión y que consta de una secuencia de 128 bit. La copia del usuario es guardada en el archivo *.Xauthority* en su directorio. Cuando se corre una aplicación, se envía junto con la información para iniciar la comunicación, la galleta. Si la galleta del cliente coincide con la que tiene el servidor, se establece la comunicación. Esta característica hace que se pueda identificar y restringir el acceso entre distintos usuarios.

Este método no es muy seguro, ya que la información es enviada por la red sin ninguna protección; por lo que cualquier persona que implemente un software para espiar la red, puede leer y obtener la galleta fácilmente y utilizarla para obtener acceso al servidor X. Este método en conjunto con el anterior, forman un sistema de protección que funciona bastante bien en la mayoría de las redes. Si se requiere una mayor protección, se pueden utilizar cualquiera de los métodos restantes, que envían la información cifrada a través de la red, evitando que alguien la pueda leer.

En el tercer método, XDM-Authorization-1, la protección se lleva a cabo de la siguiente forma: Cuando el usuario entra a sesión, se genera una llave; cuya copia del usuario es almacenada en el archivo *.Xauthority*. Esta llave consta de dos partes; la primera es una llave de 56 bit's que es utilizada para cifrar información utilizando un método DES[52]; la segunda, son 64 bit's de datos generados aleatoriamente que sirven para identificar al usuario. Posteriormente cuando el usuario trata de ejecutar aplicaciones remotamente, se genera un paquete de 192 bit's que consta de dos partes: un identificador que sirve para definir el proceso, y la fecha actual en segundos. Este paquete es cifrado utilizando la llave generada al inicio de sesión y enviada por la red al servidor X, que realizando un procedimiento inverso, puede verificar la integridad del proceso y determinar si tiene los privilegios para poder ser atendido y llevarse a cabo la conexión.

El cuarto, SUN-DES-1, es basado en un sistema de procedimientos de llamadas remotas que utilizan llaves públicas seguras, implementado en las versiones más recientes de SunOS. Mediante esta técnica, el servidor X puede identificar perfectamente al usuario que ejecutó el proceso cliente, y que solicita la conexión; determinando de esta forma, si tiene la autorización para hacerlo. La información es cifrada para viajar por la red, y la llave utilizada para cifrarla se encuentra almacenada nuevamente en el archivo *.Xauthority* de cada usuario. Esta llave identifica al servidor X y es generada por XDM al momento de arrancar. Para permitir acceso a distintas personas se utiliza el comando *xhost* que será descrito más adelante.

Finalmente el MIT-KERBEROS-5 es un método en el cual, una entidad separada, el servidor de Kerberos, autentifica a las diversas partes que intervienen en la comunicación (servidor X y cliente), mediante una llave secreta, ver pág. IV-54. Cada entidad conoce

[52] Data Encryption Standar. Estándar de Cifrado de Datos.

únicamente su llave y Kerberos la de todos, para poder certificarlos. Este método al igual que el anterior, utiliza el comando *xhost* para otorgar o negar el acceso a la pantalla, a usuarios en forma individual; ya que son basados en nivel de usuarios y no de equipos como el caso del primero.

Como se puede ver, a excepción del primero, que utiliza una base de datos donde almacena el nombre de los equipos que pueden tener acceso, los restantes guardan la información que permite verificar la integridad y determinar los derechos para efectuar la conexión, en el archivo *.Xauthority* de cada usuario. La diferencia básica de todos ellos, consiste en los métodos empleados para garantizar que no pueda ser descifrada la llave. La información almacenada en este último archivo, puede ser manipulada por el comando *xauth*, que entiende el formato de cada una de las llaves empleadas en cada método, de tal forma que permite listar, mezclar, extraer, copiar, etc. llaves de un archivo a otro.

En la versión 5.3 de IRIX se tiene un error que permite a un usuario, tener acceso a la pantalla de otro, aunque se le esté restringido por alguno de los cuatro últimos métodos (que utilizan el archivo *.Xauthority*). Para solucionarlo, se debe deshabilitar el uso de la memoria compartida de transporte, añadiendo la opción *-shmnumclients 0* a la línea que invoca al servidor X en el archivo *usr/lib/X11 xdm Xservers*. Para el servidor X de la pantalla 0, la línea debe quedar de la siguiente forma:

```
:0 secure /usr/bin/X11/X -bs -nobitscale -c -pseudomap 4sight -solidroot sgilight blue -cursorFG red -cursorBG white -shmnumclients 0
```

IV.IV.IV.I. El archivo /etc/X*.hosts

Los archivos */etc/X*.hosts* contienen la lista inicial de los equipos a los que les está permitido tener acceso a cada servidor X. El * debe ser sustituido por el número de servidor X al cual corresponda la lista. Es decir, un equipo puede tener varios servidores X corriendo y se puede definir qué equipos pueden tener acceso a qué servidor con su respectivo archivo.

/etc/X0.hosts	---	Servidor X 0
/etc/X1.hosts	---	Servidor X 1
/etc/X2.hosts	---	Servidor X 2

Por lo general, un equipo sólo tiene corriendo a un servidor X; por lo que únicamente se utilizaría el archivo */etc/X0.host*. En este archivo se debe colocar los nombres de los equipos a los que se les permitirá el acceso, uno por renglón. Si se coloca un signo más (+) se permitirá el acceso a todos los equipos remotos. Si se encuentra vacío o no existe, no se dará acceso a ningún equipo.

Este archivo debe ser editado por el administrador del sistema, y colocar en él , el nombre de los equipos que considere confiables y a los que se les dará el acceso inicialmente. Posteriormente y en forma particular, cada usuarios podrá modificar esta lista para permitir o negar el acceso al servidor que esté utilizando, de otros equipos.

IV.IV.IV.II. xhost

Éste es un comando que permite modificar la lista interna de equipos autorizados para realizar una conexión con el servidor X. Cuando un servidor X es ejecutado, lee su respectivo archivo */etc/X*.hosts* para determinar a qué equipos les estará permitido realizar conexiones con él. Esta lista es mantenida por el servidor y puede ser modificada, añadiendo o borrando equipos de ella, por el comando *xhosts*.

Por default, el sistema IRIX viene configurado para permitir el acceso a cualquier equipo; es decir, no tiene activada ninguna protección. Por lo que si se requiere, se tiene que activar de la siguiente forma:

En primer lugar, se debe editar el archivo *.etc/X*.hosts* y especificar en él, el nombre de los equipos que serán considerados como confiables. En segundo lugar, existen dos archivos que liberan el acceso y que permiten establecer una conexión a todos los equipos (desactivan la protección). Estos archivos son ejecutados cada vez que un usuario entra a sesión. Uno afecta a los usuarios que se conectan directamente desde la consola, *var X11 xdm/Xsession.d* y el otro a los usuarios que se conectan desde terminales gráficas remotas, *var X11 xdm.Xsession*. En ellos, se debe modificar la línea que da acceso a todos los equipos de la siguiente forma:

	Antes	Después
Opción 1	/usr/bin/X1/xhost +	/usr/bin/X1/xhost -
Opción 2	/usr/bin/X1/xhost +	# /usr/bin/X1/xhost +

En la primera, se cambia el signo más (+), que desactiva el mecanismo de protección, por el de menos (-), que lo habilita. En la segunda, se comenta la línea; con lo cual, permanece activa y se da acceso a los equipos especificados en el archivo */etc/X*.hosts*, o a ninguno; si es que no existe este archivo.

Por otro parte, cada usuario en forma individual puede activarla e indicar a cuáles equipos les estará permitido utilizar el servidor X que estén utilizando.

IV.IV.IV.III. xlock

Este es un comando que permite bloquear la pantalla y no dar acceso a ella, hasta que se teclee la clave secreta de la cuenta del usuario que está trabajando en ese equipo. Este comando debe ser utilizado cuando se pretenda abandonar la estación de trabajo por algunos minutos y no se desee salir de sesión; por considerar que el tiempo de inactividad será mínimo y se quiere seguir trabajando con él. Esto evita que otra persona pueda utilizar el equipo aprovechando la ausencia del dueño de la cuenta, para realizar labores dañinas sobre ella.

Así como éstos, existen otros comandos que permiten garantizar la seguridad en la comunicación cuando se trabaje en ambiente gráfico, y que deben ser considerados y analizados minuciosamente.

IV.IV.V. Software de dominio público

Al igual que en los temas anteriores, en el caso de servicios y comunicación de redes, existe una gran cantidad de software de dominio público que se puede obtener para aumentar, controlar y vigilar la seguridad en este medio. Dentro de estos destaca el producto conocido como Kerberos. Éste es un servicio de autentificación confiable desarrollado por el grupo Project Athena del Instituto de Tecnología de Massachusetts, y puede ser obtenido el código fuente del directorio *pub/kerberos* del equipo *athena-dist.mit.edu* mediante un *ftp* anónimo.

En este sistema, para poderse efectuar un servicio de red, se tienen que identificar tanto el cliente como el proveedor del servicio. Kerberos es el que lleva a cabo esta identificación, para lo cual cuenta con una base de datos en la cual se encuentra un lista de cada uno de los clientes y las llaves que le fueron asignadas; por su parte, cada uno de los clientes conoce únicamente su llave. De esto se deduce que los clientes - los servicios de red - tienen que estar diseñados para solicitar esta identificación antes de poderse efectuar la comunicación. Por este motivo, dentro del paquete de Kerberos se incluyen los servicios de Berkeley[53] más comunes que ya se encuentran modificados para realizar la autentificación.

Este sistema no es una solución al problema de la seguridad, sino un medio más de protección; ya que aunque esté instalado, puede ser violado. Esto se debe básicamente a que los clientes utilizan una llave o clave secreta para identificarse, y si una persona llega a conocerla, podrá utilizar los servicios protegidos sin mayores problemas. Esto quiere decir, que la seguridad recae nuevamente en los usuarios, quienes deben de seguir un conjunto de políticas para impedir que sean descubiertas estas llaves.

[53] Los servicios de Berkeley son más conocidos como comandos r, porque el nombre del comando empieza con una r, como *rlogin*, *rsh*, *rcp*, etc.

Otro producto que puede resultar de utilidad es el conocido como *satan* por la iniciales en inglés[54] de Herramienta de Administración de Seguridad para el Análisis de Redes. Éste es un sistema rastreador de seguridad en red que, al especificar un equipo, puede realizar varias pruebas para determinar su vulnerabilidad. Las pruebas pueden ser ligeras, en cuyo caso sólo se revisan los servicios de RPC; si se especifican pruebas normales, se checan además, los servicios asignados a los puertos definidos internacionalmente, como *telnet*, *ftp*, *smtp*, etc.; y si es una búsqueda intensa, se realizarán pruebas exhaustivas en los servicios disponibles.

Como éste y cualquier otro software de dominio público, hay que elegir el lugar de donde se obtendrá; que goce de reputación. De preferencia, obtener el software en código fuente para poderlo analizar y tener la certeza de que no contendrá ningún tipo de truco dentro de él.

IV.V. Software malicioso

La seguridad de un sistema, no sólo se debe mantener pensando en que alguna persona trate de tener acceso a él sin autorización; sino tomando en cuenta la posibilidad de que programas diseñados para causar algún tipo de daño sean introducidos al sistema; ya sea por usuarios legítimos, con o sin su conocimiento, o aprovechando alguna debilidad del sistema (agujero en la seguridad) que permita su propagación.

Aunque el tema de virus es el foco de atención en los últimos años, por el incremento de ellos y lo perjudicial de sus consecuencias, éstos existen desde hace mucho tiempo. En este contexto, existen una gran variedad de programas cuyo objetivo es el de causar daño, y que por la forma de comportarse, reciben diferentes nombres: caballos de Troya, gusanos, etc. Al conjunto de todos estos programas se les denomina software malicioso. Por este motivo, dentro de este punto se describirá la manera de actuar de estos programas, para que conociéndolos, se puedan tomar las medidas pertinentes y prevenir cualquier ataque e infección.

IV.V.I. Tipos de software malicioso

Como se mencionó, dependiendo de las técnicas de penetración que utilizan los programas para violar la seguridad de un sistema, y realizar la labor para la que fueron diseñados, es el nombre que reciben estas aplicaciones. Algunos de ellos son:

[54] SATAN (Security Administrator Tool for Analyzing Networks)

IV.V.I.I. Caballos de Troya

Este nombre es dado a aplicaciones que se comportan como el famoso caballo de Troya. Según la historia, los griegos al querer vengar un ultraje por parte de los troyanos, sitiaron su ciudad. Este sitio duró diez años, después de los cuales, los griegos fingieron retirarse y dejaron en las afueras un enorme caballo de madera; dentro del cual se hallaban soldados escondidos. Los troyanos incautamente tomaron el caballo y lo introdujeron a su ciudad sin saber lo que contenía. Por la noche, los soldados escondidos salieron y de esta forma los griegos pudieron tomar la plaza.

Por este motivo, a un programa que aparenta ser de utilidad, pero que dentro de él se encuentran escondidas instrucciones que realizan una labor diferente, suelen llamárseles caballos de Troya. Los diseñadores de este tipo de programas pretenden engañar a las personas; para ello, pueden conseguir el código fuente de alguna aplicación útil, añadirle su propio código que realiza la función oculta, compilarla y distribuirla o incitar a que sea utilizada su aplicación. Los usuarios sin conocimiento de esto, la ejecutan y por tanto, la aplicación adquiere los privilegios del usuario que la ejecutó. Sin que ellos se den cuenta, realiza la tarea para la que supuestamente fue creada, pero además, realiza la función oculta, que puede consistir en borrar los archivos del usuario, modificarle los permisos para permitirle el acceso a el intruso, etc. Los usuarios normales no suelen percatarse de esta situación, sino hasta mucho tiempo después.

Si el usuario que cae en esta trampa es root, el asunto se complica; ya que se puede alterar o tener acceso al sistema de archivos completo.. Algunos intentos para engañar al administrador consisten en utilizar programas que desempeñan la función de algún comando simple y común, como lo es *ls* y *cat*, y tratar de que root ejecute éstos en lugar de los originales. Para evitarlo, el *path* de busqueda[55] de root en especial, no suele contener al punto (.) que representa al directorio actual, dentro de él. Muchos administradores suelen ponérselo por comodidad, incurriendo en una falla de seguridad; que puede tener como consecuencias, que root caiga dentro de este tipo de trampas sin darse cuenta.

IV.V.I.II. Virus

El término de virus se da a aplicaciones que se comportan de forma similar a los virus humanos: son gérmenes patógenos que se ocultan dentro de las células, se reproducen e infectan a otras; son capaces de transferirse de un cuerpo a otro, transfiriendo la enfermedad a más personas; algunos suelen permanecer latentes hasta que algo los activa y producen la enfermedad. Por eso, a un programa que se pasa de máquina en máquina por sí mismo,

[55] search path. Es una variable de ambiente que especifica el lugar y orden en que se buscará un comando para ser ejecutado. En el SO MS-DOS, primero se busca en los comandos internos, si no se encuentra, en el directorio donde se está trabajando; si no se encuentra, en cada uno de los directorios indicados en la variable PATH. En el SO UNIX sólo se busca en los directorios indicados en la variable PATH.

ocultándose dentro de aplicaciones, infectándolas y comenzando su ataque, destruyendo información almacenada en los equipos o ejecutando las instrucciones que el diseñador colocó dentro de él, suele llamársele 'virus computacional'.

Dependiendo del autor, el virus puede ser dañino, si destruye la información contenida en el equipo; juguetón, si no afecta la información pero ocasiona interferencia con el uso del equipo; benigno, si simplemente se propaga pero no causa ningún daño ni problema, etc.

Tres características importantes son las que describen a un virus: la replicación, activación y el objetivo. La replicación es un mecanismo que permite a un programa buscar otros programas, analizarlos para determinar si pueden ser infectados; en caso afirmativo, esconde el código dentro de ellos y suelen activar una bandera para indicar que ya fue infectado. El mecanismo de activación es el que hace que el código entre en función y realice el objetivo u operación dañina para la cual fue creado.

Finalmente, los efectos que generan los virus suelen ser muy diversos y en ocasiones catastróficos, y el prevenirse de ellos suele ser un asunto muy complicado. La mayoría de los contagios son por no seguir unas políticas de uso de software adecuadas; pero otros, explotan vulnerabilidades de la seguridad del sistema y penetran sin que nadie se dé cuenta. Por ello, además de las políticas de uso de software que se describirán más adelante, se debe contar con un plan que permita recuperarse de cualquier infección.

IV.V.I.III. Gusanos de red

El término de gusanos de red se da a aplicaciones que utilizan las redes para expandirse entre los distintos equipos que la componen; una vez en ellos, se comportan como virus, infectando los equipos y realizando las labores destructivas para las que fueron creados.

Para dispersarse sobre la red, los gusanos utilizan algún tipo de transporte o aplicación de red; como suele ser el correo electrónico, la posibilidad de ejecución remota, una conexión remota mejor conocida como *telnet*, etc.; puede intentar el violar una clave secreta de una cuenta para tener acceso al sistema; puede , buscar alguna falla en la seguridad de cualquier aplicación de red, etc.

Un gusano tiene las mismas características que describen a un virus: replicación, activación y objetivo. La diferencia es que los gusanos utilizan la red como medio de distribución.

Dentro de este tema existe un gusano muy renombrado por los efectos y desenlace que tuvo, conocido como el gusano de Internet. Este gusano explotaba tres deficiencias en la seguridad de algunos sistemas operativos UNIX derivados de la versión de Berkeley. La primer deficiencia explotada por este virus era la opción *debug* del programa *sendmail*, que es el encargado del correo a través de la red. Esta deficiencia permitía enviarle correo a un

programa, el cual era ejecutado y se le pasaba como parámetro el cuerpo del correo. La segunda deficiencia se encontraba en el programa *fingerd*, que bajo ciertas condiciones ejecutaba un shell y la entrada de este shell era conectada con el equipo remoto que ejecutaba el comando *finger*. La tercer deficiencia era la del programa *rsh* y *rexec*, que permiten ejecutar shell remotos, en este caso el gusano trataba de violar claves secretas de cuentas para tener el acceso.

Este virus se expandió rápidamente infectando cientos de equipos en un espacio de 48 hrs., y aunque no fue diseñado para causar ningún daño, su multiplicación rápida ocasionó que los sistemas afectados negaran el acceso a otros usuarios y en ocasiones la caida de sistemas por el exceso de copias del virus ejecutándose. Finalmente, la conclusión de esta historia fue la detención y sentencia de la persona que lo creo e introdujo en la red.

IV.V.I.IV. Bomba de tiempo

Básicamente, este tipo de programas son virus que se activan a una fecha u hora determinada. Para ello, suelen utilizar la fecha del sistema o algún otro mecanismo, como podría ser el caso de los comando *at*, *cron* en los sistemas UNIX que permiten ejecutar una aplicación a una fecha y hora determinada sin necesidad de que esté en sesión la persona que lo programó.

Si para activarse el virus utiliza algún otro mecanismo o indicador, se dice que es una bomba lógica y cuando la condición esperada ocurre, el virus se activa y comienza su labor destructiva.

Finalmente dentro de este tema, podríamos decir que existen innumerable tipos de aplicaciones, y las formas de atacar, suelen ser muchas y muy variadas. Por otro lado, cada vez suelen aparecer más, por lo que se debe estar alerta y contar con buenas medidas de prevención, como las descritas a continuación.

IV.V.II. Métodos de prevención

Nuevamente, al igual que en el caso de ataque por personas, la prevención de posibles agresiones de este tipo de aplicaciones, no sólo es responsabilidad del administrador; sino también de los usuarios. Por tal motivo, al igual que en el otro caso, la implantación de políticas de uso de software, la educación de los usuarios y establecer rutinas de monitoreo, permitirá reducir los riesgos de infección.

En el caso de los virus, se puede adquirir software que permita detectarlos y erradicarlos, ya que se puede rastrear la secuela que dejan al infectar las aplicaciones; pero tratándose de caballos de troya, esto resulta muy difícil, porque no existe un patrón en común, a no ser que se reincida en la utilización del mismo método (caballo) para efectuar los ataques.

IV.V.II.I. Políticas de uso del software

Como se describió anteriormente, uno de los métodos que se utilizan para atacar un equipo, es el de pasar oculto dentro de una aplicación un código que realice la labor dañina. Por esto es importante establecer buenas medidas que permitan regular y controlar el tráfico indiscriminado de aplicaciones por la red. Esta medidas deben contemplar entre otras cosas:

- El adquirir y utilizar preferentemente, software con licencias (que no sea de dominio público).
- Mantener en lugar seguro los medios de distribución en que se encuentre el software.
- Someter a pruebas el software antes de ser instalado.
- Realizar respaldos del producto antes y después de su instalación.
- No utilizar software pirata.
- Mantener un registro del software instalado.
- Prohibir que los usuarios instalen software en forma particular.
- Prohibir y prevenir que los usuarios tengan acceso a los archivos fuente de las aplicaciones; ya que pueden ser utilizados para generar caballos de troya.
- Prohibir que los usuarios bajen aplicaciones públicas de la red y si se les permite, verificar:
    - Que el lugar de donde se obtiene el producto, sea confiable y goce de reputación.
    - De preferencia, obtener el código fuente en lugar del código ejecutable.
    - Que antes de ejecutar el software, éste sea analizado, o en su caso, probado por personal especializado.

- Mantener un registro y reporte de los procedimientos que se llevaron a cabo para realizar su instalación y configuración; para permitir una reinstalación más ágil en ocasiones sucesivas
- Si se desarrollan aplicaciones, documentarlas y controlar todas sus modificaciones.
- Si se detecta algún problema que pueda ser causa de algún intento de violación, desactivar, si es posible, la aplicación mientras se busca una solución que repare dicha situación.
- Reportar el problema a las personas adecuadas.
- Si se detectan errores (bug) en alguna aplicación, obtener los parches correspondientes de sitios autorizados para su distribución.

IV.V.II.II. Educación de usuarios

Nuevamente, el educar a los usuarios en la forma de prevenir infecciones, en la manera en que se propagan y atacan los distintos tipos de virus, los posibles síntomas que puede presentar tanto el sistema como las aplicaciones al ser infectadas, la manera de erradicar los virus, las políticas de uso de software que se adoptarán, así como indicarles cuáles son los planes de contingencias en caso de infección, permitirá reducir el riesgo de posibles infecciones y minimizar los daños al ser detectados rápidamente. Éste es otro método para prevenir, pero no solucionar, el problema de los virus.

IV.V.II.III. Monitoreo

El monitorear el sistema regularmente, permitirá detectar cualquier anomalía. Para ello, se pueden utilizar las herramientas de que se dispongan en forma manual o automatizada.

El monitoreo puede consistir simplemente en revisar qué programas se encuentran en ejecución; si se detecta alguna aplicación extraña habrá que investigarla. Otro síntoma puede ser el que copias de una aplicación se encuentren corriendo un número de veces fuera de lo común, por lo que es conveniente llevar un registro de los procesos normales, así como de la carga de trabajo del CPU a determinados momentos; si sale de lo normal, habrá que investigar de qué se trata. También, se puede llevar un registro de las aplicaciones almacenadas en el sistema, el cual indique la fechas de creación, última modificación y su tamaño. Este registro se debe checar continuamente y permitirá determinar si alguna aplicación ha sido modificada sin autorización. O simplemente, revisar y llevar un control de los intentos fallidos para entrar al sistema; si éstos son continuos, serán síntomas claros de que se sufre un ataque.

Finalmente, se puede adquirir y emplear software especialmente diseñado para encontrar y erradicar los virus, conocido como programas vacunas, que permitan dar una mayor tranquilidad tanto a los usuario como al administrador.

IV.VI. ORANGE BOOK

Dentro de todo sistema, debe existir un mecanismo que nos permita definir una escala para poder determinar su estado. En lo tocante a la seguridad, existe un documento creado por el departamento de defensa de los Estados Unidos, que lleva por título "Criterio de Evaluación de Sistemas de Cómputo Confiables" o mejor conocido como el *Orange Book* o Libro Naranja por el color de su portada. Esta publicación forma parte de una colección de

documentos relacionados con la seguridad, conocida como " La Colección Arco iris" del NCSC[56] por lo colorido de sus portadas.

Este documento clasifica a los sistemas de cómputo dentro de cuatro divisiones jerárquicas en lo referente a la protección implementada dentro de ellos. Fue creado con tres objetivos en mente: El de proveer un sistema de medida, del grado de confianza que se puede obtener dentro de un sistema, para el proceso y almacenamiento seguro de información clasificada o vital; para proveer una guia en la manufactura y producción de qué es lo que deben contener los productos, para satisfacer los requerimientos de seguridad; y para proveer de bases, en los requerimientos de nuevas adquisiciones.

Para poder definir niveles de seguridad, primero se tiene que definir el concepto de seguridad; que según este documento: "Un sistema seguro controlará, a través del uso de caracteristicas de seguridad especifica, el acceso a la información de tal manera que sólo individuos debidamente autorizados, o un proceso corriendo por su cuenta, tengan acceso a la lectura, escritura, creación o borrado de información". De esta definición se derivan seis requerimientos fundamentales:

POLITICAS

Políticas de seguridad	Deben existir políticas de seguridad explícitas y ampliamente conocidas y eficientemente impuestas dentro del sistema. Para ello se deben definir sujetos y objetos, de tal forma que con las políticas establecidas se pueda establecer qué sujetos pueden accesar a qué objetos
Marcaje	Se deben asociar etiquetas de control de acceso a los objetos que define su nivel de seguridad y/o los modos de acceso, de acuerdo al sujeto que trate de accesarla.

CONTABILIDAD

Identificación	Cada sujeto se debe identificar para poder reconocer cuáles son sus derechos dentro del sistema. Esta información debe ser guardada y controlada eficientemente por cada uno de los elementos del sistema, para poder brindar el acceso correspondiente al sujeto.
Contabilidad	Se debe mantener un registro de las acciones realizadas por cada sujeto, para poder rastrear y localizar el sujeto que afecte a la seguridad. El sistema debe recordar cada evento relevante; así como la posibilidad de configurar y determinar qué elementos se desean rastrear. Toda esta información debe ser protegida.

[56] National Computer Security Center.

GARANTÍAS

Garantía	El sistema debe contar con mecanismos de hardware y software para proveer la suficiente garantía de que el sistema cumple con los cuatro requerimientos anteriores. Para garantizar esto, el sistema debe contener los mecanismos para implementar las definiciones anteriores. Generalmente suelen estar incorporadas dentro del sistema operativo; por lo que toda esta información debe estar claramente documentada.
Protección continua	El mecanismo confiable que mantiene los requerimientos básicos de seguridad antes descritos, debe ser protegido continuamente contra ataques, trampas y/o cambios no autorizados.

En el establecimiento de este esquema de medición de seguridad, se establecieron cuatro divisiones que van de la A (máxima seguridad) hasta la D (mínima seguridad). Cada una de estas divisiones cuenta con varias clases, que son niveles intermedios. Por otro lado, para definir cada una de estas divisiones y clases, se utilizan los requerimientos expuestos anteriormente; y dentro de la definición de cada una de ellas en este documento, se exponen detalladamente cada uno de los requisitos que deben cumplir. Para fines de esta tesis, únicamente se dará un resumen de qué debe cumplir cada uno de estos niveles; por lo que si se requiere mayores detalles, se puede recurrir al documento original, que es de dominio público y puede ser bajado a través de la red. Se encuentra dentro del directorio */pub/info* del equipo *info.cert.org*. Para obtener una copia se puede utilizar el comando *ftp* y entrar con la cuenta de *anonymous*.

IV.VI.I. Criterio de evaluación

Como se mencionó, este criterio consta de 4 divisiones que van de la D a la A, incrementándose el nivel de seguridad en cada una de ellas. Las divisiones B y C están divididas en varias clases representando distintos niveles de seguridad dentro de cada una de ellas como se describe a continuación.

IV.VI.I.I. División D: Protección Mínima.

Esta división contiene únicamente una clase. Es utilizada para los sistemas que han sido evaluados y que no cumplen con cualquiera de las clases siguientes. Sistemas dentro de esta clase no proveen mecanismos de protección segura o son mínimos. Un ejemplo de ellos, es el sistema operativo MS-DOS, utilizado en equipos personales.

IV.VI.I.II. División C: Protección Discrecional

Esta división cuenta con dos clases que proveen una protección discrecional.

CLASE C1: Protección De Seguridad Discrecional

Estos sistemas generalmente cuentan con un método de control de acceso discrecional que les permite separar e identificar los usuarios y los datos. Incorporan mecanismos para identificarlos y establecer las limitaciones en el acceso a los datos. Este tipo de sistemas son considerados como medio ambientes de trabajo de cooperación. Ejemplos de éstos, suelen ser los sistemas UNIX típicos; los cuales cuentan con mecanismos para identificar y autentificar a los usuarios y restringir el acceso a la información basados en los permisos de lectura, escritura y ejecución que son dados a cada uno de los elementos almacenados dentro de ellos.

CLASE C2: Protección De Acceso Controlado.

Los sistemas de esta clase cuentan con mecanismos de control más refinados que los de la clase C1; implementando mecanismos que permitan llevar la contabilidad individual de las acciones de cada usuario, permitiendo establecer los elementos sobre los cuales se llevará un registro de sus eventos. Ejemplo de estos sistemas puede ser el sistema operativo IRIX 5.3 tratado en esta tesis; ya que cuenta con herramientas de contabilidad y verificación.

IV.VI.I.III. División B: Protección Obligatoria

Esta división cuenta con tres clases. En esta división resalta el uso de etiquetas asignadas a los objetos junto con una serie de reglas de control de acceso obligatorio que determina el acceso de las personas sobre los objetos.

CLASE B1: Protección De Seguridad Mediante Etiquetas

Como requerimiento primordial de esta clase, se encuentran todas las características de la clase C2. Además, debe contener un modelo de políticas de seguridad **informal**, un mecanismo para etiquetar la información, y un control de acceso obligatorio que se aplica a sujetos y objetos dentro del sistema. Básicamente, en esta clase se pueden definir niveles de seguridad adicionales a los empleados en las clases anteriores (lectura, escritura, ejecución) llamados etiquetas, que restringen el acceso a los recursos en base a los privilegios dados a los usuarios. La versión del sistema operativo Trusted IRIX/B cae dentro de esta división.

CLASE B2: Protección Estructurada.

