



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

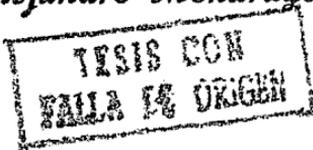
FACULTAD DE CIENCIAS

SISTEMA CONTROLADOR DE RECURSOS EN
DISCO PARA SISTEMAS UNISYS SERIE "A"

T E S I S

QUE PARA OBTENER EL TÍTULO DE
M A T E M Á T I C O
P R E S E N T A

Alejandro Mondragón Miranda



MEXICO, D. F.

1989



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 está 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.

INDICE

INTRODUCCION	1
CAPITULO 1. Marco Teórico General	1
1.1 Sistema de cómputo	1
1.2 Sistema operativo	7
1.3 Políticas de asignación	16
1.3.1 "Deadlock"	17
1.3.2 "Scheduler"	22
1.3.3 Políticas de control	25
1.4 Sistemas de software	25
1.5 Ventajas y limitaciones de las políticas provistas por el sistema	40
1.6 El desarrollo de software en forma correcta y eficiente	41
CAPITULO 2. Presentación del Problema	46
2.1 Problema a resolver	46
2.2 Descripción del equipo	48
2.2.1 Sistema operativo UNIXV9	50
2.2.2 La Inicialización	62
2.3 Herramientas a usar	63

CAPITULO 3. Solución Propuesta	65
3.1 Objetivo que cumple	65
3.2 Problema que resuelve	66
3.3 Descripción general del sistema	67
3.4 Descripción detallada del sistema	70
CAPITULO 4. Conclusiones	83
4.1 Mejoras a futuro	85
APENDICE A. Cambios necesarios a CANDE	86
APENDICE B. El almacenamiento de información en discos magnéticos	89
APENDICE C. El sistema controlador de recursos en disco para sistemas UNISYS serie "A" como un modulo del sistema SAYCUS	98
BIBLIOGRAFIA	103

INTRODUCCION

La Dirección General de Servicios de Cómputo para la Administración (DGSCA) tiene como objetivo el proveer los servicios de cómputo a las diferentes dependencias que pertenecen a la Secretaría Administrativa de la Universidad Nacional. Para lograr este objetivo existen dos subdirecciones: una de servicios, cuya función principal es la administración del equipo central, así como la administración de todos los programas de apoyo a dicho equipo; la otra subdirección es la de desarrollo que es la que se encarga de diseñar e implementar los distintos sistemas de soporte para la administración. Debido a que el ámbito de la computación así como el de las necesidades de apoyo resultan ser dinámicas, un problema que se enfrenta continuamente es la modificación, el acondicionamiento y muchas veces el rediseño de los sistemas mencionados.

Un ejemplo de ello es la necesidad de diseñar e implantar un sistema para resolver el control de recursos en disco, en este caso para sistemas UNISYS serie "A" que es el objetivo de esta tesis.

Descripción de los capítulos.

En el capítulo uno se describen las ideas teóricas que sustentan los conceptos de sistema de cómputo, sistemas operativos, políticas de asignación de recursos y sistemas de software. También se menciona una técnica para el desarrollo de software conjuntamente con la consideración de prueba de software para garantizar que sea correcto.

En el capítulo dos se describen el problema, el equipo para el cual se hace el desarrollo del sistema y las herramientas que se usaron.

En el capítulo tres se describe la solución propuesta, mediante la exposición del objetivo y problema que resuelve, conjuntamente con la descripción general y detallada del sistema.

En el capítulo cuatro se presentan las conclusiones finales y las mejoras a futuro.

En el apéndice "A" se describen los cambios que se hicieron a CANDE, para el logro de los objetivos. En el apéndice "B" se describe la forma en que se organiza la información dentro de los discos magnéticos en los equipos UNISYS serie "A". En el apéndice "C" se presenta al sistema desarrollado como parte de un sistema mayor.

CAPITULO 1

MARCO TEORICO GENERAL

1.1 SISTEMA DE COMPUTO

La mejor manera de definir a un sistema es a partir de la teoría general de sistemas que provee un enfoque sistemático, para el estudio y análisis de los mismos.

Esta teoría se define como:

"Una teoría lógica-matemática que se propone formular y derivar aquellos principios generales aplicables a todo tipo de sistemas" [8].

El modo sintético de pensamiento dice que algo para ser explicado tiene que ser visto como parte de un todo mayor, y dicha explicación es en términos de la función que tiene en el sistema mayor.

Un sistema en términos generales se puede definir como:

"Conjunto de dos o más elementos interrelacionados de cualquier especie (objetos, conceptos, personas, ...), que entre si tienen un propósito común" [12].

Las propiedades de los elementos que lo forman son:

- 1).- Las propiedades o el comportamiento de cada elemento del conjunto tiene un efecto en las propiedades o el comportamiento del conjunto tomado como un todo.
- 2).- Las propiedades y comportamiento de cada elemento, y la forma en que afecta al todo, dependen de las propiedades y comportamiento al menos de otro elemento en el conjunto.
- 3).- Cada subconjunto posible de elementos del conjunto tiene las dos primeras propiedades. En resumen cada uno tiene un efecto no independiente en el total. No se puede descomponer el total en subconjuntos independientes.

Los sistemas se engloban en dos grandes tipos que son:

- Sistemas Naturales.
- Sistemas Humanos.

Los Sistemas Naturales son aquellos creados por la naturaleza, como lo son el sistema planetario, las lluvias, el sistema nervioso, ... etc., es decir, no tienen la intervención del hombre para su creación, éste sólo describe y explica los fenómenos a través de las disciplinas científicas, puede pronosticar su comportamiento y en ocasiones incluso puede alterarlos para satisfacer sus necesidades, pero en general sólo permanece como observador ajeno al sistema.

