

00623
17



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CONTADURÍA Y
ADMINISTRACIÓN

ESTUDIO COMPARATIVO DE LOS SISTEMAS OPERATIVOS LINUX Y WINDOWS NT

TESIS PROFESIONAL QUE PARA OBTENER EL TÍTULO
DE:

LICENCIADO EN INFORMÁTICA

PRESENTA:

DANTE MURILLO DOMÍNGUEZ

ASESOR:

MTRO. LUIS EDUARDO LÓPEZ CASTRO



MÉXICO, DF.

2000
Actualizada a 2002

2003

1



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Introducción

Las computadoras Intel x86 han evolucionado de una forma notable durante la última década, influyendo directamente en actividades que realizamos a diario, ya sea como individuos o como parte de estructuras organizacionales que requieren poder de cómputo distribuido para responder a las crecientes necesidades de procesamiento de información, sin embargo, los avances del hardware han superado al software que corre bajo la arquitectura Intel x86. Lo anterior es debido a los pocos avances en el Sistema Operativo, el cual debe proporcionar a los programas de aplicación acceso a los avances del hardware, sin embargo, en el caso de las computadoras x86 los únicos Sistemas Operativos en el mercado hasta hace poco, eran los proporcionados por Microsoft dentro de una familia de productos llamada Windows, de los cuales el más representativo en este momento es "Windows .NET Server" por ser el Sistema Operativo destinado para servidores Intel x86. Por otra parte, en Internet surge un fenómeno curioso referente a las dimensiones de los proyectos del software de libre distribución, entre los cuales surge un nuevo Sistema Operativo para la plataforma Intel x86 llamado Linux. Este último, ha crecido notablemente en años recientes siendo hoy en día considerado como el proyecto de desarrollo, más ambicioso del mundo, por sus perspectivas de desarrollo y posibilidades de comercialización.

La base instalada de Linux crece rápidamente a tal punto que se ha colocando como el segundo Sistema Operativo mas vendido en los últimos años, pues ha abarcado mercado rápidamente en el terreno de los servidores x86 gracias a su inigualable estabilidad.

Por lo anterior he seleccionado a Linux dentro de este estudio, con el fin de demostrar que sus características técnicas son muy superiores a las de "Windows .NET Server" como Sistema Operativo imperante en el mercado, y que sus posibilidades comerciales para ser la plataforma del siglo XXI dentro de las empresas en el campo de los servidores x86 son reales.

Intentare demostrar lo anterior, tomando como base una estructura capitular, de la cual se da una breve explicación a continuación:

Autorizo a la Dirección General de Bibliotecas de
UNAM a difundir en formato electrónico e impres
contenido de mi trabajo recepción

NOMBRE: DANTE MURILLO

DOMINGUEZ

FECHA: 21/04/2003

FIRMA: [Firma]

Capítulo I. Daré la definición de Sistema Operativo que será usada a lo largo de este trabajo, así como el estudio de las diferentes estructuras usadas en la construcción de Sistemas Operativos. Posteriormente daré algunos conceptos de Sistemas Operativos que serán usados en este trabajo.

Capítulo II. Mostrare la historia de los Sistemas Operativos a través de las diferentes generaciones de computadoras y expondré la línea de tiempo de Windows .NET Server y Linux.

Capítulo III. Mostraré los puntos principales del estándar POSIX.1, y haré un estudio completo de las librerías y funciones principales marcadas por este estándar. Posteriormente daré una breve referencia a estándares futuros.

Capítulo IV. Expondré la arquitectura de Windows .NET Server, así como sus lineamientos de diseño y componentes fundamentales. Descompondré por completo las capas del Sistema Operativo para llegar al microkernel donde apreciaremos su tarea y mostrare que éste se dedica a las funciones más básicas de asignación de tiempos de procesamiento delegando otras tareas administrativas importantes que debería realizar.

Capítulo V. Daré una referencia de las características principales de Linux, así mismo, explicaré la arquitectura del kernel de Linux y analizaré su composición modular, explicando los puntos más importantes del kernel.

Capítulo VI. En este capítulo haré una comparación conceptual de ambos Sistemas Operativos de acuerdo a la definición de Sistema Operativo dada en el capítulo I y al análisis técnico realizado de ambos Sistemas Operativos en los capítulos IV y V.

Capítulo VII. Haré una descripción del proceso de desarrollo del Software de libre distribución y analizaré sus ventajas y desventajas debido a la importancia de este proceso en los futuros desarrollos de software y en especial del tema tratado en esta Tesis. Posteriormente haré un comparativo de la plataforma instalada de ambos sujetos de estudio y expondré sus tasas de crecimiento durante los últimos años.

Por último expondré al lector mis conclusiones de acuerdo a lo expuesto en el presente trabajo, tomando en cuenta que lo aquí presentado esta limitado exclusivamente a las definiciones conceptuales dadas y al estudio de las arquitecturas de ambos Sistemas Operativos.

Dante Murillo Domínguez
Estudiante de la Licenciatura en Informática

INDICE

INTRODUCCIÓN	2
INDICE	5
CAPÍTULO I. UN ACERCAMIENTO A LOS SISTEMAS OPERATIVOS	9
1.1 ¿QUÉ ES UN SISTEMA OPERATIVO?	9
1.2 ESTRUCTURA DE LOS SISTEMAS OPERATIVOS	11
1.2.1 SISTEMAS MONOLÍTICOS	11
1.2.2 SISTEMAS CON CAPAS	13
1.2.3 MÁQUINAS VIRTUALES	14
1.2.4 MODELO CLIENTE / SERVIDOR	15
1.3 CONCEPTOS DE LOS SISTEMAS OPERATIVOS	17
1.3.1 PROCESO	17
1.3.2 ARCHIVOS	20
1.3.3 TUBOS (PIPELINE)	21
1.3.4 LLAMADAS AL SISTEMA	22
1.3.5 INTERPRETE DE COMANDOS	22
CAPITULO II. HISTORIA DE LOS SISTEMAS OPERATIVOS	24
2.1 LAS GENERACIONES DE COMPUTADORAS	24
2.1.1 PRIMERA GENERACIÓN (1945 / 1955) BULBOS Y CONEXIONES	24
2.1.2 SEGUNDA GENERACIÓN (1955 / 1965) TRANSISTORES Y SISTEMAS DE PROCESAMIENTO POR LOTES.	25
2.1.3 LA TERCERA GENERACIÓN (1965 / 1980). CIRCUITOS INTEGRADOS Y MULTIPROGRAMACIÓN.	26
2.1.4 LA CUARTA GENERACIÓN (1980 / 2000) COMPUTADORAS PERSONALES	30
2.2 WINDOWS Y LINUX EN LA LÍNEA DEL TIEMPO	32
2.2.1 MICROSOFT WINDOWS	32
2.2.2 GNU LINUX	34
CAPITULO III. EL ESTÁNDAR POSIX 1003.1	39
3.1 INTRODUCCIÓN A LOS ESTÁNDARES POSIX	39
3.2 POSIX 1003.1	41
3.3 OTROS ESTÁNDARES POSIX	47
POSIX 2	47
POSIX 3	47
POSIX 4	48

POSIX 5	48
POSIX 9	48
ADMINISTRACIÓN DEL SISTEMA	49
3.4 ESTÁNDARES FUTUROS	49

CAPITULO IV. WINDOWS	51
-----------------------------	-----------

4.1 OBJETIVOS DE DISEÑO	51
EXTENSIBILIDAD	51
TRANSPORTABILIDAD	52
CONFIABILIDAD	52
COMPATIBILIDAD	53
SEGURIDAD	53
RENDIMIENTO	53
4.2 ARQUITECTURA DE WINDOWS	54
4.3 LOS SUBSISTEMAS DE AMBIENTE	56
4.3.1. SUBSISTEMA WIN32	56
4.3.1. SUBSISTEMA WIN32	57
4.3.2. SUBSISTEMAS OS/2 Y POSIX	57
4.3.3. SUBSISTEMA MS-DOS	57
4.3.4. SUBSISTEMA WIN16	58
4.3.5. SUBSISTEMA DE SEGURIDAD	58
4.4. NT EXECUTIVE	60
4.5 KERNEL	63
4.6 HARDWARE ABSTRACTION LAYER (HAL).	64

CAPITULO V. LINUX	65
--------------------------	-----------

5.1 ORIGEN DE LINUX	65
5.2 CARACTERÍSTICAS GENERALES DE LINUX	65
5.2.1 MULTITAREA	65
5.2.2 MULTIUSUARIO	66
5.2.3 MULTIPLATAFORMA	66
5.2.4 MULTIPROCESADOR	66
5.2.5 MODO PROTEGIDO 386	66
5.2.6 CARGA DE EJECUTABLES SOBRE DEMANDA	66
5.2.7. MEMORIA VIRTUAL USANDO PAGINACIÓN OPTIMIZADA (SIN INTERCAMBIO DE PROCESOS COMPLETOS A DISCO)	67
5.2.8 BIBLIOTECAS COMPARTIDAS DE CARGA DINÁMICA (DLL's) Y BIBLIOTECAS ESTÁTICAS	67
5.2.9 VOLCADOS DE ESTADO (CORE DUMP)	67
5.2.10 COMPATIBILIDAD	67
5.2.11 EMULACIÓN DEL COPROCESADOR MATEMÁTICO	68
5.2.12 SOPORTE DE TECLADOS	68
5.2.13 CONSOLAS VIRTUALES MÚLTIPLES	68

5.2.14 SISTEMAS DE ARCHIVOS	68
5.2.15 COMUNICACIONES	69
5.3 EL KERNEL DE LINUX	69
5.3.1 LAS DISTRIBUCIONES DEL KERNEL	70
5.4 ESTRUCTURA DEL KERNEL DE LINUX	70
5.4.1 ADMINISTRACIÓN DE PROCESOS	72
5.4.2 ADMINISTRACIÓN DE MEMORIA	72
4.4.3 SISTEMAS DE ARCHIVOS	73
5.4.4 CONTROLADOR DE DISPOSITIVOS	73

CAPITULO VI. ESTUDIO COMPARATIVO DE LINUX Y WINDOWS **75**

6.1 WINDOWS Y LINUX FRENTE A FRENTE	77
6.1.1 WINDOWS	77
6.1.2 LINUX	78
6.2 COMPARATIVO DE LOS SISTEMAS OPERATIVOS	78
6.2.1 SISTEMA DE ARCHIVOS	79
6.2.2 ADMINISTRACIÓN DE LA MEMORIA	81
6.2.3 ADMINISTRACIÓN DE PROCESOS	82
6.2.4 RENDIMIENTO	83
6.2.5 ESCALABILIDAD	83
6.2.6 CONFIABILIDAD	84
6.2.7 ADMINISTRACIÓN Y CONFIGURACIÓN	85
6.2.8 SEGURIDAD	87
6.2.9 DESARROLLO DE APLICACIONES	88
6.2.10 RESUMEN COMPARATIVO	88

CAPITULO VII. COMPARACIÓN DE ASPECTOS COMERCIALES **91**

7.1 ¿QUÉ ES EL SOFTWARE ABIERTO O LIBRE?	91
7.2 FORTALEZAS DEL SOFTWARE ABIERTO	95
7.2.1 LOS PROCESOS DE SOFTWARE LIBRE ESTÁN CRECIENDO EN EL INTERNET	95
7.2.2 LOS PROCESOS DE SOFTWARE LIBRE SON DEL TIPO "EL GANADOR SE LLEVA TODO"	95
7.2.3 LOS DESARROLLADORES BUSCAN CONTRIBUIR A LA PLATAFORMA MÁS GRANDE DE SOFTWARE LIBRE.	96
7.2.4 LOS PROYECTOS MÁS GRANDES DE SOFTWARE LIBRE TIENEN UN EFECTO CINEGÉTICO	96
7.3 REACCIONES DE MICROSOFT AL FENÓMENO DEL SOFTWARE LIBRE	96
7.4 LINUX COMO UN SOFTWARE DE DISTRIBUCIÓN LIBRE	98
7.5 PORCIONES DEL MERCADO	99
7.5.1. TASAS DE CRECIMIENTO	99
7.5.2 BASE INSTALADA	101
7.6 APOYO DE OTRAS EMPRESAS	102

CONCLUSIONES **107**

GLOSARIO	110
BIBLIOGRAFÍA	113
APÉNDICE A. LA LICENCIA PÚBLICA GENERAL GNU	114
AGRADECIMIENTOS	122

Capítulo I. Un acercamiento a los Sistemas Operativos

1.1 ¿Qué es un sistema operativo?

El Dr. Andrew S. Tanenbaum¹ hace una distinción muy importante entre las familias de programas que podemos ejecutar en una computadora:

↪ *Programas de Sistema.* Controlan la operación y administran los recursos de la máquina, tales como: disco duro, memoria, vídeo, sonido, etc.

↪ *Programas de Aplicación.* Resuelven problemas para los usuarios, y los ayudan a redactar textos, realizar cálculos, etc. En general es el software que la mayoría de los usuarios usa para producir y procesar información por medio de una computadora.

Sistema Bancario	Hojas de Calculo	Juegos	<i>Programas de aplicación</i>
Compiladores	Editores	Interprete de Comandos	<i>Programas de Sistema</i>
Sistema Operativo			
Lenguaje Maquina Microprogramación Dispositivos Fisicos			<i>Hardware</i>

El Dr. Tanenbaum ubica al sistema operativo como un programa de sistema, porque se encarga de administrar y controlar los dispositivos de la máquina teniendo una relación intrínseca con los programas de aplicación, que hacen uso de las facilidades que proporciona para poder escribir en el disco duro, imprimir, etc.

Una computadora es un sistema complejo, que necesita una pieza de software que administre y proporcione a los demás programas una forma ordenada de acceso a los recursos. Es por eso, que las funciones más importantes del sistema operativo son:

- ✓ La administración de dispositivos de hardware.
- ✓ La reducción de la complejidad inherente a la programación de controladores de hardware.

Es importante hacer énfasis en esta última tarea mencionada del sistema operativo, porque ha permitido desarrollar en tiempos humanamente aceptables, programas de aplicación que resuelven problemas a los usuarios. Es decir, los desarrolladores de programas de aplicación no tienen que resolver las complejidades inherentes al escribir rutinas para manejar dispositivos, las cuales requieren de un alto grado de especialización y aumentarían considerablemente el tiempo de desarrollo de sistemas, así como el costo de los mismos.

El Dr. Tanenbaum identifica dos funcionalidades fundamentales del Sistema Operativo¹:

El sistema operativo como máquina extendida

Esta funcionalidad permite al desarrollador de programas de aplicación, programar sin preocuparse por las complejidades del Hardware, es decir, el sistema operativo funge como una máquina extendida que oculta las complejidades del hardware al desarrollador de aplicaciones.

- ✓ Provee abstracciones de alto nivel para objetos complejos de bajo nivel.
- ✓ Da un tratamiento homogéneo a diferentes objetos de bajo nivel (Ej. Una función simple como *read* puede leer un bloque de información de cualquier disco).

El sistema operativo como administrador de recursos

El sistema operativo es el encargado de administrar todos los dispositivos de la máquina, de tal manera que los demás programas hacen uso de este para escribir en un disco, mandar una imagen a la pantalla, etc. es decir, el sistema operativo es el encargado de asignar los recursos que son solicitados por los demás programas, así como de administrarlos para que todos tengan posibilidades de usarlos.

Estas dos funcionalidades nos permiten afirmar que el sistema operativo es una capa intermedia entre los programas de aplicación y el hardware de la computadora o un software especial que nos ayuda a realizar y ejecutar programas más fácilmente.

En lo sucesivo de este trabajo tomaremos la siguiente definición como el concepto de sistema operativo:

"El sistema operativo es la capa intermedia de software entre los programas de aplicación y el hardware de la computadora que administra los recursos y dispositivos óptimamente y oculta las complejidades de la programación a bajo nivel al desarrollador de software"

1.2 Estructura de los Sistemas Operativos

Al plantear el desarrollo de un sistema operativo se pueden idear diferentes arquitecturas que permitan darle una estructura, otorgándole ventajas y desventajas que son muy importantes en el momento de evaluar las características de un sistema operativo. Las arquitecturas mencionadas a continuación, son las más usadas en el desarrollo de sistemas operativos modernos:

1.2.1 Sistemas Monolíticos

"El sistema operativo, se estructura como una colección de procedimientos, que, cada uno de los cuales puede llamar a los demás cada vez que así lo requiera".

Cuando se usa esta técnica, cada procedimiento del sistema tiene una interfaz bien definida en términos de parámetros y resultados, y cada

uno de ellos es libre de llamar a cualquier otro, si éste último proporciona cierto cálculo útil para el primero. Para construir el programa objeto real del sistema operativo usando esta estructura, se compilan en forma individual los archivos que contienen los procedimientos y después éstos son juntados en un sólo archivo ejecutable.

En términos de ocultamiento de la información, cada procedimiento es visible a los demás, siempre y cuando estos formen parte del núcleo del sistema, en contraste con una estructura de módulos o paquetes, en la cual la mayor parte de la información es local a un módulo y donde sólo los datos señalados de forma expresa pueden ser llamados desde el exterior del módulo.

Los servicios que ofrece el sistema operativo (llamadas al sistema) se solicitan colocando los parámetros en lugares bien definidos, como en un registro o en una pila, para después ejecutar una llamada al sistema a través de una instrucción especial de trampa para realizar las llamadas al supervisor. Esta instrucción es recibida por el núcleo del sistema operativo cuando cambia la máquina de modo usuario a modo kernel y transfiere el control al sistema operativo, este último examina los parámetros de la llamada, para determinar que se desea realizar. A continuación, el sistema operativo analiza una tabla que contiene en la entrada k un apuntador al procedimiento que realiza la k -ésima llamada al sistema. Esta operación, identifica el procedimiento de servicio, al cual se llama. Por último, la llamada al sistema termina y el control regresa al programa del usuario.

Lo anterior sugiere la siguiente organización del sistema operativo:

- ✓ Un programa principal que llama al procedimiento de servicio solicitado.
- ✓ Un conjunto de procedimientos de servicio que llevan a cabo las llamadas al sistema.
- ✓ Un conjunto de procedimientos utilitarios que ayudan al procedimiento de servicio.

En este modelo, por cada llamada al sistema existe un procedimiento de servicio que se encarga de ella. Los procedimientos utilitarios hacen cosas necesarias para varios procedimientos de servicio, como por ejemplo buscar los datos de los programas del usuario.

1.2.2 Sistemas con capas

Es posible organizar el sistema operativo como una jerarquía de capas, cada una construida sobre la inmediata inferior, el primer sistema construido de esta manera fue el sistema THE (Technische Hogeschool Eindhoven), desarrollado en Holanda por E. W. Dijkstra y sus estudiantes. El sistema THE era un sistema sencillo de procesamiento por lotes para una computadora holandesa, la Electrológica X8, con 32K de palabras de 27 bits. El sistema tenía 6 capas:

↳ *Capa 0.* Trabaja con la asignación del procesador y alterna entre los procesos cuando ocurren las interrupciones o expiran los cronómetros. Sobre la capa 0, el sistema consta de procesos secuenciales, cada uno de los cuales se podía programar sin tener que preocuparse por el hecho de que varios procesos estuvieran en ejecución en el mismo procesador.

↳ *Capa 1.* Realiza la administración de la memoria. Asignaba el espacio de memoria principal para los procesos y un recipiente de palabras de 512K se utilizaba para almacenar partes de los procesos para las que no existía lugar en la memoria principal. Las capas superiores a esta no debían preocuparse si estaban en la memoria o en el recipiente; el software de la capa 1 se encargaba de garantizar que las páginas llegaran a la memoria cuando fuera necesario.

↳ *Capa 2.* Se encarga de la comunicación entre cada proceso y la consola del operador. Por encima de esta capa, cada proceso tiene su propia consola de operador.

↳ *Capa 3.* Controla los dispositivos de entrada y salida y guarda en buffers los flujos de información entre ellos. Las capas superiores trabajan con dispositivos abstractos de E/S con propiedades adecuadas, en vez de dispositivos reales con muchas peculiaridades.

↳ *Capa 4.* Es donde estaban los programas del usuario. Éstos no tenían que preocuparse por el proceso, memoria, consola o control de E/S.

↳ *Capa 5.* Se encuentra el proceso operador del sistema.

Una generalización más avanzada del concepto de capas se presentó en el sistema MULTICS. En lugar de capas, MULTICS estaba organizado como una serie de anillos concéntricos, siendo los anillos interiores los privilegiados. Cuando un procedimiento de un anillo deseaba llamar a un procedimiento de un anillo interior, debía hacer el equivalente a una llamada al sistema; es decir, una instrucción TRAP cuyos parámetros eran validados con cuidado antes de permitir que procediese el llamado. Aunque todo el sistema operativo era parte del espacio de direcciones

de cada proceso del usuario en MULTICS, el hardware posibilitó el diseño de procedimientos individuales (segmentos de memoria, en realidad) de forma protegida contra la lectura, escritura o ejecución.

Mientras que el esquema de capas de THE era en realidad un apoyo al diseño, debido a que todas las partes del sistema estaban, en última instancia ligadas entre sí en un solo programa objeto, en MULTICS, el mecanismo de anillos estaba más presente durante el tiempo de ejecución y era reforzado por el hardware. La ventaja del mecanismo de anillos es su facilidad de extensión para estructurar subsistemas de usuario.

1.2.3 Máquinas Virtuales

Las versiones iniciales de OS/360 eran sistemas de procesamiento por lotes, sin embargo, muchos de los usuarios de la 360 de IBM deseaban tener tiempo compartido, de forma que varios grupos, tanto dentro como fuera de IBM, decidieron escribir sistemas de tiempo compartido para esta computadora. El sistema de tiempo compartido oficial de IBM, TSS/360, tuvo una entrega retrasada y cuando finalmente apareció, era tan grande y lento que pocos lugares lo adoptaron. Fue abandonado después de que en su desarrollo se consumieron 50 millones de dólares.

Un grupo del Centro Científico de IBM en Cambridge produjo un sistema con diferencias radicales, el cual fue aceptado por IBM como producto y que ahora tiene un uso extenso. Este sistema, cuyo nombre original era CP/CMS y que ahora se llama VM/370 se basó en un sistema de tiempo compartido que proporciona: (1) multiprogramación y (2) una máquina extendida con una interfaz más conveniente que el hardware. La esencia de VM/370 es la total separación entre estas dos funciones.

El corazón del sistema, llamado monitor de la máquina virtual, se ejecuta en el hardware y realiza la multiprogramación, proporcionando no una, sino varias máquinas virtuales a la siguiente capa superior. Sin embargo, a diferencia de los demás sistemas operativos, estas máquinas virtuales no son máquinas extendidas, con archivos u otras características adecuadas, en lugar de esto, son copias exactas del hardware, con su modo núcleo / usuario, Entrada / Salida, interrupciones y todo lo demás que posee la máquina real.

Puesto que cada máquina virtual es idéntica al hardware real, cada una puede soportar cualquier sistema operativo que se ejecute en forma directa sobre el hardware.

Las distintas máquinas virtuales pueden, y por lo general lo hacen, ejecutar distintos sistemas operativos. Algunas ejecutan uno de los descendientes de OS/360 para el procesamiento por lotes, mientras que otras ejecutan un sistema interactivo para un sólo usuario, llamado CMS (Conventional Monitor System), para los usuarios del tiempo compartido.

Cuando un programa CMS ejecuta una llamada al sistema, la llamada es atrapada por el sistema operativo en su propia máquina virtual, no en el VM/370, de la misma forma en que lo haría si se ejecutara en una máquina real en vez de una virtual. CMS proporciona entonces las instrucciones normales de E/S en hardware para la lectura del disco virtual o lo necesario para llevar adelante la llamada. Estas instrucciones de E/S son atrapadas por VM/370, que entonces las lleva a cabo como parte de la simulación del hardware verdadero. Al hacer una separación total de las funciones de multiprogramación y proporcionar una máquina extendida, cada una de las partes puede ser más sencilla, flexible y tener un fácil mantenimiento.

1.2.4 Modelo Cliente / Servidor

VM/370 ganó mucho en sencillez al trasladar gran parte del código tradicional del sistema operativo (al implantar la máquina extendida) en una capa superior, CMS. Sin embargo, VM/370 sigue siendo un programa complejo, puesto que la simulación de varios 370 virtuales no es tan simple.

Una tendencia de los sistemas operativos es el explotar más la idea de mover el código a capas superiores y eliminar la mayor parte posible del sistema operativo para mantener un núcleo mínimo. El punto de vista usual es el de implantar la mayoría de las funciones del sistema operativo en los procesos del usuario. Para solicitar un servicio, como la lectura de un bloque de cierto archivo, un proceso del usuario envía la solicitud a un proceso servidor, que realiza entonces el trabajo y regresa la respuesta.

En este modelo, lo único que hace el núcleo es controlar la comunicación entre los clientes y los servidores. Al separar el sistema operativo en partes, cada una de ellas controla una faceta del sistema, como el servicio a archivos, servicio a procesos, servicio a terminales o servicio a la memoria, cada parte es pequeña y controlable.

Además, puesto que todos los servidores se ejecutan como procesos en modo usuario y no en modo núcleo, no tienen acceso directo al hardware.

En consecuencia, si hay un error en el servidor de archivos, éste puede fallar, pero esto no afectará en general a toda la máquina.