Los modelos de seguridad **formales** dentro de esta clase, deben ser claramente definidos y documentados requiriéndose además, que los mecanismos de control de acceso discrecional y obligatorios de las clases anteriores sean extendidos a todos los sujetos y objetos del sistema. En adición, la posibilidad de utilizar canales secretos de comunicación deben existir. Debe ser estructurado adecuadamente para proteger elementos críticos y los no críticos. Este tipo de sistemas debe ser relativamente resistente a penetraciones.

CLASE B3: Dominios De Seguridad

Los mecanismos de seguridad implementados dentro de esta clase, deben satisfacer los requerimientos de monitoreo para mediar los accesos de los sujetos a los objetos, de tal forma que puedan ser resistentes a trampas y engaños; deben ser lo suficientemente pequeños para que pueda ser analizado y checado. Para llegar a este fin, se debe excluir todo el código innecesario para mantener la seguridad minimizando su complejidad. Puede soportar un administrador de seguridad, mecanismos de verificación y rastreo son expandidos en su seguridad, y debe contar con mecanismos de recuperación. Este tipo de sistemas es altamente resistente a penetraciones.

IV.VI.I.IV. División A: Protección Verificada.

Esta división se caracteriza por el uso de métodos de verificación de seguridad formales que aseguren que los métodos de control de seguridad discrecional y obligatoria puedan proteger eficientemente la información sensible o clasificada almacenada en el sistema. Para ello se requiere una documentación extensiva de todos los aspectos de su diseño, desarrollo e implementación.

CLASE A1: Diseño Verificado.

Los sistemas en esta clase, son funcionalmente equivalentes a los de la clase B3 en el aspecto de que no se requiere añadir ninguna política o función adicional. El aspecto que marca la diferencia en los sistemas dentro de esta clase reside en las especificaciones de diseño formales y las técnicas de verificación que se requieren para el diseño de éstas, resultando en un alto grado de confiabilidad que los mecanismos descritos en los niveles anteriores, están correctamente implementados.

IV.VII. Consideraciones finales

El mantener un sistema seguro es una tarea ardua y que merece atención continua, capacitarse, y estar al pendiente de nueva información que vaya surgiendo referente a posibles fallas de seguridad descubiertas continuamente. Para facilitar esta labor, es conveniente que los administradores de equipo estén en contacto mutuo a fin de poderse comunicar experiencias y plantear soluciones. A este respecto, una buena opción son las listas de correo que funcionan en distintos servidores.

Un servidor de listas permite que usuarios en diferentes equipos puedan enviarse mensajes de correo automáticamente; es decir, un usuario envía un mensaje a la lista y le llega a todos los usuarios que están subscriptos a ella. En este sentido, los servidores pueden tener listas para diferentes temas, y los miembros de esa lista platican, comparten información y experiencias relacionados con el tema de la lista.

Existe un número grande de servidores de listas y cada uno, con diversas listas relacionadas con diferentes temas. De éstos, mencionaré cuatro, por su importancia y ser medios para estar en contacto con posibles problemas de seguridad.

El primero es la lista mantenida por el CERT/CC, cuya dirección a la que se pueden enviar mensajes; si es que uno ha descubierto y comprobado una deficiencia en la seguridad de cualquier tipo de software es:

Email[57]: *cert(a)cert.org*

Teléfono: *+1 412-268-7090 (Las 24 hrs.)*

Fax *+1 412-268-6989*

Dirección Postal
CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
USA

Este organismo cuenta con distintos medios de distribución de información relacionada con la seguridad entre los que se encuentran:

[57] Es la dirección de correo electrónico (buzón) a la cual se puede enviar un correo a través de una red.

Servicios de WWW[58] y *ftp* anónimo en la red Internet

http://www.cert.org/
ftp: info.cert.org/pub/

Distribución de boletines, avisos y consejos a través de la red USENET newsgroup:

comp.security.announce

Lista de correos. Esta es una lista a la cual, las personas que se subscriben, les son enviados automáticamente los boletines, avisos y consejos que surjan relacionados con la seguridad. Para ser añadido a esta lista, se tiene que enviar un correo electrónico a:

cert-advisory-request@cert.org

e incluir en el cuerpo del mensaje:

SUBSCRIBE your-email-address

Cabe aclarar que la cuenta (email) que se utilice para enviar este mensaje, es la que quedará registrada en la lista y es a la cual, le serán enviados automáticamente los boletines y avisos que surjan.

En segundo lugar tenemos que, recientemente (abril de 1997) ha surgido el anuncio de la creación de una nueva lista de discusión sobre seguridad informática en español, auspiciada por CORE Seguridad de la Información S.A. y Secure Networks Inc. (SNI). En ella se fomentará la publicación de reportes de problemas de seguridad que incluyan tanto código demostrativo (exploits) como código para solucionar el problema en cuestión (patches). Como propósito de esta lista se publicaron los siguientes:

- Facilitar la generación de una conciencia, entre los responsables de los sistemas de información, sobre la seguridad informática en los países de habla hispana.
- Proveer un medio de comunicación e información sobre seguridad de los sistemas de información para la comunidad hispano-parlante de Internet.
- Proporcionar información en español sobre temas discutidos en otras listas, publicados generalmente en inglés.

Como ellos proponen, esta lista busca la creación de un foro de discusión abierto sobre los temas propuestos, haciendo hincapié en la objetividad y el respeto al derecho de libre expresión. Los temas de discusión son los siguientes:

[58] Word Wide Web.

- Bugs[59] en diversos sistemas operativos.
- Bugs en diversos programas (comerciales o gratuitos) de uso común.
- Firewalls.
- Consideraciones de seguridad en diversos protocolos de comunicación.
- Seguridad, auditoría y control de redes.
- Sistemas de detección de intrusiones (IDS, Intrusion Detection Systems)
- Criptografía.
- Aspectos teóricos de la seguridad de la información, estrategias de seguridad, análisis de riesgos, respuesta frente a problemas de seguridad.
- Noticias e información general sobre la seguridad de la información en países hispano parlantes.

Además se publicarán regularmente avisos, propios y generados por otras instituciones, sobre vulnerabilidades encontradas en los diversos sistemas y programas de cómputo. Para subscribirse a esta lista, se tiene que enviar un correo electrónico a la dirección:

majordomo(@)secnet.com

Incluyendo en el cuerpo del mensaje lo siguiente:

subscribe seg-l

Si se desea participar en las pláticas sostenidas por este medio, se debe enviar el mensaje a la dirección:

seg-l(@)secnet.com

Para que sea publicado y lo reciban todos los miembros de esta lista.

En tercer lugar, en la UNAM se cuenta con un servidor de listas dentro de las cuales destaca la de GASU, (Grupo de Administración y Seguridad en UNIX). El único requisito para poder pertenecer a ella, es estar a cargo de alguna máquina con sistema operativo UNIX. Su propósito es:

- Servir de medio de comunicación entre los administradores de sistemas UNIX de la UNAM, otras instituciones educativas, e incluso instituciones privadas.

[59] Un bug es un error encontrado en el código de un programa que causa una deficiencia, problema o error en su ejecución.

- Organizar distintos eventos (seminarios, cursos, pláticas, etc.) tendientes a mejorar los conocimientos de los administradores sobre distintos temas relacionados con la administración de UNIX.
- Discutir aspectos de seguridad en UNIX, tanto en general como aplicables a ciertas versiones de UNIX en particular.

Dentro de los temas que se discuten en esta lista, se encuentra el relacionado con cualquier problema de configuración u administración de un equipo UNIX; por lo que puede resultar útil el subscribirse a ésta, para obtener ayuda de expertos, y no tan expertos, en el tema de la administración cuando se requiera. Para subscribirse, se tiene que enviar un mensaje de correo electrónico a la dirección:

listproc@listas.unam.mx

E incluir en el cuerpo del mensaje:

*subscribe gasu **NOMBRE***

donde ***NOMBRE*** es el nombre real del usuario. Al igual que en todas las listas, quedará registrada la dirección electrónica desde donde se envíe este mensaje, y ahí es donde se dirigirán todos los correos publicados en la lista.

Cuando se desee publicar un mensaje en la lista, para que todos los miembros lo reciban, éste debe ser enviado a:

gasu@lista.unam.mx

Esta lista cuenta con un servicio de ftp anónimo (*ftp.super.unam.mx*) al cual puede ingresar cualquier persona. En él se puede localizar información referente a seguridad en el directorio *pub/security*, como: Documentos diversos relacionados con seguridad, herramientas para ayudar a monitorear la seguridad en un sistema, documentos y herramientas de criptografía, etc.

Finalmente, SGI[60] mantiene un servicio de listas de correo de seguridad[61], en la cual emite continuamente boletines de consejos, manifestando problemas en la seguridad encontrados en versiones de sus productos ya liberados. Para subscribirse, enviar un mensaje a:

external-majordomo@postofc.corp.sgi.com

[60] Silicon Graphics Inc.
[61] Consultar el Apéndice A, por el listado de los consejos emitidos en este último año por esta compañía.

y colocar en el cuerpo del mensaje:

subscribe wiretap ***DIR-EMAIL***

Donde ***DIR-EMAIL*** es la dirección electrónica en la cual se desean recibir los boletines generados. Si se desea reportar una falla detectada, se debe enviar un correo a:

security-alert@sgi.com

Si se desea obtener una copia de los avisos emitidos con anterioridad, así como diversos patches y otro tipo de software público de esta compañía, se puede realizar una conexión a través de *ftp* a la dirección *sgigate.sgi.com* o a *ftp.sgi.com*, que es un espejo del anterior.

SGI también mantiene diversas páginas de WEB, entre las que destaca la de seguridad, cuya página principal se encuentra en:

http://www.sgi.com/Support/Secur/security.html

Y finalmente, si se tiene dudas o preguntas relacionadas con algún documento emitido por ellos, se puede enviar un mensaje a:

cse-security-alert@csd.sgi.com.

Así como estos ejemplos; existe gran información que puede ser consultada en Internet y que todo administrador debe consultar (a su criterio) para estar actualizado; ya que la capacitación en este tema es dinámica y continuamente cambia, se descubren problemas, soluciones, nuevas aplicaciones, etc. y es conveniente mantenerse actualizado y no quedarse atrás, en este mundo inconstante de la computación y seguridad.

CAPÍTULO 5

MANTENIMIENTO

Razones, consejos y labores que permiten mantener funcionando y en óptimas condiciones un equipo; así como las medidas de recuperación a seguir, en caso de falla.

V. MANTENIMIENTO

Antes de poder ser utilizado, a un equipo se le debe instalar, configurar y poner en punto el SO. El terminar y llevar a cabo adecuadamente esta labor, no significa que todo concluyó y que la responsabilidad del administrador ha finalizado; antes bien, esto sólo es el inicio y empieza realmente su principal compromiso. El mantener seguro tanto el equipo como la información almacenada en él; el mantenerlo en su máximo rendimiento; el prevenir y solucionar los posibles problemas que se presenten; el contar como las medidas preventivas para solucionar cualquier mal funcionamiento; el monitorear constantemente la actividad, son algunas de las labores que debe realizar diariamente para garantizar la confiabilidad de las operaciones realizadas en el equipo.

Para poder efectuar estas labores, en este capítulo se presentarán los fundamentos, y posteriormente, los pasos y labores que deben realizarse.

V.I. Bitácora y libro de procedimientos

La bitácora y libro de procedimientos son una de las principales herramientas con que cuenta un administrador. Durante la vida útil de un equipo, pueden haber existido diversos administradores que lleven el control del equipo, y muy frecuentemente, el único vínculo entre ellos, que les permite seguir manteniendo el equipo en buen estado sin repetir esfuerzos y minimizando los tiempos de adaptación, suelen ser estos documentos.

El libro de procedimientos debe contener entre otras cosas:

- Un registro de los procedimientos más importantes, y sobre todo, aquéllos que lleven una secuencia especial o fuera de lo normal, por ejemplo: la creación o remoción de usuarios; la secuencia, comandos y archivos que se deben respaldar, etc..
- Un listado de las características del equipo: Tipo de video, CPU, memoria RAM y otros; número y dirección de cada dispositivo interno, en especial los SCSI; dispositivos accesados remotamente, etc. Un listado inicial puede ser obtenido mediante los comandos *hinv* y *gfxinfo*.
- Un registro del software instalado que incluya, por cada uno entre otras cosas: la fuente para obtenerlo, el procedimiento de instalación, los problemas encontrados así como su solución, las claves y licencias utilizadas, etc. Un listado general puede ser obtenido mediante el comando *versions*; éste despliega un listado del software instalado con el comando *inst* únicamente.

- Un registro de los parámetros de configuración del sistema, entre los cuales se deben encontrar: nombre, dominio y dirección IP del equipo; gateways y máscaras definidas para su red; un listado de la configuración actual del sistema obtenido mediante el comando *chkconfig*, etc..
- Los cambios más significativos hechos a la configuración básica del sistema, por ejemplo: los parámetros y valores establecidos durante la afinación del sistema, para lograr su máximo rendimiento; modificaciones hechas a archivos del sistema para adecuarlos a las necesidades propias, etc..
- Un listado de los archivos más significativos; como el de */etc/group, /etc/passwd, /etc/fstab, /etc/exports, /etc/hosts*.
- Un listado de los archivos con licencias que permitan el funcionamiento de software específico y protegido.
- Un listado de la estructura, características y tamaños de cada uno de los discos, así como de sus particiones. Se puede obtener uno mediante el comando *prtvtoc*.
- Un listado de cada una de las claves secretas de administración requeridas para la operación del sistema, por ejemplo: la clave secreta de root, y cuentas especiales; claves generadas para aplicaciones en especial; claves utilizadas para accesar ciertos recursos a través de la red; claves especiales requeridas por usuarios durante la operación del sistema, etc..

Un punto importante a considerar, es el si se debe documentar cada una de las claves secretas, y en especial la de root. La decisión será establecida por las políticas impuestas en la compañía; aunque cabe analizar los pros y los contras.

El documentar físicamente las claves secretas, puede ser un punto vulnerable; por lo que debe llevarse un estricto control sobre quién puede tener acceso a este registro y qué medidas se deben tomar para protegerlo. Por el contrario, el no plasmarlas físicamente, puede generar mayor seguridad; aunque se pueden producir otra clase de riesgos que se deben considerar. Uno de ellos, es que la memoria de una persona puede fallar, y en un momento dado, no recordar alguna clave; sobre todo aquéllas que rara vez se utilizan. Para solucionar esto, algunos administradores suelen establecer la misma clave para todos los casos donde se requiera una; lo cual es una falla en la seguridad, porque al descubrirse una, se tiene acceso a todo el sistema. Por otro lado, si sólo el administrador las conoce, la dependencia a esta persona suele incrementarse y si por alguna causa falta o es despedido, se enfrentará a un gran problema el administrador que lo suplante; ya que al no conocer ninguna de ellas, se verá limitado, y en ocasiones, no se podrá realizar ninguna otra acción hasta no reinstalar el sistema o software específico que la utilice, con la consecuente pérdida de tiempo y riesgo en la pérdida de información.

Por esto y otras razones, éste es un documento indispensable que no debe dejar de llevarse, así como la bitácora del sistema. La bitácora es un registro cotidiano que se debe llevar

sobre la actividad y problemas que se vayan presentando diariamente en el sistema. En él se debe anotar por cada entrada:

- La fecha y hora en que ocurrió el incidente.
- Una descripción del problema o listado del mensaje de error que se haya generado.
- Una breve explicación del problema así como de las medidas tomadas para solucionarlo.
- Referencias hacia documentos que informen, tanto del problema como la solución.

Generalmente, se debe registrar todo evento fuera de lo normal que ocurra, pero esta decisión queda a criterio del administrador. La bitácora permitirá llevar un historial sobre el comportamiento del sistema y, en un momento dado, el anticipar o prevenir posibles conflictos; reduciendo tiempos, ya que se puede contar con una guía de problemas y soluciones.

Como punto adicional, cabe mencionar que el sistema cuenta con una herramienta llamada *syslog* que permite llevar una especie de bitácora electrónica automáticamente. Mediante él, cada evento previamente definido, puede quedar registrado en un archivo, además de ser desplegado en la consola. Es importante que el administrador revise y dé mantenimiento a éste y otros archivos de mensajes, ya que suelen crecer generalmente sin medida, y permiten localizar posibles fallas o problemas antes que lleguen a ser críticos. Esta herramienta al igual que otras, serán explicadas más adelante.

Finalmente, muchos administradores suelen llevar estos documentos en archivos dentro del sistema, debido a la facilidad para copiar y editar. Pero no debe olvidarse el plasmarlos físicamente en papel mediante su impresión; ya que el sistema por cualquier causa, puede quedar inoperable, y en ese momento, toda la información que puede ayudarnos a solucionar el problema más rápidamente, quedará inaccesible.

V.II. Monitoreo del sistema

Como ya se ha mencionado, el administrador debe mantener una vigilancia cotidiana sobre el sistema, para detectar y poder anticipar y solucionar cualquier posible problema. Es conveniente llevar una secuencia ordenada para no dejar pasar ningún aspecto; a este respecto, es conveniente enlistar estas actividades y llevar un registro de ellas. A continuación se mencionarán algunas de las actividades, simples y sencillas, pero sobre las que se debe llevar un control.

V.II.I. Espacio en disco

El espacio en disco es un recurso muy valioso y vital que se debe monitorear continuamente. Éste es utilizado por el sistema para un gran número de funciones y, si por alguna causa llega a ser demasiado pequeño o incluso terminarse, el sistema puede presentar fallas y colapsarse (caerse). Si esto llega a suceder, cabe la posibilidad de que se pueda dañar parte o toda la información almacenada en el disco. Además, cuando esto sucede el sistema no puede arrancar normalmente, ya que no dispone de espacio en disco, y entones se tienen que realizar una serie de rutinas y procedimientos para levantarlo, liberar espacio, solucionar los problemas causados y restaurar la operación normal del equipo; esto si es que los daños causados no son mayores. Por esto y otras razones, el monitorear el espacio disponible es importante; por ello, primero se expondrán una serie de rutinas que mostrarán los puntos más vulnerables, permitiendo determinar las causa de posibles incrementos en la demanda y evitando así, que se sature el disco. Posteriormente se indicarán las medida para solucionar cualquier posible caida.

Son tres los comandos básicos para monitorear la cantidad de espacio disponible en el disco: *du*, *df* y *find*. El primero *du*, muestra un resumen de la cantidad de bloques de 512 bytes que ocupan los archivos y subdirectorios, del directorio que se especifique. La segunda *df*, muestra un resumen de la cantidad de bloques asignados a cada partición, el porcentaje ocupado y el libre; y *find* realiza búsquedas de archivos en el disco.

Al combinar estos comandos, las labores que se pueden realizar son variadas y útiles; si a esto agregamos la posibilidad de programarlas para que se ejecuten a determinados intervalos, como se verá más adelante, el resultado es una gran herramienta para el administrador. A continuación se describen una serie de labores importantes, y que sirven como muestra, implementadas con estos comandos.

- Durante el funcionamiento del equipo, existen aplicaciones que pueden fallar generando archivos que son utilizados por ellas mismas para tratar de restablecer su situación y que posteriormente sólo ocupan espacio en disco. Estos archivos pueden llegar a ser una gran cantidad, e incluso, ser de gran tamaño; se pueden encontrar dispersos por todo el SA. Un ejemplo de ellos son los archivos cuyo nombre es *core* y que genera el sistema cuando alguna aplicación falla. Estos contienen una imagen de la aplicación en memoria RAM al momento de ocurrir la falla. Si desea, el programador o persona responsable puede utilizar el programa depurador *dbx* para revisar este archivo y localizar la falla que originó el conflicto.

- El siguiente comando genera una lista que es enviada mediante correo a la cuenta de root, indicando la ubicación de cada archivo *core* en el sistema.

  # find / -name core -print | mail root &

- También se puede utilizar el siguiente comando para localizar archivos que excedan de cierto tamaño; por ejemplo, más de 5 MB:

  # find / -size +10000 -print | mail root &

- O para localizar archivos inactivos que tengan más de 2 meses de haberse creado:

  # find / -type f -mtime +60 -print | mail root &

- Si se desea saber el espacio en KB ocupado por los archivos de un usuario en su directorio de trabajo, se puede emplear el comando:

  # cd /usr/people/usuario1
  # du -ks

- Si se desea conocer el porcentaje de espacio disponible en cada una de las particiones disponibles del disco, utilizar el comando:

  # df -k

  Si el porcentaje ocupado llega a ser de 90 a 95%, el rendimiento del equipo puede decaer; por lo que será conveniente tratar de liberar espacio.

- Para determinar el número de inodos libres, utilizar el siguiente comando. Recordar que si se terminan los inodos, no se podrá crear ningún archivo más; aunque exista espacio en disco.

  #df -i

Es conveniente monitorear también, los archivos de reportes y errores que utiliza tanto el sistema operativo como cualquier aplicación, para en caso de que su tamaño sea demasiado grande, sean limpiados y evitar que saturen el sistema. Dependiendo de la carga de trabajo y demanda en la utilización del equipo, su crecimiento puede ser rápido o lento. Entre éstos es recomendable vigilar:

/var/cron/log	Mensajes de error y registro de las acciones tomadas por *cron*.
/var/adm/SYSLOG	Mensajes de errores de los demonios en ejecución y del sistema.
/var/adm/sulog	Bitácora del comando *su*.
/etc/wtmp	Contiene un historial de los intentos y accesos al sistema.

Por otra parte, se recomienda la vigilancia sobre los directorios siguientes; ya que por su utilización, pueden saturarse rápidamente. Generalmente en ellos se encuentran archivos temporales y basura:

/var/adm/crash	Se localizan archivos *core* generados por el comando *savecore* cuando el sistema falla. Estos archivos pueden ocupar un espacio considerable (MB), dependiendo de la memoria RAM de que se disponga y de la cantidad de carga de trabajo; ya que son un copia de la información contenida en RAM al momento de presentarse la falla.
/tmp	Directorio público que generalmente contiene archivos temporales.
/usr/tmp	Liga a */var/tmp*.
/var/tmp	Directorio público que generalmente contiene archivos temporales.
/lost+found	Contiene archivos perdidos y recuperados con el comando */sck*.

Finalmente, al igual que las razones anteriores, la demanda de espacio en disco puede incrementarse por malos hábitos de los usuarios, pero éstos serán cubiertos en el siguiente punto.

V.II.II. Usuarios

El monitorear a los usuarios es una gran fuente de información para el administrador. A través de esta actividad se pueden determinar fallas en la seguridad, problemas en el funcionamiento del equipo, tendencias de los usuarios y con esta información, poder implantar nuevas políticas de uso, determinar el rumbo a seguir en la capacitación que se les dará, etc. todo ello encaminado a mejorar y perfeccionar el desempeño, tanto del equipo como de los usuarios, que redunda en beneficios económicos a la empresa..

Los comandos empleados para estas labores suelen ser: el comando *who* que muestra qué usuarios están en el sistema y desde dónde se han conectado al equipo, permitiendo determinar posibles intrusos conectados remotamente o de lugares fuera de lo común; *w* que muestra qué usuarios se encuentran en sesión y qué es lo que están haciendo; *whodo* que indica qué está haciendo cada usuario; *finger* que muestra información valiosa de los usuarios y *last* que nos permite determinar cuál ha sido su actividad pasada. Los comandos *w*, *who* y *whodo* son similares, pero se debe conocer y distinguir sus pequeñas diferencias.

Además de éstos, cada uno de los sistemas suelen tener herramientas específicas de monitoreo; como la de *sar* y *audit* que ya han sido mencionadas en capitulos anteriores, y la de *acount*, que permite contabilizar la actividad del usuario, llevando un registro de los

comandos que utiliza, el tiempo de conexión, uso de CPU y otros aspectos. Esta información suele ser utilizada para generar facturas de cobro por uso del equipo.

A continuación se indicarán varios puntos que deben ser tomados en cuenta cuando se monitoree la actividad de los usuarios.

Hábitos de trabajo.

El monitorear a los usuarios puede arrojar indicios de cuáles son sus tendencias en varios aspectos. En la forma de trabajo se puede determinar malas costumbres que causen daños, tanto al equipo como a su funcionamiento. Esta vigilancia debe ser llevada a cabo tanto al uso físico que se le da, como dentro del sistema. Físicamente mediante rondas, encuestas o pláticas sostenidas con ellos, y dentro del equipo por medio de las diversas herramientas y comandos descritos tanto en este capítulo, como en los anteriores: *w*, *whodo*, *finger*, etc.. Algunos de los aspectos detectados y que no deben pasarse por alto son:

- Los usuarios tienden a almacenar una gran cantidad de archivos; aunque ya no les sean útiles. Por ello, es conveniente mantener una comunicación constante con ellos y concientizarlos de esta situación, fomentando la costumbre de realizar limpieza contante sobre sus directorios, respaldando y eliminando la información histórica y que generalmente no se ocupa, así como aquélla que rara vez se procesa, a fin de mantener el SA limpio y evitar su saturación. Esto dará como resultado un mejor rendimiento del equipo. Para este monitoreo se puede utilizar el comando *du* sobre los directorios de los usuarios, o activar el sistema de *quotas* para limitar el espacio que puede utilizar cada uno.
- De observaciones cotidianas, se ha detectado en algunos usuarios un desconocimiento de cuál es el funcionamiento real de las herramientas gráficas para borrar archivos; por ello, se les debe dar a conocer que al utilizarlas, sólo se les cambia de nombre y son colocados en un directorio especial, los documentos eliminados. Esto significa que no son borrados del disco y siguen ocupando espacio. Para eliminarlos realmente se les tiene que indicar a estas mismas herramientas mediante otro proceso, que deben ser borrados en forma definitiva; conocido generalmente como "vaciar la papelera". El desconocimiento de esto influye en la saturación del disco; por lo que se debe detectar cuál es el directorio donde son colocados por default y determinar su tamaño.

  En el SO IRIX, la papelera es el directorio *dumpster* dentro del directorio *HOME* de cada usuario, y ya que los directorios de los usuarios suelen estar dentro de */usr/people*, la combinación del siguiente comando *find* y *du*, entrega un resumen de la cantidad de KB que ocupa la basura de cada uno de los usuarios del sistema:

  # find /usr/peolple -name dumpster -exec du -sk '{}' ';'

- En lo tocante al aspecto fisico del equipo, algunos usuarios no suelen conocer cuál es el correcto funcionamiento, la forma de operarlos o el trato que se les debe dar a los equipos de cómputo. Ello conlleva a reducir su tiempo de vida así como la presencia de fallas continuamente, redundando en pérdida de tiempo y capital. Por ello, se debe vigilar este aspecto, y si se detectan anomalías, instruirlos en las labores adecuadas. Esto se aplica tanto al CPU como a cualquier periférico; como las impresoras.

Aplicaciones con gran demanda

El determinar qué aplicaciones tiene mayor demanda puede ser un aspecto importante en la capacitación que se les dará a los usuarios; ya que el correcto uso de ellas, requerirá de un menor tiempo de proceso, y por tanto, se optimizan los recursos.

Por otra parte, los requerimientos de RAM, disco, CPU y áreas de swap de las aplicaciones varía, y el determinar cuáles son las de mayor uso puede dar buenos indicios en cuál es el aspecto que se debe cuidar; si incrementar el área de swap, otro disco duro, más memoria RAM, etc. a fin de mejorar el rendimiento. Por otra parte, aplicaciones propias pueden ser perfeccionadas a fin de mejorar su rendimiento y de este análisis, se puede determinar a cuales poner mayor atención y en qué aspectos.

Al monitorear la actividad de los usuarios visualmente, se puede determinar cuáles son las aplicaciones utilizadas frecuentemente, o se puede obtener una estadística con herramientas como *audit,* o las aplicaciones en uso con comandos como *top*, *ps* y otros que serán descritos más adelante.

Horas pico

Las horas pico pueden ser un verdadero conflicto para un administrador; en ellas la demanda de servicios se incrementa, y generalmente llega a saturar el sistema. Esto ocasiona que el tiempo para procesar alguna tarea se extienda, contribuyendo a extender el período de carga, alentar el sistema, disminuir su rendimiento, y muy frecuentemente producir la caída del sistema. Todo esto contribuye al malestar de los usuarios y demorar otras labores ajenas al equipo, pero que dependen de los resultados obtenidos de estos procesos, generando finalmente, pérdidas monetarias.

Por todo lo anterior, una labor adicional de administración que redunda en grandes beneficios, es el tratar de evitar la existencia de horas pico; cuando sea posible. En el caso de que esto no pueda ser, lo recomendable es realizar un correcto análisis de las necesidades de los usuarios para en base a ello, acondicionar el equipo para soportar de la mejor manera, esa carga de trabajo; así como el capacitarlos para que realicen sus labores sin errores. Este estudio se debe basar en la demanda de las aplicaciones y en un análisis de los resultados obtenidos del constante monitorear del sistema en general.

Dependiendo del análisis de las tareas de mayor demanda, se puede educar a los usuarios que las utilizan para redistribuir la carga en diversos horarios y, si la aplicación y su prioridad lo permiten, programarlas para que se realicen en otros horarios. De esta forma, procesos que requieran de una gran demanda de CPU, pueden ser pospuestos para ser programados y ejecutados automáticamente de noche u horas de menos carga. El acceso de los usuarios puede ser distribuido y programado a lo largo del horario de trabajo, dependiendo de las prioridades de sus labores y la demanda de recursos. Se puede especializar, mediante capacitación, a los usuarios en el correcto manejo de las aplicaciones[62] que requieren para desempeñar sus labores, etc.

Seguridad

Como se indicó en el capítulo anterior, la seguridad es un aspecto muy importante, y la atención que el administrador le dé, redundará en grandes beneficios. Por ello, si se mantiene una estrecha vigilancia de los usuarios, se podrá determinar en forma más eficiente, situaciones fuera de los común que sugieran un posible intento de violación; ya sea de usuarios locales o externos. Para esto es conveniente tomar en cuenta las recomendaciones y aspectos mencionados en el capítulo anterior.

V.II.III. Procesos

Los procesos son otro recurso sobre el cual, el administrador debe poner atención. Además de que muestran las tendencias de los usuarios, un análisis correcto de ellos puede conducir a las labores de mantenimiento que den como resultado, la optimización del uso de CPU. Por el contrario, si no se contemplan, pueden existir conflictos en ellos que ocasionen una disminución sensible en el desempeño del equipo.

La herramienta fundamental para el análisis de procesos es el comando *ps*, que muestra una imagen de la tabla de procesos del kernel. Alternativamente, se puede utilizar el comando *top* o su análogo en ambiente gráfico *gr_top*, que muestran un listado e histograma de los procesos que consumen mayores recursos de CPU a intervalos específicos. La correcta interpretación y entendimiento de su significado es vital para el análisis tendiente a mejorar el uso de este recurso.

[62] Este aspecto lo he mencionado varias veces, ya que en la práctica me ha redundado en grandes beneficios: reduciendo la asistencia requerida por los usuarios en dudas, la cantidad de problemas dentro del sistema debido a programas con conflictos que reducen el rendimiento del equipo, etc.. Todo ello hasta en un 90 %; sobre todo en usuarios nuevos o con poca experiencia.

Los monitoreos a intervalos pueden mostrar procesos bloqueados o que se encuentran en ciclos iterativos consumiendo tiempo de CPU. Si al utilizar el comando *ps -ef* se localizan procesos que en la columna de *TIME* muestre un tiempo alto de ejecución, se debe investigar, ya que pueden ser procesos del sistema correctamente funcionando o procesos que se encuentran en ciclos infinitos consumiendo recursos únicamente; ya que esta columna indica el tiempo de ejecución acumulado del proceso. De igual forma, en la columna *STIME* se muestra la fecha en la que comenzó a ejecutarse el proceso; por lo que procesos con fechas antiguas deben ser investigados para determinar si no se encuentran bloqueados ocupando un lugar en la tabla de procesos del sistema.

La columna *S*, que aparece al ejecutar el comando *ps -el*, muestra el estado en que se encuentra cada uno de los procesos. Si se llegan a localizar procesos con una bandera *Z*, se tratan de zombis; éstos pueden surgir por problemas en los procesos background o por un mal uso del sistema por parte de usuarios al no cerrar correctamente las aplicaciones, o apagar el equipo indebidamente. Normalmente, el programa *init* se encarga de eliminarlos automáticamente; pero si por alguna causa éstos llegan a persistir, muy frecuentemente ni el comando *kill* los puede eliminar y el único método es re-iniciando el equipo. Por este motivo se debe investigar cuál es el programa o la causa que los crea, para erradicarla o darle solución. El único problema que causan estos programas, es el que ocupan espacio en la tabla del kernel; por lo que si son demasiados interfirieren con la creación de nuevos procesos, y si la causa que los genera persiste, pueden llegar a saturar las tablas del kernel. Por otro lado, la bandera que puede ser indicio de una deficiencia de memoria es la de X; la cual indica que un proceso está en espera de memoria para su proceso. Si el número de procesos con esta bandera se incrementa, se debe investigar la causa; posiblemente se requiera adquirir más memoria para el equipo o incrementar el área de swap del disco.

También se tiene que mantener una vigilancia sobre la cantidad de procesos en ejecución; ya que si llega a ser muy grande, pueden saturarse las entradas en la tabla de procesos del kernel, y en ese momento no se podrá ejecutar ningún otro más. La variable *nproc* del grupo *numproc (statically changeable)* que aparece al ejecutar el comando *systune* , muestra el número máximo de procesos permitidos en el equipo. Si esto llega a suceder con frecuencia, se tiene que reconfigurar el kernel y crear uno nuevo con una cantidad mayor de entradas (ver '**Reconfiguración del kernel**', pág. III-26); por el contrario, si la cantidad de procesos que normalmente o en horas pico se requieren es pequeña, se puede reconfigurar el kernel para disminuir el tamaño de la tabla de procesos y liberar memoria RAM para ser utilizada por las aplicaciones, redundando en un mejor desempeño del equipo. Esto mismo se aplica para las demás tablas, como la de archivos abiertos, que se mantienen en la RAM del equipo y que pueden ser configuradas al regenerar el kernel adecuadamente.

Si el rendimiento del equipo empieza a decaer, se puede utilizar el comando *top* para determinar cuál es, o son, los procesos que están consumiendo la mayor parte de los recursos del CPU; ocasionando el problema. Se deben de analizar para determinar si fueron mal programados o están bien diseñados; en cuyo caso, se puede pensar en posponer su

ejecución para horas de menos demanda; donde la carga que generan no afecte a los usuarios. También se puede pensar en la expansión de recursos, como RAM o Disco para solucionar el problema.

Finalmente, llegan a ocurrir situaciones en las cuales se empiezan a ejecutar copias continuas de un programa, que van creciendo hasta saturar la tabla del kernel y bloquear el sistema. Esto puede ocurrir por varias causas, pero una de ellas, es el mal uso o problemas en aplicaciones que utilizan el comando *init*. Cuando se instalan nuevos programas, y en especial cuando se añaden líneas al archivo */etc/inittab* con la bandera de *respawn*, se le dice a *init* que ejecute el proceso, y cuando éste finalice, lo vuelva a ejecutar. Se suele utilizar para activar terminales, de tal forma que aparezca el mensaje de *Login:* en la pantalla y un usuario pueda utilizarla. Al ejecutar el comando *exit* para salir de sesión, se termina el proceso e *init* lo detecta y lo vuelve a ejecutar, de tal forma que aparece nuevamente el mensaje en la pantalla. Si se instalan aplicaciones que utilicen este método, se puede presentar algún error y ocasionar que *init* esté ejecutando continuamente copias del mismo programa sin que haya terminado el anterior.

Por esto y otras razones vale la pena dedicar tiempo para el monitoreo de los procesos; así como la comprensión del correcto significado e interpretación de las columnas impresas por el comando *ps -ef* al igual que todas sus demás opciones.

V.II.IV. Rendimiento

En los puntos anteriores se han dado sugerencias que pueden conducir a optimizar el uso del equipo y mejorar su rendimiento, que es un aspecto que todo administrador busca. Por eso se debe estar al pendiente, y si se llegan a tener síntomas que muestre un decremento de la eficiencia del equipo, se deben checar los aspectos mencionados anteriormente; especialmente el de los procesos y la cantidad de memoria virtual disponible, para determinar dónde se está consumiendo la mayor parte del tiempo del CPU. Para ello se pueden utilizar las herramientas examinadas, como la de *w -u*, que muestra un promedio de los procesos en ejecución hace 1, 5 y 15 minutos; la de *top* o *gr_top* para ver qué procesos son los que tienen mayor demanda de CPU; la de *osview* o su análoga gráfica *gr_osview* que nos muestran la cantidad de demanda que tiene la RAM, CPU y otros recursos; o bien, el comando *df /proc* para obtener el estimando de uso de la memoria virtual; este último merece un descripción más detallada.

Si en el listado obtenido del comando *df /proc* se muestra un porcentaje alto de uso (*%use*), se indica que la gran mayoría de la memoria virtual se está usando. La cantidad total de bloques (*blocks*) es igual a la suma de la memoria RAM y swap de que dispone el equipo. El directorio */proc* es un sistema de archivos virtual que provee acceso a cada proceso activo en el sistema; históricamente se conocía como */debug*. Cuando el sistema arranca el

directorio */proc* es montado automáticamente, por lo que si no existe, debe ser creado. Para ello, ejecutar los comandos siguientes desde root:

```
# mkdir /proc
# mount -t proc /proc /proc
```

En este punto, cabe recordar que se recomienda habitualmente contar con una memoria swap del doble del tamaño de la RAM; es decir, que si se cuenta con 32 MB de RAM se recomienda crear un área de swap de por lo menos 64 MB. Pero, debido a las condiciones y demandas de recursos del equipo, ésta puede ser insuficiente y entonces, se tendrá que pensar en expanderla; ya sea adquiriendo más RAM, añadiendo o reparticionando el disco duro para asignar más área a la partición 1, que corresponde swap para solucionar el problema. Es importante destacar que el SA */proc* no consume espacio físico del disco; ya que es virtual.

Como última recomendación, si la carga de trabajo cotidiana es demasiada; si no puede ser expandido el equipo físicamente y si los recursos con que cuenta están optimizados al máximo, la siguiente alternativa es la de educar a los usuarios a que depuren y afinen la aplicaciones que creen, a fin de no consumir muchos recursos; el instruirlos en ejecutar sus procesos en forma lineal y no paralelamente, es decir, que ejecuten uno y al terminar otro; el que programen sus aplicaciones, que así lo permitan, mediante el comando *at* y *batch* para que sean ejecutadas durante la noche o a las horas de menor demanda; o el utilizar el comando *nice* para bajar la prioridad a los procesos menos importantes y así, se agilicen los más vitales.

V.III. Automatización de labores

Durante el proceso de mantenimiento, existen tareas que se deben realizar y que no requieren de la presencia del administrador; labores como la del monitoreo del tamaño de los archivos de errores, que por su rápido crecimiento pueden llegar a saturar el espacio libre del disco; la eliminación de archivos temporales; el monitoreo de qué usuarios están en sesión a determinados momentos; el monitoreo de la carga de trabajo a ciertos intervalos o hasta la realización de respaldos. Así como éstas, existen otras labores que pueden ser automatizadas mediante el uso del sistema de *cron*, para que se realice su tarea, en determinados momentos (cuando la carga del sistema sea mínima) y a determinados intervalos.

El *cron* permite programar una serie de comandos para que sean ejecutados a cierta hora; aunque el usuario que los programó no se encuentre en sesión en ese instante. Cabe destacar que inicialmente, root es el único al que se le da permiso para utilizar este sistema. Se

pueden utilizar los archivos */etc/cron.d/cron.allow* para definir a qué usuarios, además de root, se les ha de permitir la posibilidad de utilizarlo; por el contrario, */etc/cron.d/cron.deny* contiene un listado de los usuarios a los que se les negará el permiso. Estos dos archivos son mutuamente excluyentes, y por lo tanto, sólo puede existir uno. La decisión de cuál crear depende del administrador, si le es más fácil definir a cuáles usuarios no se les debe permitir el uso, debe crear el segundo; por el contrario, si le resulta práctico el definir únicamente a los que sí les está permitido, se debe crear el primero. La tabla siguiente muestra el uso de estos archivos.

Tabla 15 Configuración de acceso al *cron*

Quién Tiene Acceso	cron.allow	cron.deny
Sólo root	No debe existir.	No debe existir.
Todos los usuarios	No debe existir.	Sí existe; pero vacío.
Negación del acceso selectivamente	No debe existir.	Sí existe; con la cuenta de los usuarios a los que se les negará.
Acceso permitido selectivamente	Sí existe; con la cuenta de los usuarios que tendrán permiso.	No debe existir.

Para utilizar este sistema se debe proceder de la siguiente forma:

- Crear un ***archivo*** en el cual se especifique la serie de eventos que se deseen programar. Cada evento debe ocupar un renglón con el siguiente formato:

Minuto	Hora	Día del mes	Mes	Día de la semana	Evento
30	15	31	*	*	/usr/local/tarea
15	2	*	*	0-5	/bin/rm -r /tmp

En el primer ejemplo ejecutará el programa *tarea* a la 3:30 p.m. de cada día 31 de todos los meses, no importando el día de la semana en que caigan. En el segundo, se borrarán todos los archivos del directorio */tmp* a las 2:15 a.m. de todos los días de todos los meses, pero únicamente de Lunes a Viernes.

- Ejecutar el comando:

% crontab archivo

Al ejecutarlo, se creará una copia de este archivo en el directorio */usr/spool/cron/crontab* y se le pondrá como nombre, la cuenta del usuario. Además, se copiará en la memoria

RAM del equipo, quedando activado. Éste permanecerá en memoria y sólo podrá ser modificado por su respectivo dueño.

- Para listar los procesos programados:

  % corntab -l

- Para remover los procesos programados:

  % crontab -r

Cabe destacar que si el sistema es apagado, cuando arranca nuevamente, *cron* empieza a funcionar y lee todos los archivos almacenados en */etc/spool/cron/crontabs*; que es donde están almacenados los archivos programados de cada usuario. Así mismo, es importante mencionar que *cron* ejecuta cualquier evento programado mediante la utilización del shell Bourne. Esto es importante, ya que existen ligeras diferencias en la forma de definir variables y otros aspectos a través de los diferentes shell disponibles en UNIX. Por otro lado, cualquier mensaje de error que ocurra durante la ejecución del evento, es enviado por correo al usuario correspondiente y registrado en el archivo */var/cron/log*.

Este sistema es un importante medio para automatizar las labores cotidianas que no se debe pasar por alto. Por ello, el sistema al ser instalado, utiliza este medio para colocar en él, algunas de las tareas más importantes del administrador, programando ciertas actividades que serán realizadas por las cuentas de root, adm y sys.

Es recomendable analizar estos archivos para determinar si requieren cambios; ya que ellos suponen que el sistema estará funcionando las 24 hrs. del día y utiliza las horas de la madrugada, supuestamente menos saturadas, para realizar algunas labores pesadas. Esto es importante, ya que si el equipo es utilizado únicamente durante el día y por la noche es apagado, nunca se realizarán y podrían causar, a la larga, fallas o la disminución del rendimiento.

El más importantes de ellos, por las labores que desempeña, está definido para root, por lo cual lleva el nombre de */etc/spool/cron/crontabs/root*; un resumen de su listado se encuentra en la pág. V-17. Entre las labores destacan la depuracion del SA, al realizar limpiezas cotidianas y eliminar archivos viejos e inservibles que generalmente ocupan espacio, así como rotar archivos de mensajes, evitando que crezcan indefinidamente y manteniendo únicamente un respaldo de ellos. De estas labores, la penúltima es la de mayor demanda de tiempo y la que redunda en grandes beneficios, ya que reorganiza el SA completo, manteniéndolo en las mejores condiciones.

- Si por los horarios de la empresa donde se labora, estas funciones quedan en horas donde el equipo no funciona, es conveniente reprogramarlas a otra hora, en lugar de eliminarlas o dejarlas así. Para ello:

```
# $Revision: 1.37 $
# The root crontab can be used to perform accounting data collection and cleanup.
# Remove old trash
0     15    *     *     *       find / -local -type f '(' -name core -o -name dead.letter ')' -atime +7 -mtime +7 -exec rm -f '{}' ';'
# Remove old sendmail mail files
2     15    *     *     *       find /var/spool/mqueue -local -type f -mtime +30 -exec rm -f '{}' ';'
# Remove old rwhod files
2     15    *     *     *       find /var/spool/rwho -local -type f -mtime +7 -exec rm -f '{}' ';'
# Remove old vi/ex 'preserved' files
3     15    *     *     *       find /usr/preserve -local -type f -atime +30 -mtime +30 -exec rm -f '{}' ';'
# Rotate the logs
45    20    *     *     1       umask 033;cd /var/cron;if test -s log && test "`wc -c log`" -ge 10240; then mv -f log OLDlog;touch log;
killall 1 cron; fi
45    20    *     *     1       umask 077;cd /var/adm;if test -s sulog && test "`wc -c sulog`" -ge 10240; then mv -f sulog
OLDsulog;touch sulog; fi
45    20    *     *     1       umask 033;cd /var/adm;if test -s SYSLOG && test "`wc -c SYSLOG`" -ge 10240; then mv -f SYSLOG
oSYSLOG;touch SYSLOG; killall 1 syslogd; fi
# If accounting is on it will handle wtmp rotating.
# wtmp and wtmpx are always kept in sync by libe/getut so we should always do things to them together
45    20    *     *     1       if /etc/chkconfig acct; then :; else umask 033;cd /var/adm; if test -s wtmp && test "`wc -c wtmp`" -ge
10240; then mv -f wtmp OLDwtmp; mv -f wtmpx OLDwtmpx; touch wtmp wtmpx; chown adm.adm wtmp wtmpx; fi; fi
#12   4     *     *     *       sh /var/spool/lp/etc/lib/log.rotate
# If this machine is running NIS and it's a slave server, the following commands keep the NIS databases up-to-date.
7     9     *     *     *       if /etc/chkconfig yp; then find /var/yp -type f -name '.*.*' -mtime +1 -exec rm -f '{}' ';' ; fi
8     *     *     *     *       if test -x /var/yp/ypxfr_1ph; then /var/yp/ypxfr_1ph; fi
9     9,15  *     *     *       if test -x /var/yp/ypxfr_2pd; then /var/yp/ypxfr_2pd; fi
10    9     *     *     *       if test -x /var/yp/ypxfr_1pd; then /var/yp/ypxfr_1pd; fi
# If this machine is a NIS master, ypmake will rotate the log file and ensure that the databases are pushed out with some regularity.
# It is best to not build and push the databases at the same time the commands above on slave servers are pulling the databases.
0,17,30,45 * *    *     *       if /etc/chkconfig ypmaster && /etc/chkconfig yp && test -x /var/yp/ypmake; then /var/yp/ypmake; fi
# dodisk does the disk accounting
0     14    *     *     4       if /etc/chkconfig acct; then /usr/lib/acct/dodisk    /var/adm/acct/nite/disklog; fi
# Reorganize file systems
30    15    *     *     1       if test -x /usr/etc/fsr; then (cd /usr/tmp; /usr/etc/fsr) fi
# Repair mangled utmp/wtmp entries
1     14    *     *     *       /usr/sbin/chkutent
```

Listado del arcvhio *letc/spool/cron/crontabs/root*

- Copiar los archivos que sufrirán cambios a un directorio temporal.

  cp /etc/spool/cron/crontabs/root /tmp/root

- Realizar las modificaciones pertinentes en el archivo temporal, para que los procesos se realicen en horas en que se encuentre encendido el equipo, seleccionando las de menos carga de trabajo para no saturarlo.
- Activarlo mediante el comando:

  # crontab /tmp/root

  Al ejecutarlo, se copiará tanto a RAM como a */etc/spool/cron/crontabs* con el nombre del usuario; por lo que se debe estar en la cuenta de root en este caso.

Las rutinas programadas por default para la cuenta de sys, se utilizan para recolectar información del macanismo de reporte de la actividad del sistema (*sar*). Cuando este mecaniso es activado, reportes de la actividad diaria del sistema son generados y colocados en */usr/adm/sa* .

Finalmente, cuando se activa el servicio de *uucp*, entran en función las labores programadas para la cuenta uucp que realizan labores de mantenimiento de este servicio.

Al igual que estos claros ejemplos, el administrador puede agregar o generar nuevos archivos para programar labores y, automatizar de esta forma, sus actividades; mejorando y aligerando la carga de trabajo.

V.IV. Mantenimiento del disco duro

En el capítulo III se describió la estructura de un disco así como varios conceptos y términos relacionados con ellos; por lo que en éste, fijaremos la atención en las rutinas y procedimientos que permitan mantener integra su estructura, como aquéllas que nos permitan modificar o mejorar su desempeño.

En primer lùgar, vale la pena recordar que existen dos tipos de disco: los del sistema y los secundarios. En cada equipo debe existir únicamente un disco del sistema, activo. Es desde éste que arranca el SO; por lo que debe contener las herramientas necesarias en la partición de encabezado de volumen (vh), así como el SA principal (root) en la partición cero y una de swap en la uno. Por default, el disco del sistema suele ser el disco 1 de la tarjeta controladora 0 (dks0d1). Cualquier otro disco es tomado como secundario y puede, o no, tener las herramientas de arranque o cualquier tipo de distribución en la partición. Ya que las herramientas de arranque se encuentran almacenadas en la partición del volumen (vh) y no ocupan mucho espacio, es recomendable tener una copia de ellas en otro disco, aparte del principal, para en dado caso de que sufra daños el del sistema, se pueda tener una alternativa para arrancar y solucionar cualquier problema.

De todo esto, que el mantenimiento de un disco se base esencialmente en el hecho de conservar en buen estado la partición 8, correspondiente al encabezado de volumen, así como controlar las particiones existentes en el disco; temas que serán tratados a continuación.

V.IV.I. Mantenimiento del encabezado de volumen

Cuando se inicializa un disco[63] es creada en forma automática la partición 8 llamada encabezado de volumen (volhdr). Ésta contiene básicamente tres partes: Los parámetros del dispositivo, la tabla de partición y un directorio que muestra las herramientas almacenadas en él. Si la información contenida aquí se llega a dañar, se puede perder la información completa del disco; por lo que cualquier inconsistencia debe ser reparada inmediatamente.

Para el mantenimiento de esta área se cuenta con tres comandos que pueden ser utilizados para detectar y solucionar problemas: *prtvtoc*, *dvhtool* y *fx*.

El comando *prtvtoc* lee la información almacenada en el encabezado de volumen del disco especificado y muestra un resumen de la tabla de particiones y de los parámetros del dispositivo: cilindros, sectores, pistas, cabezas, etc. Es conveniente obtener un listado de cada uno de los discos instalados en el equipo y colocarlo en el manual de procedimientos; para que pueda servir de referencia durante la reparación de daños causados al disco u otras labores como: redistribuir las particiones, saber la estructura del disco, tamaños del área de swap, etc.

La segunda herramienta, *dvhtool*, nos permite modificar el contenido del encabezado de volumen. Como cualquier otra herramienta que puede ser destructiva, debe ser utilizada con cuidado y se encuentra restringida para ser usada desde la cuenta de root únicamente. Es un programa interactivo, en el que al ejecutarse, se pueden disponer de varios subcomandos:

```
# dvtool
Volume? (/dev/rvh)
Command? (read, vd, pt, dp, write, bootfile, or quit):
```

Por default es seleccionado el encabezado de volumen del disco de sistema; pero se puede elegir otro indicando el archivo de dispositivo de dicha partición. La opción *read* permite leer el encabezado de otro disco, y poder trabajar con él. La de *vd* permite actuar sobre el directorio y archivos contenidos en el volumen: leer, copiar, borrar o anexar archivos a él. La de *dp* permite trabajar con los parámetros del disco e información de la tarjeta controladora; aunque se recomienda ampliamente utilizar el comando *fx* si se desean modificar éstos. La de *pt* permite modificar el contenido de la tabla de particiones; al igual que en el caso anterior, *fx* es el recomendado para esta labor. El comando *bootfile* permite modificar cuál será el kernel que utilizará para arrancar el sistema, así como cuál será la partición de swap y la de root. El de *write* escribe cualquier cambio hecho, al disco; actualizando los valores. Y *quit* permite salir de la aplicación. A continuación se presentarán labores desarrolladas con esta herramienta.

[63] Ver Instalación del Sistema Operativo en la pág.I-21, inciso h, que corresponde al uso de la herramienta *fx* para inicializar un disco.

Mantenimiento de las herramientas instaladas en el encabezado de volumen:

- En primer lugar, se debe estar en la cuenta de root:

  *$ **su***
  passwd:
  #

- Ejecutar la herramienta *dvhtool:*

  *# **dvtool***
  Volume? (/dev/rvh)
  Command? (read, vd, pt, dp, write, bootfile, or quit):

- Para listar las herramientas instaladas:

  *Command? (read, vd, pt, dp, write, bootfile, or quit): **vd***
  *Command? (d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE, or l) ? **l***

  Current contents:

File name	*Length*	*Block #*
sgilabel	*512*	*2*
sash	*140800*	*3*
ide	*977920*	*278*
symmon	*156160*	*2188*

- Si se llega a dañar alguna herramienta instalada en el encabezado de volumen del disco, se puede remplazar por una versión correcta. Ejemplo: remplazar la versión de *sash* por la instalada en el directorio */stand*.

  *Command? (read, vd, pt, dp, write, bootfile, or quit): **vd***
  *Command? (d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE, or l) ? **c /stand/sash sash***

- Añadir el comando *fx* al encabezado de volumen:

  *Command? (read, vd, pt, dp, write, bootfile, or quit): **vd***
  *Command? (d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE, or l) ? **a /stand/fx fx***

- Si la versión del directorio */stand* es la dañada, se puede obtener una copia de la del encabezado; por ejemplo la de *sash*.

  *Command? (read, vd, pt, dp, write, bootfile, or quit): **vd***
  *Command? (d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE, or l) ? **g sash /stand/sash***

Recordar que en el directorio */stand* se encuentran almacenados programas que pueden correr en modo stand alone; es decir que no requieren de un sistema operativo UNIX para funcionar, por lo que pueden ser ejecutados desde el PROM para solucionar problemas.

La última herramienta, *fx* será detallada en el siguiente punto, pero es conveniente señalar nuevamente, que si se desea modificar la tabla de particiones o parámetros del disco, ésta es la herramienta más adecuada y recomendada. La de *dvhtool*, debe ser utilizada para manejar el directorio y programas instalados en el encabezado de volumen del disco y, aunque lo permita, abstenerse de modificar la tabla de particiones y otros parámetros con ella.

V.IV.II. Mantenimiento de las particiones del disco

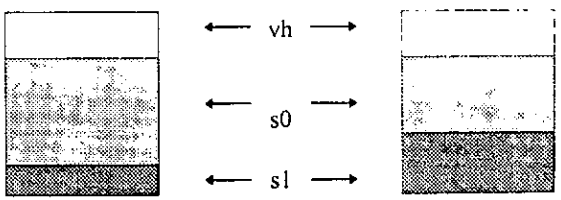
Como se explicó en el capítulo III, las particiones son la forma de dividir el espacio de un disco; de tal forma que sean tomadas lógicamente como varios discos. El comando adecuado para inicializar y particionar un disco, es *fx*. En el capítulo I y en espacial en el procedimiento de instalación inciso h, ya se explicó la forma de inicializar y particionar un disco con *fx*: por lo que en esta sección se darán algunas recomendaciones al utilizar esta herramienta.

Las razones por querer cambiar la forma en que está particionado un disco son variadas:

- Si se requiere más espacio.
- La distribución actual ya no satisface las necesidades.
- Se requiere más espacio en la partición de swap, o cualquier otra.
- Se añadirá un nuevo disco, y se tiene que inicializar y particionar.

Pero en cualquiera de los casos, es aconsejable seguir estas recomendaciones:

- Si toda el área de un disco se encuentra distribuida en particiones (lo que generalmente sucede) el incrementar el área de una, reducirá el área de otra; casi siempre es la adyacente, aunque la repartición pudiera en determinados casos afectar varias.



- Por tal motivo, el reparticionar un disco debe ser planeado cuidadosamente.

- Se debe realizar un respaldo de las particiones que se verán afectadas; pero es recomendable el hacerlo de todo el disco, aunque se suponga que los cambios no afectarán su contenido.

- Si las modificaciones hechas afectarán particiones tipo EFS que contienen sistemas de archivos, se tienen que regenerar éstos; por lo que se debe utilizar el comando *mkfs* para crear un nuevo SA y restaurar el respaldo hecho a dicha partición.

- Si el disco que se reparticionará es el de root, esta labor debe ser efectuada desde miniroot; por lo que es aconsejable el dar aviso a los usuarios de la suspensión del servicio. Este proceso debe ser planeado adecuadamente, ya que no se contarán con los diversos comandos de que se dispone cuando se trabaja desde el SO.

- Los esquemas de partición explicados en el capítulo III, son los estándares en IRIX; pero se puede utilizar cualquier otro , así como definir y crear las particiones 2,3,5,11,12,13,14 y 15 que no han sido manejadas con anterioridad[64]. Si se utilizan éstas, se deben generar los archivos de dispositivos que permitan el acceso a la información almacenada en ellas, de forma manual[65] con el comando *mknod*.

- Todo disco debe contener un encabezado de volumen (partición 8) que contiene información referente al disco, y en el cual, no se podrá almacenar datos del sistema.

- El disco de sistema debe contener una partición para swap, que debe ser la 1, y una para root, la 0.

El Procedimiento para Reparticionar un Disco es el siguiente:

- Recolectar los datos que muestren la estructura de la partición actual

  *# **prtvtoc***

- Planear la repartición.
- Respaldar la información del disco.
- Si se trata del disco de sistema, darlo de baja y entrar al PROM.

  *# **shutdown -y***

[64] Sólo se pueden crear 16 particiones en un disco; de la 0 a la 15.
[65] Los archivos de dispositivos estándares son creados automáticamente por IRIX.

- Si se trata de cualquier otro disco que no afecte al sistema, desmontarlo.

  *# **umount /aplic***

- Correr la herramienta para particionar

  *>> **boot /stand/fx - -x*** ——— Desde el PROM. Es forzoso colocar dos guiones en la opción *x*, que permite realizar acciones destructivas (modificar) en la tabla de particiones del disco.

  *# **fx -x*** ——— Desde el SO.

- Utilizar *fx* para listar y reparticionar el disco (ver pág. I-21). En el menú de repartición se puede seleccionar la opción *resize* para cambiar el tamaño de una partición en específico (root, swap, usr, o el disco entero). Al seleccionarla, el sistema mostrará un mensaje de advertencia indicando que al cambiar el tamaño de una partición, se afectarán otras.

- Si no sufrió daños o modificaciones el SA de root, arrancar en modo de un usuario; de caso contrario arrancar en modo de miniroot.

- En cualquier caso, reconstruir los SA afectados

  *# **mkfs /dev/dsk/dks0d2s0***

- Restaurar la información respaldada.

- Reiniciar el equipo de forma normal.

  *# **reboot***

Es *fx* una herramienta muy poderosa que debe ser utilizada con cuidado; es la principal para el mantenimiento de los discos duros. Con ella se pueden realizar labores como:

Inicializar un disco. Todo disco antes de poder ser utilizado debe pasar por un proceso de inicialización, en el cual se revisa su estado; eliminando los sectores defectuosos, y creando la partición 8 del encabezado de volumen. Esta operación es llevada a cabo con la opción *format*.

Pruebas y eliminación de sectores defectuosos. Todo disco tiene sectores defectuosos que son detectados y registrados desde su fabricación; pero debido al uso, pueden surgir nuevos. Estos sectores deben ser separados y registrados para que no sean utilizados y evitar la pérdida de información. Para eliminarlos se puede utilizar la opción *format*; pero se tendría

que regenerar las particiones y restaurar la información de ellas. Por tal motivo, *fx* cuenta con una opción alternativa, *badblock*; que permite separar y eliminar lógicamente estos sectores sin afectar el resto. Cabe destacar que algunos sistemas cuentan con métodos para suprimir sectores dañados en forma automática y al vuelo; es decir, cuando son detectados son eliminados inmediatamente. Para efectuarlo, por cada pista se cuenta con un número determinado de sectores que no son utilizados, pero que sí disminuyen la capacidad de almacenamiento del disco, y que sirven como reemplazos de sectores que se vayan dañando en dicha pista. Es decir, si se daña algún sector, el sistema toma uno de esta zona para reemplazarlo en forma automática. Se debe estar monitoreando, y cuando se detecten un número considerable de sectores defectuosos, o los sectores reservados se vayan a agotar, es conveniente dar un nuevo formato al disco para eliminarlos definitivamente (lógicamente).

Realizar pruebas al disco. La opción que permite realizar pruebas y detectar posibles fallas, tanto en el mecanismo de desplazamiento de las cabezas, como en la superficie de los platos que forman al disco (sectores defectuosos) es la de *debug*. La operación realizada por esta opción generalmente no afecta el contenido de los discos; a menos que se utilice la opción *-x* al ejecutar el comando *fx*, en cuyo caso las pruebas envolverán operaciones de escritura y la información almacenada será destruida.