Otra de las ventajas del modelo cliente/servidor es su adaptabilidad para su uso en sistemas distribuidos. Si un cliente se comunica con un servidor mediante mensajes, el cliente no necesita saber si el mensaje se maneja en forma local, en su máquina, o si se envía por medio de una red a un servidor en una máquina remota.

La idea anterior de un núcleo que sólo controla el transporte de mensajes de clientes a servidores y viceversa no es totalmente real. Algunas funciones del sistema operativo son difíciles, si no es que imposibles de realizar a partir de programas del usuario.

Existen dos formas de enfrentar este problema:

- ↗ Hacer que algunos procesos de servidores críticos se ejecuten en realidad en modo núcleo, con acceso total al hardware, pero de forma que se comuniquen con los demás procesos mediante el mecanismo normal de mensajes.

- ↗ Construir una cantidad mínima de mecanismos dentro del núcleo, pero manteniendo las decisiones de política relativas a los usuarios dentro del espacio del usuario. Por ejemplo, el núcleo podría reconocer que cierto mensaje enviado a una dirección especial indica que se tome el contenido de ese mensaje y se cargue en los registros del dispositivo de E/S de algún disco, para iniciar la lectura del disco. En este ejemplo, el núcleo ni siquiera inspeccionaría los bytes del mensaje para ver si son válidos o tienen algún sentido; sólo los copiaría en forma ciega en los registros del dispositivo del disco. La separación entre mecanismo y política es un concepto importante; aparece una y otra vez en diversos contextos de los sistemas operativos.

1.3 Conceptos de los Sistemas Operativos

1.3.1 Proceso

“El concepto central en todo sistema operativo, es el de proceso, el cual en su forma más simple es un programa en ejecución y consta del programa ejecutable, datos, pila, contador y algunos otros registros que contienen la información necesaria para ejecutar el programa”¹.

La forma más sencilla para tener la noción de lo que es un proceso, es pensar en los sistemas de tiempo compartido. En forma periódica el sistema operativo decide detener la ejecución de un programa y comenzar la ejecución de otro, cuando un programa se detiene en forma temporal, son guardados todos los datos correspondientes resultantes del procesamiento de tal forma de que le sea posible arrancar en su próximo turno, en el mismo estado en el que se encontraba cuando fue interrumpido. Esto quiere decir, que toda la información relativa a un programa en ejecución o proceso debe almacenarse en forma explícita en algún lugar durante la interrupción de la ejecución.

En muchos sistemas operativos, toda la información relativa a un proceso es almacenada en la tabla de procesos, la cual consta de un arreglo de estructuras, una por cada proceso existente, así un proceso que se encuentre suspendido consta de su espacio de dirección llamado imagen central y los datos de su tabla de procesos, que entre otras cosas contiene los registros de la CPU en su último estado antes de que fuera interrumpida la ejecución del programa.

Las llamadas al sistema de control de procesos son las que se ocupan de la creación y terminación de procesos. Para entender esto más fácilmente podemos verlo a través de un proceso llamado interprete de comandos o shell, el cual lee los comandos a partir de una terminal, donde el usuario los teclea manualmente y éstos crearan nuevos procesos que realizarán las actividades necesarias para efectuar la tarea que el usuario solicito; cuando estos procesos terminan el shell ejecutará las llamadas al sistema para terminar los procesos que fueron arrancados.

Si un proceso puede crear a otros es llamada proceso padre y a los procesos que fueron originados por éste se les llama procesos hijos. Si los procesos hijos pueden crear a su vez a otros procesos, entonces se tiene una estructura compleja llamada árbol de procesos. Las relaciones de dependencia entre los procesos padre e hijo son definidas por el sistema operativo de acuerdo a dos esquemas esencialmente²:

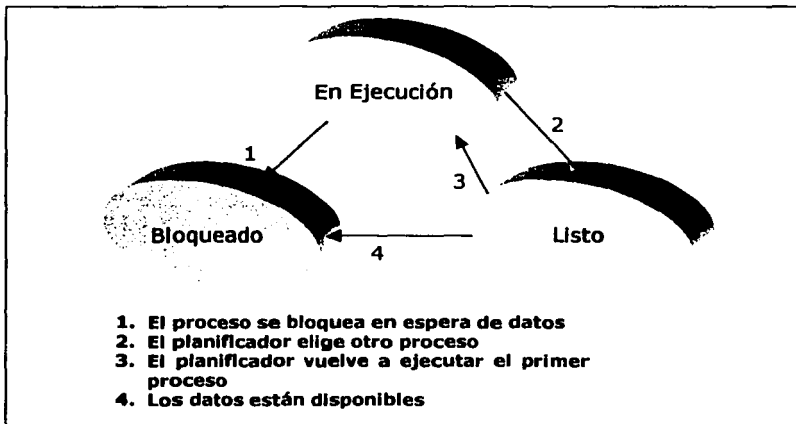
- ✓ Los procesos hijo dependen directamente de su creador, es decir estos mueren cuando su creador muere y;
- ✓ La relación de dependencia solamente se da cuando el proceso es creado, después de ese momento el proceso hijo es independiente.

Para controlar a los procesos que son iniciados y liberados por el sistema operativo, se dispone de una serie de llamadas al sistema que permiten realizar tareas como solicitar más memoria, liberar memoria, crear subprocesos o comunicarse con otros procesos con el fin de mantener una estructura dinámica.

Cada proceso es una entidad independiente de las demás que conserva sus propios datos y estado. Básicamente los procesos se pueden encontrar en tres estados distintos¹:

- ✓ En ejecución (Utiliza el CPU en el instante dado).
- ✓ Listo (ejecutable que se detiene en forma temporal para que se ejecute otro proceso).
- ✓ Bloqueado (No se puede ejecutar debido a un evento externo).

La forma en como trabajan estos estados se puede apreciar en la siguiente figura:



También el sistema operativo debe ser capaz de enviar señales a los procesos para informarles acerca del estado de una petición que hayan realizado, el estado en que se encuentra el sistema o para terminar de forma inesperada algún proceso, es decir, las señales a los procesos son una analogía a las interrupciones del hardware de una computadora, es un mecanismo mediante el cual se puede valer el sistema operativo para controlar a los diferentes procesos que se encuentran en ejecución.

En un sistema operativo que realiza multiprogramación², se mantienen registros adicionales a los mencionados como son, los identificadores del usuario que usa el proceso (UID) y del grupo al que pertenece el usuario (GID) sobre los cuales se pueden establecer mecanismos de seguridad a los procesos que corren sobre un sistema operativo de este estilo, de tal forma que si un proceso lanza una petición a otro, el sistema operativo primero hace la verificación de los identificadores mencionados, para saber si ese proceso tiene acceso a los servicios que está solicitando. Además de lo anterior el sistema operativo asigna a cada proceso en ejecución un identificador de proceso (PID) con el cual es identificado cada uno de los procesos por el sistema operativo.

TESIS CON
FALLA DE ORIGEN

Los procesos en todo momento se encuentran en competencia por el CPU, esta competencia debe de ser resuelta por el administrador de procesos del sistema operativo de que se trate. Esto se soluciona por medio de jerarquías de procesos y semáforos en el núcleo del sistema operativo.

1.3.2 Archivos

Es esencial para un sistema operativo, poder controlar archivos, ya que éstos representan la información de todo el sistema y su entorno, por lo que es evidente la necesidad de crear llamadas al sistema que permitan la manipulación correspondiente a la creación, eliminación, lectura y escritura de archivos. También es importante proporcionar un espacio donde almacenarlos y organizarlos donde se introduce el concepto de directorio como una forma de agrupar y catalogar archivos, por lo cual, también se generarán llamadas al sistema para manipular éstos últimos.

La forma de organizar a los procesos y archivos es muy similar, éstos conservan una estructura de árbol, aunque estas estructuras pueden llegar a diferir en su magnitud y profundidad, así mismo, los mecanismos de seguridad empleados para los archivos son similares a los empleados para los procesos, en los aspectos de propiedad y protección de acuerdo a las reglas implementadas por el sistema operativo.

Cada uno de los archivos que se encuentre dentro de una estructura de directorios puede ser referido mediante su "ruta de acceso", la cual indica la colocación exacta del archivo dentro de una jerarquía que comienza desde el directorio de raíz.

Todos los procesos tienen activo un directorio de trabajo en el que se buscan aquellos nombres de archivos que no sean referidos por medio de su ruta de acceso completa. Los procesos, pueden modificar su directorio de trabajo mediante una llamada al sistema que especifique uno nuevo.

Si se habla de un sistema que realice multiprogramación, es decir, varias personas hacen un acceso concurrente a la misma computadora, es importante tener un medio de protección para los archivos de cada persona. Al hablar de distintos sistemas operativos se pueden encontrar distintos esquemas de protección que son activados antes de que un archivo pueda ser leído o escrito:

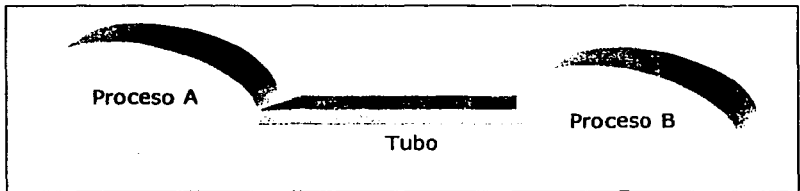
↙ Si el acceso es permitido, el sistema operativo regresara un descriptor del archivo o entrada de manejo para que sea usado en los accesos subsecuentes.

↙ Si el acceso es denegado, el sistema operativo regresara un código de error.

Muchos sistemas operativos proporcionan una abstracción para los usuarios, con el fin de que lleven a cabo las operaciones de entrada / salida sin tener que preocuparse por los detalles del hardware. Esta abstracción representa a cada dispositivo como un archivo especial, en el cual se puede escribir o leer según se desee, mediante las mismas llamadas que son usadas para lectura y escritura de archivos comunes.

1.3.3 Tubos (pipeline)

Un tubo es una especie de pseudo archivo que se puede utilizar para conectar dos procesos. Sin embargo, esta solución para la comunicación entre diferentes procesos envuelve una serie de algoritmos complejos que se refieren a la inanición de procesos o espera eterna. El algoritmo mas conocido en este caso, es el de Dijkstra titulado "la cena de los filósofos" en el cual se hace necesaria cierta planificación de procesos



Cuando un proceso A desea enviar datos al proceso B, escribe en el tubo como si fuera un archivo de salida. El proceso B puede leer los datos, al leer el tubo como si fuera un archivo de entrada, de esta forma, la comunicación entre procesos se ve como la lectura y escritura ordinaria en archivos.

TESIS
FALLA DE ORIGEN

1.3.4 Llamadas al Sistema

Los programas del usuario se comunican con el sistema operativo y le solicitan servicio mediante las llamadas al sistema. A cada una de éstas le corresponde un procedimiento de la biblioteca que puede ser invocado por los programas del usuario. Este procedimiento pone los parámetros de la llamada al sistema en un lugar específico con la finalidad de ocultar los detalles de la instrucción (o llamada) y hacer que las llamadas al sistema parezcan llamadas comunes a un procedimiento.

Cuando el sistema operativo recupera el control después una llamada éste examina los parámetros para ver si son válidos; en ese caso, desarrolla el trabajo solicitado. Al terminar, el sistema operativo coloca un código de estado en un registro, para indicar si tuvo éxito o fracaso; además, regresa el control al procedimiento de biblioteca. Este procedimiento regresa entonces a quien hizo la llamada, proporcionando el código de estado como un valor de función. A veces, dentro de los parámetros regresan valores adicionales.

1.3.5 Interprete de Comandos

Un interprete de comandos no es parte del sistema operativo, sin embargo, éste es uno de los programas que hace un uso más extenso de las llamadas al sistema y proporciona la interfaz principal de interacción entre el sistema operativo y el Usuario.

El interprete de comandos o shell lee los comandos a través de una terminal, al recibir la entrada del usuario el interprete de comandos crea un nuevo proceso que realizara el trabajo solicitado, siendo éste el que realice una llamada al sistema para terminar el trabajo.

El interprete de comandos es la herramienta mas poderosa de un sistema operativo y dependerá de la maestría y habilidad del usuario el explotar al máximo las características del interprete de comandos que se le presente. Un ejemplo de esto son los llamados "scripts de shell" los cuales hacen uso de una serie de instrucciones reservadas del interprete de comando para actuar de la misma forma que un programa lo hace.

Estos scripts son cargados por el interprete de comandos a la memoria y luego son ejecutados renglón por renglón. Dependiendo de la versatilidad y robustez del interprete de comandos que se use será la complejidad que se pueda alcanzar en los scripts, que va desde un simple archivo de procesamiento por lotes hasta programas completos que pudieran manipular a muchos otros.

Capitulo II. Historia de los Sistemas Operativos

La historia de los sistemas operativos ha estado ligada a los avances tecnológicos del hardware, a la aceptación en un mercado creciente, a las alianzas de varias compañías para desarrollar software específico para un sistema operativo, etc. Este es el medio ambiente para desarrollar sistemas operativos que sean aceptados en el mercado. Es por eso que es importante analizar el desarrollo de los sistemas operativos a través de las diferentes generaciones de computadoras, para tener una mayor comprensión del desarrollo de éstos.

2.1 Las generaciones de computadoras

2.1.1 Primera Generación (1945 / 1955) Bulbos y Conexiones

Durante la segunda guerra mundial a mitad de la década de los cuarenta, Howard Aiken, John Von Neuman, J. Presper Eckert y William Mauchley, así como Konrad Zuse, entre otros, lograron construir máquinas de cálculo mediante bulbos. Estas máquinas eran enormes y llenaban cuartos completos con decenas de miles de bulbos. En esos primeros días, un solo grupo de personas diseñaba, construía, programaba, operaba y daba mantenimiento a cada máquina. Toda la programación se llevaba a cabo en lenguaje de máquina absoluto y con frecuencia se utilizaban conexiones para controlar las funciones básicas de la máquina. Los lenguajes de programación eran desconocidos. No se oía de los sistemas operativos. El modo usual de operación consistía en que el programador reservaba cierto periodo en una hoja de reservación pegada en la pared, iba al cuarto de la máquina y realizaba una serie de conexiones correspondientes a números binarios y operaciones que hoy podríamos llamar como funciones básicas. La inmensa mayoría de los problemas eran cálculos numéricos directos, como por ejemplo el cálculo de valores para tablas de senos y cósenos.

A principios de la década de los cincuenta, la rutina mejoró un poco con la introducción de las tarjetas perforadas. Fue entonces posible escribir los programas en las tarjetas y leerlas en vez de insertar conexiones; por lo demás, el proceso era lo mismo.

2.1.2 Segunda Generación (1955 / 1965) Transistores y Sistemas de Procesamiento por Lotes.

La introducción del transistor a mediados de los años cincuenta modificó en forma radical a las computadoras, estas se volvieron confiables, de forma que podían fabricarse y venderse a clientes. Por primera vez, hubo una clara separación entre los diseñadores, constructores, operadores, programadores y personal de mantenimiento. Las máquinas correspondientes a esta generación se aislaban del exterior en cuartos de cómputo con aire acondicionado especial y un equipo de operadores profesionales para la ejecución.

Para ejecutar un programa o conjunto de programas, el programador debía escribir el programa en tarjetas perforadas usando algún lenguaje de programación como FORTRAN o Ensamblador, después debía llevar el paquete de tarjetas perforadas al cuarto de lectura y dárselas a uno de los operadores, el cual revisaba las tarjetas y si era necesario cargar algún compilador, el operador debía primero meter el paquete de tarjetas perforadas del compilador y luego alimentar el programa suministrado por el programador.

Es notable que se desperdiciaba demasiado tiempo de cómputo mientras los operadores caminaban en el cuarto de la máquina y dado el alto costo del equipo, se buscó una forma más rápida para reducir el tiempo invertido. La solución en su momento fue la del procesamiento por lotes. La idea subyacente era coleccionar varios trabajos en el cuarto de entrada y leerlos después en una cinta magnética mediante una computadora pequeña y relativamente barata, como la IBM 1401, que era muy buena en la lectura de tarjetas y copiado de cintas para impresión de la salida.

Después juntar un lote de tarjetas, la cinta se embobinaba y era llevada al cuarto de la máquina, donde se colocaba en una unidad de cinta. El operador cargaba entonces un programa especial que es de hecho el antecesor del actual sistema operativo, el cual leía cada trabajo de la cinta y lo ejecutaba. La salida se escribía en una segunda cinta, en vez de imprimirse. La impresión de las salidas se realizaba en una 1404 para impresión fuera de línea.

Las tarjetas perforadas que se usaban como entrada se estructuraban de la siguiente forma:

1. Se comenzaba con una tarjeta \$JOB, en la que se especificaba el tiempo de ejecución en minutos, el número de cuenta al que se hace el cargo y el nombre del programador.
2. Después venía una tarjeta \$FORTRAN, la cual indicaba al sistema operativo que cargara el compilador de FORTRAN de la cinta de sistema.
3. Seguía el programa a compilar y una tarjeta \$LOAD, para indicar al sistema operativo que cargara el programa objeto recién compilado.
4. Después venía la tarjeta \$RUN, que indicaba al sistema operativo que ejecutara el programa con los datos que le seguían.
5. Por último, la tarjeta \$END marcaba el final del trabajo.

Estas primitivas tarjetas de control fueron los antecesores de los modernos lenguajes de control de las tareas y los intérpretes de comandos.

Las grandes computadoras de la segunda generación se utilizaron principalmente para los cálculos científicos y de ingeniería; por ejemplo, en la resolución de ecuaciones diferenciales parciales. Por lo general, se programaba en FORTRAN y lenguaje ensamblador. Los sistemas operativos más comunes eran FMS (Fortran Monitor System) e IBSYS, el sistema operativo de IBM para la 7094.

2.1.3 La Tercera Generación (1965 / 1980). Circuitos Integrados y multiprogramación.

A principios de la década de los sesenta, la mayoría de los fabricantes de computadoras tenían dos líneas de productos, distintas y totalmente incompatibles:

- ✓ Por un lado, estaban las computadoras científicas de gran escala, orientadas a palabras, como la 7094, la cual se utilizaba en cálculos científicos y de ingeniería.
- ✓ Por el otro, estaban las computadoras comerciales, orientadas a caracteres, como la 1401, de uso común para el ordenamiento de cintas e impresión por los bancos y las compañías aseguradoras.

El desarrollo y mantenimiento de dos líneas de productos completamente distintas era una propuesta cara para los fabricantes. Además, la mayoría de los nuevos clientes necesitaban una máquina pequeña, pero con el tiempo deseaban una máquina más grande, que ejecutara sus programas anteriores pero de una manera más rápida.

IBM intentó resolver ambos problemas a la vez; para ello, introdujo el Sistema/360. El 360 era una serie de máquinas con software compatible, con un rango desde la 1401 hasta máquinas más poderosas que la 7094. Estas sólo diferían en el precio y el desempeño. Puesto que las máquinas tenían la misma arquitectura y conjunto de instrucciones, al menos en teoría, los programas escritos para una máquina podían ejecutarse en las otras. Además, la 360 se diseñó para hacer cálculos tanto científicos como comerciales. Así, una sola familia de máquinas podría satisfacer las necesidades de todos los clientes. En los años siguientes, IBM produjo sucesores compatibles como con la línea 360, pero con una tecnología más moderna, conocidas como las series 370, 4300, 3080 y 3090.

La 360 fue la primera línea principal de computadoras que utilizó los circuitos integrados, lo que proporcionó una gran ventaja de precio y desempeño con respecto de las máquinas de la segunda generación, construidas a partir de transistores individuales. Tuvo un éxito inmediato; la idea de las computadoras compatibles pronto fue adoptada por los demás fabricantes. Las descendientes de esas máquinas siguen utilizándose en los grandes centros de cómputo de la actualidad.

La enorme fuerza de la idea de una familia fue al mismo tiempo su mayor debilidad. La intención era que todo el software, sistema operativo incluido, debía funcionar en todos los modelos. Debía poder ejecutarse en sistemas pequeños, que con frecuencia reemplazaban a la 1401 en el copiado de tarjetas en la cinta; también en sistemas muy grandes, que con frecuencia reemplazaban a la 7094 en el pronóstico del tiempo y otros cálculos pesados. Debía funcionar en los ambientes científicos y comerciales. Pero por encima de todo, debía ser eficaz en todos estos diversos usos.

No había forma de que IBM pudiera escribir una pieza de software que cumpliera con todos estos requisitos en conflicto. El resultado fue un sistema operativo enorme y extraordinariamente complejo, tal vez del doble o triple de magnitud que FMS. Consta de millones de líneas de lenguaje ensamblador, escrito por miles de programadores, con miles y miles de errores, que requerían de un flujo continuo de nuevas versiones, en un intento por corregirlos.

Cada nueva versión resolvía algunos errores pero introducía otros nuevos, por lo que es probable que el número de errores fuera constante con respecto del tiempo.

A pesar de su enorme tamaño y sus problemas, OS/360 y los sistemas operativos similares de la tercera generación producidos por otros fabricantes de computadoras realmente pudieron satisfacer, en forma razonable, a la mayoría de sus clientes, también popularizaron varias técnicas fundamentales, ausentes de los sistemas operativos de la segunda generación, de las cuales la más importante tal vez sea la de multiprogramación. En la 7094, si el trabajo en curso se detenía en espera de una cinta o de que concluyera otra labor de E/S, el CPU simplemente se detenía a esperar hasta que la E/S finalizara. En los pesados cálculos científicos, con una cuota para el uso del CPU, la E/S no es frecuente, por lo que el tiempo perdido era insignificante. Pero en el procesamiento de los datos comerciales, el tiempo de espera para la Entrada / Salida podría representar 80 o 90 por ciento del tiempo total, por lo que algo debía hacerse para evitar que la CPU estuviera inactiva por mucho tiempo.

La solución que se desarrolló fue la de partir la memoria en varios segmentos, con un trabajo distinto en cada partición. Mientras que un trabajo esperaba a que concluyera la Entrada / Salida otro podía estar utilizando el CPU. Si se podían mantener en la memoria principal los trabajos suficientes a la vez, el CPU estaría ocupado el 100 por ciento del tiempo. El mantenimiento de varios trabajos en memoria a la vez requería de un hardware especial, para proteger cada trabajo, la 360 y otras máquinas de la tercera generación contaban con dicho hardware que proporcionaba ese tipo de seguridad.

Otra de las características principales de los sistemas operativos de la tercera generación era la capacidad de leer trabajos de las tarjetas al disco, tan pronto como llegaran al cuarto de cómputo. Así, siempre que concluyera un trabajo, el sistema operativo podía cargar un nuevo trabajo del disco en la partición que quedaba desocupada y ejecutarlo.

Esta técnica se llama spooling (de Simultaneous Peripheral Operation On Line) y también se utilizó para las salidas. Con esta técnica las 1401 ya no fueron necesarias y desapareció el transporte de las cintas de un lado al otro.

Aunque los sistemas operativos de la tercera generación eran adecuados para los grandes cálculos científicos y la ejecución de un procesamiento de grandes cantidades de datos comerciales, seguían siendo en esencia sistemas de procesamiento por lotes. Con los sistemas de la tercera generación, el tiempo transcurrido entre el envío de un trabajo y la obtención de la salida era a menudo de varias horas, por lo que una sola coma mal colocada podía provocar que la compilación fallara, con lo que el programador perdía la mitad del día.

El deseo de aprovechamiento al máximo por parte de los programadores preparó el camino para el tiempo compartido (time sharing), variante de la multiprogramación, en la que cada usuario tenía una terminal en línea. En un sistema con tiempo compartido, si de 20 usuarios que están conectados, 17 de ellos, no están ocupando el CPU, este puede ser asignado a los tres trabajos que requieren servicio. Puesto que las personas que depuran los programas utilizan, por lo general, comandos cortos en vez de largos, la computadora podía dar un servicio rápido e interactivo a varios usuarios y, por otra parte, tal vez realizar trabajos con grandes lotes si la CPU está inactiva. Aunque el primer sistema serio de tiempo compartido (CTSS) fue desarrollado en MIT en una 7094 con modificaciones especiales, éste no se popularizó sino hasta que se difundió, durante la tercera generación, el hardware necesario para protección de procesos.

Después del éxito del sistema CTSS, las compañías MIT, Bell Labs y General Electric decidieron emprender el desarrollo de una computadora que soportara a cientos de usuarios con tiempo compartido. Su modelo fue el sistema de distribución de electricidad. Los diseñadores de este sistema, llamado MULTICS (Multiplexed Information and Computing Service), imaginaron una formidable máquina que proporcionaría corriente a cada persona en Boston (La idea de que, dentro de 20 años, máquinas tan poderosas como la GE-645 se vendieran como computadoras personales por unos cuantos miles de dólares, no era mas que ciencia-ficción, en aquella época).

MULTICS introdujo muchas ideas seminales en la bibliografía sobre computación, pero su construcción era mucho más ardua de lo esperado. Bell Labs abandonó el proyecto y General Electric se retiró completamente del negocio de la computación. MULTICS llegó a ejecutarse de una manera que fuera útil en un ambiente de producción en el MIT y algunos cuantos lugares más, pero este concepto de computadora fue un fracaso. Aun hoy en día, MULTICS sigue teniendo una enorme influencia en los sistemas operativos subsecuentes.