Otra operaciones permiten modificar las particiones, parámetros del disco, etc.. Por todo esto, es recomendable estudiarla cuidadosamente. Información respecto a ella puede ser obtenida mediante el comando *S man fx*.

V.IV.III. Mantenimiento del área de swap

El área de swap es una porción del disco que es utilizada para almacenar pedazos del contenido de la memoria RAM; es decir, de los programas en ejecución. Esto permite que el equipo aparente tener más memoria RAM y, por tanto, atender un mayor número de procesos; es por esto que se le conoce como memoria virtual. También, la utilización de la tecnología conocida como paginación junto con el uso de memoria swap, permiten ser ejecutados programas que tienen un tamaño mayor al de la memoria RAM disponible. Esta técnica consiste en que al momento de compilar un programa, se hace de un forma inteligente, dividiendo el programa ejecutable en segmentos llamados páginas que el CPU reconoce fácilmente; de tal forma que, al ser ejecutado es cargada la mayor parte del programa en la memoria virtual (área de swap) y sólo se carga en RAM la primer página del programa. Cuando es ejecutada ésta y se requiere otra, el SO se encarga de leer de la memoria virtual la siguiente página y ponerla en ejecución en RAM y así hasta terminar la aplicación. Este proceso de carga y descarga de páginas de RAM a memoria swap es llevado a cabo para todos los programas en ejecución, incluyendo los mismos programas que forman al SO, y de manera automática por el sistema; de tal forma que para el usuario, aparenta estar cargado de forma completa e integra en RAM su aplicación.

Estos detalles son importantes y debido a ellos, mientras se tenga un medio de comunicación rápido y eficiente entre la RAM y el controlador de discos, entre éste y el disco, y que finalmente el disco muestre un tiempo de acceso a los datos idóneo, se incrementará en forma notable el desempeño del sistema.

El disco del sistema o de arranque debe contener la partición número 1 y debe estar destinada para el área de swap. No debe ser colocado en ella un SA; ya que cuando se requiere arrancar miniroot para realizar labores de mantenimiento, el sistema supone que la partición 1 es de swap y por tanto que no se encuentra información almacenada en ella y es ahí donde carga y configura a miniroot, borrando la información que pudiera encontrarse ahí.

El área destinada para swap debe ser como mínimo de igual tamaño que la cantidad de RAM disponible; pero es ampliamente recomendado que se tenga el doble y, dependiendo de las necesidades y carga de trabajo, éste se puede incrementar. Si un programa es cargado y para su ejecución es montado en la memoria virtual, y si no se cuenta con el suficiente espacio para almacenarlo, el kernel lo elimina automáticamente para evitar que quede estancado en demanda de más memoria virtual. Si se llegan a presentar casos frecuentes de esta situación, el área destinada para swap debe ser incrementada.

Cuando es instalado el SO, por default crea un área de swap que permite trabajar con él. El proceso de mantenimiento de swap incluye labores como el reparticionar el disco para incrementar el área destinada a swap, el agregar otra partición para este propósito, o el utilizar archivos para ello, cuyos procedimientos serán tratados a continuación.

V.IV.III.I. Incremento del área de swap.

Si es necesario incrementar el área de swap y solamente se tiene un disco, el único método es reparticionar el disco para incrementarla; claro está, que el área de otra partición se reducirá de manera proporcional. Si es éste el caso, se debe utilizar la herramienta *fx* para reparticionar el disco; según se indica en "Procedimiento para Reparticionar un Disco " en la pág. V-22.

Es importante mencionar que a la partición correspondiente al área de swap, no se le debe crear un SA ni llevar a cabo ningún otro procedimiento especial sobre ella. Al arrancar el sistema será reconocido el nuevo tamaño de la partición y utilizada sin ningún problema.

V.IV.III.II. Adición de una nueva partición al área de swap.

En este caso, generalmente se ha adquirido un nuevo disco que es añadido al sistema. En "Adición de un nuevo disco" pág. V-46 se explica el procedimiento para instalar uno nuevo, por lo que se partirá de este conocimiento para explicar los pasos para utilizar una partición de ese disco como área adicional de swap.

Una vez que ha sido instalado y particionado un nuevo disco, se puede utilizar una de las particiones para añadirla al área de swap; lo cual permitirá incrementar el desempeño y potencia del equipo. La partición destinada al área swap no requiere de ningún tratamiento especial; por lo que una vez definida puede ser añadida inmediatamente mediante el comando *swap*.

 # swap -a /dev/dsk/dks0d5s1

Este comando permite manipular el área de swap. La opción *-a* se utiliza para añadir una nueva partición, por lo que a continuación de ella debe estar el archivo de dispositivo de acceso en bloque. Además de esta opción, se disponen de otras como: para listar el espacio disponible (-l), eliminar una partición del área de swap (-d), etc. El efecto de este comando dura únicamente mientras el SO está funcionando; por lo que al apagar y encender nuevamente el equipo, se tendrá que ejecutar el comando cuando se desee incrementar el área. Si se desea automatizar esta labor, dejar de forma permanente, se puede utilizar el archivo */etc/fstab* para definirlo, y que al encender el equipo se cargue el espacio de esta partición al área de swap. Para ello se debe colocar una nueva entrada en este archivo similar a:

/dev/dsk/dks0d5s1 swap swap 0 0

De esta forma, la partición 1 del disco 5 de la controladora 0, será añadida al iniciar el sistema.

V.IV.III.III. Utilización de archivos para swap

Si se sufren problemas de memoria y el área de swap definido actualmente no es suficiente; si de momento no se puede detener el sistema para reparticionar el disco e incrementar el área; si no se cuenta con otro disco del cual se pueda disponer de una partición para este propósito; si dentro del sistema de archivos se cuenta con suficiente espacio, se puede utilizar un archivo de tamaño adecuado, cuyo espacio pueda ser utilizado como área de swap temporal, que añadido al actual, puede solventar la situación. El procedimiento para agregar un archivo como área de swap es simple:

- Crear un archivo del tamaño necesario para este fin.

  # mkfile 100m /swapfile

  El comando *mkfile* permite crear un archivo de cierto tamaño; en este caso de 100MB llamado */swapfile*. Seleccionar un SA con el suficiente espacio, y que posteriormente no se sufra de falta de espacio en él. De igual forma, si se cuentan con varios discos se debe elegir el de menor utilización, para de esta forma, redistribuir la carga y agilizar el sistema.

- Añadir el espacio del archivo al área de swap

  # swap -a /swapfile.

Aunque se puede establecer en forma definitiva este archivo como una extensión del área de swap, es conveniente, cuando sea posible, ya sea reparticionar el disco o agregar una nueva partición para incrementarla.

V.IV.III.IV. Problema comunes

Generalmente esta área no requiere ningún tratamiento especial para mantenerla. El SO se encarga de controlarla y únicamente se debe estar al pendiente, mediante monitoreos, de la demanda que se tiene, para de ser necesario, aumentarla. El comando indispensable para esta labor sigue siendo el de *swap*. A continuación se describen algunos de sus usos:

 # swap -l

Lista todos los dispositivos (particiones y archivos) definidos como área de swap. El comando:

 # swap -s

Da un resumen de la cantidad de área asignada, reservada, usada y disponible del sistema. Esta opción merece especial atención por la información que proporciona. En primer lugar, el área asignada es la cantidad de memoria que se encuentra utilizada realmente por las aplicaciones; la reservada es memoria que ha sido solicitada, pero que aún no se encuentra en uso. La memoria usada es la suma de las dos anteriores y la disponible es lo que resta del total del espacio destinado para swap (obtenida del comando *swap -l*). Todo esto en bloques de 512 Bytes.

La razón de la existencia de diferentes tipos de área de swap, es la siguiente: cuando un programa entra en operación, utiliza cierta memoria para su uso, donde son colocados datos, lo que se refleja en la cantidad de memoria asignada; conocida como el tamaño real de la aplicación. Por otro lado, las aplicaciones suelen reservar cierta cantidad de memoria adicional, para en caso de ser necesario, poder crecer; cantidad de memoria que puede nunca llegar a ser utilizada realmente, pero que las aplicaciones debido a la forma en que fueron creadas y trabajan, requieren. Ésta es llamada la memoria reservada, que junto con el tamaño real de la aplicación, o memoria asignada, constituyen el tamaño virtual de dicha aplicación.

Este hecho puede llegar a confundir y hacer pensar que ya no se cuenta con memoria, o que se encuentra desbordada por la demanda, siendo que esto realmente no ocurre. Por lo que se debe analizar cuidadosamente los resultados del comando. Si muestran que la memoria disponible está por agotarse, se deben analizar el resto de los campos. Ejemplo:

```
Total: 35000 allocated  +  60000 reserved  =  95000 block used,   5000 block available
         (asignada)          (reservada)          (usada)           (disponible)
```

En este ejemplo, restan únicamente 5000 bloques (2.5 MB) disponibles, del total de 100000 bloques (50 MB) del área de swap; lo que indicaría que está por agotarse. Al analizar, se puede observar que realmente se están ocupando 35000 bloques, es decir 17.5 MB. En este caso se tienen aplicaciones que reservan bastante memoria (30MB) y que muy frecuentemente no llegan a utilizar. Por lo tanto, aún se dispone de memoria física de swap. Si este caso se presentara muy seguido, puede ser utilizada una memoria de swap virtual para solucionarlo; esto es, una cantidad de memoria definida lógicamente que puede ser reservada, pero que dado el caso, no puede ser asignada. De esta forma, la memoria reservada puede ser tomada del swap virtual y la asignada del swap físico. Para ello:

```
# chkconfig vswap on
# /etc/init.d/swap start
```

La primera instrucción configura el sistema para activar la memoria swap virtual. Esta configuración entrará en función automáticamente la próxima vez que sea encendido el equipo; por lo que para activarla en este momento, una vez dada la primera, se debe ejecutar el segundo comando. Esto solventaría la situación anterior. Aquí restaría únicamente, estar al pendiente que la cantidad de memoria asignada no llegue a ser igual a la física; ya que en este caso sí se estaría llegando al límite y podría desbordarse, ocasionando conflictos.

Por otro lado, si los resultados arrojados indican una cantidad de memoria asignada casi igual a la real obtenida por el comando *swap -l*, sí se está llegando al límite de la memoria y se tendría que pensar en la posibilidad de incrementar la memoria RAM o añadir más área de swap siguiendo alguno de los métodos descritos anteriormente. Ejemplo:

```
Total: 95000 allocated  +  4000 reserved  =  99000 block used,   1000 block available
```

Un último caso que puede prestarse a confusión, es cuando la memoria disponible indica una cantidad negativa, sintoma de que se está utilizando más memoria de la disponible, por lo que se pudo haber desbordado:

```
Total: 65000 allocated + 70000 reserved = 135000 block used, -35000 block available
```

En este ejemplo se están ocupando 65000 bloques realmente de los 100000 disponibles; es decir que aún se cuenta con espacio de swap. El conflicto ocurre cuando se agrega la cantidad de memoria reservada, 70000 bloques que dan como resultado 135000; es decir que se está disponiendo de 35000 bloques de algún lugar desconocido. Esto tiene una explicación, la cual es que los resultados entregados por este comando incluyen la memoria swap y no toman en cuenta a la memoria RAM fisica del equipo; por lo que los 35000 bloques (17.5 MB) son de la memoria RAM. Si se cuentan con 128 MB de RAM, se tendrían arriba de 100 MB todavía disponibles; pero si se contase con 32, restarían únicamente 12.5 MB. En el primer caso, bastaría con activar el swap virtual para solucionarlo. El segundo caso es un poco más grâve y se tendría que revisar si realmente se está ocupando toda el área de swap. Si la cantidad de memoria asignada es muy cercana a la real, se estaría cerca de la saturación; por lo que nuevamente la solución es incrementar el área de swap o memoria RAM. Si por el contrario, existe una gran diferencia, no se estaría ocupando realmente toda la memoria swap; por lo que bastaría con activar el área virtual para solucionar este detalle.

Todo esto muestra la importancia de mantener una vigilancia sobre la memoria swap disponible, y en especial cuando sean instaladas nuevas aplicaciones. Existen algunas que debido a la tardanza en empezar a ejecutarse, o por lo muy utilizadas que pueden ser, cuando son ejecutadas se corren varias copias de ellas mismas. De tal forma que si se trata de ejecutar una segunda vez, su respuesta es inmediata; debido a que se encuentran ya instaladas en RAM. Esto brinda una rápida respuesta pero consumen y solicitan una demanda adicional de memoria fisica, que puede no ser requerida. Un caso típico de esto, es el software de WEB "WebFORCE Server 1.1.1" , que al ser instalado, por default establece como parámetros que al ser ejecutado se corran 16 copias de él en RAM, en espera de peticiones de servicio que le lleguen; ocupando 500 KB por cada una y consumiendo un total de 8 MB aproximadamente, aunque no se tenga ninguna petición.

Otras aplicaciones, por los recursos con que pueden funcionar, realizan y requieren una gran demanda de espacio de swap para su funcionamiento, por ejemplo el programa "Adobe Photoshop 3.0" que dentro de sus requerimientos, para poder ser ejecutado necesita como mínimo 32 MB de RAM y 80 MB de swap. De todo esto resalta el hecho de que la compra de un nuevo software, debe ser analizada cuidadosamente, y tomar en cuenta las necesidades de RAM, disco, CPU, versión del SO, etc., que requieren para su funcionamiento, y no encontrarse con el hecho frecuente que una vez adquirido el producto, éste no puede ser instalado por falta de recursos.

V.IV.IV. Mantenimiento del sistema de archivos

De todo lo visto anteriormente, se destaca que el mantener el sistema de archivos en perfectas condiciones es el punto primordial de la administración; ya que éste es la espina dorsal del sistema operativo. En él se encuentran tanto los programas como la información de los diversos usuarios, así como los numerosos archivos que forman al SO; por otra parte, el estado de éste, influye considerablemente en el rendimiento del equipo. Un SA sucio y lleno, puede ocasionar pérdida de información, lentitud del SO o que este último falle.

En el capítulo III se describió la estructura que debe guardar un SA; por lo que en este punto se describirán varias labores, que añadidas a las ya mencionadas, permitirán que las fallas y pérdidas de información sean nulas o mínimas, dando como resultado un sistema sano.

V.IV.IV.I. Corrupción de un SA

Como resumen de todo lo visto se dirá, que la corrupción de un sistema de archivos se da cuando la versión de las tablas del superbloque, inodos, y directorios mantenidos en memoria por el SO, son inconsistentes con las que se encuentran grabadas físicamente en la partición del disco correspondiente. Los siguientes factores pueden ser causa de una corrupción del sistema:

Falla física

Si el disco sufre alguna falla que afecte su funcionamiento, no se podrá hacer otra cosa que reemplazarlo por uno nuevo y bajar de cinta los respaldos de la información; de ahí la importancia de mantener respaldos actualizados.

Si la falla afecta sólo unos sectores, éstos pueden ser añadidos a la lista de sectores defectuosos y eliminarlos lógicamente; para ello se requiere:

⇒ Respaldar la información.
⇒ Ejecutar *fx* para marcar los sectores como defectuosos.
⇒ Si se requiere, dar formato nuevamente a la partición afectada.
⇒ Si se dio formato a un SA, se tendrá que generar uno nuevo mediante el comando *mkfs*.
⇒ Bajar de cinta la información respaldada.

Las fallas físicas son factores aleatorios difíciles de predecir; pero la mejor forma de prevenirlas, es siguiendo las recomendaciones del fabricante respecto a su correcto uso, al igual que verificar la regulación de la corriente eléctrica y las condiciones de temperatura e higiene del lugar.

Errores Humanos Este tipo de errores son los de mayor frecuencia y los que pueden ser solucionados siguiendo rutinas simples. Los problemas más frecuentes suelen ser:

El apagar el equipo sin darlo de baja adecuadamente mediante los comandos *shutdown*, *init 0*, etc. Ésta suele ser la principal causa de fallas; ya sea porque se apagó intencionalmente, por accidente o por alguna falla en el suministro de energía eléctrica.

El no dar mantenimiento a los SA. Para ello se pueden emplear las diversas herramientas tratadas en este capítulo. Ésta suele ser la segunda causa de pérdida de información.

El operar el SA sin suficiente espacio en disco; sobre todo si se trata del principal (root).

El desconectar físicamente discos duros externos sin dar de baja y apagar adecuadamente el equipo; sea por accidente o predeterminadamente.

V.IV.IV.II. Revisión

El revisar el buen estado del SA es un punto importante, y que se debe efectuar a intervalos regulares. El SA puede presentar inconsistencias debido a problemas que se presentan durante su operación; las cuales, si no son reparadas pueden causar la pérdida de información parcial o total del disco. Para evitar cualquier posible problema, es aconsejable calendarizar labores de mantenimiento tendientes a analizar su estado, y si se llegan a detectar fallas, solucionarlas antes de que causen problemas. Por otra parte, es recomendable el llevar a cabo revisiones del buen funcionamiento del SA después de realizar labores de mantenimiento del disco duro o antes de realizar actualizaciones, instalación de aplicaciones o respaldos; para asegurarse de que la información con la que se está trabajando, se encuentra en perfectas condiciones.

La herramienta principal que se cuenta para este aspecto es el comando *fsck*, (File System CheckK). Este comando puede ser ejecutado manualmente por el administrador en cualquier momento, o automáticamente por el SO cuando se enciende el equipo; si es que se encuentra activada la bandera *fs_dirty* que indica que probablemente el SA no fue desmontado adecuadamente la última ocasión que se utilizó. De preferencia, esta operación debe ser realizada en modo de un usuario (single user) o de administrador de sistema; para evitar posibles interferencias de usuarios conectados al equipo, que afecten la integridad de la operación. Por ello, se deben efectuar los siguientes pasos:

Revisión de un SA

- Dar aviso a los usuarios que el servicio será suspendido por labores de mantenimiento. De preferencia, esto debe ser dado a conocer días antes de efectuarse el evento, programando los avisos mediante el comando *corn* o *at*, o avisos mediante el uso de *wall*.

```
# wall
El servicio será suspendido de las 7:00 a.m. a las 7:30  a.m. por labores de
mantenimiento.

Atte:
      Administrador
^d
```

Se recomienda utilizar los horarios que menos afecten a los usuarios y el mínimo de tiempo.

- Antes de suspender el servicio, realizar una última inspección para verificar que ningún usuario se encuentre en sesión, y de ser necesario, dar un aviso final.

```
# who
root    ttyq0       Aug 20 17:41
#
```

- Pasar al modo de un usuario:

```
# init 1 .
The system is shutting down
Please wait.
INIT:SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for Single User Mode):
Entering Single User Mode.
TERM=(iris-ip)

#
```

Para ingresar al modo administración de sistema.

Dar la clave secreta de root.

Presionar la tecla *ENTER* para aceptar el tipo de monitor que muestra por default.

- Sincronizar los discos para asegurarse de que el contenido de todos los buffer mantenidos en RAM sean escritos al disco; que todas la modificaciones hechas a los archivos hasta ese momento, se hayan actualizado a disco.

```
# sync
```

- Para revisar un SA, éste debe estar desmontado; a excepción del SA principal (*/dev/root*). Por tal motivo, efectuar los siguientes pasos:

Cualquier otro SA	Para el SA Principal
# mount */dev/root on / type efs (rw,raw=/dev/rroot)* */proc on /proc type proc (rw)* */dev/fd on /dev/fd type fd (rw)* */dev/dsk/dks0d3s0 on /usr type efs (rw)* ***# umount /dev/dsk/dks0d3s0*** ***# fsck /dev/dsk/dks0d3s0*** *fsck: checking /dev/dsk/dks0d3s0* *** Phase 1 Check Blocks and Sizes* *** Phase 2 Pathnames* *** Phase 3 Connectivity* *** Phase 4 Reference Counts.* *FREE INODE COUNT WRONG IN SUPERBLK* *FIX?* ***Yes*** *** Phase 5 Free List.* *40326 files 2714450 Blocks 1167604 Free* *CHEEKSUM WRONG IN SUPERBLK* *FIX?* ***Yes*** ***** FILE SYSTEM WAS MODIFIED ***** #	***# fsck /dev/root*** *fsck: checking /dev/root* *** Phase 1 Check Blocks and Sizes* *** Phase 2 Pathnames* *** Phase 3 Connectivity* *** Phase 4 Reference Counts* *** Phase 5 Free List* *FREE BLK COUNT WRONG IN SUPERBLK* *FIX? yes* *BAD FREE LIST* *SALVAGE?* ***Yes*** *** Phase 6 salvage Free List* *2550 files 192704 Blocks 1620688 free* *REMOUNT ROOT? yes* *Remounting root.* #

- Una vez terminada la operación, restaurar el servicio nuevamente.

  # init 3 ó
  # reboot

- Checar la existencia de archivos huérfanos y reconectados; tratar de localizar su procedencia (ver descripción más adelante).

Es importante explicar el funcionamiento del comando, para en dado caso de que se presenten problemas, saber qué es lo que se debe hacer. El funcionamiento de *fsck* es llevado a cabo en 6 fases:

- La primera, *Block and Sizes*, se encarga de checar las listas de inodos y reporta cualquier error encontrado en ellas. Aquí, se pueden presentar mensajes informativos o preguntas que requieren una respuesta. Las posibles opciones de respuesta que presenta son:

  CONTINUE ? Si se responde ***n*** (no), el programa termina; si es ***y*** (sí), prosigue la revisión del sistema. Si se llega a presentar, es recomendable responder ***yes*** y al terminar la ejecución del comando, volverlo a ejecutar para realizar una segunda revisión.

  CLEAR? Si se responde ***n*** (no), ignora el error encontrado; si es ***y*** (sí), se limpia el inodo afectado. Si se llega a presentar este tipo de error, es aconsejable limpiarlo y erradicar el problema. La respuesta de ***n*** sólo debe ser dada si se conoce alguna otra forma más efectiva o menos ruda de solucionarlo.

- La segunda, *Path Names*, recorre el árbol jerárquico de los directorios a partir del principal (/) analizando cada inodo de cada archivo contenido en cada directorio. El caso más critico es si se encuentra en mal estado el inodo del directorio principal (root), en cuyo caso se puede correr el riesgo de perder toda la información del SA completo. Si se presenta el mensaje ROOT INODE UNALLOCATED, no hay nada que hacer, la información contenida en ese SA se ha perdido irremediablemente. Si se presenta la de DUPS/BAD IN ROOT INODE, se puede responder con ***y*** (sí) para que se trate de salvar lo más posible de la estructura de directorios. Si se presenta la de ROOT INODE NOT A DIRECTORY, nos indica que el inodo que apunta al directorio principal, parece no contener un directorio; aquí se puede responder con ***y*** (sí) par indicarle que trate la información como si lo fuera y, si la información contenida ahí era realmente el del directorio principal y ésta no ha sufrido daños, se podrá recuperar y solucionar el problema.

  Una vez revisado el directorio principal, empieza el análisis en los demás directorios. Aquí sólo puede aparecer la pregunta de *REMOVE?* para indicar una falla en un archivo; cuya única solución, es la de dar como respuesta un ***y*** (sí). Claro está, que esto eliminará los archivos que contenían el problema; por lo tanto, hay que tratar en la medida de lo posible[66] de detectar cuáles fueron para bajarlos de cinta, si se tenían respaldados. Si se responde con ***n*** (no), no serán eliminados, pero el problema persistirá; lo cual no es conveniente.

[66] Generalmente resulta inútil el intento; ya que sólo se dan como datos importantes, el dueño del archivo y su tamaño; pero se deben extenuar las posibilidades.

recomienda responder con *y* (sí), y al terminar la revisión del sistema, efectuar una nueva para tratar de solucionar adecuadamente la falla. La de FIX, que se presenta cuando existe una discrepancia en el contador de bloques libres del superbloque del SA; se recomienda responder son *y* (sí) para que sea reemplazado el contador erróneo, por el correcto. Por último la de SALVAGE, la cual al responder con *y* (sí) reemplaza y actualiza el mapa de bit's de bloques libres del SA por el mapa ya corregido y actual.

- Finalmente la fase seis, *Salvage Free List*, se encarga de reconstruir el mapa de bit's de bloque libres. Esta fase se efectúa sólo que se haya detectado una falla en el mapa de bit's y se haya respondido con *y* (sí) a la pregunta de SALVAGE del punto anterior. Cabe destacar que en esta fase no se presenta ningún error.

- Una vez concluidas las fases, termina sus funciones el comando y generalmente se despliegan mensajes informativos indicando, entre otras cosas, el número de archivos, bloques y bloques libres del SA. Se pueden presentar otros problemas en la fase de cierre del comando como: Puede ocurrir que la bandera DIRTY del superbloque se encuentre encendida, indicando que el SA no fue dado de baja adecuadamente en las sesiones pasadas (se encuentre posiblemente corrupto). Cuando esto sucede, realiza la pregunta SUPERBLOCK MARKED DIRTY; se puede responder con *y* (sí) para indicar que se limpie la bandera, ya que al correr este programa se ha verificado su integridad; o responder con *n* (no) dejándola así, en cuyo caso, cuando se inicie nuevamente el sistema operativo, se detectará que la bandera está activada y en consecuencia se correrá el comando *fsck* para verificar el SA nuevamente.

  También puede aparecer el mensaje de SECUNDARY SUPERBLOCK MISSING que indica que no existe una segunda copia del superbloque; aquí se le debe indicar que la cree para en dado caso de que se dañe la primera, se tenga de respaldo esta segunda copia. Si aparece PRIMARY SUPERBLOCK WAS INVALID, indicarle que lo reemplace con la segunda copia existente en el SA, para solucionar el problema.

Del funcionamiento del *fsck*, vale la pena destacar algunos aspectos importantes. En primer lugar, al utilizar dispositivos tipo raw, las funciones desempeñadas por este comando se realizan en forma más rápida. Así también, como ya se mencionó, para que *fsck* pueda analizar un SA, éste debe estar desmontado; a excepción del de root, que debe estar montado para que pueda funcionar el comando *fsck*. Es por este motivo, que una vez checado el SA y se hayan realizado correcciones en él, se despliega un mensaje indicando que si se desea re-montar, para que queden en operación y en memoria RAM, las tablas del SA ya corregidas.

También es importante recordar, que toda acción correctiva que realice *fsck*, puede causar la pérdida de datos; por ello, siempre despliega mensajes indicando cuál es el problema, y después pregunta si lo corrige o no. En este aspecto, es recomendable indicar siempre que se efectúen las correcciones, y tratar de recuperar los archivos afectados por la acción,

- La tercer fase, *Connectivity*, localiza cualquier directorio sin referencia encontrado en la fase anterior y lo reconecta. Un directorio o un archivo sin referencia son aquéllos que no contienen un inodo que los esté apuntando y suelen ser llamados huérfanos. Aquí, es recomendable responder *y* (sí) a la pregunta de RECONNECT?. En este punto, *fsck* tomará un inodo del directorio */lost+found* y lo asignará a este directorio o archivo; de tal forma que una vez terminada la revisión del sistema, se puede acudir a este directorio y ahí se encontrarán los archivos que reconectó. Éstos llevarán por nombre el número de inodo que se utilizó para reconectarlos; por lo que es tarea del administrador, el tratar de determinar cuál era su nombre original y restaurarlos en su ruta correcta. Cuando se tratan de archivos de texto, se pueden listar con los comando *more* o *cat* para tratar de analizar su contenido y determinar cuál era su nombre y ubicación original. El problema consiste cuando se tratan de archivos binarios; ya que no se puede contar con un método eficaz para identificarlos. Aquí se puede tratar de buscar información que nos pueda servir como referencia para identificarlos dentro de él, mediante el comando *strings*, que lista los caracteres que pueden aparecer en la pantalla únicamente; o bajo suma precaución y tomando todas la medidas, ejecutarlos para tratar de averiguar qué programa son. Si se utiliza esta última opción, es recomendable realizarlo desde la cuenta *guest*, en cuyo caso será necesario copiar el programa del directorio */lost+found* al directorio hogar de la cuenta *guest* ($HOME). Si se tratan de directorios, es conveniente cambiarse a ellos y tratar de investigar en los archivos o subdirectorios que contienen. En ocasiones resulta imposible, pero se debe intentar; ya que si eran programas importantes, su ausencia puede afectar la operación del sistema operativo.

  Es importante en este punto recordar que todo SA debe contener un directorio llamado */lost+found*, que es generado automáticamente cada vez que se crea un SA. Administradores novatos no suelen conocer su utilización y la importancia de éste para el funcionamiento de *fsck* y lo eliminan, ocasionando que no pueda reconectar los archivos o directorios perdidos y sean eliminados automática e irremediablemente al momento de efectuar la revisión del SA.

- En la fase cuatro, *Reference Counts*, se checan que sean correctas las múltiples referencias que se hacen a un archivo; esto es, la existencia de ligas al mismo archivo. Aquí se pueden presentar cuatro tipo de preguntas: la de RECONNECT, que tiene el mismo significado que en el punto anterior; por lo que es recomendable responder con *y* (sí). La de CLEAR, que al dar la respuesta de *y* (sí) limpia el inodo afectado. La de ADJUST, que indica que hay una discrepancia entre las ligas encontradas de un archivo durante la revisión, y el contador de ligas del inodo del archivo afectado; aquí es recomendable indicar *y* (sí) para que sea ajustada la cantidad de ligas del inodo, al valor real. Finalmente la de FIX, la cual debe ser contestada con *y* (sí) para que solucione el problema encontrado.

- La fase cinco, *Free List*, revisa la tabla de bloques libres del SA. En esta fase se pueden presentar tres tipos de preguntas: CONTINUE, que indica un problema serio; se

posteriormente (aunque generalmente no es posible). La opción para que no las efectúe y las deje tal y como están en cualquier caso, sólo debe ser dada si se conoce algún método para tratar de recuperar el archivo dañado; después de lo cual, se puede correr nuevamente *fsck* para erradicar el problema. Si se deja sin solucionar alguno de los errores encontrados, por pequeño o insignificante que parezca, puede acarrear consecuencias durante el transcurso del tiempo, perjudicando más información y llegar a causar la pérdida total del SA.

Una técnica utilizada por varios administradores, y en especial sobre SA con poco movimiento de archivos, para determinar cuáles pueden ser los archivos que se dañan y pierden durante la reparación de problemas encontrados por *fsck*, es la de:

- Realizar un respaldo del SA.
- Obtener un listado detallado de todos los archivos respaldados.
- Si durante una revisión del SA se encuentran problemas y son afectados archivos, una vez terminada la operación de *fsck*, se obtiene un listado detallado del SA ya reparado.
- Se comparan mediante programas como *diff*, las diferencias entre el listado del SA actual y el del respaldado.
- Las diferencias encontradas se analizan para determinar si son archivos que pudieron ser eliminados por el usuario o si pueden ser los archivos afectados y eliminados por *fsck* (que pueden ser colocados en el directorio */lost+found*); para ello se puede utilizar como referencia, el tamaño de los archivos, o comparando un listado de cada uno de ellos mediante la ayuda de los comandos *cat*, *more* o el mismo comando *diff*.
- Si se localizan, pueden ser restaurados del respaldo.

V.IV.IV.III. Defragmentación

Otra labor importante en el mantenimiento del SA es el reorganizar la información almacenada en él.

Durante la operación diaria, la información del SA es modificada, eliminándose y grabándose nuevos archivos. Esto ocasiona que bloques de datos del disco queden libres y vuelvan a ser ocupados, lo cual no sucede de la mejor forma, tendiendo a incrementar la fragmentación y a que la información de un archivo quede dispersa en bloques por todo el disco. Desde el punto de vista de la seguridad de la información, esto no representa ningún problema, ya que pueden ser recuperados sin dificultad alguna; pero desde el de funcionamiento, la fragmentación excesiva ocasiona pérdidas de tiempo al tener que desplazar las cabezas del disco de un lugar a otro para leer o grabar la información de un archivo; redundando en un decremento del desempeño del SO en general. También el movimiento excesivo de las cabezas implica un desgaste mayor, que a la larga redunda en una disminución del tiempo de vida del disco. Por el contrario, en un SA bien administrado y organizado, la fragmentación

no existe o es poca, teniendo como resultado que la información de un archivo se encuentra almacenada, generalmente junta, por lo que la lectura o escritura es continua y sin movimiento de las cabezas del disco.

El comando utilizado para reducir la fragmentación del SA es el *fsr* (File System Reorganizer). Por la importancia que representa el que se encuentre en óptimas condiciones el SA, este comando se encuentra programado por default dentro del *cron* de root al ser instalado el sistema operativo. Es importante mencionar que esta aplicación usa en forma intensiva y proporcional al tamaño del SA a reorganizar, la memoria RAM del equipo cuando entra en operación. Ésta es la razón por la cual, se encuentra programada para ser ejecutada una vez a la semana (el día domingo) y a las 3:00 a.m.; que supuestamente es una hora donde la carga de trabajo del equipo es mínima. Nuevamente, si esta hora no es la adecuada para un caso en particular, es recomendable reprogramarla; pero no es aconsejable eliminarla o dejarla así, ya que no se ejecutaría y por lo tanto, se incrementaría la fragmentación del disco. La forma de trabajar de este comando es la siguiente.

- Al ser ejecutado, examina por default el archivo */etc/mtab* para determinar los SA sobre los que efectuará la reorganización. Este archivo contiene la lista de los SA que se encuentran montados actualmente, y efectuará la operación sólo sobre los que tengan la opción de *rw* (permiso de lectura y escritura).
- El análisis, reorganización y defragmentación los efectúa sobre archivos regulares[67], y directorios, empezando por el archivo que tenga el inodo 0 del primer SA listado en */etc/mtab*. Esta operación la realiza en tres fases: En la primera reorganiza los directorios y los descriptores de extensiones indirectas, que contribuye a mejorar el rendimiento en la búsqueda de información así como en el desempeño del comando *fsck*; En la segunda compacta en lo posible, la información de los archivos, mejorando el desempeño del equipo; y durante la fase final, efectúa la defragmentación. Estas fases las lleva a cabo en forma cíclica sobre cada uno de los sistemas de archivos montados.
- Por default al ser ejecutado, trabaja durante 2 hrs. reorganizando todo lo posible. Al concluir el tiempo, guarda un registro del estado y cuál fue el archivo donde se quedó en */usr/tmp/.fsrlast*.
- Cuando vuelve a ser ejecutado (mediante el *cron*) revisa el archivo */usr/tmp/.fsrlast* para localizar dónde se quedó y continuar su operación. Si el archivo llega a ser eliminado o si encuentra alguna inconsistencia en él, empezará su operación nuevamente desde el principio.
- Los mensaje de resultados de las operaciones que efectúa son mandados por default, por medio del sistema de reportes SYSLOG, cada vez que se completa un SA; quedando registrado en el archivo */var/adm/SYSLOG*.

[67] Un archivo regular es aquél que contiene un guión (-) en la primer columna al ser listado con el comando *ls -l*.

Cabe destacar que si durante su operación se encuentra con archivos que están bloqueados por algún usuario o proceso, así como archivos especiales FIFO, ligas simbólicas, socket, etc., son saltados y no se efectúa ninguna operación sobre ellos.

Como se puede ver, ésta es una operación cíclica que se debe mantener para garantizar el óptimo desempeño del equipo. Aunque *fsr* está diseñado para operar desde el *cron*, puede ser ejecutado manualmente para efectuar la reorganización completa de cualquier SA cuando se desee.

V.IV.IV.IV. Expansión

Uno de los problemas a que se enfrentaban los anteriores administradores, era el incremento en la demanda de espacio para almacenar la información, así como el incremento en el tamaño de los archivos, y en especial, de las bases de datos. Los antiguos sistema de archivos tenían limitados estos recursos; como el hecho de que no podían sobrepasar los límites de una partición, y por tanto, el SA más grande que se podía generar correspondía al tamaño del disco, cuanto todo su espacio era asignado a una partición.

La única solución posible en esos tiempos, era la de colocar toda la información de un directorio en un disco separado para que éste fuese montado automáticamente al arrancar el sistema (ver Fig. III-13). Pero ya que las necesidades requerían en algunos casos otro tipo de solución, se tenía entre las manos un reto a resolver. De la solución a estas necesidades, surgieron técnicas ampliamente difundidas en todos los sistemas operativos, como: la creación de Volúmenes Lógicos conteniendo Sistemas de Archivos Virtuales; SA con alto desempeño y carga distribuida conocidos como Striped; o el crecimiento de los SA ya existentes.

En la actualidad existen discos de gran tamaño (GigaBytes) y los SA presentes permiten controlar particiones y la creación de archivos de un tamaño cada vez mayor, que cubre la mayoría de las necesidades en forma normal; pero si no fuera suficiente, se pueden utilizar las técnica mencionadas anteriormente y descritas a continuación, para expandir el tamaño del SA.

V.IV.IV.IV.I. Volúmenes lógicos

Un Volumen Lógico está constituido por varias particiones que pueden estar ubicadas en el mismo o en distintos discos, y que el sistema por medio del software apropiado, las reconoce y las trata como si fuera una sola. De esta forma, se puede crear un SA normal que esté ubicado en el VL, y romper así, con la limitación que impedía que los SA atravesasen los límites de la partición ocupando varias a la vez.

Viéndolo de otra forma, la técnica de particionar un disco, permite ver cada una de las particiones pertenecientes a un disco, como un disco separado; y la de volúmenes lógicos, permite que varias particiones de diferentes o el mismo disco, sean reconocidas como una sola.

Son cuatro los archivos y programas que se encargan de controlar y definir los VL: */etc/lvtab*, *mklv*, *lvinit* y *lvck*. */etc/lvtab* es el archivo donde se definen los VL; es análogo al */etc/fstab* que se utiliza para definir los SA normales. El contenido de éste es leído por los restantes programas que se encargan de mantener y crear los VL. Para definir un nuevo VL, basta con editar este archivo y agregar una nueva entrada; o para modificar uno, sólo es necesario modificar la entrada que lo define. Cada línea de este archivo debe guardar el siguiente formato:

Nombre_del Dispositivo_del_VL:[*Nombre_del_VL*]:[*opciones*:] **devs=***Nombre_de_dispositivos*

Nombre_del Dispositivo_del_VL es el nombre del dispositivo que controlará el volumen lógico. Éste debe empezar con **lv** y seguido de un número que puede ser del cero al nueve (lv0, lv1,...,lv9). Es requisito indispensable para que pueda ser creado y funcionar, que lleve este nombre; esto quiere decir que en un equipo, sólo pueden existir 10 VL. Estas restricciones se deben a que dentro del Kernel, ya se encuentran definidos estos parámetros, que normalmente satisfacen la mayoría de las necesidades; si se desea crear un mayor número, se tiene que cambiar la variable que define la cantidad máxima de VL en el kernel y regenerar uno nuevo (ver "**Reconfiguración del kernel**", pág. III-26). Vale la pena destacar que en este campo se debe poner el nombre del dispositivo, aunque éste no exista; ya que las siguientes aplicaciones se encargarán de crearlo.

Nombre_del_VL es un nombre que el administrador puede dar al VL para identificarlo. Éste es agregado al encabezado de cada partición que conforma al volumen. Los paréntesis cuadrados indican que este campo es opcional.

Las *opciones* son parámetros que definen la forma en que se tratará al volumen. Éstas se describirán cuando se analicen los volúmenes striped.

Nombre_de_dispositivos es una lista separada por comas, del nombre de los dispositivos de tipo bloque (archivos de dispositivos especiales) que pertenecen o sirven de acceso a las particiones que conformarán el VL. Estos archivos generalmente se encuentran ubicados en el directorio */dev/dsk*. Esta lista debe ser introducida por la palabra clave **devs=**.

El programa *mklv* se encarga de crear los VL; para lo cual, lee el contenido del archivo */etc/lvtab*. Entre las tareas que desempeña, está la de colocar etiquetas a cada una de las particiones para que puedan ser identificadas como un VL; la de crear los archivos de

dispositivos especiales (lv0 -lv9) dentro del directorio */dev/dsk* y */dev/rdsk* para que puedan ser accesados en forma normal los VL; y la de ejecutar la aplicación de *lvinit* para que lo inicialice y pueda ser utilizado.

El programa de *lvinit*, como ya se mencionó, se encarga de inicializar los controladores de dispositivos pertenecientes a VL; sin esta inicialización, no se podrá grabar o leer ninguna información en el VL. Este programa es ejecutado por el SO cada vez que arranca, y cuando lo hace, examina el contenido del archivo */etc/lvtab* para determinar que VL debe inicializar; por lo que la ejecución manual de este comando generalmente nunca se realiza.

Finalmente, *lvck*, se encarga de revisar la consistencia del VL. Cabe destacar que este comando no revisa el estado del SA colocado en el VL; sino la estructura del volumen en sí. Esto quiere decir que *lvck* revisa que las etiquetas que identifican a cada partición como miembros de un VL, estén correctas. Este comando debe ser ejecutado cuando se realicen las labores de mantenimiento y en especial, antes de ejecutar *fsck* sobre el SA montado en el VL. También, si al arrancar el SO aparecen mensajes de error generados por *lvinit*, se debe utilizar este programa para detectar y solucionar, si es posible, cualquier error que esté ocasionando el mal funcionamiento.

Como podemos observar, los VL ofrecen grandes beneficios; pero al igual que otros recursos, éste tiene algunos inconvenientes que se deben analizar. El primero es que como se requieren correr aplicaciones para que el SO pueda reconocer los VL, el SA perteneciente a root no puede ser colocado en un VL; esto es, los directorios */etc*, */dev*, */bin*, etc. En segundo lugar, es indispensable que todos los discos conteniendo particiones que pertenezcan a un VL, siempre deben estar funcionado; si en algún momento se daña un disco o una partición, toda la información del VL quedará inaccesible. Esto es, que la información de particiones pertenecientes al VL que no sufrieron daños, no podrán ser recuperadas; de ahí la importancia de realizar respaldos.

Cuando se crea un VL, el SO a través de los controladores de dispositivos del volumen, se encargará de grabar o leer infamación, siendo todo este proceso transparente para el usuario. Ellos se encargarán de distribuir la información a través de las distintas particiones del volumen. A continuación, se presentará el proceso para colocar un SA sobre un VL.

Procedimiento de creación de un Volumen Lógico:

- Si el VL estará sobre nuevos discos, éstos deben ser agregados tanto física como lógicamente al equipo (ver pág. V-46).

- Planear y seleccionar las particiones que formarán el volumen lógico

- Añadir una nueva entrada en el archivo */etc/lvtab* para definir el VL. Por ejemplo, se creará el vl1 llamado *aplicaciones* conteniendo 2 particiones; la primera es la número 7

del disco con dirección SCSI 2 y la segunda es la 0 del disco SCSI 3. El archivo quedaría de la siguiente forma:

lv1:aplicaciones:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks0d3s0

- Ejecutar el programa que se encarga de construir el VL:

  # mklv lv1

  Entre otras cosas, este programa creará los archivos de dispositivos */dev/dsk/lv1* y */dev/rdsk/lv1*, creará el VL y lo inicializará.

- Crear un SA sobre el VL de forma normal:

  # mkfs /dev/rdsk/lv1

- Ahora se puede montar el VL de forma normal:

  # mount /dev/dsk/lv1 /aplic

- Es conveniente agregar una nueva entrada al archivo *'etc/fstab* para que sea montado automáticamente al momento de iniciar el sistema:

  /dev/dsk/lv1 /aplic efs rw,raw=/dev/rdsk/lv1 0 0

Si lo que se intenta es modificar la definición de un VL existente para agregarle o eliminar de él particiones, se realiza el mismo proceso modificando la definición de dicha partición en el archivo */etc/lvtab*. Es indispensable el realizar un respaldo de la información contenida en el volumen para posteriormente restaurarla.

V.IV.IV.IV.II. Volúmenes strip

Éste es un tipo especial de volumen que intenta optimizar su acceso. En este tipo, la información es distribuida equitativamente sobre cada una de las particiones que forman el volumen lógico. El rendimiento obtenido sobre una tarjeta controladora SCSI en la cual se utilizan dos particiones en dos discos, es desde un 20% hasta un 50%. Esto se puede mejorar si se utilizan diversos discos, cada uno manejado por su propia controladora; en este caso, se tienen diferentes rutas para mandar la información a los diversos discos. Todo esto se ve afectado por las velocidades de transmisión y los tiempos de acceso de los discos y tarjetas.

Lo que intentan los volúmenes tipo strip, es el que bloques de información fijos sean grabados en una partición de un disco, y mientras se está grabando esta información, el CPU grabe otro bloque de datos de igual tamaño en un segundo disco. Esto se puede realizar debido al hecho de que la velocidad del CPU es mayor que la de la tarjeta controladora, y mucho más sobre el tiempo de acceso de los discos. De ahí que si se utilizan particiones ubicadas en diferentes discos, permitirá mejorar el rendimiento; y aún mejor, si se utilizan distintas controladoras para cada uno de los discos, se obtendrá el máximo desempeño. Por lo tanto, la peor elección es seleccionar dos o más particiones ubicadas en el mismo disco.

Cuando se utiliza esta tecnología se tiene que definir dos aspectos importantes. En primer lugar, se debe indicar cuántos serán los discos a través de los cuales, se distribuirá la información equitativamente. Esto se realiza mediante la opción *stripes* que se coloca en la línea que define al VL en el archivo */etc/lvtab*. Por ejemplo, si se utilizaran dos particiones, la definición del volumen quedará así:

lv1:aplicaciones:stripes=2:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks0d3s0

Cabe destacar que se hace referencia a dos particiones en diferentes discos. En este caso, se grabará un bloque de datos en la partición 7 del disco 2 y otro en la partición 0 del disco 3. Para que esto pueda llevarse a cabo, es indispensable (es un requisito) que las particiones que forman un VL tipo striped, sean del mismo tamaño; ya que de caso contrario, una pudiera llenarse y la otra no, impidiendo la distribución equitativa de la información para la cual fue diseñado este tipo de volúmenes. Por tal motivo, si el tamaño de alguna partición no coincide, no podrá ser generado el VL.

Otro punto adicional dentro de este aspecto, se tiene que el número de particiones (archivos de dispositivos) colocados en el último campo, debe ser múltiplo del valor dado a la opción stripes. Esto es, que si se da un factor de tres (*stripes=3*), se podrán colocar en el campo *devs=* 3, 6, 9, etc. archivos de dispositivos.

*lv1:aplicaciones:stripes=3:devs=/dev/dsk/dks0d2s0,/dev/dsk/dks0d3s0,/dev/dsk/dks0d4s0, *
/dev/dsk/dks0d2s7,/dev/dsk/dks0d3s7, /dev/dsk/dks0d4s7

En este ejemplo la información será repartida equitativamente en grupos de 3; por lo que primero se utilizarán las particiones 0 de los discos 2, 3 y 4. Cuando se llenen éstas, se utilizará el siguiente grupo y la información será almacenada en la partición 7 de los discos 2, 3 y 4. Todo esto impone otra condición: que si se desea agrandar el tamaño de este volumen (lv1 en este caso), se deberán agregar particiones en grupos de 3. Esto es un factor importante al considerar la creación de este tipo de volúmenes.

El segundo aspecto a considerar es el tamaño que se utilizará como unidad de almacenamiento; esto es, qué cantidad de información será grabada equitativamente en cada una de las particiones del volumen. Este parámetro es dado con la opción de *step=* en unidades de bloques de 512 Bytes. El valor que toma por default, generalmente es el adecuado y es el recomendado; por lo que este parámetro puede ser omitido. Para explicar su funcionamiento asumamos que se definió un VL con las variables stripes=2 y step=1, y 4 archivos de dispositivos declarados. El sistema grabará un bloque de 512 Bytes en la primer partición del grupo, después otro bloque de 512 Bytes en la segunda; esto se estará repitiendo hasta llenar las dos primeras particiones, después de lo cual, se realizará el mismo proceso sobre el segundo grupo.

Finalmente, el procedimiento de creación de VL tipo striped es idéntico al de los VL (ver Procedimiento de creación de un Volumen Lógico en la pág. V-41), con la excepción de que debe ser colocada por lo menos, la opción de *stripes=* en la declaración del volumen.

V.IV.IV.IV.III. Crecimiento de un SA

El proceso de crecer un SA ya existente, está basado en el concepto de volúmenes lógicos. Este proceso permite que a una partición conteniendo un SA se le pueda agregar, o pueda ser expandido, con el espacio de otra. Este método suele ser empleado para expandir los SA sin la necesidad de reparticionar los discos, y sobre todo, cuando el tamaño de la partición requerida excede el tamaño físico del disco con que se cuenta, permitiendo que varios discos o particiones puedan ser tomadas como una sola. El método es sencillo y no afecta el contenido (información almacenada) de la partición con el SA actual, por lo cual, puede representar una alternativa de solución rápida y efectiva al conflicto que supone la saturación del espacio de almacenamiento.

El comando que permite la expansión es el *growfs*. Para que un SA pueda ser expandido es requisito indispensable que no se encuentre dañado, o que tenga activada la bandera *fs_dirty* indicando que se encuentra sucio; por lo que es conveniente ejecutar el comando *fsck* antes de realizar esta operación. A continuación se presenta el:

Proceso para expandir un SA

- Si es necesario, desmontar el SA que se intenta expandir, para que sea checada su integridad; suponer que es el directorio */usr* cuyo archivo de dispositivo es el */dev/dsk/dks0d2s7*.

  #umount /usr
  # fsck /dev/dsk/dks0d2s7

- Siempre que se vayan a realizar modificaciones sobre un SA es conveniente realizar un respaldo de su información; aunque se suponga, como en este caso, que no será afectada. Por lo cual, se debe montar nuevamente el SA y realizar el respaldo.

  #mount /dev/dsk/dks0d2s7 /usr
  #bru

- Planear y decidir qué partición y de cuál disco, se utilizará para expandir el SA. Esta partición, claro está, no debe estar siendo utilizada por ningún proceso ni por el sistema operativo. Si se adquirió un nuevo disco para este propósito, debe ser instalado físicamente en el equipo y adecuadamente particionado.

- Desmontar el SA que se va a crecer.

  # umount /usr

- Definir el nuevo VL en el archivo */etc/lvtab*, para lo cual es necesario agregar la siguiente entrada; suponer que será el lv3.

  lv3:usr:devs=/dev/dsk/dks0d2s7, /dev/dsk/dks0d3s0

  El primer dispositivo representa la partición conteniendo el SA actual y la segunda es la partición que será añadida; es decir que el espacio de /dev/dsk/dks0d3s0 será utilizado para expandir el SA.

- Crear el VL.

  #mklv lv3

  mklv indicará que una de las particiones contiene un SA por lo que pregunta que si desea proceder. Ya que se intenta crecer el SA, se debe responder con *y* (sí).

- Crecer el SA para que ocupe el espacio total del VL.

  #growfs /der/rdsk/lv3

  Al ejecutar el comando, el espacio de la segunda partición será añadido al de la primera, y el contenido de la primera no se verá afectado.

- Montar el nuevo volumen.

  # mount /dev/dsk/lv3

- Modificar el contenido del archivo */etc/fstab*, ya sea eliminando la entrada que montaba el antiguo SA, o comentándola para que no sea ejecutada.

  # /dev/dsk/dks0d2s7 /usr efs rw 0 0

- Añadir una nueva entrada al archivo */etc/fstab* para que sea montado automáticamente el nuevo VL.

  /dev/dsk/lv3 /usr efs rw,raw=/dev/rdsk/lv3 0 0

Si posteriormente se desea expandir aún más el espacio de este volumen lógico añadiendo otra partición, bastará con realizar este proceso nuevamente.

Un punto importante es que todos los discos que contengan particiones pertenecientes al VL, deben estar funcionando para que opere y pueda ser montado y utilizado el SA. Si alguno falla, dañará la información total del VL.

Una vez creado el VL y colocado un SA sobre él, se podrán utilizar las herramientas normales para mantener la integridad de este último de forma normal; como la de *fsck*, *fsr*, etc.

V.IV.V. Adición de un nuevo disco

La instalación de un nuevo disco, puede ser dividida en dos grandes pasos: la instalación física del equipo, que implica el abrir el CPU si se trata de un disco interno, o mediante cables unirlo al CPU si es externo. Y la instalación lógica, que permite que el sistema reconozca y utilice el nuevo disco. Ya durante el transcurso de todo este documento se han tratado los comandos y temas relacionados con este proceso en forma aislada; por lo que se tocarán en este punto las cuestiones importantes únicamente.

El procedimiento para instalar, tanto física como lógicamente un disco es:

- Siempre antes de toda labor de mantenimiento, es recomendable realizar un respaldo; no importando si las operaciones realizadas supuestamente no la afectan.

- Añadir físicamente el disco. Recordar que si se trata de un disco SCSI externo, se debe colocar un terminador al final (ver **Conexión física**, pág. I-6).

- Crear el archivo de dispositivo que permita accesar cada una de las particiones de ese disco. Generalmente el sistema ya cuenta con los archivos de dispositivos de las particiones estándares para los posibles discos SCSI en los directorios */dev/dsk* y */dev/rdsk*. Si éste no fuera el caso, se puede utilizar el comando *mknod* para crear las necesarias.

- Usar el comando *fx* para inicializar el disco, si se trata de uno nuevo, o reparticionarlo a las necesidades actuales ver V-22.

- Crear los SA mediante el comando *mkfs* en las particiones deseadas.

  # mkfs /dev/dsk/dks0d2s0

- Montar los nuevos SA; para lo cual es necesario crear los directorios que servirán como punto de montaje.

  #mkdir /disco2
  #mount /dev/dsk/dks0d2s0 /disco2

- Si este disco cuenta con una partición que será utilizada como swap, añadirla.

  # swap -a /dev/dsk/dks0d2s1

- Añadir las entradas necesarias al archivo */etc/fstab* para que sean reconocidos y montados al iniciar nuevamente el sistema.

  /dev/dsk/dks0d2s0 /disco2 efs rw 0 0
  /dev/dsk/dks0d2s1 swap swap 0 0

Además de este procedimiento manual, se pueden utilizar dos comandos que realizan algunos de los pasos mencionados anteriormente. El primero de ellos es el *add_disck* que permite añadir lógicamente un disco SCSI conectado a la tarjeta controladora que traen los equipos interconstruida en la tarjeta principal. Este comando realiza lo siguiente al ser ejecutado:

- Crea y liga los archivos de dispositivos adecuados para el disco.
- Crea un SA en el disco.
- Crea el directorio donde será montado el nuevo disco.
- Monta el nuevo SA.
- Añade una nueva entrada en el archivo */etc/fstab* para el disco.

Y si se tratase de otra clase de disco, esdi, ipi, jag, etc. (ver **Unidades de almacenamiento**, pág. III-3), se puede utilizar el comando *makedev* cuyo único propósito es el de crear el juego completo de los archivos de dispositivos necesarios para el funcionamiento de dicho disco.

V.V. Respaldos

Los respaldos desempeñan un papel muy importante en la vida de un equipo; ya que gracias a ellos, se pueden recuperar datos perdidos, en el supuesto caso de que ocurra una falla y la información del equipo se pierda. Este tema es tan basto, que se puede realizar un tratado sobre los diferentes métodos, técnicas, herramientas, etc. utilizadas en él, por lo que se tratarán las bases y fundamentos de ellos. Se darán ejemplos de los diversos comandos existentes, aunque no se cubrirán a detalle todas las opciones, por lo que es conveniente, una vez asimilado los conceptos presentados aquí, el estudiar detalladamente cada uno de éstos mediante el uso de la ayuda del comando *man*.

V.V.I. Tipos de Respaldos

Antes de poder realizar un respaldo, es importante conocer los diversos tipos de respaldos que existen. Podemos clasificarlos en: Remotos y locales; Completos o parciales; Incrementales y finalmente de Usuarios.

Un respaldo local es aquél en el que la unidad de respaldo, se encuentra conectada directamente al equipo donde se localizan los archivos que se respaldarán; éste es el caso más común y simple que se puede presentar. El respaldo remoto por el contrario, implica que la unidad de respaldo está conectada en un equipo diferente al que contiene los archivos que se desea respaldar, y ambos equipos deben estar unidos por algún medio. La utilización de redes para este fin, permite el poder compartir dispositivos como las unidades de cinta u otro medio; de tal forma que no es indispensable el contar con una unidad por cada equipo, reduciendo costos. Es importante mencionar que cada SO cuenta con elementos que permiten realizar respaldos remotos, algunos más complicados que otros; pero abren la posibilidad de utilizar una red para dicho fin. En el caso particular del SO IRIX, en los comandos empleados para respaldar localmente, se encuentran implementadas opciones que permiten especificar como unidad de respaldo, un dispositivo localizado en un equipo remoto; para lo cual se debe especificar el nombre del equipo, una cuenta que permita al software local tener acceso al equipo remoto (generalmente se utiliza la cuenta de *guest*), y el nombre del archivo de dispositivo que identifica la unidad que será utilizada para realizar

el respaldo. Para llevar a cabo su labor, utiliza el Protocolo de Cinta Remota; por lo que cualquier equipo que soporte el comando BSD *rmt*, realizará la operación de forma transparente para el usuario. Algunos ejemplos son dados en el punto referente "Herramientas de respaldo".

Independientemente de si el respaldo es local o remoto, éste puede ser completo o parcial. Un respaldo completo debe ser efectuado en general, cuando se haya configurado y probado la eficiencia del sistema actual; y en especial, antes de realizarle cambios (ver "Estrategias de respaldo" en la pág. V-53). Este tipo implica el respaldar en forma completa todo el contenido del SA, y en el caso de IRIX, respalda de igual forma el contenido de la partición 8 correspondiente al encabezado de volumen de disco; que como se mencionó, contiene información relevante sobre la forma en que está particionado y distribuido el disco. Esto permite que si por alguna causa se llega a dañar irremediablemente el SA o el disco, se tenga un método para que desde el PROM, se pueda restaurar el sistema tal y como estaba antes del incidente.

Si es el disco el que se averió, para poder utilizar este método, debe ser sustituido por otro disco de la misma capacidad y caracteristicas; ya que el PROM restaura el contenido del encabezado de volumen, así como la información del SA. Si únicamente sufrió daños el SA, con el empleo de este método quedará tal y como se encontraba funcionando al momento de realizar el respaldo. Es por este hecho que los respaldos completos deben ser programados periódicamente, o por lo menos cuando se realicen cambios importantes tanto en la configuración como en el software instalado.

Los respaldos completos, por lo que se mencionó, podrían satisfacer todas las necesidades; sin embargo, existe un pero en ellos. Por lo general son muy tardados y para realizarlos, es recomendable tener al SO en modo de un sólo usuario; por lo que el servicio debe suspenderse por varias horas, dependiendo de la velocidad y cantidad de información. Es por este hecho que se cuentan con los respaldos parciales que permiten respaldar cierta información; la que ha sufrido cambios o la que por su importancia merece respaldarse frecuentemente. Esto puede ser realizado en un tiempo mucho menor y generalmente, sin afectar las labores de los usuarios.

Los respaldos incrementales es un técnica que emplea los dos métodos mencionados anteriormente. En ésta, se realiza un respaldo completo del sistema, y posteriormente, mediante elementos que emplean cada una de las herramientas, se reconocen los archivos que han sufrido cambios desde la última ocasión que se utilizó el comando y son los únicos que se respaldan. Esto permite que se realice un respaldo completo del sistema y posteriormente y en forma sucesiva, respaldos de los últimos cambios hechos. Podría pensarse que ésta es la mejor solución, pero se tienen algunas condiciones que se deben cumplir al restaurar los archivos. Por ejemplo, si se llega a dañar el sistema, para poderlo restaurar se tiene que bajar la información de la cinta que contiene el respaldo completo, y posteriormente y en secuencia, bajar los archivos de cada uno de los respaldos parciales que

se hicieron hasta la fecha actual. Si se omite uno o se invierte la secuencia de alguna cinta, podría dar como resultado que cierta información quede alterada o perdida. Más adelante se expondrán estrategias de respaldo que emplean los métodos y técnicas vistas.

Finalmente, los respaldos de usuarios forman una parte integral del mantenimiento, y ello consiste en capacitar y educar a los usuarios a que en la medida de lo posible, realicen respaldos de su información; ya que cada uno sabe bien la importancia que tienen los datos que manejan. El realizar estos respaldos no implica que el usuario tenga directamente acceso al equipo, ya que pueden realizarse de forma remota a través de la red, y utilizarse cualquier medio de almacenamiento; dependiendo del tamaño de su información, desde un simple disquete hasta cintas y discos ópticos. Además, si es necesario restaurar un archivo perdido o borrado accidentalmente, es más rápido el hacerlo desde un respaldo hecho por el propio usuario que contiene solamente su información, que utilizar cintas que contienen un respaldo de todo el SA.