Otro desarrollo fundamental durante la tercera generación fue el crecimiento fenomenal de las mini computadoras, a partir de la DEC PDP-1 en 1961 que sólo tenía 4K de palabras de 18-bits, que para cierto tipo de trabajo no numérico, era casi tan rápida como la 7094 y dio lugar a toda una nueva industria. Le siguió una serie de otras PDP, hasta llegar a la PDP-11.

Uno de los científicos de Bell Labs que trabajó en el proyecto MULTICS, Ken Thompson, se encontró una pequeña mini computadora PDP-7 que nadie utilizaba e intentó escribir una versión desprotegida de MULTICS para un solo usuario. Este trabajo desembocó en el sistema operativo UNIX, que en la actualidad domina los mercados de las mini computadoras y estaciones de trabajo, así como otros mercados.

2.1.4 La cuarta generación (1980 / 2000) Computadoras Personales

Con el desarrollo de los circuitos LSI (Large Scale Integration), y VLSI (Very Large Scale Integration) tabletas (chips) con miles de transistores en un centímetro cuadrado de silicio, se inició la era de la computadora personal y las supercomputadoras. En términos de arquitectura, las computadoras personales no eran muy distintas. Mientras que la mini computadora permitió que un departamento de una empresa o universidad tuviera su propia computadora, el chip microprocesador hizo posible que una sola persona tuviera su propia computadora personal. Las computadoras personales más poderosas utilizadas por empresas, universidades e instalaciones de gobierno reciben el nombre genérico de estaciones de trabajo, pero en realidad sólo son computadoras personales grandes. Por lo general, se conectan entre sí mediante una red.

La amplia disponibilidad del poder de cómputo, en particular, el poder de cómputo de gran interacción y que, por lo general, cuenta con gráficos excelentes, condujo al crecimiento de una gran industria de producción de software para las computadoras personales. Gran parte de este software es amigable con el usuario, lo que indica que está destinado al usuario que no sabe nada acerca de las computadoras y que además no tiene la más mínima intención de aprender. Este fue un cambio fundamental con respecto a OS/360, cuyo lenguaje de control de tareas, JCL, era tan misterioso que se habían escrito libros enteros acerca de él.

Dos sistemas operativos han dominado la escena de las computadoras personales y las estaciones de trabajo: MS-DOS de Microsoft y UNIX. MS-DOS tiene un amplio uso en la IBM PC y otras máquinas con la CPU 8088 de Intel y sus sucesores sean, 80286, 80386 y x86. Aunque la versión inicial de MS-DOS era relativamente primitiva, las subsecuentes versiones han incluido características más avanzadas, entre ellas, algunas de UNIX.

El otro contendiente principal es UNIX, que domina en las computadoras que no utilizan a Intel, así como en las estaciones de trabajo, en particular en las que poseen chips de alto desempeño con tecnología RISC. Estas máquinas tienen en general la potencia de cómputo de una mini computadora.

Un interesante desarrollo que comenzó a llevarse a cabo a mediados de la década de los ochenta ha sido el crecimiento de las redes de computadoras personales con sistemas operativos de red y sistemas operativos distribuidos.

En un sistema operativo de red, los usuarios están conscientes de la existencia de varias computadoras y pueden conectarse con máquinas remotas y copiar archivos de una máquina a otra. Cada máquina ejecuta su propio sistema operativo local y tiene su propio usuario.

Por el contrario, un sistema operativo distribuido es aquél que aparece ante sus usuarios como un sistema tradicional de un solo procesador, aun cuando esté compuesto por varios procesadores. En un sistema distribuido verdadero, los usuarios no necesitan saber del lugar donde su programa se ejecute o del lugar donde se encuentran sus archivos; eso debe ser manejado en forma transparente automática y eficaz por el sistema operativo.

Los sistemas operativos de red no tienen diferencias fundamentales con los sistemas operativos de un solo procesador. Es obvio que necesitan un controlador de interfaz de la red y algo de software de bajo nivel para dirigirlo, al igual que programas que permitan la conexión y el acceso a un archivo remoto, pero estas características adicionales no modifican la estructura esencial del sistema operativo.

2.2 Windows y Linux en la línea del Tiempo

A continuación se presenta una de manera concisa un breve resumen de la historia de ambos sistemas operativos, con el fin de comparar el enfoque de su desarrollo así como su crecimiento a través del tiempo.

2.2.1 Microsoft Windows

1993: Windows NT Advanced Server 3.1

En julio de 1993 Microsoft libero su primer sistema operativo para servidores el cual fue diseñado para actuar como un servidor dedicado en ambientes cliente / servidor ofreciendo como puntos principales del sistema: escalabilidad, tolerancia a fallas e interoperabilidad basada en estándares.

Microsoft coloco en el mercado este producto como un servidor de aplicaciones para Novell Netware, Banyan VINES, y Microsoft Networks, con el objeto de proveer una plataforma para soluciones de negocio a gran escala.

1994: Windows NT Server 3.5

Esta versión incluyo mejoras en rendimiento del servidor y mas elementos de conectividad, especialmente para Novell Netware y ambientes UNIX, además de incluir aplicaciones administrativas para el servidor.

1995: Windows NT Server 3.51

Microsoft incluyo utilerías para ayudar a sus usuarios a administrar sus licencias de acceso a clientes para la familia de productos de Microsoft BackOffice y utilerías para instalar a través de la red Windows 95.

1996: Windows NT Server 4.0

En esta actualización del sistema operativo Microsoft libero por primera vez un sistema operativo de 32 bits reales con las siguientes características: mejoras en la velocidad de comunicación a través de la red, servicios de archivos e impresión mas rápidos, soporte mas robusto A las aplicaciones, un servidor de web integrado, herramientas de administración para intranets.

1997: Windows NT Server 4.0, Enterprise Edition

Este sistema operativo esta completamente basado en el anterior, agregando capacidades para ambientes corporativos de misión critica entre las cuales se incluyen mejoras en la escalabilidad y disponibilidad, además de proporcionar servicios para facilitar el desarrollo de aplicaciones.

Entre las facilidades de desarrollo esta el Microsoft Transaction Server que ayuda en el desarrollo de aplicaciones para Intranets en internet y el Microsoft Message Queue Server que permite comunicación asíncrona entre aplicaciones para el intercambio de información.

Dentro del sistema operativo se incluyeron servicios como Microsoft Cluster Service y Microsoft Windows NT Server Load balancing service.

1998: Windows NT Server 4.0, Terminal Server Edition

Básicamente esta versión del sistema operativo incluye 3 componentes:

- ✓ Windows NT Server multi-user core, el cual hace posible tener varias sesiones de manera simultanea dentro del servidor.
- ✓ Remote Desktop Protocol. Este habilita en el cliente la comunicación con el servidor que tiene el servicio de terminal a través de la red.
- ✓ Windows based client software. Permite desplegar la interfaz de Windows de las aplicaciones remotas.

2000: Windows 2000 Server Family

La presentación de una familia de servidores para cumplir con diferentes necesidades del Mercado fue el resultado de evolución natural de los sistemas operativos de Microsoft los cuales son:

- ↗ Windows 2000 Server. Es un sistema operativo diseñado para trabajos departamentales de acceso a archivos, Impresión, Web, y aplicaciones.
- ↗ Windows 2000 Advanced Server. Ofrece soluciones para aplicaciones críticas en Web y de línea de negocios.
- ↗ Windows 2000 Datacenter Server. Es un sistema operativo para sistemas a gran escala, que requieran de alta disponibilidad y tolerancia a fallas.

2002: Windows .NET Server

Como su predecesor Windows .NET Server fue diseñado para cubrir necesidades de negocio a todos los niveles, de tal manera que esta versión del sistema operativo se puede encontrar en 4 diferentes configuraciones:

- ↗ Windows .NET Standard Server. El cual es un sistema operativo para servidor de propósito general que cubre con las necesidades básicas de todo negocio.
- ↗ Windows .NET Enterprise Server. Incluye capacidades para el desarrollo de Web Services e infraestructura de conectividad.
- ↗ Windows .NET Datacenter Server. Como en su versión anterior incluye capacidades para aplicaciones de misión crítica y tolerancia a fallas.
- ↗ Windows .NET Web Server. Esta versión da facilidades para el desarrollo de aplicaciones en Internet.

2.2.2 GNU Linux

El inicio del desarrollo de este sistema operativo se sale un poco de lo convencional, tomando en cuenta que como la mayoría de los proyectos de software de libre distribución comenzó con una buena intención por parte de algún desarrollador en Internet.

En Agosto de 1991 un estudiante finlandés puso un mensaje en el News Group comp.os.minix con las siguientes palabras:

"Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones."

El nombre del estudiante era Linus Torvalds y el sistema operativo al que se refería era lo que hoy en día conocemos como Linux.

Este sistema operativo tipo POSIX, fue creado por cientos de desarrolladores alrededor del mundo que creyeron en la idea expresada en ese mensaje de Linus Torvalds.

Este esfuerzo masivo de desarrollo alrededor del mundo no se encuentra coordinado por nadie y es uno de los fenómenos más inexplicables que ha sucedido en Internet, un estudiante en Rusia compra una nueva Tarjeta Madre y escribe un controlador de dispositivo que necesita para una nueva característica; un Administrador del sistema necesita un sistema de respaldo y escribe un programa que lo ayude a facilitar esa tarea.

Linux puede ser obtenido de Internet sin necesidad de pagar dinero por el, y la mayoría del software ya sea de sistema o de aplicación puede ser bajado de Internet de la misma manera que el sistema operativo.

Un grupo de programadores del Norte de Carolina en los Estados Unidos se pusieron como objetivo hacer a Linux más fácil de usar. Lograron hacer una recopilación ordenada de software en diversos paquetes que les permitieron sacar al mercado una distribución de Linux llamada Red Hat, la cual ha sido por mucho tiempo la distribución de Linux más usada alrededor del mundo, debido a características que hacen más fácil su instalación administración y distribución.

La compañía Red Hat fue fundada en Marzo de 1993, a diferencia de otras distribuciones de Linux su éxito se debió principalmente a que la forma de distribución fue por medio de paquetes que encerraban funcionalidades específicas. Este tipo de distribución permite una forma fácil y ordenada de actualización y organización del software que hasta ese momento no existía para una distribución de Linux.

A continuación se presenta una breve descripción de los acontecimientos mas importantes en la distribución de Red Hat:

1993: Fundación de Red Hat.

En Marzo de este año se funda legalmente la compañía Red Hat alcanzando en ese año la cantidad de 100,000 usuarios de Linux.

1994: Creación de la forma de distribución

A finales de 1994 se creo por fin la forma de distribución de Linux que impulsaría a esta compañía: el sistema Red Hat Package Manager (RPM).

1995: El inicio de una compañía sólida alrededor del mundo.

para este momento existían 1,500,000 usuarios de Linux alrededor del mundo y los creadores de Red Hat Bob Young y Marc Ewing deciden poner todos sus esfuerzos en una aventura llamada Red Hat Software, Inc.

1998: Red Hat 5.2

Esta es probablemente la versión mas conocida en el mundo de la distribución de Linux llamada Red Hat, debido a que las facilidades de instalación y administración superaban por mucho a sus predecesoras.

Adicionalmente compañías como Intel, Netscape Communications invierten en Red Hat grandes cantidades de dinero e IDC informa al mundo que los 750,000 servidores Linux instalados alrededor del mundo representan un incremento del 212 por ciento con respecto al año anterior dejando dramáticamente por debajo las tasas de crecimiento de base instalada de competidores como Windows, NetWare y otros servidores UNIX.

1999: Red Hat 6.0

Se contabilizan 15,000,000 de usuarios de Linux alrededor del mundo, debido a esto, compañías como Compaq, IBM, Novell, Oracle, SAP toman posiciones equitativas en cuanto al sistema operativo sobre maquinas x86.

Red Hat 6.0 es vendido como sistema operativo de fabrica de computadoras Dell dando igualdad de condiciones de competencia contra la distribución de Windows para servidores imperante en ese momento.

El 4 de octubre de este año es entregada la versión 6.1 que incluía mejoras significativas a la versión 6, en cuanto a las facilidades de instalación y administración.

2000: Red Hat 7.0

En febrero 9 de este año IDC anuncia a Linux como el sistema operativo con la tasa de crecimiento de plataforma instalada mas elevada llegando a un 90% y teniendo un 25% del total del mercado de sistemas operativos para servidor.

Linux Red Hat obtiene importantes premios de la industria por separado:

- ✓ El premio de la Network Magazine "Product of the Year"
- ✓ El premio de Computer Reseller News "Editors' Choice"

IBM anuncia que Linux será soportado por completo en su línea de servidores eServer incluyendo dentro de Linux productos como Lotus y Tivoli

2001: El kernel 2.4

En enero 5 de este año Linus Torvalds libera la versión 2.4 del kernel de Linux, que incluye mejoras de performance, seguridad y una nueva y mejorada versión de la estructura del kernel principal de Linux haciéndolo mas eficiente y estable.

Es liberada a principios de este año la versión 7.1 de Linux Red Hat que incluye la nueva versión del kernel.

Es liberado en octubre de este año la versión de Linux Red Hat 7.2 que provee a desarrolladores y usuarios una plataforma homogénea para todo tipo de dispositivos de computo que van desde Hand Helds hasta servidores OS/390.

El Wall Street Journal revela que Microsoft esta usando código de Open Source para algunos de sus productos principales

2002: Otras plataformas de hardware

La nueva estructura del kernel y las mejoras a este durante todos estos años han permitido a este sistema operativo ser fácilmente adaptable a nuevas plataformas de hardware, un ejemplo de esto es el anuncio de las versiones de Linux Red Hat 7.2 para procesadores Titanium de Intel ofreciendo una gran escalabilidad para empresas de nivel mundial, o la migración de Linux para procesadores Alpha de Compaq en el primer cuarto del año 2002.

Capítulo III. El estándar POSIX

1003.1

El continuo desarrollo de los sistemas operativos y la gran variedad de plataformas hizo necesario desarrollar estándares en la industria del software para proporcionar cierta facilidad de transporte del código. Éstos estándares fueron desarrollados centrados principalmente en el sistema operativo UNIX, ya que era el más difundido en ese entonces. Actualmente existen muchos estándares generados para proporcionar independencia de plataforma a nivel código fuente, como son ANSI C, XPG3, SVR4, BSD y POSIX. Este último es el más aceptado y difundido; no solamente las plataformas tipo UNIX sino también otros sistemas operativos como Windows, que cumplen con las especificaciones de este estándar que se refiere a la interfaz de los programas de aplicación con el sistema operativo, es decir, las llamadas al sistema.

En el presente capítulo centraremos la atención principalmente en el documento 1003.1 del estándar POSIX que define para un sistema operativo: una interfaz y un ambiente estándar para soportar el transporte de aplicaciones a nivel código fuente.

El apartado 1003.1 (en adelante llamado POSIX.1) es soportado parcialmente por Windows permitiendo que aplicaciones desarrolladas bajo POSIX.1 corran en este sistema operativo, pero con amplias modificaciones de profundidad para funciones muy específicas como los conceptos de abstracción de hardware (HAL). Por su parte Linux cumple con todas las características del estándar.

3.1 Introducción a los estándares POSIX

La existencia de diferentes versiones de UNIX ha permitido crear mucho software innovador, sin embargo, esto también ha sido causa de que el sistema operativo haya tomado diferentes tendencias de tal forma que muchos programas que fueron escritos para una versión de UNIX en específico, no corrieran en otra, por lo que se hizo necesario crear una serie de estándares que permitieran a programadores y usuarios unificar los diferentes conceptos obtenidos para crear un UNIX estándar.

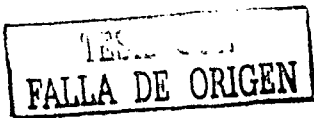
En 1984 el grupo de usuarios "/usr/group" ahora llamado UNIFORM comenzó la tarea de la especificación de un UNIX estándar, que hoy en día se conoce como la serie POSIX, la cual estandariza las interfaces para los programas de aplicación, sin embargo, este estándar no dice nada acerca de como deben ser implementadas las interfaces definidas.

La especificación de esas interfaces se encuentra organizada en nueve secciones, cada una tratando un punto muy específico como se explica a continuación:

- POSIX 1003.1 Interfaces básicas del sistema para el lenguaje C incluyendo extensiones para programación en tiempo real (1003.1b) e hilos (1003.1c)
- POSIX 1003.2 Especifica el shell y sus herramientas así como el procesamiento por lotes (1003.2d)
- POSIX 2003 Métodos de prueba para mediciones adoptado en 1991
- POSIX 2003.1 Métodos de prueba para POSIX 1003.1
- POSIX 2003.2 Métodos de prueba para POSIX 1003.2
- POSIX 1003.5 Adecuación del lenguaje Ada
- POSIX 1003.9 Adecuación del lenguaje FORTRAN
- POSIX 1003.17 Estándar para los servicios de directorio X.500, un protocolo que permite buscar en múltiples directorios distribuidos como si fuera una entidad única.
- POSIX 1387.2 Administración del sistema: este es principalmente un estándar para la instalación de software.

POSIX es el nombre de una colección de estándares de software basados pero no limitados al sistema UNIX sus siglas en inglés significan "Interfaz Portable del Sistema Operativo" (Portable Operating System Interfaces). Este estándar es desarrollado por grupos de trabajo de la IEEE en los cuales la participación esta abierta a cualquier persona. Por éstas y otras razones los estándares POSIX se refieren a los sistemas abiertos.

El punto principal tratado en los estándares POSIX es promover el transporte de aplicaciones entre diferentes sistemas. El desarrollo de este estándar es a través de 16 comités distribuidos en las secciones que trata el estándar, en las cuales puede participar cualquier persona.



El estándar POSIX es desarrollado bajo un enfoque de sistemas abiertos, lo cual da libertad a los desarrolladores de software de agregar características que no se encuentren en el estándar a su sistema operativo y que este siga siendo compatible con POSIX, siempre y cuando estas características sean bien identificadas y no interfieran con las bases que presenta el estándar.

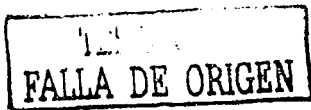
3.2 POSIX 1003.1

Este documento es considerado el más importante de toda la serie POSIX, esta compuesto de tres partes principales:

1. Definiciones para terminología y objetos referidos en el documento.
2. Interfaces del sistema y subrutinas
3. Usos de las interfaces, incluyendo el transporte de software entre plataformas distintas
4. Formatos de almacenamiento
5. Manejo de errores y recuperación

Originalmente POSIX.1 especificaba la sintaxis y la semántica de 203 funciones y el contenido de varias estructuras de datos, así como una serie de constantes necesarias para plantear las bases del estándar organizadas en diferentes librerías del lenguaje C de las cuales presentamos las más importantes a continuación:

Biblioteca	Descripción
<cpio.h>	Valores para archivos tipo cpio
<dirent.h>	Entradas para directorios
<fcntl.h>	Control de archivos
<grp.h>	Funciones para el manejo del archivo "group"
<pwd.h>	Funciones para el manejo del archivo "passwd"
<tar.h>	Valores para archivos ".tar"
<termios.h>	Funciones de Entrada / Salida para terminales
<unistd.h>	Constantes simbólicas del sistema
<utime.h>	Funciones de fechas para archivos
<sys/stat.h>	Estado de los archivos
<sys/times.h>	Tiempos de proceso
<sys/types.h>	Tipos nativos de datos
<sys/utsname.h>	Nombre del sistema
<sys/wait.h>	Control de procesos



Las bibliotecas presentadas anteriormente corresponden a los procesos principales del sistema operativo como tiempos de proceso, tipos nativos de datos, funciones de Entrada / Salida, así como a las interfaces de sus formatos de archivo principales (tar, cpio).

Cabe destacar dentro del estándar, la definición de un par de bibliotecas las cuales representan la columna medular del estándar en cuanto a su implantación real en una plataforma específica de hardware:

En las primeras páginas del estándar se introduce la definición de una biblioteca llamada <limits.h> la cual establece los valores máximos de los tipos de datos para una implementación de un sistema operativo tipo POSIX o compatible con POSIX.

Esta biblioteca es uno de los puntos claves para fomentar la el transporte del código a otras plataformas de hardware sin necesidad de hacer modificaciones substanciales. Entre los valores más importantes de ésta se encuentra WORD_BIT, el cual establece el tamaño de palabra para el sistema operativo y el número de bits para el tipo de dato entero (int), independientemente de la plataforma en donde se realice la implementación.

A continuación damos la tabla de valores definida por el estándar:

Nombre	Descripción	Valor
ARG_MAX	Número máximo de argumentos para la función exec() en bytes.	4096
CHAR_BIT	Número máximo de bits de un tipo char.	8
CHAR_MAX	Valor máximo del tipo de datos char. <input type="checkbox"/> Si no tiene signo <input type="checkbox"/> Si tiene signo	UCHAR_ MAXSCHAR_MAX
CHAR_MIN	Valor mínimo del tipo de datos char.	OSCHAR_MIN
CHILD_MAX	Número máximo de resultados por UID	6
CLK_TCK	Número de pulsaciones del reloj por segundo	10
FCHR_MAX	Tamaño máximo de un archivo en bytes	100000
INT_MAX	Valor máximo del tipo de datos int	32767

Nombre	Descripción	Valor
INT_MIN	Valor mínimo del tipo de datos int	-32767
LOCK_MAX	Número máximo de entradas a la tabla de cerrojos del sistema.	32
LINK_MAX	Número máximo de ligas a un archivo	8
LONG_MAX	Valor máximo del tipo de datos long	2147483647
LONG_MIN	Valor mínimo del tipo de datos long	-2147483647
MAX_CHAR	Número máximo de caracteres permitidos para una entrada de terminal.	256
NAME_MAX	Número máximo de caracteres en un nombre de archivo	14
NGROUPS_MAX	Numero máximo de GID por proceso	33
OPEN_MAX	Número de archivos que un proceso puede tener abiertos al mismo tiempo	16
PASS_MAX	Numero máximo de caracteres permitidos en un password	8
PATH_MAX	Numero Máximo de caracteres en una ruta	255
PID_MAX	Valor máximo para el identificador de procesos (PID)	30000
PIPE_BUF	Número máximo de bytes que se garantizan como atómicos cuando se escribe a un tubo.	512
PIPE_MAX	Número máximo de caracteres que pueden ser escritos a un tubo.	4096
PROC_MAX	Número máximo de procesos simultáneos	8
SCHAR_MAX	Valor máximo del tipo de datos signed char	127
SCHAR_MIN	Valor mínimo del tipo de datos signed char	-127
SHRT_MAX	Valor máximo para el tipo de datos short	32767

TESIS CON
 FALLA DE ORIGEN

Nombre	Descripción	Valor
SHRT_MIN	Valor mínimo para el tipo de datos short	-32767
STD_BLK	Número máximo de bytes en una transacción atómica de escritura a un dispositivo de bloque.	256
SYS_NMLN	Máximo número de caracteres para cadenas regresadas por la función uname()	9
SYS_OPEN	Número máximo de archivos que pueden ser abiertos en el sistema al mismo tiempo.	16
UCHAR_MAX	Valor máximo del tipo de datos unsigned char	255
UID_MAX	Valor máximo para un identificador de usuario (UID)	32000
UINT_MAX	Valor máximo del tipo de datos unsigned int	65535
ULONG_MAX	Valor máximo del tipo de datos unsigned long	4294967295
USHRT_MAX	Valor máximo para el tipo de datos unsigned short	65535
WORD_BIT	Número de bits en una palabra o en un tipo de datos int	16

Los valores presentados acotan al sistema operativo resultante en sus tipos de datos mas importantes. Cabe hacer notar que el seguimiento siego de estos límites puede llevar a una considerable decaída en el rendimiento del sistema. Por ejemplo:

El tamaño de palabra usado actualmente por computadores PC es de 32 bits, lo cual implica que siguiendo el estándar se limita la capacidad de las PC actuales a 16 bits. En este aspecto la mayoría de los sistemas operativos respetan el tamaño de palabra de la plataforma en donde se hace la implantación. Este es también el caso de los dos sistemas sujetos de estudio, los cuales usan palabras de 32 bits, el cual es el estándar para computadoras x386 y superiores.

Así mismo, el estándar deja asentada la necesidad de la implementación de una biblioteca llamada <inistd.h> la cual, marca la definición de constantes que son usadas en todo sistema operativo que cumpla con el estándar.

TESIS CON
FALLA DE ORIGEN

La definición de estas constantes se refieren a la verificación de permisos de usuario, bloqueo de recursos, lectura de archivos, etc. En este archivo es donde se registra la revisión del estándar bajo la cual se realiza la implementación.

A continuación damos una breve descripción de la biblioteca y sus aspectos mas importantes:

Constantes simbólicas para la función `access()`, usada para verificar los permisos de usuario sobre un archivo:

Constante	Descripción
R_OK	Verificar permisos de lectura
W_OK	Verificar permisos de escritura
X_OK	Verificar permisos de ejecución
F_OK	Verificar la existencia de un archivo

Constantes simbólicas para la función `lockf()`, la cual permite bloquear una región de un archivo para uso exclusivo, así como preguntar por un bloqueo ya existente y quitar un bloqueo.

Constante	Descripción
F_ULOCK	Quitar el bloqueo de una región
F_LOCK	Bloquear una región para uso exclusivo
F_TLOCK	Probar y bloquear una región para uso exclusivo
F_TEST	Verificar si una región esta bloqueada

Constantes simbólicas para la función `lseek()`, que permite posicionarse en diferentes partes de un archivo.