V.V.II. Medios de Respaldo

Los medios de respaldo comprenden los diversos dispositivos que serán utilizados para realizar el respaldo. El medio más usado son cintas, de las cuales, 4 formatos son los utilizados. A continuación se muestra una tabla en la que se describen estos tipos de cinta, así como las capacidades de almacenamiento de cada una de ellas.

Tabla 16 Tipos de medios

MEDIO	TAMAÑO	LONGITUD	CAPACIDAD*
Carrete de cinta	½ pulgada, 9 pistas	2400 pies	20 - 45 MB
Cartucho de cinta	¼ pulgada, 4 pistas	600 pies	60 - 150 - 250 MB
Cartucho	8 mm	112 m	2.3 GB
DAT	4 mm.	60 - 90 m	2 - 5 GB

* La capacidad depende del tipo de unidad de cinta que se utilice y la longitud de la cinta.

Como se puede observar, la tendencia es la de crear cintas cada vez más pequeñas, pero que permitan contener una mayor cantidad de datos. Este hecho hace que cada vez se creen dispositivos mejores y más veloces, ya que la capacidad de almacenamiento de la cinta, se ve sujeta a la capacidad para grabar la información de la unidad de cinta que se utilice; así como técnicas para condensar la información, que permiten en un mismo espacio, guardar mayor cantidad de datos. Por esto mismo, se deben utilizar las cintas de capacidad adecuada para la

unidad de cinta que se esté utilizando, y evitar posibles problemas al almacenar o recuperar la información.

En la actualidad está preponderando el uso de cintas DAT, o también conocidas como DDS, por la gran capacidad y lo pequeño de su tamaño; aunque equipos anteriores siguen utilizando cartuchos de ¼ de pulgada, y los más antiguos, carretes de cintas.

La información que procesan los equipos, representa lo más valioso; por lo que se debe extremar precauciones para que ésta no sufra alteraciones ni pérdidas. De ahí la importancia de realizar respaldos de ella, y mucho más, el que se tengan ciertas precauciones antes, durante y después de efectuarlas. Por tal motivo a continuación se presentarán una serie de

Recomendaciones en el cuidado de las cintas:

- Etiquetar las cintas indicando claramente, la información que contiene, el comando empleado para realizar el respaldo, el que debe ser utilizado para bajar la información y no debe olvidarse la fecha en que se realizó. Si es un respaldo completo del sistema, es aconsejable que se anote la clave secreta de la cuenta de root; ya que si se llega a dañar el sistema y se utiliza esta cita para restaurarlo, se debe conocer esta clave para poder accesar al sistema y continuar las labores de restauración. Éste es un punto importante y que suele ser un conflicto si no se lleva a la práctica; especialmente en aquellos lugares que por seguridad, se cambia continuamente la clave secreta de root.

- Usar la cinta adecuada para cada unidad de respaldo. Al adquirir la unidad, ésta incluye un manual que especifica la densidad con la que será almacenada la información, así como el tipo y características de la cinta recomendada por el fabricante. Se deben acatar estas recomendaciones, ya que si no se hace, se corre el riesgo de que al querer respaldar o restaurar los datos, se presenten problemas de lectura o escritura ocasionando pérdidas.

- Utilizar una muy buena estrategia de respaldo que incluya el tener varios juegos de cintas; de tal forma que si se daña una, se pueda contar con otra; o por lo menos no se pierda toda la información.

- Mantener en buen estado la unidad de respaldo; para lo cual se pueden utilizar cintas de limpieza y el comando *mt* para diagnosticar posibles fallas.

  # mt stat

- Ajustar la tensión de la cinta, en especial antes de realizar un respaldo; si la cinta es colocada en la unidad y se deja mucho tiempo en esa posición sin utilizarse, la tensión puede disminuir y ocasionar posibles problemas de inconsistencia al tratar de recuperar los primeros datos almacenados en ella. Para ello, el comando *mt* dispone de una opción:

  # mt ret

- Almacenar las cintas en un lugar seguro; ya que si una persona ajena tiene acceso a ellas, puede restaurar la información en otro equipo y tener acceso a los datos más importantes, brindando la posibilidad de violar la integridad del equipo.

- Es recomendable el tener por lo menos, una segunda copia almacenada en un lugar diferente al de la primera, de preferencia en otro edificio; para en caso de un siniestro en uno de ellos, se tenga una segunda copia intacta.

- Es recomendable mantener las cintas en lugares fuera de la influencia de campos magnéticos, así como de la luz directa del sol; ya que éstos afectan su contenido.

- Si se tienen las cintas almacenadas en un sitio acondicionado, y en especial a diferente temperatura y humedad que el del lugar donde se encuentra el equipo donde se utilizarán, es conveniente que sean trasladadas a este sitio con varias horas de anticipación para que se aclimaten; ya que las características de las cintas pueden variar de forma sensible con la temperatura, y ocasionar problemas al tratar de grabar o leer la información.

- Activar el mecanismo de protección contra escritura que tiene consigo cada cinta, para evitar que por accidente sea alterada su información.

- El tiempo juega un factor importante en la seguridad y confiabilidad de la información almacenada en las cintas; ya que ésta se encuentra grabada en forma de campos magnéticos, que con el paso del tiempo, pueden sufrir cambios; aunque se mantengan en las mejores condiciones. Se estima que el contenido de una cinta puede durar como máximo 2 años sin sufrir alteraciones o defectos; por lo que cuando se piense en guardar su contenido por varios años, se tienen que implementar estrategias de mantenimiento, en las cuales, por lo menos cada año se regrabe la información (copiar de la cinta a disco y nuevamente a la cinta; o de cinta a cinta) para mantener la calidad de grabación.

- Las cintas con el uso pueden llegar a sufrir daños, por lo que deben ser desechadas cada determinado tiempo. Cada fabricante de cintas establece ciertos límites, basados en pruebas que se les realizan, que indican el número de veces que su cinta puede ser reutilizada, o su tiempo de vigencia. Éstos permiten garantizar la integridad de los datos almacenados en ellas; pòr lo que deben ser respetados y reemplazar las cintas cuando se alcancen o se note cualquier deficiencia en ellas. Ya que el valor de la información almacenada en ellas, no se compara con lo que pueda costar otro juego de cintas.

- De estos puntos resalta el hecho de que deben ser empleadas cintas de reconocida calidad.

- Una vez realizado un respaldo, se debe comprobar que la información almacenada en ellas quedó bien grabada. Esto suele ser un conflicto común, que cuando se requiere restaurar los respaldos, los datos no puedan ser leídos por encontrarse dañados. Esto

puede ser evitado consumiendo unos minutos más al momento de realizar los respaldos y checar la integridad de la grabación. Los comandos empleados para respaldar suelen tener opciones que efectúan esta labor.

- Antes de realizar un respaldo completo es ampliamente recomendado checar, y en dado caso, reparar cualquier inconsistencia en el SA. Para ello se emplea el comando *fsck*.

Éstas son algunas y las más importantes recomendaciones que se deben tomar con el cuidado de cintas y, dependiendo de la importancia que la información merece para la empresa, se pueden extremar o simplificar.

Por otro lado, aunque las cintas son el medio más ampliamente difundido y empleado para realizar respaldos, también pueden utilizarse los discos flexibles cuando se trate de poca información. Estos permiten de una manera simple y sencilla, a los usuarios respaldar y restaurar sus archivos, o al administrador información valiosa como los archivos principales del sistema: */etc/hosts*, */etc/passwd*, */etc/group* y archivos de licencias entre otros.

También suelen ser empleados para esta labor, unidades de discos ópticos, que representan un medio más rápido, aunque un poco más caro; o incluso discos duros que permiten mantener copias de la información. En este último rublo, en la actualidad se cuentan con sistemas y utilerías que permiten mantener la misma información en varios discos a la vez; para en caso de que falle uno, se active la copia alternativa y el servicio no se suspenda; claro está, que el costo de implantación de estas técnicas es considerablemente elevado, pero dependiendo de las necesidades e importancia, vale la pena considerar.

Como resumen: no importando el medio empleado, es indispensable tener un cuidado y mantenimiento sobre el medio elegido para evitar pérdidas de información.

V.V.III. Estrategias de respaldo

Una estrategia de respaldo es la técnica, método o procedimiento empleado al realizar los respaldos. Es importante establecer una estrategia, y mucho más, el que se lleve a cabo. Existen diversas estrategias ampliamente difundidas, como la conocida como "La del Abuelo", que consiste en utilizar tres juegos de 7 cintas cada uno. El primer juego es utilizado para realizar un respaldo diario de cada día de la semana, que en este momento representa el respaldo actual o hijo. La segunda semana se utiliza el segundo juego, que ahora ocupa el lugar del respaldo actual y el respaldo anterior se convierte en el padre; porque fue efectuado una semana antes. La semana siguiente se utiliza el tercer juego, el cual se convierte en el respaldo actual; el de la semana anterior en el padre y el de hace dos

semanas en el abuelo. Al terminar este recorrido, se empieza a utilizar el primer juego de cintas, el que era el abuelo, para convertirse en el respaldo actual o hijo y asi sucesivamente. De tal forma que siempre se tiene un respaldo actual, uno de hace una semana y el de dos semanas atrás.

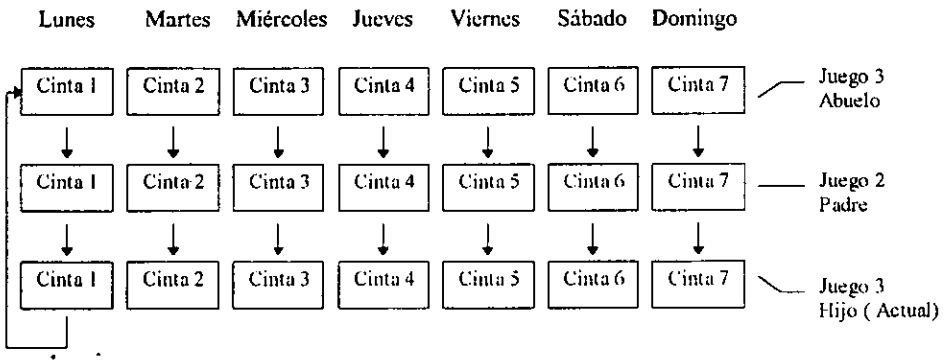


Fig. V-1 Estrategia de respaldo del Abuelo (3 Juegos de 7 cintas cada uno).

De este hecho resalta la importancia de tener no sólo el respaldo actual de la información; sino el de almacenar por algún tiempo, los respaldos pasados. Dependiendo de la importancia de la información; de la cantidad que se respaldará; de lo frecuente que ésta pueda sufrir cambios, es la estrategia que se debe utilizar. Nadie más que el administrador que conoce las características de su equipo, puede tomar la mejor decisión al elegir una estrategia de respaldo; pero siempre es importante almacenar, por lo menos, dos juegos de ella. Por ejemplo, si los datos del sistema no son demasiados y su contenido generalmente no varía, se pueden realizar respaldos completos cada mes y tener almacenadas 12 cintas correspondientes a cada uno de los 12 meses; éstos pueden ser rotados cada año. Si la información no cambia con frecuencia, pero es extremadamente importante, se pueden utilizar dos juegos de cintas en las que se llevan a cabo respaldos incrementales cuando se efectúen cambios considerables. Si la cantidad de información que contienen los SA que se respaldarán es demasiado grande, la solución es nuevamente implementar una estrategia a base de respaldos incrementales. En este método se puede realizar un respaldo completo del sistema, que puede ser muy tardado, y se utilizan cintas para realizar respaldos incrementales diarios durante una o dos semanas, que suelen ser muy rápidos. Al término de este período, se utiliza el segundo juego de manera semejante, y al finalizar éste, se rotan las cintas empezando el ciclo nuevamente. En fin, la estrategia seleccionada debe contemplar el tener por lo menos los dos juegos de respaldos anteriores y ser llevada a cabo puntualmente.

Por otro lado, no se debe olvidar el mantener respaldos (copias) de la información contenida en cintas, CD y demás medios de distribución; especialmente de software que se ha adquirido, como el SO, lenguajes de programación y cualquier otra aplicación. De preferencia se deben utilizar las copias durante los procesos de instalación, y guardar en un lugar seguro los originales. Esto es importante, ya que como se mencionó, el tiempo es un factor primordial en la conservación de las cintas, y si no se generan respaldos de ellas, puede darse el caso de que se dañe alguna y se pierda la información y la posibilidad de ser utilizada para instalar la aplicación que contenía nuevamente.

Un caso importante es el de los medios que contienen el software de instalación del SO; ya que éstos contienen una serie de herramientas que le permiten arrancar desde el PROM. Es por ello que es indispensable el generar juegos de cinta que permitan al administrador, en dado caso de sufrir una caída de sistema y no disponer de los medios de distribución originales, el arrancar de cinta para solucionar el problema. Personas que reciben el software en CDROM, suelen omitir este paso pensando que el tiempo no puede dañar el contenido del CD; pero se olvidan que pueden existir otros factores o circunstancias que sí lo hagan, quedando sin la posibilidad de ser utilizados. Por ello, es aconsejable generar este tipo de cintas conocidas como "cintas de boot". Se pueden generar dos tipos de estas cintas: las que contienen una copia fiel de la cinta o CD original y las que contienen las herramientas para arrancar miniroot únicamente.

El software de instalación ocupa bastante espacio, por lo que generalmente es distribuido en CDROM o cintas de gran capacidad. Si se disponen de medios de almacenamiento de baja capacidad, como carretes de cinta o cartuchos, la única alternativa de recibir el software es a través de CD, y por tanto, no será posible generar una copia fiel de la información contenida en él. En este caso es conveniente crear una cinta que permita únicamente arrancar, y poder de esta forma, restaurar respaldos hechos previamente, en caso de que se dañe el SO y no se cuente con el CD original. Para ello se utiliza el comando *distcp*; ya que los programas de arranque se encuentran almacenados en un formato y estructura especial que no cualquier herramienta de respaldo conoce.

 # distcp /CDROM/dista/sa /dev/tape

En este ejemplo la unidad de CD conteniendo el disco de instalación debe estar montada en el directorio *CDROM*, y el archivo de dispositivo */dev/tape* corresponde a la unidad donde se encuentra insertada la cinta. */CDROM/dista/sa* indica que únicamente se copie a *sash*, para poder arrancar desde el PROM y solucionar posibles problemas con esta cinta.

Si se dispone de unidades de cinta de suficiente capacidad se puede realizar una copia de la información completa del medio original. Por ejemplo, si el software fue recibido en cinta y se cuenta con dos unidades de este tipo, el siguiente comando realiza el copiado:

#distcp /dev/nrtape1 /dev/tape2

La cinta original debe ser insertada en el dispositivo */dev/nrtape1* y la cinta donde se efectuará la copia en */dev/tape2*. Si solamente se cuenta con una unidad, se debe utilizar como intermediario un archivo en disco para pasar la información de una cinta a otra; es decir, se copia la información de cinta a disco, se coloca la nueva cita y se copia de disco a cinta:

 # distcp /dev/nrtape /tmp/dist
 # distcp /tmp/dist/* /dev/tape

Finalmente, si se recibe en CD y se cuenta con una unidad de cinta, y claro está, el CDROM, se puede utilizar el siguiente comando:

 # distcp /CDROM/dist/* /dev/tape

Esto representa una seguridad adicional para el administrador que permite garantizar una rápida respuesta a problemas que se presenten durante la vida del equipo. La Tabla 17 muestra un sumario de las recomendaciones para efectuar un respaldo. Las herramienta que pueden ser utilizadas son descritas en el próximo punto.

Tabla 17 Tipos de respaldos

Tipo	Cuándo realizarlo	Archivos a respaldar
Completo	Después de configurar y acondicionar el sistema a las necesidades propias. Antes de instalar cualquier software adicional. Después de haber probado que un paquete instalado funciona adecuadamente sin causar perjuicios. Antes de realizar cualquier mantenimiento sobre el equipo; especialmente si atañe al disco. Cuando los cambios hechos al sistema sean considerables, desde la última fecha del respaldo.	Todo el Sistema de Archivo. (root, usr y cualquier otro SA)
Parcial	Semanal o quincenalmente; según necesidades	*/usr* ; Archivos de Usuarios
Incremental	Diariamente.	Todos los cambios
Usuario	Cuando así lo requiera cada uno.	Archivos del usuario.

Es importante resaltar que cuando se haga un respaldo completo, no debe desecharse o utilizarse la cinta del respaldo completo anterior; es conveniente tener varios juegos de versiones pasadas, para de ser necesario, recurrir a alguna de ellas.

V.V.IV. Herramientas de respaldo

Una vez cubiertos y analizados los temas anteriores sólo resta describir las posible herramienta o comandos de que dispone el sistema para efectuar los respaldos. Éstas las podemos clasificar en dos grandes grupos:

- Orientada a SA.
- Orientadas a Archivos.

Las primeras están diseñadas para respaldar SA completos, por lo que utilizan técnicas para agilizar la operación, evitando detalles que pudieran retardar el proceso. Por tal motivo, herramientas de este tipo no tienen implementados mecanismos, o son muy complejos y rígidos, para bajar[68] uno o varios archivos específicos del respaldado; fueron diseñadas para bajar el SA completo. Las segundas permiten grabar archivos en forma individual; por lo que guarda características propias de cada uno, como su ruta (path) o como el colocar un encabezado al inicio de cada archivo grabado que permita su identificación y facilite el proceso para localizarlo. Esto permite que en determinado momento, se pueda bajar de cinta sólo uno o varios archivos, de todo el conjunto almacenado en ella; claro está que el proceso de grabado es un poco más lento.

Existe un tercer grupo de programas que llamaremos orientados a bit y que generan copias imágenes del original. Éstos son mucho más rápidos, ya que leen los sectores de un medio y los copian a otro sin importar su contenido; es decir no lo interpretan. Las herramienta anteriores sí interpretan el contenido de los discos que se grabarán; ya sea en menor (orientadas a SA) o mayor grado (orientadas a archivos), por lo que leen la estructura del disco, de los directorios , de los inodos, etc., para localizar la información que respaldarán, y cuando la restauran, leen la información de cinta y la colocan en el lugar especificado, para lo cual tiene que analizar nuevamente la estructura del SA y poder colocarla en el bloque de datos adecuado. En cambio la orientada a bits, lee un sector de datos y lo copia en otro medio sin saber cual es realmente la información que contenía.

Los detalles mencionados, hacen que las herramienta orientadas a bit sean utilizadas para generar copias fieles de un disco a otro medio (disco o cinta). Estas copias pueden ser del disco completo o parciales, lo que permite el poder respaldar particiones individuales. Si se trata de otro disco, éste debe ser de las mismas características físicas, sectores, cabezas, pistas, etc. para que no exista conflicto. En épocas anteriores en donde las velocidades de transmisión eran bajas y los procesos para respaldar eran muy, pero muy tardados, estas herramientas representaban una gran alternativa debido a la velocidad para copiar que se obtenía al no interpretar la estructura en que se encontraba grabada; aunque su limitante era que se tenía que bajar el respaldo en forma íntegra.

[68] bajar es un término utilizado para indicar que se copiará información contenida en una cinta hacia un disco.

Con los adelantos y la llegada de nuevos medios que nos permiten tener tiempos de acceso cada vez menores y altos desempeños en los procesos de lectura y escritura de los discos, así como las velocidades a que trabajan los CPU y demás tecnología, las herramienta orientadas a bit, podríamos decir que han desaparecido por lo rígidas que son (bajar todo o nada), y se encuentran en operación las orientadas a SA y Archivos.

Cuando surgieron estas herramientas, sí existía una diferencia entre estas tres clases, pero por las mismas causas anteriores, en la actualidad podemos encontrar aplicaciones de respaldo orientadas a SA que también permiten respaldar archivos en forma individual y viceversa. En el caso de IRIX las herramienta de que dispone son:

Tabla 18 Tipos de comandos para respaldar

Comando de RESPALDO	Comando para RESTAURAR	Orientado A
System Manager Backup	Completo: PROM Parcial: System Manager Backup	Sistemas de Archivos (Ambiente Gráfico)
bru	bru	Sistemas de Archivos
Backup	Restore	Sistemas de Archivos
dump	restore	Sistemas de Archivos
cpio	cpio	Archivos
tar	tar	Archivos
dd	dd	bit o fisico

Una vez comprendido estos conceptos analizaremos algunos comandos de que se puede disponer en el SO IRIX, indicando sus caracteristicas, ventajas y desventajas así como ejemplo de su uso.

V.V.IV.I. System Manager Backup and Restore Tool

Esta herramienta está diseñada para trabajar en el ambiente gráfico de IRIX. Es muy simple e intuitiva; lo que permite que su manejo sea sencillo. Está diseñada para respaldar SA completos; pero también puede trabajar sobre archivos individuales.

Al ser ejecutada presenta una pantalla en la que se puede seleccionar, tanto el tipo de operación (respaldo o restauración de archivos), como el dispositivo, de los disponibles, que

se utilizará para efectuarlo; ya sea local o remoto. Una vez seleccionado, presenta otra pantalla en la que se puede indicar lo que se desea respaldar y efectuar la operación con un simple click. Si se desea respaldar un directorio o un archivo, basta con escribir su ruta o arrastrar y depositar en él, el icono de la carpeta que se desea respaldar y dar un simple click en el botón de backup.

Esta herramienta es la recomendada por SGI para efectuar respaldos, ya que fue creada para optimizar los recursos del sistema y, de esta forma, obtener un mejor rendimiento. También dentro de sus características se encuentra que:

- Efectúa operaciones de chequeo de la información; por lo que requiere aproximadamente un 20% más de espacio para respaldar cierta información.

- Es la única que permite crear una cinta de recuperación para ser utilizada desde el PROM durante la restauración o rescate el sistema, si éste ha sufrido un daño irreparable que imposibilite el arranque del equipo. Para ello se debe realizar un respaldo completo, indicado al dar como ruta de lo que se respaldará, el directorio raíz (/).

- Está basada sobre el comando *bru*; podríamos decir que es una interfaz gráfica para la utilización del comando *Backup* que es una interfaz del comando *bru*; por lo que

- Reconoce el formato de cintas creadas con *bru* y *Backup*. Es posible restaurar información respaldada con estas herramientas.

- La restauración de archivos puede efectuarse hacia diferentes directorios; no necesariamente a su directorio original.

V.V.IV.II. bru

Es una herramienta diseñada para respaldar SA que presenta mejoras significantes sobre las demás (*tar*, *cpio*). Si se está trabajando en un equipo que no posea capacidades gráficas para poder utilizar el *System Manger Backup*, ésta es la herramienta recomendada (junto con la de *Backup* por su sencillez). Entre sus características, además de las comunes se encuentran:

- Permite la compresión de archivos (opción -*Z*).
- Permite checar la integridad de los datos de la cinta.
- Calcula el espacio requerido para efectuar un respaldo (opción -*e*); por lo que de antemano podemos determinar cuántas cintas se requieren, y no presentarnos ante el hecho de que varias horas después de iniciado el proceso, es necesario introducir otra cinta que ya no tenemos.

- Permite el respaldo o restauración de archivos basados en las fechas de su última modificación (incremental).
- El restaurar los archivos que han sufrido algún cambio únicamente.
- Permite respaldar archivos individuales.
- Permite respaldar archivos especiales de tipo carácter o bloque, ligas simbólicas, archivos tipo fifo y cualquier otro archivo.
- Si se especifica un guión (-), los archivos a respaldar son leídos del dispositivo estándar de lectura; lo cual le permite ser usado junto con comandos como *find*, para formar estructuras que permitan seleccionar archivos que cumplan con cierto patrón.
- Posee varios niveles de reporte, que permiten obtener información detallada del proceso.

Asume que el dispositivo de respaldo será el de */dev/tape*; por lo que si éste no es, es conveniente crear una liga de éste, hacia el dispositivo que sí lo sea. Por ejemplo:

 # ln /dev/mt/tps0d6 /dev/tape

En este caso si se graba algo sobre el dispositivo *tape*, será enviado a *tps0d6*. También debe existir una línea apropiada de */dev/mt/tps0d6* en el archivo */etc/brutab* que puede ser anexada con cualquier editor de texto.

Es importante mencionar que esta herramienta no es estándar, por lo que probablemente no se localice en otras versiones del SO UNIX. Si se deseá exportar a otros SO, es conveniente verificar su existencia en ellos, y de lo contrario, utilizar otra que sí lo sea. Por otra parte, probáblemente no se tengan problemas con versiones posteriores de esta herramienta en IRIX; pero si se trata de versiones de otras plataformas, cabría la pena probar su compatibilidad antes de efectuar respaldos formales.

Un factor importante, es el poder ser utilizada en modo no interactivo, que brinda la posibilidad de programar labores de respaldo automáticamente desde el *cron*[69] o cualquier otro medio. En este modo, cuando se llena la cinta donde esté grabando, es rebobinada y no pregunta ni da la opción para que la cinta sea cambiada, por lo que sobrescribe sobre su contenido. Por tal motivo, cuando se programen respaldos, se debe revisar que su contenido no esté sobre los límites de capacidad de la cinta o lo sobrepasen; ya que se corre el riesgo de perder la información respaldada.

EJEMPLOS

Para respaldar el SA completo:

 # bru -c /

[69] Ver Automatización de labores en pág. V-14

La opción *-c* se utiliza para indicar que desea crear un nuevo respaldo en la cinta. Para respaldar el archivo llamado *resultado*:

 # bur -cv /usr/peolple/proyecto/resultado

La opción *-v* indica que se desplieguen mensajes informativos del proceso en la pantalla. Para crear un respaldo incremental se utiliza la opción *-n*, que permite respaldar archivos modificados a partir de una fecha especifica. El siguiente ejemplo respalda los archivos modificados a partir del 20 de Marzo de 1997 contenidos en el directorio *people*:

 # bru -c -n 20-Mar-97 /usr/people

Para respaldar remotamente en un equipo llamado *jupiter* :

 # bru -cvf guest(@)jupiter:/dev/tape /usr/prog

La opción *-f* permite especificar el dispositivo donde se realizará la grabación; que en este caso se trata de */dev/tape* del equipo *jupiter*. Para efectuar la operación se requiere una cuenta en dicho equipo y de preferencia sin clave secreta; generalmente suele ser *guest*. Cualquiera de los dos ejemplos siguientes, copia todos los archivos existentes perteneciente al usuario *juan*:

 # find / -user juan -print | bru -c -
 # bru -c -o juan /

La opción *-o* especifica que sean los archivos pertenecientes al usuario *juan*. El siguiente, restaura todos los archivos de lenguaje C, contenidos dentro del directorio */usr/people/cmd*:

 # bru -xw '/usr/src/cmd/.c'*

La opción *-w* especifica que se confirme antes de bajar cualquier archivo; por lo que aparecerán mensajes preguntando si se desea bajar o no cada uno. Para obtener un listado del contenido de una cinta:

 # bru -t

Y finalmente para restaurar la información completa de la cinta a su posición original:

 # bru -x

Y un archivo:

 # bru -x /usr/peolple/proyecto/resultado

V.V.IV.III. Backup

Como ya se mencionó, es una interfaz hacia el comando *bru*. Cabe mencionar que cuando se realiza un respaldo completo, además de la información del SA, crea un archivo (*/tmp/volhdrlist*) que almacena un listado del contenido de la partición 8, encabezado de volumen, que también es respaldado para en caso de que se llegue a dañar ésta, se pueda arrancar de cinta y efectuar operaciones de restauración.

EJEMPLOS

Para generar un respaldo completo:

 # Backup /

Cuando se especifica un respaldo completo, *Backup* crea el archivo */etc/lastbackup* donde coloca la fecha en que se realizó el respaldo. Esto permite el poder realizar respaldos incrementales; para lo cual se utiliza la opción *-i*, que especifica que se respalden los archivos modificados desde la fecha indicada en este archivo. Si se desea hacer el respaldo en forma remota, se utiliza la opción *-h* para especificar el equipo:

 # Backup -h jupiter /

Backup asume que existe una cuenta *guest* en la máquina remota y que el dispositivo será */etc/tape*; si es otro, se debe especificar mediante la opción *-t*.

Si se desea restaurar un respaldo hecho con *Backup*, debe utilizarse el comando *Restore*, el cual restaura los archivos sobre el directorio actual, si se utilizó una ruta relativa al realizarlos. Una ruta relativa es aquélla que no empieza con un / al especificarla, por ejemplo: *programa/prog1* es relativa y */programa/prog1* es absoluta; ya que empieza con el directorio raíz.

 # Restore

Para restaurar un archivo o directorio:

#Restore prog1

donde *prog1* es un archivo en este caso.

V.V.IV.IV. dump

Es utilizado para respaldar todos los archivos del sistema a cinta. Junto con este comando se encuentra el de *restore*, que permite recuperar información creada con *dump*; cabe mencionar que *Restore* se utiliza para los que fueron creados con *Backup*. Estos programas fueron introducidos a partir de la versión 4.0 de IRIX, para proveer de flexibilidad y compatibilidad al sistema; ya que suelen encontrarse implementados sobre varias plataformas o versiones de otros fabricantes de SO UNIX.

El comando *dump* permite varios niveles de respaldos incrementales divididos del 0 al 9; donde el nivel 0 respalda el SA completo. La opción *-u* le indica a *dump* que después de efectuar adecuadamente el respaldo, grabe la fecha, el nivel y el SA que respaldó en el archivo */etc/dumpdates*. Esto permite llevar el control de los respaldos incrementales; ya que cada nivel respalda todos los archivos modificados a partir de la fecha del último respaldo de menor nivel. Por ello, es importante el incluir esta opción al efectuar respaldos incrementales, ejemplo: Si se realiza un respaldo de nivel 0, contendrá todo el SA; si posteriormente se efectúa uno de nivel 2, se respaldará cualquier modificación a partir de la fecha del último respaldo de nivel inferior, el de 0; si después se realiza uno de nivel 6, contendrá las modificaciones desde la fecha cuando se realizó el de menor nivel, el de 2; y si posteriormente se efectúa otro de nivel 4 contendrá todas las modificaciones desde cuando se hiso el de nivel 2, es decir cubrirá también lo respaldado por el nivel 6 anterior, ya que éste es de mayor nivel.

Una estrategia de respaldo sencilla de 4 semanas, consisten en que cada inicio de ciclo se efectúe un respaldo completo, posteriormente de lunes a sábado, efectuar respaldos diarios que guarden los cambios realizados en el día, y cada domingo, efectuar un respaldo que guarde los cambios efectuados de toda la semana.

	Inicio	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Semana 1	0	3	3	3	3	3	3	2
Semana 2		3	3	3	3	3	3	2
Semana 3		3	3	3	3	3	3	2
Semana 4		3	3	3	3	3	3	2

El de nivel 0 del inicio, respalda todo el SA; el de nivel 3 efectuado el lunes, los cambios hechos desde que se efectuó el de 0; el del martes, los cambios hechos desde el lunes; el del miércoles, los efectuados desde el martes y así hasta el sábado, que guarda los cambios efectuados desde el respaldo del viernes. El respaldo de nivel 2 efectuado el domingo, respalda todos los cambios realizados desde el último de menor o igual nivel, por lo que respaldó todo lo efectuado en esa semana. Esto se aplica para cada semana del ciclo.

Dentro de las opciones destaca la de *-c*, que debe ser utilizada para especificar que el respaldo será hecho sobre un dispositivo que soporta cartuchos; de lo contrario asume que maneja carretes[70]. La de *-f* que permite especificar el tipo de dispositivo donde se efectuará el respaldo; el default es */dev/tape*, que puede ser cualquier dispositivo físico. Dentro de esto, es importante mencionar que cuando se utilicen varias opciones que requieren parámetros, el orden de las opciones debe concordar con el de los parámetros; el ejemplo 3 lo ilustra.

Este comando es muy dependiente del tipo de medio donde se grabará la información; por lo que se debe tener cierto cuidado al utilizarlo (ver el segundo ejemplo). Si el respaldo será utilizado localmente, generalmente no se tienen problemas; pero si será portado a otras plataformas, es conveniente el estudiar las demás opciones que permiten controlar tanto la densidad, factor de bloqueo, etc., que pueden en algún momento no concordar y, por tanto, deben ser especificados directamente.

restore permite restaurar el respaldo completo o archivos específicos. Cuenta además con un modo interactivo, que permite listar y seleccionar los archivos que se desea restaurar.

EJEMPLOS

Para realizar un respaldo completo a un cartucho:

 # dump 0uc /

Para realizar un respaldo incremental de nivel 2 del SA */usr* sobre un dispositivo QIC-150; que soporta cartuchos de alta densidad (18 pistas y 600 pies de largo, por ejemplo):

 # dump 2ucs 10800 /usr

el valor de 10800 sale del hecho de que la cinta es de 600 pies y tiene 18 pistas; es decir que la longitud total es 18 x 600 = 10800. Si no se especifica, *dump* asume una longitud total de 5400, que suele ser para dispositivos de 9 pistas y 600 pies de largo como las QIC-24 de baja densidad[71]. Si se tiene dudas al respecto se puede consultar la ayuda mediante el comando *man*. Si el archivo de dispositivo es otro, se debe especificar:

 # dump 2ufsc /dev/mt/tps0d6 10800 /usr

[70] ver Medios de Respaldo en pág. V-50

[71] Cuando se habla de la capacidad de un cartucho, se suele describir en términos de su longitud física medida en pies (600 ft), o mediante el término de ftpi (Flux Tracks Per Inch) que especifica el largo total, que es igual a todas las pistas del cartucho unidas consecutivamente y expresado en pulgadas; 10800 en el ejemplo anterior.

Para restaurar un respaldo completo:

 # restore -x

Para restaurar sólo un archivo, éste debe ser especificado:

 # restore -x ./etc/group

Se puede entrar al modo interactivo para ver el contenido de una cinta respaldada con *dump* y poder restaurar archivos en forma individual; para entrar en este modo:

 # restore vi

V.V.IV.V. tar

Es una herramienta orientada hacia archivos específicamente. Permite salvar y restaurar múltiples archivos o directorios en un sólo archivo; o en cintas. Si se especifica un directorio, son respaldados todos los archivos y subdirectorios que contiene en orden alfabético. Por estas características, además de ser utilizada para generar respaldos de información, suele ser empleada para facilitar la distribución de archivos, gráficas, programas, en fin, datos sobre una red. En este sentido, se utiliza para almacenar en un sólo archivo, todo el conjunto de programas que forman una aplicación determinada. A este archivo se le suele colocar la extensión *.tar*, el cual generalmente suele ser compactado por herramientas para reducir su tamaño y facilitar su transporte, quedando algo como *archvio.tar.Z* dependiendo de la herramienta utilizada para compactar. Cuando es trasladado al lugar deseado, se suele descompactar; posteriormente se utiliza el comando *tar* para extraer las piezas de la aplicación contenidas en él.

Es una herramienta que puede ser considerada con cierto grado de dificultad por algunos, debido a las diferentes opciones que posee. Por otro lado, suele ser muy utilizada por administradores expertos ya que fue una de las primeras y, por tanto, se encuentra implementada en una gran variedad de plataformas y versiones de UNIX, por no decir que todas.

Uno de los aspectos que puede ser considerado como complicado es el del factor de bloqueo y la capacidad para grabar varios respaldos, archivos, sobre una misma cinta. En cuanto al factor de bloqueo, si no se especifica *tar* determina cuál es el adecuado para poder formar los registros donde se grabará la información. Si se trata de una cinta, utiliza llamadas para determinar este parámetro; si se trata de los dispositivos estándar de lectura y escritura (*stdio* y *stdout*) utiliza un factor de bloque de 1; y si se trata de un archivo o cualquier dispositivo que no pueda soportar las llamadas que realiza, utiliza un factor de 20, el cual causa que el proceso de lectura ignore los límites de cada registro y los concatene y tome

como una sola secuencia de datos para ser grabados. Si los archivos respaldados serán restaurados en el mismo equipo, no se presentará ningún problema en este aspecto; pero si serán restaurados en otro, puede ser que la unidad que se utilice no soporte dicho factor de bloqueo, por lo que no lo pueda determinar automáticamente, en cuyo caso, se tendrá que especificar mediante la opción *-b*. Si se desea determinar el factor de bloqueo de algún dispositivo, se puede utilizar el comando:

 # mt blksize

Es importante colocar estos datos en la etiqueta de la cinta para evitar contratiempos. Por otra parte, cuando se utiliza el comando *tar* para respaldar un conjunto de archivos en una cinta, los graba como un sólo registro. Si posteriormente se anexa otro conjunto de archivos al final mediante la opción *-r*, se tendrá que físicamente en la cinta se encuentran dos registros. Mediante este proceso, se puede seguir agregando respaldos a la cinta sin borrar los existentes. Posteriormente si se desea restaurar el segundo, por ejemplo, la cinta debe ser avanzada al inicio del segundo registro; es decir, leer el primero pero no restaurarlo. Cuando se coloca la cinta en el lugar adecuado, se procede a leer la información, bajando todos los archivos que fueron almacenados en el segundo respaldo. El ejemplo 6 muestra cómo leer los dos primeros únicamente; y el ejemplo 7, cómo leer el 5°. Aunque parece un poco complicado, con la práctica se puede adquirir experiencia en su manejo.

EJEMPLOS

Para respaldar los archivos del directorio */usr/people*:

 # tar -cv /usr/people

La opción *-v* especifica que se desplieguen mensajes indicando los archivos que son grabados; la *-c* indica que se desea crear un nuevo respaldo. Por default utiliza el dispositivo */dev/tape* para grabar la información. Si se especifica como nombre de un archivo un guión (-) *tar* lee los archivos a respaldar del dispositivo de lectura estándar; lo que permite que sea utilizado en conjunción con otros, como se ilustra en el siguiente ejemplo:

 # tar cv 'find /usr/people -name proyecto'
 # find /usr -mtime 7 -type f -print | tar cvf -

En el primer comando, *find* localiza todos los archivos llamados *proyecto* dentro del directorio */usr/people*; los cuales son pasados a *tar* para que sean respaldados. En el segundo, *find* localiza todos los archivos que han sido modificados desde 7 días atrás. Estos archivos son impresos, y ya que se encuentra unido a *tar* median un pipe (|), le son pasados para que sean respaldados. Para respaldar el directorio */usr* en forma remota a un dispositivo (*/dev/tape*) de cinta colocado en la máquina *jupiter*:

 # tar -cvf guest@jupiter:/dev/tape /usr

Para listar el contenido de una cinta:

 # tar -t

Para restaurar un archivo:

 # tar -xv prog1

Si se desea restaurar los dos primeros archivos (registros) de la cinta:

 # tar xvf /dev/nrtape
 # tar xv /dev/tape

El primer comando restaura el primero utilizando el dispositivo *nrtape*. Por convención, el nombre de este dispositivo lleva una ***nr*** que indica No- rewind; que no se regrese o rebobine la cinta después de ejecutar el comando; por lo que la cinta permanece en su lugar y al ejecutar el siguiente comando, se restaura el segundo, y posteriormente sí se rebobina la cinta; ya que como no se indicó, se utiliza el dispositivo normal *tape*. Si se desea restaurar el quinto archivo de la cinta y está rebobinada; en su punto inicial:

 # mt fsf 4
 # tar xv

El primer comando le indica que se salte los primeros cuatro registros; por lo que el siguiente restaura el quinto archivo.

V.V.IV.VI. cpio

El comando *cpio* (**c**opy **i**n **o**ut) es orientado hacia archivos; por lo que permite respaldar directorios y archivos a cinta o disco. Básicamente posee tres funciones que son mutuamente excluyentes:

-i Que se utiliza para restaurar información copiada previamente a cinta: Le indica a *cpio* que extraiga archivos del dispositivo estándar y que copie únicamente los que coincidan con un patrón especificado; el default es *, todo.

-o Que se utiliza para respaldar: Le indica a *cpio* que lea del dispositivo estándar de entrada, una lista de rutas; para que el contenido de dichos archivos, sea copiado al dispositivo estándar de salida. En esta copia se agregan encabezados e información estadistica.

-p Que se utiliza para restaurar archivos condicionalmente; es decir, que son leídos del dispositivo estándar de entrada, una serie de rutas que definen los archivos que serán creados y copiados si cumplen ciertas condiciones especificadas mediante parámetros adicionales.

Son por estas razones que este comando suele formar parte de expresiones compuestas, en las que se utiliza el pipe (|) o redireccionamientos como < y >, para conectar los comandos que definen qué respaldar o bajar, y los dispositivos de donde se realizará el proceso. Esto se puede apreciar mejor en los siguientes ejemplos.

EJEMPLOS

Para respaldar todo el contenido de un directorio:

 # cd /usr/people
 # find . -type f -print | cpio -ovBc > /dev/tape

El primer comando coloca al usuario en el directorio */usr/people*. En el segundo, el comando *find* lista el contenido de todos los archivos ubicados dentro de ese directorio; los cuales le son pasados mediante el dispositivo de entrada estándar a *cpio*, ya que se encuentran conectados mediante un pipe (|). La opción *-o* de *cpio*, le indica que lea el dispositivo de entrada estándar para obtener una lista de rutas, o sea, lo que despliega el comando *find*, para copiar dichos archivos al dispositivo estándar de salida, que está redireccionado a */dev/tape* mediante el signo > . La opción *-v* indica que se desplieguen mensajes informativos del proceso en la pantalla, y la *-B* establece un bloqueo de 5120 bytes; es decir, que sean tomados 5120 bytes del dispositivo de entrada como un registro de salida en la grabación (ver el comando *tar*). La opción *-c* indica que sea leida y grabada la información del encabezado en formato ASCII, que brinda una mayor portabilidad; es recomendable siempre utilizar esta opción. Para copiar el archivo *programa* únicamente:

 # cat programa | cpio -o >/dev/tape

Para copiar los archivos que han sido modificados de hace 15 días a la fecha del directorio */usr*:

 # find /usr -mtime 15 -depth -print | cpio -oO /dev/tape

La opción *-O* especifica que sea direccionada la salida al siguiente parámetro, en lugar del dispositivo estándar; en este caso se trata de */dev/tape*, pero puede ser un equipo remoto, por ejemplo: *guest@jupiter:/dev/tape*. Para restaurar un archivo llamado *proyecto* de cinta a disco:

 # cpio -id proyecto < /dev/tape

La opción *-i* hace que *cpio* extraiga archivos del dispositivo de entrada, que en este caso se encuentra conectado mediante el símbolo < a */dev/tape*, y que sólo los archivos que concuerden con el patrón sean seleccionados para ser copiados o restaurados, en este caso *proyecto*; puede ser utilizada cualquier expresión encerrada entre comillas (" "). La opción de *-d* le indica que cree los directorios que sean necesarios para colocar el archivo en su lugar. Para restaurar archivos de una cinta colocada en el equipo remoto *jupiter*

 # cpio -ivc -C1024000 -I guest@jupiter:/dev/tape

Aquí la opción *-I*, le indica que tome el dispositivo remoto *guest@jupiter:/dev/tape* como dispositivo de entrada en lugar del estándar, para leer los archivos a restaurar. Como no se indicó ningún patrón, se asume que es *; por lo que son restaurados todos los archivos contenidos en la cinta. Para llevar a cabo este proceso, establece un buffer de 1024000 bytes de largo.

V.V.IV.VII. dd

Aunque se utiliza ampliamente para realizar copias literales (bloques de datos) de un medio a otro, fue diseñada para realizar conversiones de formato sobre los datos al momento de copiarlos. Por default lee y escribe hacia los dispositivos de lectura y escritura estándar (*stdin* y *stdout*), pero se pueden especificar otros mediante el uso de las opciones *if* y *of* respectivamente como se podrá apreciar en los ejemplos. Entre las ventajas que posee se encuentran:

- Poder cambiar el tamaño de los bloques durante el proceso de lectura y escritura. Esto permite realizar las operaciones con el máximo desempeño de los dispositivos tipo carácter (raw), cuando son utilizados.
- Copiar un número específico de bloques.
- Realizar conversiones sobre los datos, entre otros.

Cuando se hace una copia literal de un disco, ésta es una imagen del original. Cuando es restaurada, la información queda intacta; es decir, que si tenía inconsistencias el SA al momento de efectuar el respaldo, o la información se encontraba muy fragmentada, ésta quedará de igual forma al restaurarla. Esto no sucede cuando se realiza un respaldo lógico de archivos o directorios; donde cada archivo, que pudiera estar fragmentado, es leído para ser almacenado en un sólo bloque continuo de datos a cinta. Por tal motivo, cuando es restaurado, si hay suficiente espacio en disco, éste puede quedar en un sólo bloque de datos.

EJEMPLOS

Para crear una copia imagen del SA */usr*, cuyo archivo de dispositivo que utiliza es */dev/usr*:

 # dd if=/dev/usr of=/dev/tape

La opción *if* especifica que sea tomado como dispositivo de entrada, el archivo especial */dev/usr*; la de *of*, que sea tomado como salida el de */dev/tape*. Para realizar una restauración de los datos, basta con invertir los parámetros de las opciones.

V.V.V. Recomendaciones finales

Aunque algunas de las herramientas son portables, ya que se encuentran difundidas entre los diversos SO UNIX, no siempre será posible leer una cinta respaldada en otro equipo; ya que también interviene su densidad y tamaño. Como se puede observar de la Tabla 16, pág. V-50 existen distintos dispositivos que manejan diversas capacidades y densidades de cintas: baja, normal y alta densidad, con compresión de datos o no, etc. Es por ello que es extremadamente importante el colocar en la cinta el comando con el cual fue grabada la información, el tipo de cinta y características adicionales que permitan, en un momento dado, facilitar el proceso de restauración de la información.

La Tabla 19 muestra una serie de recomendaciones en la utilización de los diversos comandos para efectuar respaldos.

Tabla 19 Recomendación de uso de herramienta de respaldo

HERRAMIENTA	UTILIZACIÓN
System Manager Backup Tool	Si los respaldos serán utilizados únicamente en el equipo o portados a otros que trabajan con el SO IRIX, ésta es la que se recomienda ampliamente para cualquier tipo de respaldo. Se requiere ambiente gráfico para funcionar; si no se posee éste, utilizar las siguientes.
Backup	Si no se es experto, ésta es la recomendada por su sencillez y ventajas que ofrece. Si se trata de un respaldo completo que permita arrancar de cinta, ésta es la única que lo hace. Trabaja sobre SO IRIX.
bru	Si se requiere de mayor flexibilidad que Backup, ésta es la ideal. Realiza cualquier tipo de respaldo, y fue diseñada para optimizar los recursos de IRIX.
dump	Si se requieren realizar respaldos incrementales con cierto grado de portabilidad ésta es la adecuada. Está enfocada para este tipo de respaldos; por lo que es fácil de usar.
cpio	Para respaldar archivos o copiarlos de un SA a otro, ésta es la elegida; ya que el comando cp no lo puede hacer. Posee portabilidad entre diversas versiones de UNIX.
tar	Si se requiere portabilidad entre diversos SO UNIX, ésta es la que debe elegirse. Contiene bastantes opciones que amplían su eficiencia y complejidad.
dd	Es ideal para realizar copias imágenes y durante el cambio de tamaño de bloque así como la conversión de formatos. Es ampliamente difundida entre los SO.

Para mayor referencia de cualquiera de los comandos mencionados en este capítulo, referirse a la ayuda mostrada por el comando *man*.

CONCLUSIONES

CONCLUSIONES

De lo expuesto en la presente tesis se puede concluir lo siguiente:

- Las labores principales del administrador son: instalación, administración de los recursos, mantenimiento y seguridad del equipo; todos ellos enfocados tanto al hardware como al software.

- Debido a la forma de trabajar del SO IRIX así como lo delicado y costoso del equipo, la colocación de un UPS que proteja y alargue su vida, proporcionando energía eléctrica constante y regulada, es indispensable.

- La administración de un equipo es una labor cotidiana sobre la cual se debe llevar un estricto control a fin de mantener en óptimo estado su eficiencia, así como prevenir y en su caso, solucionar cualquier falla en el menor tiempo posible. Para lograrlo, se debe contar con la ayuda de tres documentos que son indispensables en todo equipo: la bitácora, libro de procedimientos y las políticas de uso.

  La importancia de cada uno así como su eficiencia se ven limitados por el rigor conque sean respetados y llevados a la práctica; es decir, de nada sirve contar con una bitácora si en ella no se registran tanto los eventos y fallas que sufra el sistema como las soluciones a cada uno, que nos permitan en un futuro, solucionar cualquier problema rápida y eficazmente; ni el tener un conjunto de políticas y reglas que controlen y rijan el uso del equipo, brindando permisos y restricciones, obligaciones y derechos, responsabilidades y seguridad, si no son llevadas a acabo; o el no plasmar en papel los procedimientos esenciales que permitan facilitar el uso del equipo en situaciones críticas, como puede ser la ausencia del administrador o cualquier otro imprevisto.

- Existen labores que pocas veces se requiere efectuar (instalación del SO, afinación del sistema, configuración del kernel, ...), pero cuando es necesario realizarlas, hay que poseer los conocimientos fundamentales que nos permitan entender qué es lo que estamos haciendo, en qué punto del proceso nos encontramos y qué consecuencias traerán consigo las acciones que tomemos. Por ello, la capacitación es un punto importante que no hay que pasar por alto, y de hacerlo, puede acarrear pérdida de información, tiempo y dinero; ya que existen una gran cantidad de métodos y herramientas que se emplean para la gran diversidad de servicios que se brindan, y desafortunadamente, no existe una que lo haga todo.

- Las listas de correo brindan un gran servicio al permitir estar en contacto y compartir experiencias con otros administradores, dando soluciones más ágiles a los problemas. Claro está, esto no reemplaza a la capacitación; sino que la fortalece.

- Un acción adecuada y a tiempo, permitirá detectar, evitar y solucionar fallas antes que puedan terminar en tragedias. Por ello se debe establecer un seguimiento continuo y

calendarizado sobre los diversos puntos críticos que requieren mantenimiento, dando como resultado, un sistema sano. El mantenimiento debe cubrir aspectos como el hardware, el SO, el SA, los datos de usuarios, procesos, respaldos, rendimiento del equipo, la seguridad, etc. ; todo ello se debe documentar.

- Si hacemos una comparación con los elementos de la vida diaria, podemos decir que los respaldos son algo así como un seguro de vida; ya que al igual que ellos, durante el transcurso de la vida útil del equipo hay que estarlos efectuando, siguiendo una calendarización rigurosa y deseando nunca llegar a necesitarlos. De presentarse esa situación, representan un alivio; ya que son los que nos permitirán restaurar la información del equipo minimizando o nulificando las perdidas, dependiendo de la estrategia seguida (respaldos diarios, semanales, mensuales, etc.) y lo riguroso de su seguimiento. Si uno se confía y no los efectúa, ya sea por considerar que el equipo se encuentra físicamente en óptimas condiciones, porque el SO ha funcionado y funciona perfectamente, por considerarlo una perdida de tiempo, por subestimar la confiabilidad del suministro de energía eléctrica y aparatos relacionados, etc., al presentarse cualquier imprevisto que afecte la integridad de uno, varios o incluso todo el Sistema de Archivos y que no pueda ser reparado, será el final; ya que nos tendremos que enfrentar a la tragedia que implica la perdida de la información así como al enojo de los usuarios afectados. Ya lo dice un dicho "Aquellos que no respalden su información, estarán condenados a reescribirla ...". Por todo ello, los respaldos son otro punto importante sobre el que el administrador debe poner interés y verificar su cumplimiento, ya que la información de los usuarios es lo más valioso; cualquier otro software en determinado momento, se puede volver a instalar.

- La seguridad es un campo extenso y controversial, en el cual, los expertos en él tardan más en reparar y crear parches para solucionar problemas y deficiencias que, en que los crackers encuentren otros; debido en parte a la gran variedad de servicios, sobre todo en el ámbito de redes, que el SO UNIX brinda. Por ello, expertos como Eugene Spafford dicen, en su frase tan conocida en el mundo de la seguridad:

  "El único sistema totalmente seguro es aquel que está apagado, desconectado, guardado en una caja fuerte de titanio, encerrado en un bunker de concreto, rodeado por gas venenoso y cuidado por guardias armados muy bien pagados. Aún así, no apostaría mi vida por él."

  Por ello, el administrador tiene que poner en una balanza el nivel de seguridad deseado así como los servicios y datos que desea proteger, y del otro, el tiempo requerido o que está dispuesto a otorgar a ello, para encontrar el punto de equilibrio que permita disminuir la vulnerabilidad del equipo sin afectar otras labores; es decir, no olvidarse del tema y dejarlo a la deriva, ni obsesionarse con él y dedicarle el 100%, ya que por lo expuesto se dice que no hay un sistema totalmente seguro.

- La seguridad es una responsabilidad conjunta en la que cada uno, administrador y usuarios, deben de poner su granito de arena para mantenerla.

Para lograr lo anterior, el administrador de sistemas debe ser una persona integra, capaz y con ética, que administre eficientemente los recurso del equipo (software, hardware y en su caso humanos) a fin lograr los objetivos de la organización donde desempeña sus funciones. De este párrafo resaltan lo siguiente:

- La administración implica que a esa persona se le han conferido los poderes para coordinar, dirigir y controlar los recurso del equipo a fin de garantizar su óptimo desempeño. Esto suele ser confundido por algunos administradores de sistemas, que se consideran los dueños del equipo y en lugar de brindar servicios adecuados, ponen trabas y restricciones innecesarias que degradan su rendimiento.

- Ya que en el equipo se mantienen y procesan distintas clases de información y al administrador se le ha otorgado autoridad sobre él, la integridad y la ética, junto con las políticas establecidas, le indicaran hasta donde llegar y no sobrepasar ese límite, robando, alterando, espiando o cualquier otra acción dañina, sobre los datos e información de los demás, dejando por un lado, los aspectos personales que puedan interferir con su labor.

- Ahora bien, para poder administrar eficientemente los recursos de un equipo, además de conocer adecuadamente las características, capacidades y desventajas tanto del equipo como del sistema operativo y cualquier software o hardware conectado a él, se requiere la chispa que le permita poner en práctica todos sus conocimientos eficientemente. Por ello, la capacitación así como el ingenio y experiencia, son las herramientas que le permitirán alcanzar las metas deseadas.

Finalmente, el administrador debe estar consciente de la responsabilidad que lleva en hombros, y aunque no es una tarea fácil, siempre debe dar su máximo esfuerzo al cumplir con sus obligaciones, para que la gran diversidad de usuarios que pueden existir, tengan la certeza de que los datos e información que procesan, están en las mejores manos.

APÉNDICE A

Consejos Emitidos por SGI

Consejos Emitidos por SGI

Diversas compañías, entre ellas SGI, cuentan con listas de distribución mediante las cuales, se envían mensajes, consejos o avisos a las personas interesadas. En este apéndice se presentarán dos casos, referentes a avisos de seguridad emitidos por Silicon Graphics, en los cuales se indican agujeros en la seguridad que permiten a un usuario normal, bajo ciertas características, adquirir privilegios de root.

Los avisos se encuentran impresos tal y como fueron emitidos (en inglés) para no mal interpretar alguna parte. Están compuestos por varias secciones, como: el encabezado, que describe datos importantes del mensaje (la fecha en que fue emitido, título, número de identificación, etc.); La denominada impacto, que muestra cuáles son las consecuencias que tiene dicha falla sobre la seguridad del sistema; La de solución, que indica cuál es el remedio, permanente o temporal, para erradicar el problema y, si es necesario, se indica cuál es el método adecuado para cada versión de sistema que se ve afectado; La de agradecimientos, y la denominada contactos e información de seguridad, que indica cómo subscribirse a la lista para que le sea enviado cualquier otro mensaje emitido, referente a seguridad entre otros. Esta última sección, muestra sitios de interés que se pueden visitar a través del *web*, o vía *ftp anónimo*, para obtener parches e información referente a seguridad.

Es importante subscribirse a este tipo de listas, si es que se tiene acceso a la red, o utilizar cualquier otro método para mantenerse al día con respecto a problemas de todo tipo, y en especial de seguridad, relacionados con el equipo que se administra.

Silicon Graphics Inc. Security Advisory

Title: Possible Vulnerabilities in systour and OutOfBox
Title: Subsystems for IRIX 5.x, 6.0.x, 6.1, 6.2 and 6.3
Number: 19961101-01-I
Date: November 6, 1996

Silicon Graphics provides this information freely to the SGI user community for its consideration, interpretation, implementation and use. Silicon Graphics recommends that this information be acted upon as soon as possible.

Silicon Graphics will not be liable for any indirect, special, or consequential damages arising from the use of, failure to use or improper use of any of the instructions or information in this Security Advisory.

Recently, potential security vulnerabilities in the OutOfBox and systour subsystems have been advertised in several public forums. Additionally, the Australian Computer Emergency Response Team (AUSCERT) released an advisory (AA-96.08) on this issue.

Silicon Graphics Inc. has investigated the issues and recommends the following steps for neutralizing exposure. It is HIGHLY RECOMMENDED that these measures be implemented on ALL SGI systems running IRIX versions 5.0.x, 5.1.x, 5.2, 5.3, 6.0.x, 6.1, 6.2 and 6.3. This issue will be corrected in future releases of IRIX.

- ----------
- --- Impact ---
- ----------

The Silicon Graphics Indigo Magic System Tour and OutOfBox Experience packages are factory installed on all Silicon Graphics Indy systems.

The Indigo Magic System Tour and OutOfBox Experience packages are not factory installed with any Silicon Graphics Indigo2 systems however, CDs with these packages are provided with the systems.

The OutOfBox Experience subsystem is factory installed on all Silicon Graphics O2 systems. The System Tour subsystem is not part of the software provided for the O2 system.

Note that either or both the Indigo Magic System Tour and OutOfBox Experience subsystems maybe be installed from CD on any Silicon Graphics system.

The purpose of these two packages, systour and OutOfBox, are to demonstrate and highlight the features and capabilities of the user environment and system.

Due to the disk space requirements of these subsystems, most sites will remove these subsystems for disk space reclamation as part of initial system setup. Those sites which have done this will not be vulnerable.

On those systems that the subsystems are still installed on, both subsystems provide background setuid root programs to perform a subsystem removal when a user decides to remove the software. This removal is done using the standard IRIX /usr/sbin/inst program that manages IRIX software.

Provided with the right environment, the inst program could be manipulated to execute arbitrary commands with root privileges.

An account on the vulnerable system is required for exploit. With an account, these vulnerabilities are exploitable by both local and remote access.

- ----------
- --- Solution ---
- ----------

There are no patches for these issues. However, using the information below steps can be taken to eliminate the exposure.

To determine if the OutOfBox and systour subsystems are installed on a particular system, the following command can be used:

```
% versions OutOfBox.sw systour.sw
I = Installed, R = Removed

  Name                    Date     Description

I OutOfBox                11/05/96 OutOfBox Experience, 1.1
I OutOfBox.sw             11/05/96 OutOfBox Experience Software, 1.1
I OutOfBox.sw.complete    11/05/96 Complete OutOfBox Experience
I OutOfBox.sw.intro       11/05/96 OutOfBox Intro Movies

I systour                 02/12/96 Indigo Magic System Tour, 5.2
I systour.sw              02/12/96 System Tour Execution Environment
I systour.sw.eoe          02/12/96 System Tour Execution Environment
```

In the above case, the subsystems of concern are installed and the steps below should be performed. If no output is returned by the command, the subsystems are not installed and no further action is required.

**** IRIX 4.x ****

The 4.x version of IRIX is not vulnerable as the System Tour and OutOfBox Experience subsystems are not part of available software for this IRIX version. No action is required.

**** IRIX 5.x, 6.0, 6.0.1, 6.1, 6.2 ****

There are no patches for this issue.

The steps below can be used to remove the vulnerability by either changing the program permissions (use step 2a) or by removing the subsystems (use step 2b).

1) Become the root user on the system.

```
% /bin/su -
Password:
#
```

2) Choose either step 2a or 2b depending on which has the desired result.

2a) Change the setuid root permissions on the programs of concern.

```
# /bin/chmod u-s /usr/lib/tour/bin/RemoveSystemTour
# /bin/chmod u-s /usr/people/tour/oob/bin/oobversions
```

```
************
*** NOTE ***
************
```

Removing the setuid root permissions from these tools will prevent non-root users from removing the subsystems. Removal of the subsystems will only be possible if the systour or OutOfBox user is a root user or if the inst IRIX software manager is used by root for removal.

2b) Remove the vulnerable subsystems.

```
# /usr/sbin/versions -v remove systour OutOfBox
```

4) Return to previous level.

```
# exit
$
```

**** IRIX 6.3 ****

The IRIX operating system version 6.3 does not have the System Tour subsystem but does have the OutOfBox Experience subsystem.

There are no patches for this issue.

The steps below can be used to remove the vulnerability by either changing the program permissions (use step 2a) or by removing the subsystems (use step 2b).