Constante	Descripción
SEEK_SET	Posición de inicio del archivo mas k-número de bytes.
SEEK_CUR	Posición actual del archivo mas k-número de bytes.
SEEK_END	Ultima posición del archivo mas k-número de bytes

Constante simbólica para la posición de las bibliotecas del estándar POSIX

Constante	Descripción
IN_PATH	Directorio que contiene los encabezados de las librerías marcadas por el estándar.

Constante simbólica que define la versión del estándar bajo la cual se realizó la implementación.

Constante	Descripción
IEEE1003	Valor correspondiente a la fecha de la versión del estándar bajo el cual se realizó la implementación la cual consiste en 4 dígitos para el año y dos para el mes

El estándar en esta sección especifica también la abstracción de la estructura del sistema de archivos, el cual, debe ser jerárquico a través de un sistema de directorios, archivos tipo FIFO y archivos Regulares. Cada archivo debe tener los atributos típicos de un sistema UNIX como lo son, los bits de permisos, el propietario y los identificadores de grupos a sí como el contador de ligas.

Uno de los compromisos del estándar POSIX 1 es especificar los formatos de archivo, sin embargo, ya que en UNIX System V el formato era cpio y en BSD el formato preferido era tar. Ambos usados ampliamente, POSIX.1 requiere que ambos formatos sean soportados. En la especificación no se mencionan las herramientas cpio y tar, solamente son mencionados los formatos de archivo en el estándar.

El estándar POSIX solamente define una interfaz y no la forma de implementar cada una de las funciones y funcionalidades especificadas, por lo que el estándar no hace distinción entre funciones y llamadas al sistema. Por esta razón, todas las rutinas especificadas en el estándar se llaman funciones.

Las revisiones del estándar a través del tiempo, han incrementado el número de funciones originales con el objeto de poder incrementar (en cualquier implementación del estándar) funcionalidades nuevas como son: programación en tiempo real y programación para soportar aplicaciones multi-hilo.

El comité de POSIX.1 fue responsable de especificar y estandarizar el sistema UNIX que ya estaba hecho, apoyándose en la participación de representantes de diversos sistemas UNIX existentes como BSD y System V. Para conciliar las diferencias entre las versiones de UNIX existentes fue necesario reunir los puntos comunes que ofrecían los participantes en la realización del estándar, después en los puntos de discrepancia entre las diferentes implantaciones del sistema se hacía un consenso para decidir cual era la mas apropiada de acuerdo al modelo planteado.

Sin embargo en los casos en que se consideraba que las interfaces no eran las óptimas o no compatibles con el modelo propuesto se definían otras nuevas que dejaran ver una mejor solución a la situación.

3.3 Otros estándares POSIX

El documento 1003.1 es considerado la base de todos los estándares POSIX, por lo que los documentos subsecuentes lo toman como base, los cuales se presentan a continuación.

POSIX 2

Esta parte del estándar se refiere al shell y sus herramientas. Fue formalmente aprobado en septiembre de 1992 por IEEE y como un estándar internacional en junio de 1993. POSIX 2 es el estándar más importante para los usuarios, ya que su propósito principal es especificar la semántica del interprete de comandos o shell, basado en el Korn Shell de UNIX y una colección de bibliotecas que pueden ser usadas para desarrollar programas de Shell que puedan ser ejecutados en diferentes plataformas POSIX 1. El segundo propósito de este estándar es promover la transparencia hacia el usuario referente a la especificación de utilerías como vi, man y who.

POSIX 2 es independiente de POSIX 1 y cualquier sistema operativo puede cumplir completamente con este estándar sin necesidad de referirse al anterior.

POSIX 3

Los desarrolladores de los estándares POSIX han reconocido que es necesario definir métodos de prueba que vayan de acuerdo a los estándares planteados y con medidas que sean estándar. Es por eso que el estándar POSIX 3 especifica los principios generales de pruebas conforme a medidas establecidas para formar una base de comparación con los demás estándares.

POSIX 4

Este estándar describe las interfaces del lenguaje C para aplicaciones de tiempo real en adición y modificación a las ya establecidas por POSIX 1. Este estándar puede ser definido como la habilidad del sistema operativo de proveer el nivel de servicio requerido para respuestas en tiempo real. Los sistemas en tiempo real han sido históricamente implementados en sistemas tipo stand-alone los cuales controlan los procesos o las máquinas por sí mismos.

Este estándar fue adoptado en septiembre de 1993; las características especificadas en éste son ampliamente usadas para aplicaciones en tiempo real como semáforos, cronómetros, intercomunicación de procesos, memoria virtual y muchos otros.

Este estándar está estructurado como una colección de extensiones opcionales del estándar POSIX1. Es decir, que un sistema puede cumplir con algunas partes y otras no.

POSIX 5

Ésta es la versión para el lenguaje Ada del estándar POSIX 1. Es decir, especifica las rutinas en el lenguaje Ada que proveen esencialmente la misma funcionalidad de dicho estándar. Adicionalmente se está trabajando en la versión POSIX. 4^a, la cual permitirá utilizar las características de tiempo real para los usuarios del lenguaje Ada.

POSIX 9

Es la versión de FORTRAN del estándar POSIX. 1 y ésta especifica las rutinas en el lenguaje FORTRAN 77 que proveen la misma funcionalidad que las rutinas especificadas en C del estándar POSIX. 1

Administración del Sistema

La administración del sistema ha sido explícitamente omitida del marco del estándar POSIX, sin embargo los grupos de desarrollo del estándar POSIX notaron la importancia y la necesidad de definir estándares para la administración del sistema de tal forma que estos quedaron en la siguiente organización compuesta por tres comités principales:

- ↗ POSIX 1387.2 Distribución del Software. Este incluye estándares para empaquetar aplicaciones así como la instalación, desinstalación y mantenimiento de éstos paquetes.
- ↗ POSIX 1387.3 Administración de Usuarios y Grupos. El cual incluye estándares para añadir y borrar usuarios y grupos a los sistemas, así como administrar cuentas de correo electrónico y cuotas de disco.
- ↗ POSIX 1387.4 Administración de Impresión. Esta intenta estandarizar las interfaces de la aplicaciones para acceder a los dispositivos de impresión.

3.4 Estándares Futuros

Los siguientes estándares POSIX están siendo desarrollados de acuerdo a las necesidades presentadas por los usuarios, siendo una de las áreas en las que ha aumentado la necesidad de definir estándares es la seguridad.

Desde el desarrollo del sistema operativo UNIX, el cual tenía un simple pero relativamente efectivo sistema de seguridad que consistía en permisos de acceso en tres niveles de seguridad (propietario, grupo de trabajo y otros) no se ha modificado, lo cual para grupos de trabajo pequeños es bastante efectivo, pero para las necesidades actuales este esquema comienza a carecer de efectividad.

Un ejemplo de la necesidad de estandarización de la seguridad es el libro naranja (orange book), en el cual, se definen diferentes niveles de seguridad requeridos por los sistemas. Es por eso que el comité del estándar POSIX está desarrollando mecanismos, para proveer y definir niveles más altos de seguridad que incluyan las siguientes áreas generales:

✓ Least Privilege. (Privilegio Mínimo) Este se refiere a la idea de que un proceso debe tener únicamente los privilegios que le son necesarios para su función.

✓ Discretionary Access Controls (Control de Acceso Discrecional). Este se refiere a la idea de añadir a las restricciones tradicionales de accesos a archivos una lista de control de accesos la cual contendrá los identificadores de usuario que podrán acceder al archivo.

✓ Mandatory Access Controls (Control de Acceso Obligatorio). Estas se refieren a dos niveles de seguridad. Un objeto es creado en algún nivel de seguridad sin el control de su creador y puede ser accedido sólo por procesos con el mismo o más alto nivel. El nivel puede ser determinado por los identificadores de proceso usuario grupo o por la naturaleza del proceso en sí mismo.

✓ Auditability Mechanism. (Mecanismo de Auditoría) Este cubre los tipos de objetos o acciones necesarias para auditar y los mecanismos para llevar el flujo de la auditoría.

En todas las áreas mencionadas anteriormente, existen sistemas UNIX que ya implementan algunas de éstas características para probar y mejorar cada una de éstas. Adicionalmente el comité del estándar POSIX. 22 está trabajando en características muy similares pero aplicables a ambientes distribuidos.

Capitulo IV. Windows

No cabe duda que Windows de Microsoft es el sistema operativo más exitoso comercialmente hablando, pero ¿Cuales son las características técnicas que lo han llevado a ocupar ese lugar?, ¿Cuáles fueron los objetivos de sus creadores?, ¿Cuáles son sus deficiencias como sistema operativo?. A todas estas preguntas trataremos de introducirnos en este capítulo con el objeto de proporcionar una base de comparación con otros sistemas operativos.

4.1 Objetivos de Diseño

Los creadores de Windows se centraron en 6 características clave para desarrollar este Sistema Operativo:

- ✓ Extensibilidad
- ✓ Portabilidad
- ✓ Confiabilidad
- ✓ Compatibilidad
- ✓ Seguridad
- ✓ Performance

Estas características fueron la piedra angular del desarrollo de Windows, pero ¿Bajo qué enfoque fueron vistas? Y ¿En realidad fueron cubiertos cada uno de los puntos?:

Extensibilidad

En el mercado de las Computadoras Personales el desarrollo de nuevo hardware es algo cotidiano, por lo que se creyó necesario que Windows fuera un sistema operativo con alta extensibilidad para soportar los nuevos dispositivos que fueran siendo introducidos al mercado.

Este punto clave en todo sistema operativo fue cubierto por Microsoft en mayor parte por medio de alianzas comerciales, de las cuales casi todas las compañías de hardware en la actualidad ponen sus dispositivos a la venta con controladores para Windows.

Transportabilidad

Es la habilidad de un sistema operativo de ser migrado a diferentes plataformas de hardware con un mínimo esfuerzo, para lo cual Windows presenta características que le permiten poder ser migrado a varias plataformas:

- ↗ Código modular.
- ↗ En su mayoría el sistema operativo fue escrito en lenguaje C.
- ↗ Se desarrolló un concepto llamado HAL (Hardware Abstraction Layer) el cual permite al sistema operativo obtener una abstracción de alto nivel del hardware que es manejado realmente.
- ↗ Actualmente Windows corre sobre las siguientes plataformas: Intel x86, MIPS y Alpha.

Confiabilidad

Hoy en día es necesario que los sistemas operativos sean robustos y tolerantes a fallas tanto del usuario, como de las aplicaciones o el hardware. Windows proporciona desde su diseño los siguientes mecanismos para cumplir con este cometido:

- ↗ Aislamiento de cada una de las aplicaciones.
- ↗ Aislamiento de las aplicaciones respecto al kernel.
- ↗ Sistema de archivos con de recuperación, incluyendo errores críticos en el disco.
- ↗ Uso de almacenamiento redundante basado en un modelo transaccional para el procesamiento de datos.

Además de que Windows también puede ejecutarse en un sistema de archivos vfat de 32 bits (Estándar para Windows 95 y 98), la paginación de memoria se realiza hacia un archivo en el sistema de archivos elegido al instalarlo.

Compatibilidad

La compatibilidad es la habilidad de un sistema operativo de poder ejecutar en él, programas desarrollados en otros sistemas operativos.

Windows se desarrolló para ser compatible con los demás sistemas operativos de Microsoft, incluso soportando aplicaciones de 16 Bits y aplicaciones para OS/2 y aplicaciones en modo texto bajo el estándar POSIX.

Esta compatibilidad fue desarrollada por medio de subsistemas que trabajan en modo usuario donde los módulos principales de otros sistemas pueden ejecutarse aisladamente.

Seguridad

La seguridad se refiere a la protección del uso no autorizado de recursos locales y de red.

Windows proporciona seguridad de las siguientes maneras:

- ✓ Monitor de Seguridad. En el que es posible controlar quien esta accedendo a los recursos de la máquina.
- ✓ identificadores de usuario y de grupo.
- ✓ Listas de control de acceso del sistema.
- ✓ Listas de control de acceso discrecional

Rendimiento

Los desarrolladores de Windows definieron el rendimiento como la capacidad del sistema operativo de dar respuesta a los requerimientos de las demás aplicaciones por medio de la una óptima utilización de los recursos.

Para cumplir con este punto los desarrolladores de Windows se centraron en las siguientes características:

- ✓ Una arquitectura real de 32 bits.
- ✓ Un mecanismo de llamadas a procedimientos locales (LPC).
- ✓ Optimización de procesos críticos del sistema tales como llamadas al sistema y manejo de páginas de memoria.
- ✓ Los mecanismos para trabajar en red son críticos para el sistema por lo que han sido construidos en la parte privilegiada del sistema operativo.

4.2 Arquitectura de Windows

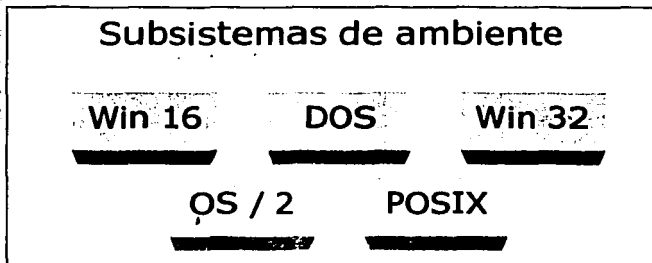
Los puntos principales que comprenden las características arquitectónicas de Windows son los siguientes:

- ✓ Multitarea prioritaria.
- ✓ Soporte para procesamiento en paralelo.
- ✓ Servicios de Red Integrados.
- ✓ Soporte para varios Sistemas de Archivos

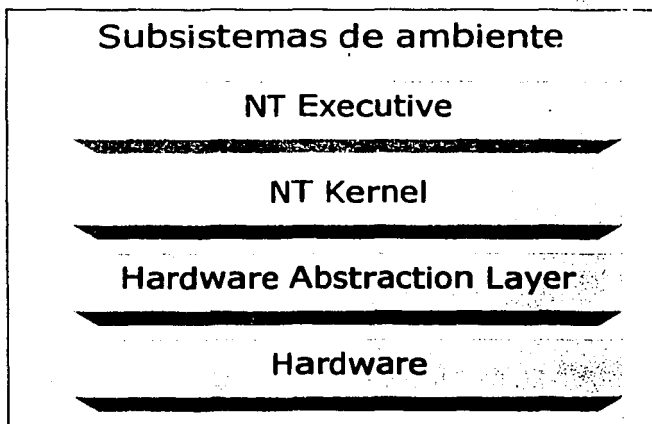
Windows esta basado en un modelo estructural cliente servidor, sin embargo este modelo incorpora algunas de las características del modelo de máquinas virtuales a través de HAL el cual hace posible construir proveedores de servicios del sistema operativo usando esta interfaz sin preocuparse de como es que realmente el servicio se comunica con el hardware de la computadora.

En la siguiente gráfica podremos observar como esta estructurada la arquitectura de este sistema operativo.

Modo Usuario



Modo Kernel



FALLA DE ORIGEN

Windows realiza sus funciones a través de dos modalidades que encontramos en el sistema:

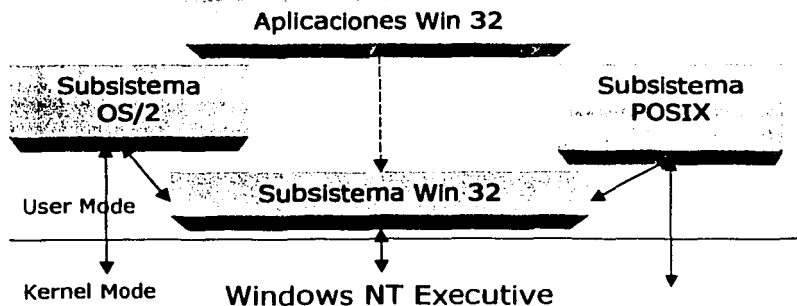
- ✓ **Modo usuario.** En este se realiza la mayor parte de la interacción con las aplicaciones del usuario a través de los subsistemas de ambiente, los cuales realizan la traducción de las llamadas del usuario a servicios ofrecidos por el kernel. En este modo de operación, cada aplicación se ejecuta de forma no privilegiada para el sistema operativo.
- ✓ **Modo kernel.** Cuando los subsistemas de ambiente son atendidos por alguno de los servicios del sistema operativo este cambia a modo kernel donde se llevan a cabo tareas privilegiadas.

Es notable como Windows da tratamiento a las llamadas al sistema a través de los subsistemas de ambiente, los cuales presentan las siguientes características:

- ✓ Son procesos en modo usuario que proveen de servicios a las aplicaciones.
- ✓ Emulan la conducta del sistema operativo.
- ✓ Traducen las solicitudes de una aplicación a una llamada al sistema que el kernel pueda entender.
- ✓ Son implementados usando un modelo cliente / servidor tradicional

4.3 Los subsistemas de ambiente

Los subsistemas de ambiente de Windows son los que hacen que las aplicaciones se comuniquen con el kernel, por lo que profundizaremos un poco más en su estudio.



TESIS CON
FALLA DE ORIGEN

4.3.1. Subsistema WIN32

Este subsistema es el más importante de todos, ya que no solamente atiende a aplicaciones de 32 bits. Este subsistema maneja las entradas del teclado y la salida a pantalla de todo el sistema operativo, de tal forma que los demás subsistemas tienen que hacer uso de éste para manejar los dispositivos mencionados.

Cuando se desarrolla software que interactúa directamente con el kernel de Windows es necesario usar el API de Win32 el cual fue escrito para correr bajo este subsistema de ambiente. Además de que éste actúa como servidor para los demás subsistemas de ambiente.

En este subsistema de ambiente las aplicaciones se ejecutan en su propio segmento de memoria de manera separada de otros subsistemas Win32 que estén siendo ejecutados por Windows.

4.3.2. Subsistemas OS/2 y POSIX

Estos subsistemas básicamente soportan aplicaciones en modo texto y hacen uso del subsistema de ambiente Win32 para obtener las entradas del teclado y la salida a pantalla, sin embargo, al igual que el subsistema Win32 las aplicaciones se ejecutan en su propio segmento de memoria el cual es independiente del subsistema que las este atendiendo.

4.3.3. Subsistema MS-DOS

Este subsistema es diferente a los demás por que asocia a cada aplicación con una máquina virtual de DOS la cual provee los siguientes servicios a las aplicaciones:

- ✓ Ejecución de instrucciones para procesadores intel x86.
- ✓ Servicio de emulación de la interrupción 21.
- ✓ Servicio de emulación de interrupciones del BIOS de la máquina.
- ✓ Servicio de emulación de dispositivos de hardware.

4.3.4. Subsistema Win16

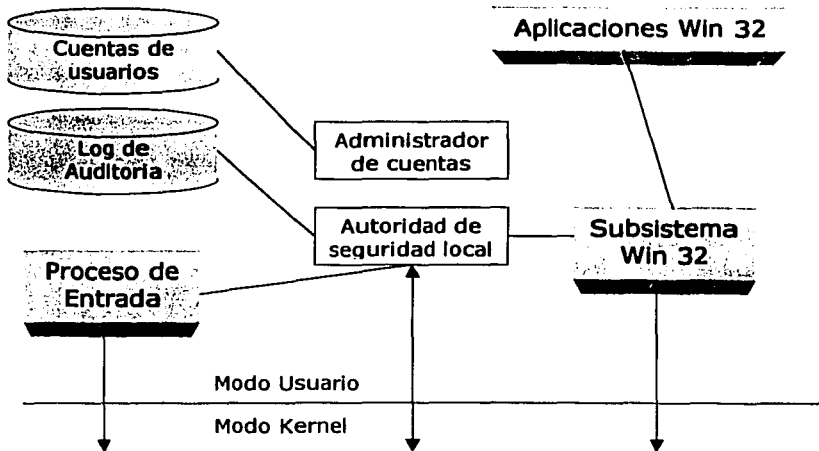
Este subsistema realiza una asociación entre las aplicaciones para Windows de 16 bits con una máquina virtual llamada WOW (Win16 on Win32) que podríamos definir mejor como una máquina virtual Windows. Es importante mencionar que esta máquina virtual soporta aplicaciones en modo gráfico a diferencia de algunas de las anteriores (POSIX, OS/2, MS-DOS). Las principales características que se hacen presentes al ejecutar una aplicación de 16 bits dentro de Windows son las siguientes:

- ↙ Cada aplicación se asigna a un hilo de ejecución diferente con una máquina virtual WOW.
- ↙ Cada hilo de ejecución debe ceder el control antes de que otra aplicación Win16 pueda ejecutarse.
- ↙ Si una aplicación Win16 aborta en modo erróneo es posible que el subsistema Win16 quede inutilizable, sin embargo a las aplicaciones Win32 no les afecta.

La función principal de este subsistema de ambiente es convertir las llamadas al API Win16 en llamadas al API Win32 para que puedan ejecutarse las solicitudes de las aplicaciones.

4.3.5. Subsistema de Seguridad

El subsistema de Seguridad se ejecuta en modo usuario y es el responsable de brindar la seguridad del sistema operativo. Como ya hemos visto en este capítulo existen subsistemas de ambiente para dar servicio a las aplicaciones, sin embargo, es necesario que cada uno de esos sistemas proporcione cierto grado de seguridad antes de mandar a ejecutar algo al modo kernel del sistema operativo. En la siguiente gráfica lo explicamos más claramente:



El subsistema de seguridad se compone de las siguientes partes:

- ✓ **Autoridad de Seguridad Local (LSA).** Genera un identificador de acceso para cada requisición que vaya a pasar a modo kernel, además de escribir en el log de acceso lo que le sea solicitado por el modo kernel del sistema operativo.
- ✓ **Identificador de Acceso.** Describe lo que el usuario puede y no puede hacer, es decir proporciona información acerca del usuario con respecto al proceso que intenta ejecutar.
- ✓ **Proceso de Entrada.** Este proceso está en espera de requisiciones para entrar al sistema, a su vez este proceso se comunica con la autoridad de seguridad local para validar al usuario que ha hecho la petición de entrada al sistema.
- ✓ **Administrador de Cuentas.** Este provee acceso a la base de datos de usuarios y grupos y es el proceso encargado de dar mantenimiento y administrar dichas bases de datos.

1985 CTW
FALLA DE ORIGEN

4.4. NT Executive

Hasta este momento hemos revisado los componentes de Windows que son ejecutados en modo usuario (o modo no privilegiado), sin embargo, de acuerdo a nuestra definición de sistema operativo del primer capítulo podemos decir, que la mayoría de las funciones como sistema operativo Windows las realiza en modo kernel, ya que es ahí donde se realizan las tareas de administración de los recursos de la computadora. El punto de entrada al modo kernel de Windows es la capa llamada NT Executive que analizaremos a continuación:

Como ya se dijo antes, NT-Executive es el punto de entrada hacia el modo kernel de Windows. La interfaz que proporciona el punto de entrada a este modo, es la API nativa de Windows la cual provee las principales interfaces para poder usar los servicios ofrecidos por el NT-Executive. El cual esta dividido en módulos perfectamente identificados de los cuales daremos una breve explicación más adelante.

Servicios del sistema NT Executive
Administrador de Objetos
Llamadas a procesos locales
Administrador de procesos
Monitor de seguridad
Administrador de la Memoria virtual
Administrador de Entrada / Salida
✓ Administrador de cache
✓ Sistemas de archivo
✓ Controladores de dispositivos

Es importante mencionar algunas características comunes a todos los componentes de NT Executive:

- ✓ Cada componente proporciona un API distinto.
- ✓ Algunos de estos API's fueron diseñados para usarse directamente desde el modo usuario.
- ✓ La mayoría de los API's proporcionados por los módulos del NT-Executive son solamente visibles en modo kernel y se usan para la comunicación interna entre cada uno de los módulos del sistema operativo.

NT Executive se compone principalmente como se pudo apreciar de 6 módulos principales, donde cada uno de ellos realiza una tarea muy específica:

↙ **Administrador de Objetos (Object Manager).** Un objeto es la representación conceptual de una entidad física del sistema, es decir, un archivo o un semáforo del sistema es representado como un objeto; dichos objetos son administrados por este módulo.

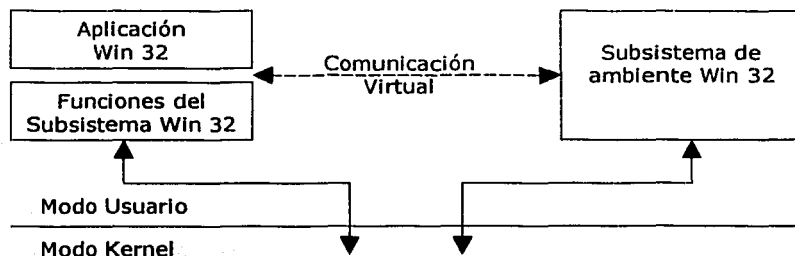
↙ **Monitor de Seguridad (Security Reference Monitor).** Este verifica los permisos de ejecución antes de que un objeto pueda usarse. Cabe mencionar que el Monitor de seguridad es un cliente del Administrador de Seguridad de los recursos.

↙ **Administrador de Procesos (Process Manager).** Es el encargado de proporcionar los mecanismos para crear, modificar o terminar procesos o hilos de ejecución en el sistema. Cada proceso se representa como un objeto del sistema llamado "Objeto de Procesamiento" el cual contiene:

- ↙ El espacio virtual de procesamiento.
- ↙ Los recursos visibles al proceso.
- ↙ Los hilos del proceso.

Es importante mencionar que NO se mantienen relaciones entre procesos padre e hijo.

↙ **Llamadas a Procesos Locales (Local Procedure Call).** Cualquier mensaje entre diferentes procesos se hace a través de este módulo, incluyendo la comunicación entre los programas de aplicación y sus subsistemas de ambiente como por ejemplo:



TRABAJO
FALLA DE ORIGEN

✓ **Administrador de la memoria virtual (Virtual Memory Manager).** Este es el encargado de administrar la memoria virtual, así como de realizar la paginación de memoria cuando es requerida. Este administrador proporciona mecanismos de búsqueda para los segmentos de memoria usados por cada proceso, así como la implementación de una página seccionada, es decir, no es necesario subir a la memoria real toda una página si solamente va a ser usado un segmento de la misma. Así mismo fueron implementados mecanismos para generar espacios de memoria compartidos los cuales se usan principalmente para las llamadas a procesos locales, las cuales fomentan una buena comunicación entre procesos. El Administrador de Memoria Local es dependiente del procesador que se encuentre corriendo ya que de éste dependen el tamaño de las páginas, o el número de espacios en memoria reales que pueden ser direccionados.

✓ **Administrador de Entrada / Salida (I/O Manager).** Este administra los controladores para sistemas de archivos, dispositivos de la máquina y dispositivos de red. Entre sus mas destacadas cualidades están:

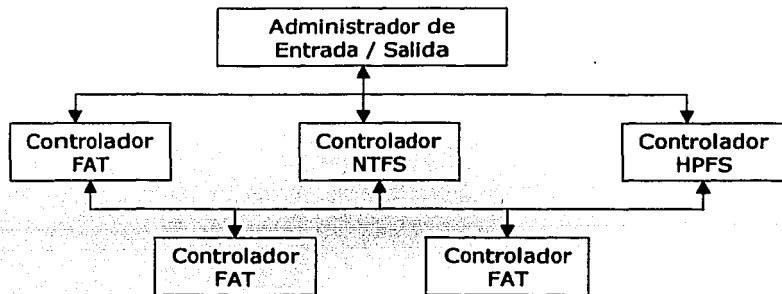
✓ Las requisiciones se manejan como paquetes llamados IRP los cuales van de un componente de Entrada / Salida a otro.

✓ Se maneja un buffer para cada IRP.

✓ Mantiene un control de tiempos de espera hacia los dispositivos.

✓ Mantiene un registro de los componentes instalados en el sistema

La forma más común de la labor desempeñada por el Administrador de Entrada / Salida es la que se muestra en el siguiente diagrama:



El Administrador de Entrada / Salida envía un IRP al controlador de HPFS (High Performance File System) el cual procesa el paquete y a su vez envía un nuevo paquete IRP al controlador del disco flexible para que sea escrita la información. Para concluir con este punto es necesario hacer un breve paréntesis en el sistema de archivos NTFS, el cual se constituyó para que ser usado por Windows como el sistema de archivos principal.

Las características que NTFS ofrece son principalmente las siguientes:

- ✓ Recuperación rápida de datos después de una falla en el sistema.
- ✓ Habilidad para manejar archivos grandes.
- ✓ Seguridad
- ✓ Soporte para sistemas operativos basados en POSIX.

4.5 Kernel

Como se vio en el Capítulo I los sistemas tipo Cliente / Servidor llevan consigo un kernel muy pequeño dedicado principalmente a administrar la relación proceso-CPU. Cuando un sistema operativo tiene un kernel basado en estas características, se dice que es un sistema operativo de micro-kernel, porque las funciones comúnmente atribuibles al kernel han sido descentralizadas de éste, actuando como servicios ofrecidos por otras entidades, a las cuales el kernel provee de tiempos de procesamiento.

- ✓ El kernel de Windows provee de los siguientes servicios:
 - ✓ Programa los procesos que serán ejecutados por la CPU.
 - ✓ Maneja las interrupciones del hardware y las direcciona a los drivers correspondientes.
 - ✓ Maneja las fallas que se presenten, tanto de hardware como de software.
 - ✓ Provee de un API para comunicarse con los servicios del NT-Executive.

4.6 Hardware Abstraction Layer (HAL).

HAL. El concepto de abstracción del hardware permite aislar los problemas inherentes al manejo de un dispositivo en específico de la computadora, de esta manera el kernel del sistema operativo no se encarga en su totalidad del manejo del dispositivo, e incluso no necesita saber las características específicas de ese dispositivo como marca, modelo, etc., puesto que el kernel maneja todas estas características de acuerdo a una interfaz estándar para ese dispositivo, la cual es implementada en las parte más bajas del sistema operativo.

Esta parte del modo kernel es la que se comunica directamente con los dispositivos de la computadora y provee de las siguientes características al microkernel del sistema operativo:

- ✓ Arranque del procesador.
- ✓ Soporte para arreglos simétricos de procesadores.
- ✓ Espacios de almacenamiento de paso para instrucciones y/o datos.
- ✓ Soporte para drivers.
- ✓ Manejo de errores a bajo nivel.

Capítulo V. Linux

5.1 Origen de Linux

Linux es un sistema operativo para PC basados en la familia de procesadores Intel x86, el cual, ha sido desarrollado por cientos de programadores alrededor del mundo con el objetivo de crear un símil de UNIX completamente funcional, sin la ayuda de software comercial bajo una licencia de derechos de autor que permita a cualquiera usarlo y modificarlo sin restricción alguna.

Originalmente el módulo principal de Linux fue creado por Linus Torvalds de la Universidad de Helsinki con el objetivo de crear un sustituto de Minix que tuviera mayor funcionalidad. Sin embargo, Linus Torvalds terminó escribiendo la base del kernel del sistema operativo y al ponerlo bajo la licencia pública GNU un gran número de desarrolladores han participado en el proyecto y éste ha crecido hasta llegar a ser una alternativa real en constante desarrollo y que compite con los mejores del mercado por su estabilidad, portabilidad y escalabilidad.

5.2 Características Generales de Linux

Linux, como símil de UNIX ha heredado de este último características que lo conforman como un sistema operativo flexible y potente:

5.2.1 Multitarea

La palabra multitarea describe la capacidad de ejecutar muchos programas al mismo tiempo sin detener la ejecución de cada aplicación. Se le denomina multitarea prioritaria porque cada programa tiene garantizada la oportunidad de ejecutarse, y se mantiene corriendo hasta que el sistema operativo da prioridad a otro programa para que se ejecute. Linux y otros sistemas operativos multitarea prioritaria realizan esto supervisando los procesos que esperan para ejecutarse, así como los que se están ejecutando. El sistema operativo programa para cada proceso tiempo para que disponga de las mismas oportunidades de acceso al microprocesador, produciendo el efecto de que las aplicaciones se están ejecutando casi al mismo tiempo porque transcurren solo unas billonésimas de segundo entre la ejecución de unas y otras.

5.2.2 Multiusuario

La capacidad de Linux para asignar el tiempo de microprocesador simultáneamente a varias aplicaciones ha derivado en la posibilidad de ofrecer servicio a varios usuarios a la vez, ejecutando cada uno de ellos una o varias aplicaciones.

5.2.3 Multiplataforma

El desarrollo de Linux ha permitido la migración del este sistema operativo a diferentes plataformas como son 386, 486. Pentium, Pentium Pro, Pentium II, Amiga, Atari, Alpha, ARM, MIPS, PowerPC y SPARC.

5.2.4 Multiprocesador

Linux es capaz de correr en sistemas con más de un procesador en plataformas Intel y SPARC dando una gran escalabilidad.

5.2.5 Modo Protegido 386

Linux es posible en gran parte porque las computadoras Intel 386 y subsecuentes permiten trabajar en modo protegido, el cual permite proteger la memoria compartida entre varios procesos, de tal forma que ningún proceso pueda bloquear el sistema de manera inesperada.

5.2.6 Carga de ejecutables sobre demanda

Esta característica permite a Linux leer del disco solamente aquellas partes de un programa que están siendo usadas, lo cual repercute en un mejor rendimiento en la ejecución de programas grandes. Es decir Linux permite usar una política de copia en escritura para compartir páginas entre diferentes programas permitiendo que varios procesos puedan usar la misma zona de memoria para ejecutarse sin violar la seguridad del sistema. Cuando alguno intenta escribir en esa memoria, la página (4096 bytes de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.

5.2.7. Memoria virtual usando paginación optimizada (sin intercambio de procesos completos a disco)

Linux permite establecer particiones o archivos de intercambio e incluso, permite añadir más áreas de intercambio en tiempo de ejecución.

La memoria se administra como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede usarse para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.

5.2.8 Bibliotecas compartidas de carga dinámica (DLL's) y bibliotecas estáticas

El uso de bibliotecas de carga dinámica aumenta el rendimiento y permite a los programadores estructurar sus desarrollos de tal forma que el código sea reutilizable y compartido entre varias aplicaciones.

5.2.9 Volcados de estado (core dump)

Posibilita los análisis post-mortem de un proceso, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.

5.2.10 Compatibilidad

Es compatible con POSIX, System V y BSD a nivel de código fuente. La emulación de iBCS2 es casi completamente compatible con las versiones SCO, SVR3 y SVR4 de UNIX a nivel binario.

5.2.11 Emulación del coprocesador matemático

Emulación del circuito 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Si la computadora ya tiene una FPU (unidad de punto flotante), esta se usará en lugar de la emulación, pudiendo incluso compilar un kernel sin la emulación matemática y conseguir ahorro de memoria.

5.2.12 Soporte de teclados

Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.

5.2.13 Consolas virtuales múltiples

Se pueden tener varias sesiones de login a través de una sola consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de vídeo). Es posible tener hasta 64 terminales activas en la consola.

5.2.14 Sistemas de archivos

Linux tiene un avanzado sistema de archivos propio, con una capacidad de hasta 4 Tera Bytes y nombres de archivos de 255 caracteres de longitud; soporta varios sistemas de archivos incluyendo minix-1, Xenix y todos los sistemas de archivos típicos de System V, tiene acceso transparente a particiones MS-DOS mediante un sistema de archivos especial: no es necesario ningún comando especial para usar la partición MS-DOS, esta parece un sistema de archivos normal de UNIX (excepto por algunas restricciones en los nombres de archivo, permisos, etc.), se ha añadido al núcleo soporte para VFAT. Un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS, así como soporte en sólo lectura de HPFS-2 del OS/2 2.1 y sistema de archivos ISO8660 que lee todos los formatos estándar de CD-ROM.

5.2.15 Comunicaciones

El kernel contiene diversos protocolos de red con soporte completo de TCP/IP, Appletalk, Netware, Lan Manager, Windows Native (SMB), IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.

5.3 El Kernel de Linux

El kernel o núcleo de Linux se puede definir como el corazón del sistema operativo, es el encargado de que el software y el hardware de la computadora puedan trabajar juntos. Las funciones más importantes del mismo, aunque no las únicas, son:

- ↙ Administración de la memoria, para los programas en ejecución.
- ↙ Administraciones del tiempo del procesador para los programas en ejecución.
- ↙ Administración de equipo periférico.

Es decir, existen una serie de procesos que atienden diferentes tareas, cada proceso le presenta al sistema solicitudes de recursos como memoria, conectividad de red, o algún otro recurso que necesite, en general el sistema se encuentra compuesto de una serie de módulos que se encargan de tareas específicas.

El kernel de Linux esta construido bajo un modelo estructural monolítico, es decir, el kernel del sistema operativo es una sola pieza de software donde se realizan todas las operaciones (ver Capítulo 1) correspondientes al sistema operativo y este a su vez atiende las peticiones de los programas de aplicación mediante llamadas al sistema. Sin embargo, aunque Linux usa un modelo monolítico en su estructura podemos identificar cada una de las partes que constituyen el kernel del sistema operativo, como lo analizaremos mas adelante en este capítulo.

Por propósitos de control y de desarrollo del sistema operativo existen dos versiones del kernel de Linux:

↙ **Versión de producción:** Es la versión considerada estable hasta el momento la cual es producto de las versiones finales de desarrollo y que han pasado por una etapa de pruebas exhaustiva por usuarios de todo el mundo. Cuando el equipo de desarrollo del kernel experimental, decide que ha conseguido una versión estable y con la suficiente calidad, se lanza una nueva versión de producción. Esta versión es la que se debe de usar bajo condiciones normales del sistema.

↙ **Versión de desarrollo:** Esta es una versión experimental y es la que utilizan los desarrolladores para programar, comprobar y verificar nuevas características y/o correcciones. Estos núcleos suelen ser inestables y por lo regular son usadas por desarrolladores expertos y voluntarios para pruebas de versiones beta.

5.3.1 Las distribuciones del kernel

Las versiones del kernel son identificadas con 3 números, de la siguiente forma: XX.YY.ZZ

↙ **XX:** Indica la serie principal del kernel. Hasta el momento solo existen la 1 y 2. Este numero cambia cuando la manera de funcionamiento del kernel ha sufrido un cambio muy importante.

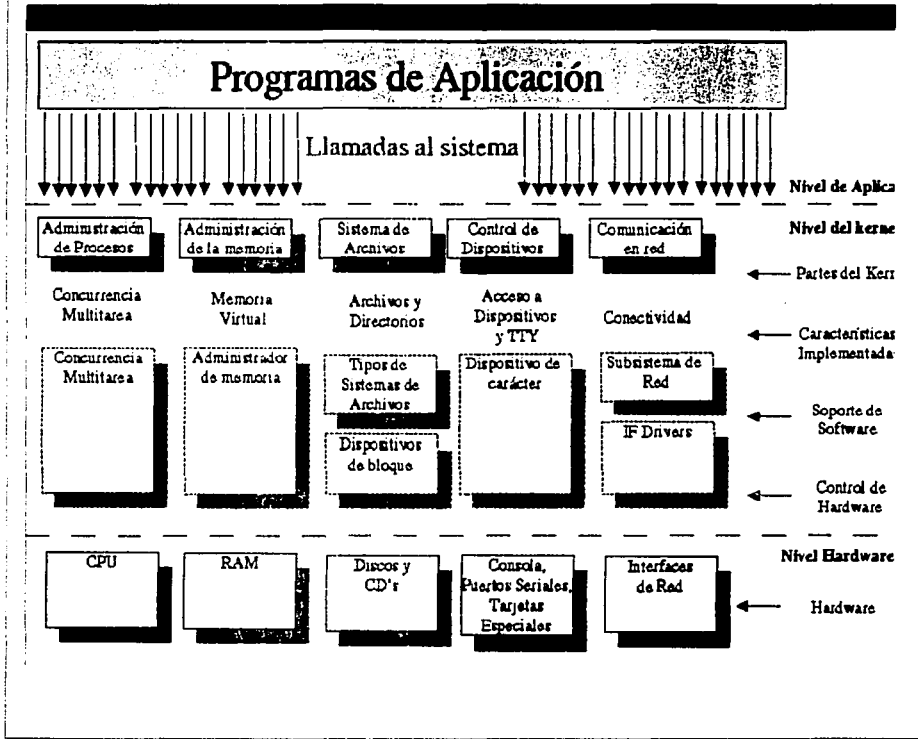
↙ **YY:** Indica si la versión es de desarrollo o de producción. De tal forma que un número impar, significa que es de desarrollo, uno par, que es de producción.

↙ **ZZ:** Indica nuevas correcciones dentro de una versión, en las que lo único que se ha modificado, son errores menores.

5.4 Estructura del Kernel de Linux

Como se dijo anteriormente Linux es un sistema monolítico, sin embargo presenta una serie de características en su estructura que lo hacen sumamente flexible. Como podemos apreciar en la siguiente gráfica el kernel de Linux esta dividido en módulos:

TESIS CON
 FALLA DE ORIGEN



5.4.1 Administración de Procesos

Este módulo corresponde a la parte más valiosa del sistema operativo pues es el que se encarga de crear, destruir y cargar procesos; así como de manejar las conexiones de entrada y salida de la computadora y la comunicación entre procesos; estas corresponden a la funcionalidad básica de todo el sistema. Adicionalmente este modulo también contiene la agenda de tareas la cual administra a todos los procesos que son ejecutados en el CPU.

5.4.2 Administración de Memoria

Probablemente la memoria de la computadora sea el recurso mas usado en cualquier sistema después del CPU lo cual hace al Administrador de Memoria uno de los procesos más críticos para el rendimiento de la computadora.

El administrador de memoria de Linux proporciona a los programas de aplicación un direccionamiento virtual de memoria para todos los procesos que se encuentren corriendo en el sistema, el cual alcanza hasta 2Gb de memoria real en su forma más simple sin necesidad de hacer paginación. Adicionalmente es posible construir 16 particiones de 128 Mb cada una, como espacio para realizar paginación de memoria.

Linux realiza la paginación de memoria con base en segmentos de pagina, es decir, la memoria es administrada sobre demanda donde no es necesario vaciar toda una pagina a un espacio de memoria virtual en disco para poder subir otra, sino que solamente son vaciadas partes de la memoria real para ser substituidas por otras que se encuentran almacenadas. Por esta razón Linux usa un tipo de partición especialmente diseñada para realizar paginación a disco en vez de usar solamente archivos dentro de su partición principal.

4.4.3 Sistemas de Archivos

Linux al igual que UNIX se basa fuertemente en el concepto de Sistema de Archivos haciendo que todo en el sistema pueda ser tratado como un archivo. El kernel de Linux genera un sistema de archivos estructurado por encima del hardware, el cual, es manejado por todo el sistema en términos de archivos. Linux presenta en las capas intermedias de este módulo los mecanismos que corresponden a los diferentes Sistemas de Archivos (NFS, vfat, ext2 NTFS, etc.), y en la capa mas baja los controladores de hardware de los dispositivos de bloque.

5.4.4 Controlador de Dispositivos

Casi toda la operación de una computadora termina realizándose hacia algún dispositivo, cada uno de éstos debe de ser administrado por un proceso central que se haga cargo del control y marque las reglas de uso general de todos los dispositivos. Esta tarea es realizada por medio del Controlador de Dispositivos el cual hace uso a su vez de los drivers necesarios para el control del dispositivo.

En todo sistema UNIX los dispositivos pueden ser accedados como si fueran archivos, estos archivos son conocidos por el nombre de Archivos Especiales a los cuales les corresponde un driver en específico, que el kernel de Linux puede tratar de dos formas distintas según haya sido compilado el driver. Es decir el driver puede ser tratado como un módulo del sistema que es cargado a memoria bajo demanda o como parte del kernel, en cuyo caso permanecerá residente en memoria todo el tiempo.

Como hemos mencionado anteriormente los dispositivos son vistos en Linux como Archivos Especiales, éstos pueden ser de tres tipos distintos: de carácter, de bloque o de red. A continuación se describe cada uno de ellos:

Dispositivos de carácter

Son aquellos que pueden ser accedados como un archivo, los drivers para estos dispositivos usualmente implementan las llamadas open, close, read y write. La consola y los puertos paralelos son buenos ejemplos de estos dispositivos. La única diferencia en la forma de acceso

entre los archivos especiales de carácter con un archivo normal es que los primeros solamente pueden ser leídos secuencialmente debido a que estos archivos solamente representan un canal de datos continuo.

Dispositivos de Bloque

Estos son aquellos que pueden contener un sistema de archivos dentro de ellos mismos. En un sistema UNIX un dispositivo de bloque solamente puede ser accesado con múltiplos de un bloque (usualmente 1K), en Linux es permitido leer y escribir en un dispositivo de bloque cualquier número de bytes.

Interfaces de Red

Las interfaces de red difieren de los dispositivos anteriores, ya que como en todo sistema UNIX el acceso a los dispositivos de Red se hace por medio de un nombre único (como eth0) el cual no tiene una representación en el sistema de archivos.

Capitulo VI. Estudio comparativo de Linux y Windows

En años recientes el mercado de las computadoras basadas en procesadores x86 comenzó a explorar una nueva área: los servidores. Gracias a su gran desarrollo estas computadoras llamadas PC han adquirido una madurez en el hardware que antes solamente correspondía al mercado de las estaciones de trabajo con sistemas operativos UNIX, VMS y UNICO entre otros. Ahora las maquinas x86 cuentan con sistemas operativos de alto rendimiento capaces de dar facilidades para fungir como servidores para las empresas, universidades, etc.

Todos los administradores de sistemas de hoy en día se han planteado la posibilidad de optar por un sistema UNIX o por Windows en sus servidores x86. Esto determina una realidad se ha debatido desde hace mucho tiempo. La solución a este problema se encuentra básicamente en dos puntos: costo y escalabilidad, entendiéndose a esta ultima como capacidad de crecimiento a menor costo.

Entre los sistemas operativos existentes para este tipo de máquinas en el ámbito de servidores encontramos dos principalmente: Windows y Linux, pero cabe hacerse algunas preguntas respecto de estos dos sistemas operativos ¿Cuál da mejor rendimiento, es más confiable, proporciona mayor escalabilidad, bajos costos de mantenimiento, etc.? Al tratar de responder a estas preguntas podemos encontrar diferentes opiniones de los administradores de sistemas como por ejemplo:

- ✓ Windows es más fácil de administrar y tiene mejor soporte del fabricante.
- ✓ Linux ofrece mayor escalabilidad y es más confiable.
- ✓ Windows proporciona mejor rendimiento para aplicaciones de alto desempeño.
- ✓ Linux tiene mejores herramientas de automatización de tareas administrativas.
- ✓ Windows es más amigable para realizar tareas administrativas.
- ✓ Linux tiene mayor conectividad en redes y ofrece mejor rendimiento en ambientes de red.
- ✓ Windows hace parecer a la red una extensión de tu computadora.

En general un administrador de Windows seguramente argumentará que un sistema UNIX es viejo, con un sistema de comandos críptico y caduco. Sin embargo un administrador UNIX nos dirá que Windows no es fiable y que no se adapta bien a las necesidades que puedan exigírsele en el camino.

Y así podríamos seguir escuchando comentarios sobre ambos sistemas operativos sin llegar a una conclusión fundamentada. Es por eso que nos referiremos en este capítulo a la comparación objetiva de las características más importantes de ambos sistemas operativos.

Los sistemas operativos ideados para dar servicio a las PC's y a las pequeñas computadoras han abierto un mercado de servidores intel x86 en el mundo empresarial. Windows es una buena muestra de la evolución que ha sufrido la arquitectura PC, aunque buena parte ha tenido que ver con la agresiva política de mercadeo de Microsoft. Por su parte Linux hace gala de la administración de los recursos de este tipo de máquinas aprovechando los avances en la arquitectura x86 haciendo comparable una computadora personal con estaciones de trabajo.

Antes de proseguir con esta comparación debemos distinguir la forma como están enfocados los desarrollos de ambos sistemas operativos:

✓ **Windows.** Se ha caracterizado por sus constantes mejoras en la interfaz del usuario, siendo el sistema operativo para servidores PC más fácil de usar en todo el mercado, sin embargo es un sistema poco confiable por sus ya conocidas fallas y la poca escalabilidad de sus aplicaciones.

✓ **Linux.** Se ha caracterizado por ser un sistema operativo difícil de configurar y poner a punto, además de los problemas de compatibilidad de hardware. Sin embargo la estabilidad y escalabilidad que presenta este sistema operativo son únicas para servidores PC.

Como podemos constatar el desarrollo de ambos sistemas operativos lleva un enfoque diferente por parte de sus creadores, unos están dedicados principalmente a la mercadotecnia y facilidad de uso de su sistema operativo y los otros están empeñados en hacer el sistema operativo (técnicamente hablando) más perfecto. Lo anterior no quiere decir que ambas partes descuiden el lado opuesto, por ejemplo, Windows en su última versión incorpora un sistema de archivos novedoso y eficiente; por su lado Linux presenta una serie de escritorios gráficos de trabajo muy amigables entre los cuales se encuentran principalmente G-NOME y KDE.

Un punto importante de comparación entre ambos sistemas operativos es el conjunto de aplicaciones usadas en estos: Windows ofrece a los administradores de Sistemas interfaces que hacen la administración sencilla y fácil de entender, sin embargo ha de reconocerse que la sencillez también tiene su costo: Existen para Windows una serie de herramientas de administración para servidores WWW, DNS, Correo electrónico, etc. que no son más que versiones reducidas de las originales donde el objetivo principal de esta reducción es la sencillez.

En cambio la mayoría de estas aplicaciones han sido reproducidas o migradas de sus originales en UNIX para ser usadas con Linux y muchas de ellas han sido incluso mejoradas, o se les han agregado nuevas funcionalidades.

Con lo anterior queremos decir que existe una serie de factores clave en Windows que podrían afectar seriamente su escalabilidad a mediano y largo plazo de las aplicaciones de soporte al sistema operativo.

6.1 Windows y Linux frente a frente

6.1.1 Windows

El líder de la familia Windows, distribuido a gran escala en el mercado fue la versión Windows 3.1, después se abordó la falta de ínter conectividad entre los distintos usuarios, de tal forma que Microsoft desarrolló Windows para trabajo en grupo o Windows 3.11 que permitía compartir directorios, impresoras y otros dispositivos en una red local. El siguiente paso de Microsoft fue construir un sistema operativo que ofreciera mayores prestaciones en el ámbito de la conectividad, dotado de una mayor robustez. Los servidores Windows incorporan aplicaciones de gestión de red, la interfaz de usuario propia de Windows 95 y servicios de conectividad con otras redes y con Internet.