1) Become the root user on the system.

```
% /bin/su -
Password:
#
```

2) Choose either step 2a or 2b depending on which has the desired result.

2a) Change the setuid root permissions on the program of concern.

```
# /bin/chmod u-s /usr/people/tour/oob/bin/oobversions
```

```
************
*** NOTE ***
************
```

Removing the setuid root permissions from this program will prevent non-root users from removing the subsystem. Removal of the subsystem will only be possible if the OutOfBox user is a root user or if the inst IRIX software manager is used by root for removal.

2b) Remove the vulnerable subsystem.

```
# /usr/sbin/versions -v remove OutOfBox
```

4) Return to previous level.

```
# exit
$
```

- ----------
- --- Acknowledgments ---
- ----------

Silicon Graphics wishes to thank AUSCERT and FIRST members worldwide for their assistance in this matter.

- ----------
- --- SGI Security Information/Contacts ---
- ----------

If there are questions about this document, email can be sent to cse-security-alert@csd.sgi.com.

------oOo------

Silicon Graphics provides security information and patches for use by the entire SGI community. This information is freely available to any person needing the information and is available via anonymous FTP and the Web.

The primary SGI anonymous FTP site for security information and patches is sgigate.sgi.com (204.94.209.1). Security information and patches are located under the directories ~ftp/security and ~ftp/patches, respectively. The Silicon Graphics Security Headquarters Web page is accessible at the URL http://www.sgi.com/Support/Secur/security.html.

For issues with the patches on the FTP sites, email can be sent to cse-security-alert@csd.sgi.com. For assistance obtaining or working with security patches, please contact your SGI support provider.

------oOo------

Silicon Graphics provides a free security mailing list service called wiretap and encourages interested parties to self-subscribe to receive (via email) all SGI Security Advisories when they are released.

Subscribing to the mailing list can be done via the Web (http://www.sgi.com/Support/Secur/wiretap.html) or by sending email to SGI as outlined below.

```
% mail wiretap-request@sgi.com
subscribe wiretap <YourEmailAddress>
end
^d
```

In the example above, <YourEmailAddress> is the email address that you wish the mailing list information sent to. The word end must be on a separate line to indicate the end of the body of the message. The control-d (^d) is used to indicate to the mail program that you are finished composing the mail message.

------oOo------

Silicon Graphics provides a comprehensive customer World Wide Web site. This site is located at http://www.sgi.com/Support/Secur/security.html.

------oOo------

For reporting *NEW* SGI security issues, email can be sent to security-alert@sgi.com or contact your SGI support provider. A support contract is not required for submitting a security report.

Silicon Graphics Inc. Security Advisory

Title: IRIX webdist.cgi, handler and wrap programs
Title: CERT Advisory CA-97.12, AUSCERT Advisory AA-97.14
Number: 19970501-02-PX
Date: August 26, 1997

Silicon Graphics provides this information freely to the SGI user community for its consideration, interpretation, implementation and use. Silicon Graphics recommends that this information be acted upon as soon as possible.

Silicon Graphics provides the information in this Security Advisory on an "AS-IS" basis only, and disclaims all warranties with respect thereto, express, implied or otherwise, including, without limitation, any warranty of merchantability or fitness for a particular purpose. In no event shall Silicon Graphics be liable for any loss of profits, loss of business, loss of data or for any indirect, special, exemplary, incidental or consequential damages of any kind arising from your use of, failure to use or improper use of any of the instructions or information in this Security Advisory.

- - ----------
- - --- Issue Specifics ---
- - ----------

Several programs provided with the Outbox Environment subsystem have been found to be insecure. These are the cgi-bin programs webdist.cgi, handler and wrap available for IRIX 5.x and 6.x. Each of these programs can be manipulated to execute arbitrary commands with potentially elevated privileges.

Silicon Graphics Inc. has investigated the issue and recommends the following steps for neutralizing the exposure. It is HIGHLY RECOMMENDED that these measures be implemented on ALL vulnerable SGI systems. This issue will be corrected in future releases of IRIX.

- - ----------
- - --- Impact ---
- - ----------

In general, the Outbox subsystem is install by default on all SGI systems starting with IRIX 6.2. However, IRIX 5.x and pre-IRIX 6.2 systems may have the Outbox subsystem as part of the Desktop software package.

For these particular vulnerabilities, a local account is not required. Furthermore, each of these vulnerabilities can be exploited remotely.

Utilizing these vulnerabilities, arbitrary commands can be executed with httpd daemon privileges. Depending on configuration of the http server, privileged access may be possible.

This issue has been publically disclosed and discussed in several public forums including the BUGTRAQ mailing list in addition to security advisories CERT CA-97.12 and AUSCERT AA-97.14

- - ----------
- - - --- Software Check ---
- - ----------

To determine if the Outbox software is installed on a particular system, the following command can be used:

```
% /usr/sbin/versions outbox.sw

I = Installed, R = Removed

  Name                Date      Description

I outbox              03/23/97  Outbox Environment, 1.2
I outbox.sw           03/23/97  Outbox End-User Software, 1.2
I outbox.sw.outbox    03/23/97  Outbox Software Tools, 1.2
I outbox.sw.webdist   03/23/97  Web Software Distribution Tools, 1.2
```

In the above case, the Outbox software is installed and the steps in either the "Temporary Solution" or "Solution" section should be completed.

- - ----------
- - - --- Temporary Solution ---
- - ----------

Although patches are available to address the vulnerabilities in these programs, it is realized that there may be situations where installing the patches immediately may not be possible.

Below, two possible solutions are provided to remove the vulnerabilities. In Solution A program permissions are changed while in Solution B the Outbox subsystem is removed. Either solution can be used depending on site requirements.

Solution A - Change program permissions.

The steps below can be used to remove the vulnerabilities by removing the permissions of the vulnerable programs. The default installation places these files in /var/www/cgi-bin, however, all cgi-bin directories on a system should be checked for these programs.

1) Become the root user on the system.

```
% /bin/su -
Password:
#
```

2) Change the permissions on the programs.

```
# /bin/chmod 400 /var/www/cgi-bin/webdist.cgi
# /bin/chmod 400 /var/www/cgi-bin/handler
# /bin/chmod 400 /var/www/cgi-bin/wrap
```

```
************
*** NOTE ***
************
```

By changing the permissions on these programs as above, these programs can not be executed by any user.

3) Return to previous level.

```
# exit
$
```

Solution B - Removal of the Outbox software.

1) Become the root user on the system.

```
% /bin/su -
Password:
#
```

2) Remove the vulnerable outbox subsystem.

```
# /usr/sbin/versions -v remove outbox
```

3) Return to previous level.

```
# exit
$
```

- - ----------
- - --- Solution ---
- - ----------

OS Version	Vulnerable?	Patch #	Other Actions
IRIX 3.x	no		
IRIX 4.x	no		
IRIX 5.0.x	no		
IRIX 5.1.x	no		
IRIX 5.2	no		
IRIX 5.3	yes	2315	
IRIX 6.0.x	yes	not avail	Note 1
IRIX 6.1	yes	not avail	Note 1
IRIX 6.2	yes	2314	
IRIX 6.3	yes	2338	
IRIX 6.4	yes	2338	

NOTES

1) upgrade operating system or see "Temporary Solution" section.

Patches are available via anonymous FTP and your service/support provider. The SGI anonymous FTP site is sgigate.sgi.com (204.94.209.1) or its mirror, ftp.sgi.com. Security information and patches can be found in the ~ftp/security and ~ftp/patches directories, respectfully.

Patch File Checksums

The actual patch will be a tar file containing the following files:

```
Filename:              README.patch.2315
Algorithm #1 (sum -r): 25011 11 README.patch.2315
Algorithm #2 (sum):    39892 11 README.patch.2315
MD5 checksum:          9B5B74574022FEE0259307C44A4602C0

Filename:              patchSG0002315
Algorithm #1 (sum -r): 44474 2 patchSG0002315
Algorithm #2 (sum):    48055 2 patchSG0002315
MD5 checksum:          0C9FA667D42B3FC6895C5AD612CE4FB1
Filename:              patchSG0002315.idb
Algorithm #1 (sum -r): 27036 3 patchSG0002315.idb
Algorithm #2 (sum):    23448 3 patchSG0002315.idb
MD5 checksum:          3DEEA538437CC5D96488269715948F31

Filename:              patchSG0002315.outbox_sw
Algorithm #1 (sum -r): 38552 35 patchSG0002315.outbox_sw
Algorithm #2 (sum):    34937 35 patchSG0002315.outbox_sw
MD5 checksum:          0655A31A55306B4272FD00796CF69466

Filename:              README.patch.2314
Algorithm #1 (sum -r): 10949 11 README.patch.2314
Algorithm #2 (sum):    39885 11 README.patch.2314
MD5 checksum:          E2BFB467EF18F1D5B5CDCE5FBDC3F36D

Filename:              patchSG0002314
Algorithm #1 (sum -r): 18667 2 patchSG0002314
Algorithm #2 (sum):    51210 2 patchSG0002314
MD5 checksum:          FE7C82E22CD63CC278C011FE80F3265A

Filename:              patchSG0002314.idb
Algorithm #1 (sum -r): 23116 3 patchSG0002314.idb
Algorithm #2 (sum):    23091 3 patchSG0002314.idb
MD5 checksum:          CB3E71D4FB3D86192E15EE59AED8A296

Filename:              patchSG0002314.outbox_sw
Algorithm #1 (sum -r): 56643 35 patchSG0002314.outbox_sw
Algorithm #2 (sum):    28391 35 patchSG0002314.outbox_sw
MD5 checksum:          5CC76625AB89FF2862EB21111E7924F0

Filename:              README.patch.2338
Algorithm #1 (sum -r): 39046 13 README.patch.2338
Algorithm #2 (sum):    48961 13 README.patch.2338
MD5 checksum:          D568381EF9948399D50663C2CB9175E8
```

```
Filename:               patchSG0002338
Algorithm #1 (sum -r):  41914 3 patchSG0002338
Algorithm #2 (sum):     15225 3 patchSG0002338
MD5 checksum:           18845B0651DC6CCED5E92694915DB36D

Filename:               patchSG0002338.idb
Algorithm #1 (sum -r):  32107 4 patchSG0002338.idb
Algorithm #2 (sum):     19674 4 patchSG0002338.idb
MD5 checksum:           351DFB48864622C3B8219CC88526F5BE

Filename:               patchSG0002338.outbox_sw
Algorithm #1 (sum -r):  36841 107 patchSG0002338.outbox_sw
Algorithm #2 (sum):     12878 107 patchSG0002338.outbox_sw
MD5 checksum:           2269AB01C700BB39E1CCF1849FAC70AF
```

- - ----------
- - - --- Acknowledgments ---
- - ----------

Silicon Graphics wishes to thank the CERT Coordination Center, and AUSCERT for their assistance in this matter.

- - ----------
- - - --- Silicon Graphics Inc. Security Information/Contacts ---
- - ----------

If there are questions about this document, email can be sent to cse-security-alert@sgi.com.

------oOo------

Silicon Graphics provides security information and patches for use by the entire SGI community. This information is freely available to any person needing the information and is available via anonymous FTP and the Web.

The primary SGI anonymous FTP site for security information and patches is sgigate.sgi.com (204.94.209.1). Security information and patches are located under the directories ~ftp/security and ~ftp/patches, respectively. The Silicon Graphics Security Headquarters Web page is accessible at the URL http://www.sgi.com/Support/security/security.html.

For issues with the patches on the FTP sites, email can be sent to cse-security-alert@sgi.com.

For assistance obtaining or working with security patches, please contact your SGI support provider.

------oOo------

Silicon Graphics provides a free security mailing list service called wiretap and encourages interested parties to self-subscribe to receive (via email) all SGI Security Advisories when they are released.

Subscribing to the mailing list can be done via the Web (http://www.sgi.com/Support/security/wiretap.html) or by sending email to SGI as outlined below.

```
% mail wiretap-request@sgi.com
subscribe wiretap < YourEmailAddress >
end
^d
```

In the example above, < YourEmailAddress > is the email address that you wish the mailing list information sent to. The word end must be on a separate line to indicate the end of the body of the message. The control-d (^d) is used to indicate to the mail program that you are finished composing the mail message.

------oOo------

Silicon Graphics provides a comprehensive customer World Wide Web site. This site is located at http://www.sgi.com/Support/security/security.html.

------oOo------

For reporting *NEW* SGI security issues, email can be sent to security-alert@sgi.com or contact your SGI support provider. A support contract is not required for submitting a security report.

APÉNDICE B

Directorios Principales

Directorios Principales

Directorio	Utilización
/	Directorio principal "root". Contiene archivos específicos del hardware y los necesarios para arrancar el sistema. También sirve de directorio hogar de root; por lo que puede contener archivos de trabajo, configuración e información de esta cuenta en especial.
/bin	Liga a */usr/bin*
/CDROM	Sirve de punto de montaje para los SA colocados en Discos Compactos.
/debug	Liga a */proc*.
/dev	Archivos de Dispositivos. Se encuentran todos los archivos especiales de dispositivos que permiten el acceso del sistema a las diversas partes del equipo; tanto lógicas como físicas, internas o externas.
/dev/dsk	Archivos de dispositivos para acceso a discos en forma de bloques.
/dev/rdsk	Archivos de dispositivos para acceso a discos en forma raw; flujo de caracteres.
/etc	Archivos, tablas y comandos de configuración y administración estándar.
/etc/config	Archivos de configuración del sistema.
/etc/init.d	Programas de inicialización que permiten en conjunto, colocar al sistema en un estado preciso, mediante el uso del comando *init*.
/etc/rc*.d	Liga a determinados archivos del directorio *'etc init.d*.
/lib	Librerías públicas.
/lost+found	Utilizado por *fsck* para conectar nuevamente, los directorios y archivos perdidos. Todo sistema de archivos contiene uno, el cual es creado automáticamente cuando es generado el SA mediante el comando *mkfs*.

Directorio	Utilización
/proc	Es un directorio que sirve como punto para montar un sistema de archivos ficticio; es decir, que no consume espacio. En versiones anteriores era conocido como */debug*. Provee una interfaz a los procesos del sistema a fin de poder analizar su estado mediante herramientas de depuración como *dbx*.
/stand	Contiene copias de reserva de los programas stand alone requeridos para arrancar y dar mantenimiento al sistema, localizados en el encabezado de volumen del disco principal.
/tmp	Archivos temporales. Utilizado por diversas aplicaciones para crear archivos temporales que serán utilizados durante su ejecución.
/unix	Kernel de IRIX.
/usr	Contiene los directorios de las diversas aplicaciones y servicios instalados en el equipo, empleados en modo multiusuario. Suele ser colocado en un SA separado.
/usr/bin	Programas y comandos ejecutables de IRIX.
/usr/bin/X11	Programas de X windows.
/usr/bsd	Comandos Berkeley.
/usr/demos	Programas de demostración.
/usr/etc	Comando y archivos de administración de la red.
/usr/include	Archivos y librerías utilizadas en lenguajes de C, FORTRAN, etc., para ser incluidos.
/usr/local	Utilizado frecuentemente para colocar en él, comandos no estándares y específicos del equipo; locales.
/usr/people	Contiene los directorios de los usuarios dados de alta en el sistema.
/usr/relnotes	Contiene las notas de actualización de último momento (release note) de los productos instalados.
/usr/sbin/inst	Base de datos de los productos instalados mediante el comando *inst*.
/usr/spool	Contiene el sistema de cola de impresión.
/var	Contiene archivos configurables por el administrador. Algunos directorios dentro de éste, son ligas a directorios dentro de */usr*.
/var/adm	Archivos administrativos de mensajes y archivos del sistema de contabilidad (*syslog* y *acct*).

Directorio	Utilización
/var/sysgen	Archivos, librerías, tablas, etc. fuentes para la generación de un nuevo kernel.
/var/X11	Archivos de configuación del ambiente gráfico X11. Es una liga a un directorio dentro de */usr*.
/var/tmp	Archivos temporales.

GLOSARIO

GLOSARIO

Administrador del sistema	Persona encargada de instalar, configurar, mantener y solucionar cualquier problema que ocurra en un sistema. Esta persona suele utilizar la cuenta de root para efectuar la mayoría de sus tareas.
Archivo	Contenedor en el cual se puede almacenar información como texto, programas o imágenes que son creadas con aplicaciones.
Archivo de Acceso en Bloque	Es un tipo de Archivo de Dispositivos, que permite la comunicación en forma de bloque.
Archivo de Acceso en Carácter.	Tipo de Archivo de Dispositivos, que permite la comunicación en forma de un flujo de caracteres.
Archivo de dispositivo	Archivo especial que permite el acceso a diferentes componentes físicos y lógicos de un equipo de cómputo (RAM, particiones, puerto serial y paralelo, red, etc.), ver pág. III-23.
Background	Fondo o segundo plano. Se emplea para referirse a los procesos no interactivos; es decir, que no requieren la intervención de una persona para funcionar, por lo que corren en el fondo sin ninguna interferencia.
Bit	Dígito binario. Una variable que puede tener sólo dos posibles valores: 0 ó 1.
Bitácora	Libro en el cual se lleva un registro de los problemas y acontecimientos relacionados con el funcionamiento y mantenimiento del equipo.
Block device	ver Archivo de Acceso en Bloque.
Bloque	Sector de un disco compuesto por 512 bytes de largo.
Bus	Canal de transmisión eléctrica que sirve para transportar información entre varios dispositivos conectados a él.
Caída de sistema	Cuando el SO falla y no acepta ninguna entrada (comando) del teclado o ratón.

CDROM	Compact Disc Read Only Memory; Disco Compacto de Sólo Lectura. Dispositivo de almacenamiento de tecnología láser.
CERT	Computer Emergency Response Team; Equipo de Respuesta a la Emergencia de Computadoras. Grupo proyectado para facilitar a la comunidad, respuesta a los acontecimientos relacionados con la seguridad en computadoras que involucran a los equipos conectados a Internet.
Cilindro	Conjunto de todas las pistas equidistantes de todos los platos que forman un disco; ver **Fig. III-3**.
CPU	Central Processing Unit. Término utilizado para nombrar al procesador central de una computadora; o al módulo físico que lo contiene.
Cracker	En el argot computacional, persona que intenta o logra violar la seguridad de un equipo de cómputo.
Cron	Sistema que permite ejecutar programas a intervalos específicos, sin la intervención de una persona.
DARPA	Defense Advanced Research Projects Agency. Agencia de Proyectos de investigación Avanzada de la Defensa de los Estados Unidos.
DAT	Casete de cinta de 4 mm de ancho que permite almacenar una gran cantidad de información, tanto digital como de audio. Utiliza una tecnología de grabación digital con calidad de CD.
DDS	Digital Data Storage. Almacenamiento de datos digitales. Ver, DAT.
Default	Conjunto de valores que son especificados inicialmente por el hardware o software, si el usuario no indica otra cosa; posteriormente se pueden cambiar a un valor adecuado. Conjunto de valores que son asignados por omisión.
Desfragmentar	Reunir la información de un archivo en bloques de datos contiguos; de tal forma que la lectura de él resulta ágil.

Directorio	Tipo de archivo especial que sólo puede ser modificado por el SO. Contiene el nombre y número de inodo de los archivos que se encuentran almacenados dentro de él. Un contenedor en un SA en el cual se pueden almacenar otros directorios o archivos.
Directorio Hogar (HOME)	Directorio en que coloca a un usuario el SO IRIX, cada vez que entra a sesión.
Disco Principal, Disco de Sistema.	Disco físico que contiene el SO IRIX así como las herramientas necesarias para poder arrancar el sistema.
EFS	Extent File System; Sistema de Archivo Extendido. Principal tipo de SA utilizado en la versión 5.3 de IRIX.
Email	Dirección de correo electrónico, a la cual se puede enviar un mensaje a través de una red.
Encabezado de volumen	ver Volhdr.
Firmware	Conjunto de programas que se encuentran almacenados dentro de una ROM e incorporados en la tarjeta principal del equipo. Realizan una función específica.
Fragmentación	Se dice que un archivo se encuentra fragmentado cuando la información que contiene se encuentra dispersa en bloques de datos por todo el disco.
Gateway	Dispositivo que permite conectar dos redes con diferente protocolo. Cambia al menos, los 4 primeros protocolos del modelo OSI.
Grupo de Cilindros	Conjunto de cilindros contiguos que se utilizan para almacenar datos de un SA; contiene una sección de inodos y otra de bloques de datos (ver **Fig. III-7**).
Gusano	Aplicaciones que utilizan la red para expandirse entre los distintos equipos que la componen. Una vez en los equipos, se pueden comportar como virus (ver pág. IV-57).
Hacker	Persona que se deleita aprendiendo los detalles de la programación de sistemas y cómo extender sus capacidades.

Hardware	Equipo físico: Está formado por todos los componentes electrónicos y mecánicos que integran a un equipo.
Icono	Pequeña figura gráfica que representa un objeto; como un archivo, directorio, aplicación u proceso dentro del equipo.
Inodo	Estructura de datos de 128 bytes que el SA usa para almacenar toda la información referente a un archivo, a excepción de su nombre: tipo, permisos, ligas, ubicación, tamaño, etc.
Integridad	Característica de la información de reflejar datos congruentes con la realidad.
Internet	Red mundial compuesta por una gran cantidad de redes menores. Sus orígenes se encuentran en la red ARPANET fundada por DARPA en 1969.
IRIX	Sistema operativo de Silicon Graphics. Es la versión UNIX implementada por esta compañía.
Kerberos	Sistema de seguridad desarrollado por MIT, el cual otorga autenticidad a los usuarios.
Kernel	Corazón del sistema operativo; sistema operativo base; código que implementa las llamadas al sistema.
Login	Cuenta requerida para accesar un equipo multiusuario de cómputo. También, proceso que se sigue para verificar la autenticidad de un usuario y darle acceso al sistema.
Memoria Virtual	La combinación de la memoria física y el área de swap.
miniroot	Miniature root. Sistema operativo reducido que permite, a través de él, realizar labores de mantenimiento, recuperación de fallas e instalación del sistema operativo IRIX (ver pág. I-16).
MIT	Massachusetts Institute of Technology; Instituto de Tecnología de Massachusetts.
Monitoreo	Mantener vigilancia sobre los diversos recursos del sistema. Existen aplicaciones que permiten efectuar estas labores, conocidos como monitores.
Montar (SA)	Proceso llevado a cabo mediante el comando *mount* para que se reconozca y sea añadido al SA principal, uno nuevo, pág. III-24

OSI	Open System Interconnect; Interconexión de Sistemas Abiertos. Estándar que consta de 7 niveles o capas de protocolos que permiten la comunicación a través de redes, de sistemas heterogéneos.
Partición	Porción del área de un disco, que puede ser tratada como una entidad propia; como un disco.
password	Clave secreta que sirve para validar la autenticidad del dueño de una cuenta.
PID	Número de Identificación del proceso. Cada vez que se crea un proceso, se le asigna un número (PID) que lo identifica. No pueden existir dos procesos con el mismo pid; ya que es un número consecutivo que nunca se repite. Cuando un proceso termina, desaparece, y el pid que se le asignó ya no vuelve a ser asignado a otro proceso. Cuando los PID asignados son excesivamente grandes, es aconsejable reiniciar el equipo para restaurar la numeración.
Pista	Ruta que genera una cabeza de lectura sobre la superficie de un plato del disco al girar éste. Círculos concéntricos formados en la superficie de un plato; ver **Fig. III-2**.
Portabilidad	Capacidad de utilizar un SO o software de aplicación, en sistemas de cómputo de diferentes proveedores.
Proceso	Es un programa en ejecución.
PROM	Programmed Read-Only Memory. Firmware utilizado para realizar labores de mantenimiento y arranque del sistema operativo.
prompt	El carácter o palabra que el sistema despliega en un shell para indicar que está listo para recibir comandos. Mensaje del software indicando que requiere alguna acción por parte del usuario.
Protocolo	Conjunto de reglas convencionales utilizadas para entablar una comunicación entre dos dispositivos.
RAM	Random Access Memory. Memoria de acceso aleatorio, que puede ser escrita y leída en forma dinámica.

Raw device	ver Archivo de Acceso en Carácter.
Red	Conjunto de equipos de cómputo enlazados a través de algún medio que permite la intercomunicación entre ellos.
ROM	Read Only Memory. Memoria de sólo lectura, cuya información es no-volátil; por lo que no puede ser modificada una vez grabada en él.
SA	Sistema de Archivos
SO	Sistema Operativo
sash	Stand Alone Shell; Intérprete de comandos que no requiere la presencia de un software bajo él para funcionar (ver pág. I-16)
SCSI	Small Computer System Interface; Interfaz de Sistema de Cómputo Pequeña. Es otro estándar de comunicación de controladores de discos que permite conectar 7 dispositivos, además de la propia tarjeta controladora, mediante el empleo de un Bus de comunicación.
Sesión	Conexión activa entre un usuario y una computadora o entre dos computadoras. Lapso que transcurre desde que un usuario ingresa al sistema mediante su cuenta y clave secreta, hasta que sale de ella utilizando el comando adecuado; como el de *exit* en UNIX.
SGI	Silicon Graphics Inc.
Shadow	Sistema de seguridad adicional mediante el cual, se extrae el campo de la clave secreta y otra información del archivo */etc/passwd*, y es colocada en otro archivo al cual los usuarios normales no tienen acceso; por tanto, no pueden leerla para tratar de descifrarla.
Shell	Interprete de comandos. Ventana en la cual se pueden escribir y ejecutar comandos. Concha, caparazón. Capa exterior de un programa, que proporciona la interfaz del usuario, o medio para gobernar la computadora.

Sistema de Archivos	Estructura y forma en la cual se organizan los datos dentro de una unidad de almacenamiento; como un disco. Estructura de datos jerárquica que contiene y organiza directorios y archivos dentro de ella.
Sistema Operativo	Conjunto de programas que administran los recursos del equipo de cómputo. Se ocupa de aspectos como la memoria, seguridad, procedimientos de entrada/salida, calendarización de procesos y administración de archivos y procesos.
Software	Se conoce como software a la diversidad de programas que corren dentro de un computadora, y hacen que ésta funcione. Se tienen dos categorías principales: de sistemas y de aplicaciones. Los primeros son el conjunto de programas que controlan y administran los recursos de la computadora, y los segundos son el resto de aplicaciones que nos permiten procesar los datos para obtener resultados.
Super bloque	Sección de datos localizada dentro de todo sistema de archivos, la cual describe su estructura. Se encuentra ubicada a partir del bloque 2 en SA tipo EFS. (ver pág. III-16).
Swap	Porción de espacio de disco utilizada como memoria secundaria para almacenar pedazos del contenido de la memoria principal; RAM.
Terminador	Para enviar una señal a través de una línea de transmisión en forma segura, debe existir una impedancia al final que iguale la impedancia de la fuente y de la línea misma. Errores de amplitud, respuesta de frecuencia, y distorsión y reflexión de pulso (fantasmas) ocurren en una línea sin un terminador apropiado.
UNIX	Marca registrada por AT&T para nombrar a su sistema operativo multiusuario; considerado muy flexible, poderoso y altamente portable. Usado equivocadamente como término genérico para varias versiones del sistema operativo UNIX de AT&T implementado por otras compañías: de Hewlett-Packard es HP-UX; de Microsoft es el XENIX; de la Universidad de California, Berkeley, es BSD; de DEC es Ultrix, etc.

UPS	Uninterruptible Power Supply; Fuente de Energía Ininterrumpible. Fuente de energía alterna que sirve de respaldo para que cuando se presente una falla de energía, no se suspenda el suministro en los dispositivos que se encuentren conectados a éste. Puede suministrar corriente por varios minutos u horas, y generalmente cuenta con sistemas de regulación y supresión de picos.
Virus	Aplicaciones que se comportan de forma similar a los virus humanos, causando interferencias o daños a los procesos e información almacenada en los equipos de cómputo (pág. IV-56)
VL	Volumen Lógico
Volhdr	Partición 8 de un disco, llamada Encabezado de Volumen, que almacena la información referente a la distribución, organización y características del disco.
Volumen Lógico	Arreglo de particiones que constituyen un disco lógico.
Window	Ventana; porción de la pantalla que puede ser manipulada individualmente y que puede contener gráficos o texto.
Workstation	Estación de Trabajo; Equipo de cómputo de escritorio, con gran potencia de proceso, que ha sido diseñado para trabajar en ambiente gráfico. Compuesto generalmente por un CPU, que contiene un disco principal y una tarjeta gráfica entre otras cosas, además de teclado, ratón y monitor.
WWW	Worl Wide Web, Gran Telaraña Mundial.
X windows	Sistema de ventanas transparente sobre red, que corre en una gran variedad de equipos de cómputo y utiliza un protocolo tipo cliente/servidor. Suele conocérsele como: X window System, X11, X version 11, X. Originalmente fue desarrollado en el Laboratorio de Ciencias de Cómputo de MIT, a partir del 1 de Enero de 1994 todos los derechos fueron asignados al X Consortium Inc.
XDM	X Display Manager. Manejador de pantallas X, para el ambiente gráfico X windows. Si el equipo posee capacidades gráficas, al correr este software se permite el desplegado y ejecución de ventanas X en la pantalla.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- An Evening with Berferd *
  In Which a Cracker is Lured, Endured, and Studied
  Bill Cheswick
  AT&T Bell Laboratories

- Computer Emergency Response - An International Problem *
  Richard D. Pethia
  Kenneth R. van Wyk

- Improving The Security Of Your UNIX System *
  David A. Curry, Systems Programmer
  Information and Telecommunications Sciences and Technology Division
  ITSTD-721-FR-90-21

- Computer Virus and Related Threats:
  A Management Guide *
  John P. Wack
  Lisa J. Carnahan

- Department of Defense Trusted Computer System Evaluation Criteria *
  DoD 5200.28-STD
  December 1985

- Practical Unix Security
  Garfinkel S. and Spafford G.
  O' Reilly & Associates, Inc.

- Indy Workstation Owner's Guide
  Document Number 007-9804-040

- IRIS Essentials
  Manual del Sistema

- IRIS Glossary of Terms
  Manual del Sistema

* Documentos públicos obtenidos del CERT: http://www.cert.org

- Software Installation Administrator's Guide
  Manual del Sistema

- Páginas de Referencia
  Ayuda en línea mediante el comando
  $ man

- IRIX Advanced Site and Server Administration Guide
  Manual del Sistema

- IRIX System Administration
  Student Handbook
  Part Number: LBT111
  Revision: 1.0 Beta3
  May 1995