6.1.2 Linux

Linux es un sistema operativo tipo UNIX, que se incorpora a toda una familia de sistemas operativos en la que están incluidos AIX, Digital Unix, SCO, Solaris, etc. Sólo que Linux fue diseñado originalmente para correr en plataformas x86. Linux tiene su herencia de aquellos viejos sistemas multitarea de tiempo compartido para mini computadoras y súper computadoras de mediados de los 70's y de las últimas versiones de los UNIX más comerciales. Desde aquellos 70's el UNIX original ha cambiado y se ha convertido en uno de los sistemas más aceptados en el mercado y Linux ha roto varios record en las tasas de crecimiento avanzando a pasos agigantados.

Linux es una versión de UNIX de distribución libre desarrollada en sus inicios por Linus Torvalds de la Universidad de Helsinki en Finlandia. Inicialmente sólo fue un proyecto inspirado en Minix (un pequeño UNIX) pero gracias a Internet pudo desarrollarse con la colaboración de muchos programadores de todo el mundo creciendo tanto que se ha convertido en un sistema operativo para los negocios, la educación y aplicaciones personales, caracterizándose por ser una implementación de UNIX versátil, y con grandes ventajas desde el punto de vista de la distribución.

6.2 Comparativo de los Sistemas Operativos

Para hacer este comparativo, hemos buscado una forma fiable y objetiva, de forma contraria a la información proporcionada por sitios como <http://www.msn.com> y <http://www.linuxbenchmarks.org> que es manipulada de manera comercial en su mayor parte, puesto que estas pruebas de rendimiento se han hecho manipulando las virtudes y deficiencias de ambos sistemas operativos.

Por las razones dadas; fundamentaremos la comparación en las características arquitectónicas y estructurales de ambos sistemas operativos expuestas en los capítulos IV y V de este trabajo.

6.2.1 Sistema de Archivos

El sistema de archivos de todo sistema operativo es uno de los puntos esenciales que deben de tomarse en cuenta ya que es ahí donde queda guardada la información producida. Un sistema de archivos debe ser robusto, confiable y sobre todo recuperable ante fallas, ya sean estas del sistema o externas al sistema.

Windows	Linux
<p>Soporta de forma nativa varios sistemas de archivos como son:</p> <ul style="list-style-type: none"> ✓ NTFS ✓ FAT32 ✓ FAT 16 ✓ ISO9660 ✓ NFS 	<p>Soporta de forma nativa una gran variedad de sistemas de archivos entre los principales tenemos:</p> <ul style="list-style-type: none"> ✓ ext3fs ✓ ex2fs ✓ BSD ✓ Minix ✓ NTFS ✓ NFS ✓ FAT32 ✓ FAT16 ✓ ISO9660
<p>El sistema de archivos de Windows es NTFS el cual proporciona un alto nivel de seguridad por medio de ACL's y permisos de grupo y de usuario.</p>	<p>El Sistema de archivos de Linux es ex3fs el cual proporciona un nivel de seguridad a nivel de identificadores de usuario y de grupo.</p>
<p>Para crear particiones adicionales es necesario crearlas dentro de otra llamada <u>partición extendida</u>.</p>	<p>Soporta 4 particiones principales de 2TB como máximo en cada disco duro.</p>
<p>Proporciona una herramienta de auto-recuperación, en caso de que halla ocurrido una contingencia con el sistema. Esta herramienta solamente arregla problemas menores con el sistema y será necesario usar herramientas de terceros si el problema es grave.</p>	<p>Proporciona varias herramientas de recuperación para el sistema de archivos entre ellas una de auto-recuperación llamada fsck, si el problema es grave se tendrá que recuperar por medio de sistemas de archivos alternos como lost+found.</p>

NTFS es el sistema de archivos de Windows, aunque también puede trabajar con el sistema tradicional de FAT utilizado de forma habitual. NTFS admite un mayor grado en el control de acceso del que es posible bajo Linux. En NTFS, el usuario que creó el archivo o el directorio es el único que tiene permisos para acceder, modificar o cambiar los permisos de los mismos. Un archivo posee además de los permisos una lista de control de acceso (ACL: Access Control List) que define los permisos de acceso de usuarios o grupos a los archivos o directorios.

I.- Permisos individuales

En Windows el propietario puede conceder los siguientes permisos a otro usuario para los archivos y directorios:

1. Sin acceso alguno.
2. Lectura. El usuario puede leer y ejecutar, pero no se le permite la modificación.
3. Cambio. El usuario puede ver, leer, ejecutar, modificar, cambiar atributos y eliminar archivos o directorios.
4. Control total: Leer, ejecutar, modificar, eliminar archivos y tomar el control sobre ellos.
5. Acceso especial. Permite conceder el permiso P para la creación de un registro de control de acceso dentro de la ACL.

Exclusivos para directorios

1. Listado. El usuario puede ver los nombres de los archivos en el directorio pero no puede acceder a los archivos mismos.
2. Agregar. El usuario puede añadir un archivo al directorio.

II.- Permisos de grupo:

Existen dos tipos de grupos en Windows : los locales y los globales. Los grupos locales son aquellos que quedan definidos por el usuario y se usan para asignar permisos a los recursos locales. Un grupo local puede contener usuarios y grupos globales. Para facilitar la gestión de los grupos existen grupos predeterminados con una serie de derechos ya preestablecidos.

En Linux se sigue el estándar de seguridad tradicional de todo Sistema UNIX existen tres categorías de permisos: usuarios, grupo y otros, además de poder generar listas de control de acceso por archivo. A cada categoría se le pueden asignar los privilegios de lectura, escritura y ejecución. Para cada archivo y directorio se almacena el identificador de usuario(UID) y de grupo(GID), de tal forma que la combinación de ambos permisos privilegios da como resultado los privilegios de acceso de los usuarios al sistema de archivos.

6.2.2 Administración de la Memoria

La tarea de administrar la memoria del sistema es un factor crítico en el desempeño de todo sistema operativo, sobre todo hablando de servidores x86 que por razones de Hardware solamente pueden direccionar de manera real 640k y lo demás es manejado por software.

Windows	Linux
Soporta en su forma inicial hasta 256MB de memoria, siendo necesario controlador adicional para soportar mas memoria.	Soporta en su forma inicial 2GB y es necesario modificar el kernel para llegar a una extensibilidad máxima de 16GB.
Es necesario mantener espacio disco para las rutas especificadas como de archivos temporales (variable TMP) porque en ese lugar se crean los archivos de intercambio de memoria los cuales son manejados por porcentajes de espacio libre en disco duro.	Es necesario crear particiones de intercambio de memoria, las cuales tienen un formato especial exclusivo para esa tarea.
La asignación de memoria se hace dentro de los primeros 640K teniendo que ser trasladados posteriormente a la memoria alta.	Asignación directa de la memoria alta.

6.2.3 Administración de Procesos

La administración de procesos corresponde a la tarea más crítica de todo sistema operativo la cual se lleva a cabo en modo privilegiado dentro del CPU. Es importante balancear la carga de información que conlleva cada proceso ya que mucha información acerca de un proceso puede ayudar a administrarlo mejor; pero puede también repercutir en el rendimiento y si vemos el caso contrario puede: aumentar mucho el rendimiento pero se pueden ver afectados la estabilidad y rendimiento.

Windows	Linux
La calendarización de procesos es una de las tres tareas principales del micro-kernel de Windows haciendo una multitarea sobre demanda al sistema operativo.	La calendarización, seguridad y administración de procesos es desempeñada en su totalidad por el kernel del sistema operativo realizando una multitarea prioritaria, guardando relaciones jerárquicas entre procesos, grupos de procesos e hilos.
La mayor parte de la información relacionada con los procesos es administrada por el Object Manager de Windows, el cual no guarda relaciones jerárquicas entre procesos y estos son tratados como objetos del sistema, teniendo cada uno un segmento protegido y su tabla de estado.	Toda la información relacionada con un proceso se administra en el módulo administrador de procesos del kernel.

6.2.4 Rendimiento

Un sistema operativo de alto rendimiento es necesario para atender las demandas del mercado, sin embargo al aumentar considerablemente el número de procesos es importante también el funcionamiento que ofrezca el sistema operativo.

Windows	Linux
Es el sistema operativo para servidores con mayor rendimiento si se ejecutan pocos procesos debido a su administración de memoria y a su administración de procesos, sin embargo, cuando el número de procesos aumenta considerablemente, el rendimiento del sistema operativo baja considerablemente.	Es un sistema operativo con muy buen rendimiento si se ejecutan pocos procesos (Por debajo de Windows), sin embargo el funcionamiento óptimo se logra cuando un gran número de procesos se encuentran corriendo al mismo tiempo debido a su administración de procesos y de memoria que le permite tener mayor capacidad.

6.2.5 Escalabilidad

La escalabilidad de un sistema operativo no solamente está en función del hardware que puede ser agregado para ir cubriendo las demandas de crecimiento en el uso del servidor, sino que este hardware realmente sea utilizado de forma optima guardando una relación casi lineal entre el hardware agregado y el rendimiento de la computadora.

Windows	Linux
Windows presenta una serie de deficiencias en cuanto a escalabilidad se refiere; gran parte de ellas derivadas del problema de rendimiento expuesto anteriormente.	Linux provee de una alta escalabilidad derivada de su herencia como sistema UNIX.
Windows soporta hasta 16 procesadores en arreglo simétrico, así como el manejo de agrupamientos a través de RPC. Sin embargo ha quedado	Soporta 32 procesadores en arreglo simétrico, de los cuales el número máximo conocido que se ha usado realmente es de 8 para hacer los efectos especiales de la película

demostrado en la práctica que después de 4 procesadores en arreglo SMP Windows no ofrece una alta escalabilidad ni una relación costo beneficio rentable.	TITANIC. Así mismo también soporta el manejo de agrupamientos por medio de conexiones RPC.
---	---

6.2.6 Confiabilidad

Hoy en día la confiabilidad es uno de los puntos más importantes a ser evaluados en un sistema operativo, ya que es relativamente fácil aumentar el hardware de una maquina para obtener mayor rendimiento, sin embargo es inaceptable que el sistema se caiga o pierda datos.

Windows	Linux
Es un sistema confiable, sin embargo su estabilidad deja mucho que desear debido a un control de procesos poco robusto y a numerosas fallas del sistema que tardan varios años en ser arreglados y son puestos como parches al sistema operativo (Service Packs).	Es un sistema catalogado de confiabilidad probada, además de que la corrección de fallas en el sistema avanza a pasos agigantados todos los días e incluso es posible arreglar las fallas uno mismo.

Seguridad, confiabilidad y rendimiento son tres variables que compiten unas contra otras. ¿Qué preferimos?, un servidor que funcione rápido pero que sepamos de que de vez en cuando se caerá; o estamos dispuestos a ceder un poco en el rendimiento en aras de la confiabilidad.

Hoy por hoy la confiabilidad es una de las características más importantes de un servidor, puesto que en caso de no estar conformes con el rendimiento siempre podremos aumentar las características del hardware de nuestro equipo. Por lo tanto, podemos considerar la fiabilidad como crítica en la elección de nuestro servidor y sistema operativo, teniendo en cuenta que viene dada por la consistencia de los procesos que están corriendo en el servidor, la integridad de los datos en memoria, los chequeos de la máquina, algoritmos de integridad, etc.

Windows representa un sistema estable pero aún no ha alcanzado lo que ofrece un sistema Linux, al que se puede considerar como de fiabilidad probada. ¿Cuántos de nosotros hemos podido disfrutar de una pantalla azul en un Windows que amablemente nos invita a reiniciar nuestro equipo?.

Però un punto en contra de Linux consiste en la falta de facilidad en la recuperación de datos borrados del disco por error, tarea que en Windows resulta mucho más sencilla, siempre y cuando se cuente con las herramientas adecuadas. Tanto Linux como Windows trabajan ejecutando sus procesos en espacios de memoria distintos, lo que representa un nivel de seguridad amplio ante el fallo de cualquiera de ellos, ya que si un proceso falla no afecta a la ejecución a los demás procesos.

6.2.7 Administración y configuración

La administración de un sistema operativo tiene que ser robusta y debe de abarcar todos los aspectos del sistema y no solamente los más comunes. Indudablemente el punto más fuerte de Windows en comparación con todos los demás sistemas operativos, es que es el sistema operativo más fácil de configurar y administrar. Sin embargo estas facilidades deben de tener un costo oculto ¿Cuál es?

Windows	Linux
Es un sistema operativo fácil de administrar y de configurar.	Se requieren conocimientos técnicos especializados para su administración.
Poco grado de especialización de aplicaciones servidoras tales como WWW, Mail, DNS, DHCP, SMB etc.	Alto grado de especialización de aplicaciones.
Poca automatización de las tareas de administración.	La automatización de tareas de administración es total, el único requisito son los conocimientos técnicos del administrador de sistema.
Interprete de comandos poco robusto.	Se puede elegir entre una gran variedad de interpretes de comandos los cuales incluyen un lenguaje de programación completo.

Interfaz amigable e intuitiva que solamente corre en modo gráfico.	Opción de trabajar en modo texto o en gráficos. Para el caso de modo gráfico existen una serie de interfaces distintas entre las que puede escoger el usuario.
Opción de comprar el equipo con el sistema operativo ya instalado con diferentes proveedores.	Opción de comprar el Equipo con el sistema operativo ya instalado, sin embargo todavía no son muchos los proveedores que ofrecen este servicio.
Administración remota parcial quedando algunos detalles pendientes como archivos y configuraciones especiales.	Administración remota total de las funcionalidades del sistema.

A primera vista podemos pensar que la administración de Windows es muy sencilla y cualquiera está preparado para afrontarla, sin embargo administrar servidores Windows no es tarea fácil, aunque sí resulta sencillo administrarlo mal. Lo que parece indudable es que un administrador calificado tiene (en las tareas más comunes del administrador) más facilidades a la hora de trabajar bajo Windows. Una interfaz gráfica ayuda en gran medida a agilizar esta tarea, apoyada en una ayuda sensitiva al contexto que verifica si estamos seguros de las elecciones realizadas antes de hacerlas efectivas. Pero la diferencia no es mucha, puesto que con Linux también podemos utilizar un sistema de administración gráfico que va tomando robustez muy rápidamente. Con todo, podemos decir que la administración en un sistema Linux es una tarea más especializada de lo que es una administración de Windows.

Linux está equipado con un varios shell mucho más versátiles que el de Widows, además debemos destacar la facilidad que ofrece a la hora de automatizar tareas, asignando los tiempos en los cuales se lanzan las ejecuciones de los programas. Las tareas cronológicas que podemos programar son de muy amplia versatilidad, y podemos indicar cualquier tipo de intervalo en el que serán efectuadas. La suma del shell y el cronab suponen un gran paso para la tarea de la administración. Tanto es así que estos dos puntos son uno de los objetivos que Microsoft ha previsto solucionar con el lanzamiento de sus futuros Windows para que tengan incorporado un lenguaje de script más versátil.

6.2.8 Seguridad

No cabe duda de que la seguridad es uno de los puntos claves para todo sistema operativo.

Se podría definir el concepto de seguridad como la habilidad del sistema para defenderse de los ataques e intrusiones exteriores, siendo condición indispensable imprescindible que el sistema operativo ofrezca al administrador un subsistema de configuración de permisos para los distintos usuarios que sea robusto y altamente configurable. En este punto los dos sistemas operativos obtienen una alta calificación y podemos considerar a ambos sistemas operativos razonablemente seguros. Windows tiene la certificación de seguridad C2, certificación que no ha adquirido Linux por falta de recursos económicos, lo cual es comprensible tomando en cuenta la forma de desarrollo en Linux.

El estándar C2 de seguridad exige las siguientes características:

- ✓ Requiere que los usuarios se identifiquen introduciendo un único y exclusivo identificador de acceso y una contraseña antes de que le sea permitido el acceso al sistema.
- ✓ El propietario de un recurso puede determinar quién puede acceder al recurso y lo que puede hacer con él. Dicho dueño consigue esta característica garantizando derechos de acceso a usuarios o a grupos de usuarios.
- ✓ La auditoria proporciona la posibilidad de detectar y registrar eventos importantes relacionados con la seguridad al crear, acceder o eliminar recursos del sistema. Utiliza identificadores de acceso para registrar la identidad del usuario que llevó a cabo la acción.
- ✓ La protección de memoria evita que cualquiera lea información escrita por otra persona, incluso después de que una estructura de datos haya sido devuelta al sistema operativo. La memoria es reinicializada antes de volver a ser utilizada.

Windows	Linux
Certificación C2 de seguridad para este sistema operativo.	El sistema operativo no se ha certificado por razones monetarias (\$15,000 USD).
Sistema expuesto a virus.	No existe riesgo de virus.
Sensible a ataques de caballos de troya en puertos específicos.	Sensible a ataques de seguridad interna si no son usadas herramientas como shadow y md5.

Las fallas de Windows en este sentido han sido muy conocidas y han dado alguno que otro dolor de cabeza a los administradores. Pongamos como ejemplo el ataque a los servidores web mediante el método "denial of service" (Rechazo de servicio), también conocido como Bonk, Boink y New Tear. El método se valía de un fallo de Windows, que deja inestables a los servidores al obligarles a auto bombardearse con información.

6.2.9 Desarrollo de Aplicaciones

En este punto Windows se encuentra a la cabeza con respecto a Linux si hablamos de aplicaciones comerciales, ya que el mercado de aplicaciones para Windows lleva ya mucho tiempo expandiéndose. Sin embargo la cantidad de proyectos de desarrollo en que se está trabajando sobre Linux es inmensa y el resultado de estos comienza a verse conforme estos van madurando a través de Internet.

6.2.10 Resumen Comparativo

Características	Windows	Linux
Estabilidad	Buena: Comparado con los demás sistemas operativos de PC es bastante fiable, pero no alcanza los niveles ofrecidos por Linux.	Muy buena: Es uno de los puntos fuertes de este sistema operativo. podemos considerarlo como de estabilidad probada.
Facilidad de uso	Muy buena: Gracias a la ideología de Microsoft que apoya las buenas interfaces mediante las cuales facilita la vida al usuario.	Requiere de sus usuarios un mayor grado de especialización.
Fabricantes	Microsoft es el único propietario, por lo cual la evolución de este sistema operativo está sometida a las decisiones de una única compañía.	Dadas sus características no existen fabricantes para este sistema operativo y el desarrollo se da a través de individuos en internet.

		Sin embargo las compañías encargadas de las distribuciones compiten entre si para hacer mejor a Linux.
Administración	Buena: la administración de toda la red puede realizarse desde un único equipo y está apoyada por la facilidad que aportan las interfaces de Windows, aunque aún es obligatorio reiniciar el servidor cuando se realizan nuevas instalaciones o cambios en la configuración.	Muy buena: No requiere reiniciar el servidor ante los cambios de configuración o nuevas instalaciones. Destacan las aplicaciones que facilitan la vida al administrador tales como el cron y el telnet.
Telnet	Sí, tanto la parte del cliente como la del servidor.	Sí, tanto la parte del cliente como la del servidor.
Automatización	Mala, pues se debe recurrir a aplicaciones externas para poder desarrollar guiones complicados y programar los eventos temporales.	Muy buena, las aplicaciones necesarias para la automatización ya vienen incorporadas además de ser muy poderosas y flexibles.
Variantes entre versiones	Pocas, al ser un único fabricante se mantiene coherencia.	Solamente las especificadas por la distribución que se utilice, sin embargo el kernel del sistema operativo es el mismo para todos los fabricantes.
Seguridad	Buena: cumple con el estándar de seguridad C2.	Buena: no esta certificado con el estándar de seguridad C2.
Limitación del espacio de disco de usuario	Windows no dispone de esta posibilidad por si solo	Sí, uso de cuotas de usuario.
Rendimiento	Excelente.	Bueno

Soporte de multiprocesadores simétricos	Sí.	Sí.
Concurrencia	Sí.	Sí.
Apoyo al desarrollo	Excelente, el inmenso número de aplicaciones existentes para desarrolladores en este sistema es apoyado por la calidad de las mismas y la velocidad con que Microsoft actualiza sus diferentes herramientas.	Excelente al no estar ligado su desarrollo a objetivos Empresariales únicos. Actualmente un gran número de empresas están migrando sus aplicaciones a Linux además de los desarrollos GNU.
Precio	Económico, pero debemos tener en cuenta que si no se compra software adicional únicamente vamos a poder explotarlo como servidor de archivos y/o impresoras.	Ninguno.
Rsh	SI. RSHSWC es el equivalente al rsh de Unix, aunque no distribuido como parte de Windows. Se distribuye con el Kit de recursos de Windows.	Sí.
Control de acceso al sistema de archivos	Muy bueno, admite mayor grado de configuración en el control de acceso.	Bueno.
Fiabilidad	Los procesos se ejecutan en espacios de memoria distintos y el fallo de uno de ellos no provoca la caída del otro.	Excelente, podemos considerarlo como de fiabilidad probada.

Capítulo VII. Comparación de aspectos Comerciales

El Software libre está sujeto a un proceso de desarrollo que estimula la rápida creación e implantación de funcionalidades que aumentan la capacidad del software y depuran el código base. En años recientes, como respuesta al crecimiento de Internet, los proyectos de software libre han adquirido una complejidad y profundidad tradicionalmente asociadas con proyectos comerciales, tales como sistemas operativos aplicaciones de misión crítica, etc.

Este software libre representa a corto plazo una amenaza directa a la economía y al dominio de la plataforma Intel x86, para la empresa Microsoft dentro del ámbito de los servidores. El libre intercambio de ideas y el paralelismo intrínseco al desarrollo del software libre tiene beneficios que no puede reproducir el actual modelo de licenciamiento de compañías comerciales como Microsoft. Sin embargo, las debilidades del proceso de construcción del software libre proporcionan un camino para que compañías como Microsoft puedan obtener ventaja en áreas clave tales como mejoras de arquitectura, integración, facilidad de uso y apoyo organizativo.

7.1 ¿Qué es el Software Abierto o libre?

El Software Abierto es software cuyo código fuente y binarios de un producto dado, están disponibles, generalmente de manera gratuita. Es común que no se distinga la diferencia entre shareware o freeware y software abierto, pero existen varias diferencias significativas entre los modelos de licenciamiento y el proceso mediante el cual se desarrolla cada tipo de producto como se muestra en la siguiente tabla:

Características de la licencia:	Precio cero	Redistribuir	Uso Ilimitado	Código fuente disponible	Código fuente modificable	Revisión publica el código fuente	Todos los derivados deben de ser gratuitos
Comercial	X	X					
Software de prueba	X	X					
Software de uso no comercial	X	X					
Shareware	X	X					
Binarios Gratuitos	X	X	X				
Librerías Gratuitas	X	X	X	X			
Fuente abierta (estilo BSD)	X	X	X	X	X		
Fuente abierta (estilo Apache)	X	X	X	X	X	X	
Fuente Abierta (estilo Linux)	X	X	X	X	X	X	X

Las características de licenciamiento incluyen:

✓ **Software Comercial.** Este software debe ser comprado, no puede ser redistribuido, y típicamente se pone a disposición del usuario final sólo los binarios o ejecutables.

✓ **Software de Prueba.** El software de prueba se refiere a versiones limitadas del software comercial que se distribuye libremente y cuya intención es la compra del código comercial. Como muestra de ello está el software de evaluación de 60 días.

TESIS CON
FALLA DE ORIGEN

✓ **Shareware.** Los productos de software son totalmente funcionales y no tienen restricciones en cuanto a su distribución, pero existe una licencia que obliga eventualmente a los individuos y a las corporaciones a la compra de la licencia para su uso legal. Muchas herramientas del Internet (tales como WinZip) utilizan este método de distribución.

✓ **Software de Uso No Comercial.** El software de uso no comercial está disponible sin limitaciones, y se puede redistribuir por entidades sin fines lucrativos. Las corporaciones deberán adquirir este producto. Un ejemplo de esto sería Netscape Navigator.

✓ **Binarios Gratuitos.** Los binarios gratuitos consisten de software cuyo binario se puede usar y distribuir libremente. Los binarios del Internet Explorer y NetMeeting entran dentro de este modelo.

✓ **Bibliotecas Gratuitas.** Las bibliotecas gratuitas son productos de software cuyos binarios y fuentes pueden ser utilizados y distribuidos libremente, pero no pueden ser modificados por el usuario final sin violar la licencia. Ejemplos de esto son las bibliotecas de clases como el "Java Development Kit" que proporciona una serie de bibliotecas para programar en Java, etc.

✓ **Fuente Abierta (Estilo BSD).** Un pequeño y cerrado equipo de desarrolladores estilo BSD desarrolla productos de software abierto y permite el libre uso y redistribución de binarios y códigos. A pesar de que los usuarios pueden modificar el código, el equipo de desarrollo generalmente no revisa lo que aporta el público.

✓ **Fuente Abierta (Estilo Apache).** Apache asume el modelo de desarrollo de software abierto de BSD y lo extiende al permitir revisiones al código fuente por parte de terceras personas.

✓ **Fuente Abierta (Estilo GNU/Linux).** El software basado en el GPL lleva un paso crítico más allá. Mientras que el software estilo BSD y Apache permiten a los usuarios utilizar el código base y aplicar sus propios términos de licencia al código modificado (e.g. hacerlo comercial), la licencia GPL obliga que todos los trabajos derivados sean a su vez código GPL. "Eres libre de usar este código en tanto que el derivado se pueda usar bajo las mismas condiciones."

El software libre es radicalmente diferente de otras formas de licenciamiento en dos aspectos muy importantes:

✓ Siempre existe un camino de adquisición completamente gratuito del código base.

✓ A diferencia de los binarios libremente distribuidos, el Software libre estimula un proceso alrededor del cual está el código base y estimula también las extensiones del código base por parte de otros desarrolladores.

TESIS CON
FALLA DE ORIGEN

Actualmente el software libre ha representado una preocupación latente a compañías comerciales de software por tres razones básicas y fundamentales:

1. **Los proyectos de software libre han logrado calidad comercial.** Una de las principales barreras para el software libre en muchos ambientes de clientes es que se ha percibido una falta de calidad. Los entusiastas del software libre plantean, en cambio, que una mayor revisión y depuración del código del software libre da como resultado un código de mayor calidad que el software comercial. Estudios recientes de casos (el Internet) proporcionan evidencia de que la calidad comercial puede lograrse e incluso superarse por los proyectos independientes.
2. **Los proyectos de software libre se han convertido en proyectos de gran escala y complejidad.** Otra barrera que ha sido abordada por el software libre es la complejidad del proyecto. Los equipos de desarrollo de este tipo de software están llevando a cabo proyectos cuyo tamaño y complejidad había sido hasta ahora el dominio exclusivo de equipos de desarrollo comerciales, organizados o motivados por intereses económicos. Ejemplos de estos lo constituyen el sistema operativo Linux y el XFree86 GUI. Esto se debe a que el desarrollo del software libre esta intrínsecamente ligado a Internet el cual permite distribuir los recursos de desarrollo a gran escala.
3. **La calidad de los desarrolladores participantes suele ser mayor.** El software libre tiene un proceso de desarrollo único, con fortalezas y debilidades únicas. El proceso de construcción de software libre es único en cuanto a sus participantes, ya que se pueden proporcionar las motivaciones y los recursos para abordar los problemas. Por lo tanto, el software libre tiene algunas características no reproducibles que deben ser comprendidas cabalmente.

7.2 Fortalezas del Software Abierto

¿Cuáles son las fortalezas centrales de los productos de software libre por los que deben estar preocupado Microsoft y cuales son su atributos principales?

A semejanza del negocio del sistema operativo de Microsoft, el ecosistema de software libre tiene varios atributos exponenciales:

7.2.1 Los procesos de software libre están creciendo en el Internet

La limitación más grande a la que se enfrenta cualquier proyecto de software libre es encontrar la cantidad necesaria de desarrolladores dispuestos a invertir su tiempo en el proyecto. Internet es indispensable como medio que haga posible este proceso y reunir la cantidad necesaria de gente para un proyecto de la escala de un sistema operativo. El avance de este tipo de proyectos esta determinado por el crecimiento Internet. Las mejoras en las tecnologías de comunicación van aceitando directamente el motor del software libre. Dicho de otra manera, el crecimiento del Internet hará que los proyectos software libre existentes se hagan cada vez más grandes, y también hará posible que vuelvan viables los proyectos de software libre en categorías más reducidas.

7.2.2 Los procesos de software libre son del tipo "el ganador se lleva todo"

A semejanza del software comercial, el proyecto de software libre más viable eliminará a los demás de su misma categoría. Por ejemplo, Linux está eliminando a BSD UNIX y ha absorbido la mayor parte de sus ideas centrales (así como las ideas de los UNIX comerciales). Esta característica le da ventajas comparativas enormes a un proyecto en particular.

7.2.3 Los desarrolladores buscan contribuir a la plataforma más grande de software libre.

Entre más grande sea el proyecto de software libre, mayor será el prestigio que adquirirá el desarrollador con su contribución de algún componente importante de alta calidad. Este fenómeno contribuye a reforzar el hecho de que "el ganador se lleva todo" en los procesos de software libre para determinado segmento.

7.2.4 Los proyectos más grandes de software libre tienen un efecto cinegético

Entre mayor sea un proyecto, mayor será la cantidad de desarrollo, prueba y depuración que recibirá el código. Entre más grande sea la cantidad de depuración involucrada, mayor será el número de gente que querrá emplearlo.

7.3 Reacciones de Microsoft al fenómeno del software libre

Sin duda el software abierto ha marcado una pauta en el estilo de desarrollo de software, además de que ha sido el centro de atención de muchas compañías desarrolladoras de software, las cuales trabajan incansablemente en modelos de licenciamiento parecidos para poder desarrollar software bajo los esquemas de software libre.

Una serie de documentos denominados "Halloween" muestran la preocupación de Microsoft hacia el fenómeno del software abierto, y como los altos directivos de esta compañía ya están viendo a esta clase de software como una amenaza para sus objetivos comerciales.

Dentro de los puntos tratados en estos documentos se pueden citar los siguientes como los más importantes:

- ↳ El software libre representa a corto plazo una amenaza directa, a los ingresos económicos y al dominio de la plataforma de Microsoft, especialmente en el ámbito de los servidores. Además, el paralelismo intrínseco, y el libre intercambio de ideas dentro del software libre tiene beneficios que no puede reproducir nuestro actual modelo de licenciamiento, y por tanto, representan una amenaza por el intercambio de ideas entre los desarrolladores.

✓ Estudios recientes de casos (el Internet) proporcionan evidencia de que la calidad comercial puede lograrse e incluso superarse por parte de los proyectos de software libre.

✓ Para comprender cómo competir en contra del software libre, debemos dirigirnos en contra de un proceso, en vez de una compañía.

✓ Linux y los entusiastas del software libre están haciendo cada vez más creíble el planteamiento de que el software libre es al menos tan robusto, si no es que más que las alternativas comerciales. Internet proporciona el escaparate ideal, de gran visibilidad para el mundo del software libre.

✓ Linux ha sido empleado en ambientes comerciales, de misiones críticas con excelentes resultados y testimonios públicos. Linux tiene un mejor desempeño que muchos otros UNIX, Linux está en camino de ser eventualmente dueño del mercado UNIX de las x86

✓ Linux puede ganar mientras los servicios y los protocolos sean mercancías sujetas a la venta.

✓ Los proyectos de software libre han sido capaces de ganar una amplia ventaja en muchas aplicaciones de servidores debido a la amplia utilización de protocolos sencillos y personalizados. Al extender estos protocolos y desarrollar otros nuevos, podemos evitar la entrada de los proyectos de software libre al mercado.

✓ La capacidad del proceso de software libre para reunir y de aprovechar el coeficiente intelectual colectivo de millares de individuos a través de Internet es simplemente sorprendente. Es más, reclutamiento de adeptos al software libre crece cada vez más rápido con Internet de lo que los esfuerzos de Microsoft parecen crecer.

Los anteriores, son solamente fragmentos del documento completo, sin embargo a través de ellos se puede notar que la preocupación de Microsoft hacia el software abierto ya tiene un nombre y una forma: Linux.

Microsoft expresa su preocupación principalmente en el ámbito de servidores Windows, los cuales a pesar de su facilidad de uso presenta aún grandes deficiencias que ya han sido superadas por Linux, sobre todo en los rubros de escalabilidad, estabilidad y conectividad.

Actualmente Windows ha acaparado el mercado de los servidores X86 de una forma arrebataadora lo cual ha causado que los administradores de redes y las empresas en general hayan escogido a este sistema operativo como plataforma de base para sus operaciones diarias.

Linux por su parte ha sido probado por un gran número de empresarios y administradores de redes con resultados excelentes, los cuales se han hecho públicos, al grado de que se dice que la estabilidad de Linux es legendaria.

7.4 Linux como un software de distribución libre

Como otros productos de software libre, la clave real para Linux no es una versión estática del producto, sino un proceso alrededor de él. Este proceso le da su credibilidad y un aire de futura seguridad a las inversiones de los clientes en Linux.

Linux ha sido empleado en ambientes comerciales de misión crítica con un gran número de testimonios públicos favorables.

Se puede decir que Linux es la Mejor Variante de UNIX. Linux tiene mejor desempeño que muchos otros UNIX (interfaz de redes, Entrada / Salida de disco, interruptores control de procesos, etc.). Para hacer crecer sus características base, Linux ha tomado características de otros UNIX (características del shell, sistemas de archivos, gráficos, puertos de CPU) adaptándolas y mejorándolas a una velocidad increíble.

El único sistema operativo UNIX en ganar una parte sustancial del mercado. Linux está en el camino de adueñarse del mercado x86 de UNIX y ha sido la única versión de UNIX en obtener una parte respetable del mercado de sistemas operativos para servidores actualmente.

Cuando Microsoft se enfrenta de Linux lo hace en el marco de los servidores por algunas características clave:

- ✓ Linux utiliza hardware de PC común. Debido a la modularidad del sistema operativo, puede ser ejecutado en sistemas de menor tamaño que Windows. Se utiliza frecuentemente a Linux para servicios tales como DNS corriendo incluso en las antiguas 486.
- ✓ Debido a su herencia UNIX, Linux representa un menor costo de cambio para algunas organizaciones que Windows.
- ✓ Las ventajas de escalabilidad, interoperabilidad, disponibilidad y facilidad de administración (SIAM) son superiores a las de Windows.

Linux ofrece una mayor gama de servicios y protocolos compatibles 100% con los comerciales.

Actualmente los procesos de desarrollo de software de distribución libre son extremadamente grandes y es incontable el número de desarrolladores involucrados.

IDC estima que las ventas de Linux fue de 1.65 millones de licencias durante el 2000 a través de los 6 mayores distribuidores de Linux y 100 organizaciones pequeñas. Además IDC estima que para finales de ese año se vendieron 422,000 servidores con versiones de servidor de Linux instalado.

Las compañías que han visto el negocio en Linux y otros productos de software libre han encontrado una buena remuneración, y se estima que en algunos casos las ventas se dupliquen rápidamente, lo cual implica un crecimiento en el desarrollo de Linux ya que cada vez son más los desarrolladores, administradores y sobre todo los directivos de grandes compañías que ven a Linux como una opción barata y poderosa para satisfacer sus requerimientos.

7.5 Porciones del Mercado

7.5.1. Tasas de Crecimiento

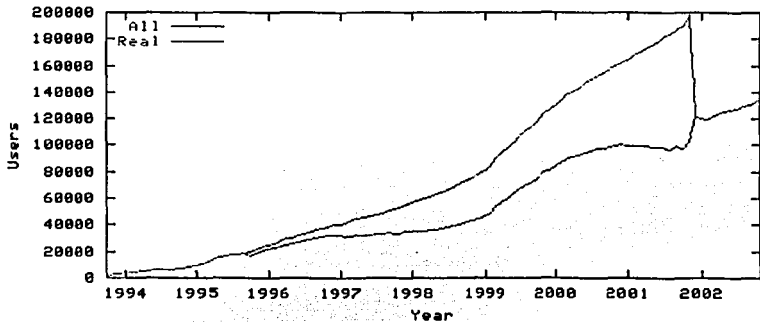
Linux se esta usando ya en grandes proyectos que van desde formar la infraestructura informática de una cadena de hoteles en Sudáfrica hasta hacer producciones de cine millonarias como Titanic donde el 90% de las máquinas utilizadas en su realización fueron máquinas Linux.

La tasa de crecimiento de los servidores Linux en el mundo ha sido alta durante los últimos años, IDC estima que la tasa de crecimiento de plataforma instalada de Linux es de 25% durante el periodo 2000 - 2003, aunque es bien sabido que el número de servidores Windows esta muy por encima de los servidores Linux y que por ende su tasa de crecimiento debe moverse a un ritmo mas bajo. De cualquier manera Linux se acerca rápidamente y a este ritmo en pocos años Linux competirá fuertemente con Windows seccionando el mercado de forma importante en cuanto a servidores x86 se refiere.

A pesar de estos datos la porción exacta del mercado es muy difícil de calcular debido a que:

- ✓ La mayor parte de las instalaciones de Linux son bajadas de sitios anónimos de FTP o son distribuidas en CD's de regalo en revistas y otros medios. Por lo tanto, no hay muchos datos de ventas publicados que seguir que sean confiables.
- ✓ Algunas adquisiciones comerciales de Linux pueden ser utilizadas en varias máquinas.
- ✓ Debido a las revisiones tan frecuentes de Linux, existe una alta probabilidad de un doble conteo de instalaciones reales vs adquisiciones por la red o compras.
- ✓ Para Linux no existen distribuciones separadas de clientes y servidores. Por lo tanto, es difícil comparar numéricamente la base instalada de Linux vs la base instalada de Windows en cuanto a servidores se refiere.

Lo que se muestra a continuación son datos de uno de los intentos más importantes de contar el número de usuarios de Linux.



** Fuente <http://counter.ll.org>

TESIS CON FALLA DE ORIGEN

7.5.2 Base Instalada

El estudio más serio sobre la porción que ocupa Linux fue publicado en marzo de 1998 utilizando los datos disponibles de otras distribuciones, Red Hat calculó la venta al menudeo de CD's en:

- 450,000 durante 1996
- 750,000 durante 1997

Otras estimaciones han colocado la base instalada de Linux en alrededor de cinco millones (Ziff Davis) a diez millones (entusiastas de Linux).

Un reporte reciente de IDC denominado "Microsoft Puts Pressure on Server Operating Environments Market Despite Tough Conditions in 2001, According to IDC" (Microsoft pone presión en el mercado de los sistemas operativos de servidores a pesar de las duras condiciones en el 2001, de acuerdo con IDC) proporciona el siguiente desglose de ventas dentro del rubro de los sistemas operativos:

El licenciamiento de servidores Windows paso del 42% en el 2000 al 49% en el 2001 avanzando sus posiciones en tres aspectos: ganancias, ventas de licencias y porción de mercado quedando sus competidores con los cifras como Linux con el 25.7%, Netware con el 11.7%, servidores UNIX con el 11.6% y otros con el restante 2%.

7.6 Apoyo de otras empresas

Ahora Linux no solamente esta avanzando por si solo frente a Windows, sino que también los gigantes del software están considerándolo seriamente; tal es el caso de las siguientes compañías:

Bea Systems	Sus principales productos WebLogic y TUXEDO corren en Linux, además de generar alianzas con empresas como Penguin Computing para vender estos productos ya instalados en servidores.
Sun Microsystems	Solaris Intel 2.8 es totalmente compatible con Linux e incluso ha tomado muchas bibliotecas del sistema Linux. Es importante mencionar también la adquisición por parte de Sun Microsystems de blackdown.org que distribuía la versión de java para Linux de forma gratuita. Y la incorporación de GNOME como escritorio propuesto para Solaris
IBM	Ha abierto una nueva división de soporte técnico para Linux y tiene versiones de DB/2 y WebSphere para Linux. Adicionalmente IBM ha incluido toda una serie de soluciones completas sobre Linux.
Informix	Ha hecho una versión completa de su motor de base de datos para Linux
Oracle	Ha migrado una gama de productos a Linux dentro de los que mas destacan son: El Motor de base de datos y el Jserver para persistencia de objetos java.
Compaq	Hoy en día los servidores de Compaq se pueden comprar con Linux ya instalado y listo para usarse en lugar de Windows
Dell	Dell provee toda una línea de servidores especializada en Linux

El siguiente es un fragmento de un amplio reportaje de IBM publicado en el Universal:

UN SISTEMA OPERATIVO DE 24 HORAS

"Desde Santiago de Chile, Steven C. Solazzo, vicepresidente de IBM Software América, aseveró: "Apoyamos a Linux porque lo vemos como el centro de la plataforma tecnológica. Hoy el mercado pide este Sistema Operativo".

Cabe señalar que el 2 de marzo de 1999, IBM anunció su apoyo a Linux, el sistema operativo de fuente abierta. El anuncio sobre las alianzas clave que se planean , los productos más importantes y las ofertas de servicios más completas de la industria hacen de IBM la única compañía que provee soluciones completas de hardware y apoyo técnico para Linux, apuntó el entrevistado.

Las noticias son:

- ✓ IBM ofrece apoyo global a las principales versiones de Linux, con lo que los clientes tendrán un solo punto de contacto para todas sus necesidades de apoyo técnico.

- ✓ IBM colabora con los distribuidores comerciales de Linux -Caldera Systems Inc., Pacific HiTech Inc., Red Hat Software Inc., y SuSe Holding AG- para abrirle el camino a las iniciativas de mercadotecnia conjunta, desarrollo, capacitación y apoyo que ayudarán a que sus clientes instalen el sistema Linux.

- ✓ Productos clave de IBM como WebSphere se venden en versiones para Linux, e incluso dos servidores de Aplicaciones y un paquete de rendimiento para que los clientes de Linux puedan explotar la Web y desempeñar tareas que pueden ser desde una simple publicación en la Web hasta el procesamiento de transacciones basadas en Java. Esta disponible el primer emulador comercial de la industria basado en Java para Linux, el IBM Host On-Demand, que proporciona un acceso seguro a los principales datos y aplicaciones empresariales por medio de un navegador de Web.

- ✓ En el segundo Trimestre de 1999, IBM comenzó las pruebas beta del servidor On-Demand para Linux que maneja el acceso de usuarios, grupos y dispositivos a las aplicaciones de negocios electrónicos; además, IBM está trabajando con la comunidad Linux para llevar el Linux a selectos modelos RS/6000. Según IDC Research, Linux es el ambiente operativo para servidores de más rápido crecimiento con una tasa del 212% aproximadamente de todos los nuevos servidores instalados. Hay más de 10 millones de

usuarios utilizando el Linux, y muchos de los clientes de IBM están pidiendo soluciones Linux, por lo que este último desea ofrecerles a sus asociados de negocios, la tranquilidad y seguridad de que, si desean usar Linux, IBM siempre les ofrecerá apoyo, al igual que hace con sus otros sistemas."

Este reportaje deja ver que las grandes compañías de software del mundo han puesto sus ojos en Linux y que éste, esta destinado a ser la barrera para Microsoft en el presente Milenio.

De acuerdo a lo anterior ¿Cuál puede ser la respuesta de Microsoft al fenómeno Linux? La respuesta a esta pregunta la encontramos en los documentos "Halloween" en los cuales el autor plantea los siguientes puntos como estrategias a seguir por parte de Microsoft donde todo surge de la siguiente pregunta "¿Cómo puede Microsoft capturar alguna parte de la exuberante capacidad intelectual para enfocarse sobre los productos software libre?"

A esta pregunta Microsoft plantea las siguientes estrategias dentro del documento Halloween:

- ✓ Capturar los beneficios de la depuración en paralelo mediante un licenciamiento más amplio. El ser más liberal al proporcionar licencias de códigos fuentes para Windows a organizaciones tales como las universidades y algunos socios.
- ✓ Proporcionar herramientas de entrada de bajo costo o gratuitas. El efecto secundario de las herramientas es generar una plataforma común que los desarrolladores puedan usar. La amplia disponibilidad de un conjunto consistente de herramientas de desarrollo en Linux es un medio crítico para coordinar de manera implícita el crecimiento del sistema.
- ✓ Publicar partes del código fuente. Tratar de generar el interés en los hackers de agregar valor a las bases del código fuente de MS. Un primer candidato podrían ser partes de TCP/IP. Sin embargo, el reto es encontrar alguna parte del código fuente de MS que sea capaz de generar una atmósfera con la masa crítica suficiente para generar interés.
- ✓ Proporcionar mayor extensibilidad. El desarrollador entusiasta de Linux es amante de comprender API y partes internas sin documentar. La documentación / publicación de algunos API internos como "sin soporte" puede ser un medio de generar innovaciones externas a los que podríamos sacar provecho. En particular, al asegurar una mayor cantidad de componentes para una mayor

cantidad de equipos para que sean automatizables para asegurar que los usuarios puedan jugar con nuestros componentes.

✓ Crear una comunidad comprometida. El MSDN llega a una población extremadamente grande. ¿Cómo podemos crear las estructuras sociales que proporcionen beneficios en la red para sacar provecho de esta gigantesca base de desarrolladores? Por ejemplo, ¿qué pasaría si tuviésemos un muestrario de VB en Microsoft.com que permitiese a los desarrolladores de VB publicar el código fuente de sus proyectos para compartirlos con otros desarrolladores de VB? Yo pienso que muchos desarrolladores de VB tendrían una enorme gratificación a su ego por el hecho de que pudiesen ser calificados como expertos dentro de la comunidad de desarrolladores de VB dentro de microsoft.com.

En los puntos anteriores es notorio que Microsoft plantea incluso la publicación de partes de su código fuente, sin embargo el esquema de licenciamiento será el que hará la diferencia entre Linux y Microsoft ya que para su caso como se vio al principio de este capítulo es el esquema de Fuente Abierta al estilo de BSD o Apache el que podría usar Microsoft, sin embargo a los desarrolladores de Software Libre estos esquemas no les han parecido lo suficientemente buenos, puesto que estos no garantizan una distribución libre y pueden ser objeto de piratería informática. Además si fijamos nuestra mente en el desarrollo actual de Linux y sus principales componentes en cuanto a líneas de código se refiere, podremos darnos cuenta de las cifras que se han alcanzado y que difícilmente se pueden lograr con un esquema de licenciamiento diferente a GPL.

Proyecto	Líneas de código
Kernel de Linux	500,000
Servidor Apache	80,000
Send Mail	57,000
Xfree86	90,000
Distribución completa de Linux	+/- 10 millones

A la pregunta de ¿Quién ganará esta lucha en el mercado? Desde mi punto de vista creo que es solamente cuestión de tiempo para que Linux comience a tomar ventaja, además de que durante 1999 y 2000 ocurrieron algunos sucesos importantes en el mercado informático como son:

✓ La terminación de los contratos de exclusividad entre Microsoft e Intel.

- ✓ La lucha entre Microsoft y Sun Microsystems por Internet (Donde Sun apuesta a Linux)
- ✓ El nacimiento de compañías especializadas en consultoría Linux.
- ✓ Las empresas Compaq y Dell comienzan a distribuir Servidores Linux instalados
- ✓ IBM hace una fuerte inversión de varios millones de dólares en el apoyo a Linux.
- ✓ La creciente credibilidad en el sistema operativo Linux.

Las cifras presentadas en el presente capítulo permiten visualizar el mercado de servidores Intel x86 como el preludio de la batalla mas encarnizada que se haya vivido jamás en el mercado informático, los contendientes: Windows y Linux.

Hasta hace algunos años nadie habría pensado que Microsoft tendría un contendiente tan duro como lo es ahora Linux para su producto estrella, sin embargo el mercado es caprichoso y existen muchas compañías hoy en día que necesitan ver a Microsoft en un estado de decadencia y que están apostando e invirtiendo varios millones de dólares por ello.

Conclusiones

El tema del Sistema Operativo dominante del siglo XXI en computadoras Intel x86 es en este momento una página en blanco de la historia de la informática, sin embargo, a lo largo del presente trabajo hemos expuesto de una forma clara y objetiva a los dos candidatos principales: Windows y Linux.

De acuerdo al estudio presentado, las características técnicas de ambos Sistemas Operativos han sido comparadas, así manifiesto mis conclusiones:

✓ **Primera.** Linux como máquina extendida proporciona un tratamiento homogéneo a todo el hardware, debido a que todos los dispositivos dentro de Linux son vistos como archivos en los que se puede leer y escribir por medio de llamadas al sistema bien definidas divididas modularmente como se mostró en el capítulo V; además de cumplir integralmente con todo el estándar POSIX.1. A diferencia de Windows el cual como máquina extendida proporciona API Win32 que muestra inconsistencias en su estructura debido al soporte que da a diferentes subsistemas de ambiente, además de no tratar a los dispositivos de hardware de una forma homogénea.

✓ **Segunda.** Como administrador de recursos, Linux proporciona un alto grado de eficiencia y confiabilidad debido a su manejo jerárquico de procesos, a la alta capacidad de administración de memoria y a la óptima administración del CPU, características que no presenta Windows, puesto que su microkernel solamente asigna tiempos de procesamiento.

✓ **Tercera.** Siendo los procesos la materia prima de todo Sistema Operativo, el control y la administración de los mismos es tan importante, que la rapidez que ofrece Windows no es tan importante como la administración de éstos procesos. La eficiencia, confiabilidad que proporciona Linux al tener un estricto control de procesos, le permite un buen rendimiento con una estabilidad ya legendaria.

✓ **Cuarta.** Refiriéndonos al punto anterior podemos decir que Windows por su arquitectura es más rápido debido a la estructura de anillos concéntricos del microkernel que actúan como clientes, sin embargo el esta estructura no le permite escalar como es debido, puesto que su control de tareas es deficiente y los cambios de modo usuario a modo kernel son constantes por cada llamada del sistema. Por su parte Linux conserva una relación coherente entre el hardware y la rapidez obtenida, proporcionando un excelente rendimiento.

• **Quinta.** El interprete de comandos representa una pieza fundamental en todo Sistema Operativo. Puesto que da al usuario acceso a todas las funciones del mismo. Windows es sumamente deficiente en este aspecto debido a que solamente se cubren las funciones mas elementales; y están divididas entre su línea de comandos y su ambiente gráfico. Por su parte, Linux se presenta como un Sistema UNIX en este aspecto, proporcionando varios interpretes de comandos que incluyen lenguajes de programación completos y acceso a todas las funciones del kernel de una forma estructurada, coherente y poderosa.

El éxito de un Sistema Operativo esta dado no solamente por las características técnicas, además debe de consolidarse comercialmente y presentar una base instalada considerable, es por eso que mi conclusiones al respecto, basadas en este trabajo se presentan como sigue:

• **Sexta.** Debido a su forma de desarrollo Linux presenta un serio problema con respecto al soporte al hardware, puesto que la mayoría de los fabricantes de equipo no han considerado el desarrollo de controladores para éste Sistema Operativo, por lo que se hace difícil el soporte a hardware y el avance en este aspecto. Por su parte Windows se presenta en este aspecto con una serie de alianzas estrategias hechas por Microsoft que le permiten soportar casi todos los dispositivos de hardware existentes.

• **Séptima.** Las posibilidades de la amplia expansión de Linux sobre el terreno de los servidores Intel x86 son cada vez mas claras debido a la crisis que Microsoft esta viviendo: ataques comerciales de muchas empresas importantes a áreas de negocio clave, como son: Navegadores de Web, Herramientas de Oficina, Servidores de correo electrónico, Clientes de correo electrónico, Bases de Datos, Herramientas de desarrollo, etc. Estos factores afectarán el lanzamiento de nuevas versiones Windows, donde los usuarios finales actuarán más desconfiados y muy probablemente se inclinen por Linux.

• **Octava.** Linux se ha vuelto un Sistema Operativo viable para grandes proyectos de enorme valor comercial debido a que los principales distribuidores de Linux como RedHat, Caldera y SlackWare se han vuelto empresas estables y que están aportando gran valor a la comunidad Linux, proporcionando una serie de servicios adicionales que hasta hace poco representaban una barrera para su desarrollo como son: Soporte Técnico, Servidores Pre-Instalados, Soporte en Línea, Consultoría y en general una serie de respaldos que hacen al usuario final sentirse confiado y con el respaldo de empresas serias que avalan su producto.

• **Novena.** Hoy en día el proceso de desarrollo del software libre ha demostrado ser confiable y capaz de alcanzar una calidad igual o superior a la del software comercial.

• **Décima.** La situación ventajosa de Windows en el mercado, se ha dado principalmente por su posición comercial y sus agresivas campañas de mercadeo; sin embargo, Linux cuenta con todo Internet para promoverse libremente con el apoyo, cada vez mas fuerte de grandes empresas de software, generando así las bases para asegurar que se esta dando una situación de igualdad comercial entre ambos Sistemas Operativos, la cual es la causa principal de la posición de Windows, como el Sistema Operativo mas vendido del mundo

• **Décimo primera.** Linux muy probablemente será el único Sistema Operativo capaz de acaparar el mercado de los servidores Intel x86 en un mediano plazo, pasando incluso sobre Windows, debido a sus inigualables características técnicas y el gran apoyo comercial que esta teniendo por parte de todo tipo de empresas.

Glosario

ACL. Abreviación de "Access Control List" (Lista de control de accesos), que permite asignar permisos a elementos como procesos, archivos, etc.

Archivo. Dispositivo lógico que permite almacenar datos o programas de manera persistente, es decir no debe verse afectado por el inicio o terminación de un proceso.

Automatización. Característica del sistema operativo que permite generar tareas que puedan ser ejecutadas de manera sistemática sin intervención humana.

Base Instalada. Porción del mercado acaparada por el sistema operativo a través de instalaciones en máquinas.

CPIO. Formato que permite manipular y administrar archivos que contienen a su vez otros (el formato es binario).

Compatibilidad. Capacidad de un Sistema Operativo para ejecutar programas que fueron desarrollados en otro diferente a él.

Confiabilidad. Característica del Sistema Operativo que le permite mostrarse robusto y tolerante a fallas.

Controlador. Pieza de software que permite el control de un dispositivo físico de la computadora.

Dispositivo. Pieza de hardware que es controlada por el Sistema Operativo a través de un controlador.

Escalabilidad. Relación existente entre el hardware y la capacidad de procesamiento ofrecida por el Sistema Operativo.

Extensibilidad. Capacidad del Sistema Operativo de soportar una amplia gama de dispositivos de hardware.

FSF. Siglas de la organización Free Software Foundation que es el principal promotor del proyecto GNU.

GNU. Es un proyecto que empezó en 1984 para crear un sistema operativo tipo UNIX. Actualmente a GNU se le ve como un acrónimo recursivo de "GNU's not Linux".

Hacker. Persona que se dedica a violar sistemas mediante diferentes técnicas para obtener información de forma ilegal.

Halloween. Conjunto de documentos publicados internamente por Microsoft que hablan acerca de las posiciones y estrategias de esta compañía con respecto al fenómeno del software de libre distribución.

Interprete de comandos. Interfaz de interacción con el usuario que permite recibir instrucciones a través de los dispositivos de entrada de la computadora.

Kernel. Núcleo del Sistema Operativo, donde se realizan las operaciones más importantes.

Linus Torvalds. Creador del kernel de Linux.

Linux. Sistema Operativo desarrollado bajo la Licencia Publica General (GNU) en Internet.

Llamada al sistema. Instrucción que envía un programa en modo usuario al sistema operativo y es ejecutada en modo núcleo.

Modo Kernel o Modo Núcleo. Modalidad del sistema operativo en el que el control del procesador lo tiene el sistema operativo.

Modo Usuario. Modalidad del sistema operativo en el que el control del procesador lo tiene la aplicación.

Multitarea. capacidad de ejecutar muchos programas al mismo tiempo sin detener la ejecución de cada aplicación.

Paginación de memoria. Técnica de administración de memoria que permite extender la memoria física de la maquina aprovechando medios de almacenamiento secundario.

Performance. Referente a la curva de aprovechamiento que el Sistema Operativo haga del hardware con respecto al número de procesos manejados.

Pipeline. Ver tubo.

Plataforma. Ver plataforma de ejecución.

Plataforma de ejecución. Referente a la plataforma de Hardware donde se ejecuta el sistema operativo.

Portabilidad. Termino empleado para referirse a la posibilidad de escribir aplicaciones que puedan ser usadas en distintos tipos de hardware.

POSIX. Conjunto de estándares desarrollados por "/usr/group" para generalizar y difundir la portabilidad de aplicaciones.

Procesamiento por lotes. Se refiere a la ejecución de un programa de forma secuencial y sin interrupciones.

Proceso. Programa en ejecución que consta de las siguientes partes: Programa ejecutable, datos, pila, contador y otros registros.

Robustez. Característica del Sistema Operativo que ofrece al usuario una amplia gama de servicios de una manera confiable.

Scripts de Shell. Archivos que contienen comandos agrupados del Sistema Operativo de una forma estructurada, para cumplir un fin específico.

Seguridad. Referente a la capacidad del Sistema Operativo de protegerse ante ataques de Hackers no deseados.

Shareware. Productos de software que son totalmente funcionales y no tienen restricciones en cuanto a su distribución, pero existe una licencia que obliga eventualmente a la compra de la licencia.

Shell. Ver interprete de comandos.

Sistema de Archivos. Parte del sistema operativo que le permite la gestión, almacenamiento y manipulación de la información.

Sistema Operativo. Capa intermedia de software entre los programas de aplicación y el hardware, que administra los recursos y dispositivos

de una forma óptima y que oculta la complejidad de la programación a bajo nivel del hardware.

Subsistemas de ambiente. Los subsistemas de ambiente de Windows son los que hacen que las aplicaciones se comuniquen con el kernel.

Tar. Formato que sirve para crear y manipular archivos que son colecciones de muchos otros respetando la estructura y organización original.

Telnet. Aplicación que permite tener una emulación de terminal de una máquina a través de la red.

Tolerante a fallas. El Sistema Operativo es capaz de recuperarse automáticamente de errores que ocurran durante la ejecución de un proceso.

Tubo. Seudo archivo que permite comunicar dos procesos.

Windows. Sistema Operativo desarrollado por Microsoft para servidores Intel x86.

/usr/group. Organización fundadora de los estándares POSIX

Bibliografía

1. TANENBAUM, Andrew S., "Sistemas Operativos Modernos", México: Prentice - Hall Hispanoamericana, S.A., 1993.
2. STEVENS, W. Richard, "Advanced Programming in the UNIX Enviroment", United States of America: Addison- Wesley Publishing Company, 1994.
3. VOLKERDING, Patrick, FOSTER- JOHNSON, Eric, REICHARD, Kevin, "Linux Programing", United States of America: MIS:Press A Subsidiary of Henry Holt and Company, Inc., 1997.
4. MATTHEW, Neil & STONES, Richard, "Beginning Linux Programming", Canada: Wrox Press Ltd., 1997.
5. VOLKERDING, Patrick, "Halloween", <http://www.gnu.org/halloween>
6. KELLY- BOOTLE, Stan, "Cómo Usar UNIX", México: Editorial Limusa, S.A de C.V. Grupo Noriega Editores, 1993.
7. TACKETT, Jack Jr., GUNTER, David, "Utilizando LINUX", Campus 400, Maylands Avenue: Prentice Hall, 1996.
8. RUBINI, Alessandro, "Linux Device Drivers", United States of America: O' Reilly & Associates, Inc., 1998.
9. BECERRA, José Luis, "Linux para Todos" **EL UNIVERSAL (Universo de la Computación)**", México, Lunes 5 de julio de 1999, p.3.
10. Francisco Javier, "Unix vs Windows NT" **SOLO PROGRAMADORES**, España, Año V Número 50, p.37.
11. Menacé, Daniel A. "Windows NT Architecture", USA, <http://www.cs.gmu.edu/~menasec/osbook/nt/sld001.html>
12. Anónimo, "Windows Time Line", <http://www.msn.com/os/articles/osd5423.html>
13. Anónimo, "Linux History"
<http://www.kernelnotes.org/old/archive/12/10/1998lh.html>

Apéndice A. La licencia Pública general GNU

Versión 2, junio 1991.

Derechos reservados © 1989, 1991 free Software Foundation, Inc., 675 Mass Ave., Cambridge, MA 02139, USA

Todas las personas están autorizadas a copiar y distribuir copias de ésta licencia palabra por palabra, pero no se permite su alteración.

Preámbulo

Las licencias de la mayoría del software están diseñadas para quitarle la libertad para compartirlo y cambiarlo. En contraste, la Licencia Pública General (GNU) tiene el propósito de garantizarle su libertad para compartir y modificar sin restricciones el software (para asegurarse de que el software sea libre para todos los usuarios). Esta Licencia Pública General se aplica a la mayoría del software de la Free Software Foundation y cualquier otro programa cuyos autores se comprometan a usarla. (Otro software de la Free Software Foundation está cubierto por la GNU Library General Public License en su lugar.) Usted también puede aplicarla a sus programas.

Nuestras licencias están diseñadas para asegurarle la libertad de distribuir copias de software libre de restricciones (y cobrar por este servicio si lo desea), que reciba el código fuente o lo obtenga si lo desea, que pueda modificar el software o utilizar partes de él en nuevos programas libres de restricciones; y que sepa que pueda hacer esto.

Para proteger sus derechos, debemos imponer condiciones que prohíban a cualquier persona negarle estos derechos o pedirle que ceda los suyos.

Esas condiciones se traducen a ciertas responsabilidades para usted si distribuye copias del software o si lo modifica.

Por ejemplo, si distribuye copias de un programa como éste, ya sea gratis o por una cuota, deberá ofrecerle a las personas que reciben el software todos los derechos que usted tiene. Debe asegurarse también de que ellos reciban o puedan obtener el código fuente. Además, debe mostrarles estos términos para que conozcan sus derechos.

Protegemos sus derechos mediante dos pasos: (1) al establecer los derechos reservados del software, y (2) al ofrecerle esta licencia, la cuales le da la autorización legal para copiar, distribuir y/o modificar el software.

Asimismo, para la protección de cada autor y la nuestra, queremos asegurarnos de que todos entiendan que no existe garantía para este software libre de restricciones. Si el software es modificado y distribuido por alguien más, queremos que los poseedores de este software sepan que éste no es el original, así que cualquier problema introducido por otros no afectará la reputación de los autores originales.

Por último, cualquier programa libre de restricciones está amenazado en forma constante por las patentes de software. Deseamos evitar el peligro de que redistribuidores de un programa libre de restricciones obtengan licencias de patente individuales, y hacer con esto un programa patentado. Para prevenir esta situación, hemos aclarado que cualquier patente debe ser autorizada para el uso libre de todos o no tener licencia.

Los términos precisos y condiciones para copiar, distribuir y modificar se describen a continuación.

Licencia Pública General GNU

Términos y condiciones para copiar, distribuir y modificar

Esta Licencia se aplica a cualquier programa u otro trabajo que contenga una nota colocada por el propietario de los derechos de autor, donde se menciones que el programa o trabajo podrá ser distribuido bajo los términos de esta Licencia Pública General. El "Programa", presentado a continuación, se refiere a cualquier programa o trabajo como éste, y un "trabajo basado en el Programa" significa el Programa o cualquier trabajo derivado bajo las leyes de derechos reservados; es decir, un trabajo que contenga el Programa o una parte de él, ya sea palabra por palabra o con modificaciones y/o traducido a otro lenguaje. Cada persona con licencia será nombrada como "usted".

Actividades diferentes a copiar, distribuir y modificar no se cubren en esta Licencia; se encuentran fuera de su objetivo. La acción de ejecutar el Programa no está restringida, y su salida se cubre sólo si su contenido constituye un trabajo basado en él (independiente de haber sido hecha al ejecutar el Programa).

Que esto sea cierto depende de lo que el programa lleve a cabo.

Puede copiar y distribuir copias exactas del código fuente del Programa como lo recibe, por cualquier medio, siempre y cuando publiquen en algún lugar visible y adecuado de cada copia una nota de derechos reservados y de ausencia de garantía; mantenga intactas todas las notas que se refieran a esta Licencia y a la ausencia de cualquier garantía; ofrezca a cualquier otro receptor del Programa una copia de esta Licencia junto con el Programa.

Puede cobrar una cuota por la acción física de transferir una copia, y ofrecer la opción de una protección de garantía a cambio de una cuota.

Puede modificar su copia o copias del Programa o alguna parte del mismo, y formar un trabajo basado en el Programa, y copiar y distribuir estas modificaciones o trabajo basado en el programa bajo los términos de la Sección I anterior, siempre y cuando también cumpla con estas condiciones:

- Debe hacer que los archivos modificados cuenten con notas visibles y especificar que usted cambió los archivos, así como la fecha de cualquier modificación.

- Debe hacer que cualquier trabajo que usted distribuya o publique, que como unidad o partes contenga o se derive del programa o de alguna parte de él, sea autorizado como unidad sin costo para otras personas bajo los términos de esta Licencia.

- Si el programa modificado lee por lo regular comandos en forma interactiva al ejecutarse, debe hacer que, cuando se emplee a ejecutar para este uso interactivo en la manera más ordinaria, imprima o despliegue un anuncio que incluya una nota adecuada de derechos de autor y otra de que no hay garantía (o de lo contrario, que diga que usted ofrece la garantía), y que los usuarios puedan redistribuir el programa bajo estas condiciones, e indicarle al usuario cómo ver una copia de esta Licencia. (Excepción: Si el Programa en sí es interactivo, pero no imprime en forma regular un anuncio como éste, su trabajo basado en el Programa no necesita la impresión de un anuncio.)

Estos requerimientos se aplican al trabajo modificado como unidad. Si secciones identificables de ese trabajo no se derivan del Programa y son considerados en forma razonable como trabajos independientes y separados, entonces esta Licencia y sus términos no se aplican a esas secciones cuando las distribuya como trabajos por separado. Pero cuando distribuye las mismas secciones como parte de una unidad, la cual es un trabajo basado en el programa, la distribución de la unidad debe efectuarse en los términos de esta Licencia, cuyos permisos para otras personas autorizadas se extienden a la unidad completa, y con esto a cada una de las partes sin importar quién lo escribió.

Además, no es el objetivo de esta sección reclamar derechos o disputar sus derechos de trabajos escritos por completo por usted; en lugar de ello, la intención es aplicar el derecho para controlar la distribución de trabajos derivados o colectivos basados en el Programa.

Además, la sola adición de otro trabajo no basado en el Programa al Programa (o trabajo basado en el Programa) en un volumen de un medio de almacenamiento o distribución no traerá al otro al campo de acción de esta Licencia.

Puede copiar y distribuir el Programa (o un trabajo basado en él, bajo la Sección 2) en código objeto o en forma ejecutable bajo los términos de las Secciones 1 y 2, siempre y cuando también realice una de las acciones siguientes:

Acompañarlo con el código fuente completo correspondiente legible para la máquina, que deberá distribuirse bajo los términos de las Secciones 1 y 2 en un medio usado en forma común para el intercambio de software; o,

Acompañarlo con un ofrecimiento escrito, válido por lo menos por tres años para ofrecerte a cualquier otra persona, por una cuota no mayor al costo de la distribución física de la fuente, una copia completa del código fuente correspondiente legible para la máquina, para distribuirse bajo los términos de las Secciones 1 y 2 en un medio usado en forma común para el intercambio de software; o,

Acompañarlo con la información que recibió como el ofrecimiento para distribuir el código fuente correspondiente. (esta opción es permitida sólo para distribución no comercial y sólo si recibió el programa en código objeto o forma ejecutable con un ofrecimiento como éste, de acuerdo con la Subsección b anterior.)

El código fuente para un trabajo significa la forma preferida del trabajo para hacerle modificaciones. Para un trabajo ejecutable, el código fuente completo significa todo el código fuente para todos los módulos que contiene, así como cualquier archivo de definición de interfaz asociada, además de los scripts usados para controlar la compilación e instalación del ejecutable. Sin embargo, como una excepción especial, el código fuente distribuido no precisa incluir nada que sea por lo regular distribuido (ya sea en forma binaria o fuente) con los principales componentes (compilador, núcleo, etc.) del Sistema Operativo sobre el cual opera el programa ejecutable, a menos que ese componente acompañe al ejecutable.

Si la distribución del ejecutable o código objeto se hace al ofrecer acceso para copiar desde un lugar designado, entonces ofrecer un acceso equivalente para copiar el código fuente, aunque otras personas no están obligadas a copiar el código fuente junto con el objeto.

No podrá copiar, modificar, subautorizar o distribuir el Programa, excepto lo expuesto bajo esta Licencia. Cualquier intento de copiar, modificar, subautorizar o distribuir el Programa está prohibido y terminará en forma automática con sus derechos bajo esta Licencia.

Sin embargo, las personas que hayan recibido copias o derechos de usted bajo esta Licencia, no tendrán sus licencias terminadas siempre que estas personas la acepten por completo.

No está obligado a aceptar esta Licencia, ya que no la ha firmado. Sin embargo, no hay nada más que garantice el permiso para modificar o distribuir el programa o sus trabajos derivados. Estas acciones están prohibidas por la ley si no acepta esta Licencia. En consecuencia, al modificar o distribuir el Programa (o cualquier trabajo basado en el Programa), expresa su aceptación de esta Licencia y todos sus términos y condiciones para copiar, distribuir o modificar el Programa o trabajos basados en él.

Cada vez que redistribuye el Programa (o cualquier trabajo basado en el programa), el poseedor recibe en forma automática una licencia del propietario original para copiar, distribuir o modificar el Programa sujeto a estos términos y condiciones. No tendrá que imponer más restricciones sobre el ejercicio de los derechos otorgados a los nuevos poseedores.

Usted no es responsable de hacer cumplir a las demás personas esta Licencia.

Si, como consecuencia de un juicio de corte o un alegato por la infracción de una patente o por alguna otra razón (no limitada a temas sobre patentes), se le imponen condiciones (ya sea por orden de la corte, acuerdo o alguna otra razón) que contradigan las condiciones de esta Licencia, aquellas no lo liberan de las condiciones en esta Licencia. Si no puede distribuir para satisfacer en forma simultánea sus obligaciones bajo esta Licencia y alguna otra obligación pertinente, entonces no podrá distribuir el Programa. Por ejemplo, si una Licencia de patente no permitiera una redistribución libre de derechos de autor del Programa, hecha por todos los que reciban copias en forma directa o indirecta de usted, entonces la única manera en que podría satisfacer tanto a esta Licencia como a la otra sería abstenerse por completo de distribuir el Programa.

Si alguna parte de esta sección se invalida o no es aplicable bajo alguna circunstancia en particular, el balance de la sección se intentará aplicar y la sección como un todo se aprovechará en otras circunstancias.

El propósito de esta sección no es inducirlo a infringir alguna patente u otros reclamos de derechos de propiedad, o disputar la validez de estos reclamos; esta sección tiene el único propósito de proteger la integridad del sistema de distribución de software gratuito, el cual está implantado por prácticas de licencia pública. Muchas personas han hecho generosas contribuciones al amplio rango de software distribuido mediante este sistema con la confianza en la aplicación consistente de este sistema; es decisión del autor/donador determinar si está dispuesto a distribuir software por medio de algún otro sistema; una licencia no puede imponer esta elección.

Esta sección tiene la intención de aclarar lo que se cree es una consecuencia del resto de esta Licencia.

Si la distribución y(o) uso del Programa está restringido en algunos países, ya sea por patentes o por interfaces con derechos reservados, el propietario original de los derechos que coloca el Programa bajo esta Licencia puede agregar una limitación explícita de distribución geográfica al excluir a esos países, así que esta distribución está permitida sólo en o entre países no excluidos. En este caso, esta Licencia incorpora la limitación como si estuviera escrita en el cuerpo de la misma.

La Free Software Foundation publica versiones revisadas y(o) nuevas de la Licencia Pública General de vez en cuando. Estas nuevas versiones serán similares en esencia a la presente versión, pero pueden diferir en detalles para acentuar nuevos problemas o preocupaciones.

A cada versión se le da un número de versión distinto. Si el Programa especifica un número de versión de esta Licencia, el cual se aplica a ella y "a cualquier otra versión posterior", tiene la opción de seguir los términos y condiciones de esa versión o cualquier otra publicada con posterioridad por la Free Software Foundation. Si el programa no especifica un número de versión de esta Licencia, puede escoger cualquier versión publicada por la Free Software Foundation.

Si desea incorporar partes del Programa a otros programas libres de restricciones, cuyas condiciones de distribución son distintas, escríbale al autor para pedirle permiso. Para el software cuyos derechos de autor pertenecen a la Free Software Foundation, escriba a dicho organismo; algunas veces hacemos excepciones para esto. Nuestra decisión será dirigida por los dos objetivos de preservar el estado de libertad de todos los derivados de nuestro software y el de promover la compartición y reutilización del software en general.

NO GARANTÍA

COMO EL PROGRAMA ES AUTORIZADO SIN NINGÚN COBRO, NO EXISTE GARANTÍA PARA EL MISMO, HASTA DONDE LO PERMITE LA LEY APLICABLE. A EXCEPCIÓN DE CUANDO SE MENCIONE POR ESCRITO A LOS PROPIETARIOS DE LOS DERECHOS DE AUTOR Y(O) OTRAS PERSONAS OFREZCAN EL PROGRAMA "COMO ÉSTA" SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPRESA O IMPLÍCITA INCLUSO, PERO NO LIMITADA A LAS GARANTÍAS DE COMERCIO IMPLICADAS O DE PROPIEDAD PARA UN PROPÓSITO EN PARTICULAR. EL RIESGO COMPLETO SOBRE LA CALIDAD Y EL RENDIMIENTO DEL PROGRAMA ES DE USTED. SI EL PROGRAMA ESTA DEFECTUOSO, USTED ASUME EL COSTO DE TODO EL SERVICIO, REPARACIÓN O CORRECCIÓN NECESARIOS.

EN NINGÚN EVENTO, A MENOS QUE SEA REQUERIDO POR LA LEY APLICABLE O ACUERDO, CUALQUIER PROPIETARIO DE DERECHOS, O ALGUNA OTRA PERSONA QUE PUEDA MODIFICAR Y (O) REDISTRIBUIR EL PROGRAMA COMO ES PERMITIDO, SERÁ RESPONSABLE POR DAÑOS, INCLUIDO DAÑO GENERAL, ESPECIAL, INCIDENTAL O CONSECUENTE QUE SURJA DEL USO O INCAPACIDAD PARA USAR EL PROGRAMA (INCLUIDO, PERO NO LIMITADO A LA PÉRDIDA DE INFORMACIÓN O

INFORMACIÓN IMPRECISA O PÉRDIDAS SUFRIDAS POR USTED U OTRAS PERSONAS O UNA FALLA DEL PROGRAMA PARA OPERAR CON OTROS PROGRAMAS), INCLUSO SI ESTE PROPIETARIO U OTRAS PERSONAS HAN SIDO ADVERTIDOS DE LA POSIBILIDAD DE ESTOS DAÑOS.

FIN DE TÉRMINOS Y CONDICIONES

Agradecimientos

A Alfredo Murillo, por haberme mostrado siempre con el ejemplo el camino a seguir a través de lecciones de vida que solamente un padre puede dar.

A Rebeca Domínguez, por enseñarme el valor de las cosas y darme el cariño y la formación que solamente una madre puede dar.

A Yadira Ruiz por su paciencia, apoyo y ayuda para la realización de este trabajo como colega, esposa y madre de mi hija.

A Sofía Murillo por motivarme en todo momento a terminar este trabajo.

A Sergio Paniagua por haberme acercado a este mundo de la informática de la mejor manera posible.