Los Sistemas Humanos son aquellos creados por el hombre como lo son: el sistema de transporte, el sistema educativo, los sistemas de cómputo, ... etc., aquí el hombre no sólo describe y explica los fenómenos, altera y predice su comportamiento, sino que también los crea.

Para esta creación de sistemas el hombre cuenta con técnicas o teorías afines tales como:

- . Ingeniería de Sistemas.
- . Análisis de Sistemas.
- . La Cibernética.
- . Teoría de la Información.
- . Investigación de Operaciones.
- . Metodología de Sistemas "Suaves".

Mediante la Ingeniería de Sistemas se diseñan sistemas en el mundo real. Una definición formal sería: "La ciencia que diseña sistemas complejos en su totalidad, su tarea es la de concebir, diseñar, evaluar, e implementar un sistema que satisfaga metas específicas, y que operen de la manera más eficiente" [8].

El Análisis de Sistemas es una técnica que consiste en la evaluación económica de todos los costos y consecuentemente de varias alternativas para llegar a un fin, definiendo estudios de análisis de costo-beneficio comúnmente utilizados. Se hace gran énfasis en sistemas de información y el desarrollo de sistemas computarizados como resultado de estudios de análisis de sistemas.'

"La Cibernética entendida como una teoría de control de sistemas trata de demostrar que los mecanismos de retroalimentación son la base del comportamiento teleológico o con propósito tanto en máquinas como en organismos vivos y sistemas sociales. El concepto básico de la cibernética es la retroalimentación, la cual es importante para comprender cómo mantiene el equilibrio dinámico un sistema" [8].

"La metodología de sistemas suaves hace la diferenciación entre los sistemas "duros" y "suaves". Los sistemas "suaves" son aquellos que le dan mayor importancia a la parte social, pues un individuo o grupo social se toma como un sistema teleológico, con fines, con voluntad, un sistema pleno de propósitos, capaz de desplegar comportamientos, actitudes y aptitudes múltiples" [8].

"Los sistemas duros se identifican como aquellos en que interactúan hombres y máquinas y en los que se les da mayor importancia a la parte tecnológica en contraste con la parte social" [9].

La Investigación de Operaciones es la aplicación de los métodos científicos a problemas complejos que surgen en la conducción y administración de grandes sistemas de hombres, máquinas, materiales y dinero en la industria, negocios, gobierno y defensa. Se distingue por la incorporación de factores de riesgo e incertidumbre.

La Teoría de la información se desarrolló en los años 40, principalmente WIENER, FISHER y SHANNON, reconoció éste último como su creador. Surgió por problemas de radio comunicación.

En la década de los 40's surgen las computadoras digitales y se crean los sistemas de cómputo, que para muchos científicos son más que una simple herramienta para calcular. Y consideran a los sistemas de cómputo como una extensión de la inteligencia humana, ya que desaparecen errores humanos como lo serían la lentitud y la ambigüedad.

Un sistema de cómputo puede definirse como:

"El conjunto de elementos interrelacionados que a través de interfaces bien definidas proveen control exacto de señales (datos, signos, efectos u otras interacciones) que circulan a través de ellos".

Hay dos términos que engloban a los elementos de un sistema de cómputo que son el concepto hardware y el concepto software; es decir el conjunto de todos los elementos físicos (hardware) y el conjunto de programas y sus métodos de aplicación o soporte lógico (software).

Un sistema de cómputo puede realizar una amplia variedad de trabajos, desde cálculos aritméticos rutinarios hasta la resolución de problemas muy complicados de administración, de ciencia y tecnología, y de ciencias sociales, entre otras. Las funciones que puede efectuar una computadora y la velocidad para desempeñarlas dependen del tipo de máquina o sistema de cómputo.

El elemento Hardware del sistema lo constituye:

- El conjunto de dispositivos de entrada/salida.
- El conjunto de dispositivos para almacenamiento.
- El conjunto de dispositivos de proceso.

De los elementos del sistema de entrada/salida se pueden mencionar: el subsistema manejador de disco "duro", el subsistema manejador de disco flexible, las impresoras (como subsistema de impresión), el teclado (como subsistema de entrada), el subsistema manejador de cintas magnéticas,... etc.

Entre los componentes del sistema de memoria, se encuentra lo que se conoce como memoria principal, memoria secundaria (discos y cintas magnéticas).

Entre los componentes de sistema de proceso, se puede mencionar lo que se conoce como "CPU" (la unidad de procesador central). Y con respecto al sistema controlador de procesos se puede mencionar al Sistema Operativo.

Actualmente se cuenta con sistemas de cómputo que:

- . Atienden a un solo usuario.
- . Atienden a varios usuarios al mismo tiempo (multiusuarios).

Un sistema de cómputo puede ser visto como un conjunto limitado de recursos, entre estos podemos considerar la memoria principal, memoria secundaria, dispositivos de entrada y salida, procesador, etc. Así también el sistema de cómputo puede verse como el medio ambiente en donde se ejecutan los procesos que pueden ser programas de usuario o bien programas del sistema.

En los sistemas de cómputo multiusuarios para que un proceso pueda ser ejecutado se requiere que se le hayan asignado todos los recursos que necesita, sin embargo como un sistema de cómputo tiene un conjunto limitado de recursos es necesario diseñar políticas para compartir de manera óptima los recursos del sistema entre todos los procesos.

1.2 SISTEMA OPERATIVO

Un Sistema Operativo (S. O.) es el software que controla el conjunto de operaciones efectuadas por un sistema de cómputo.

En los sistemas de cómputo multiusuarios "un S. O. es un conjunto de procedimientos manuales y automáticos que permiten compartir un computador eficientemente" [2].

Compartir: La manera por la cual las personas competirán para el uso de los recursos físicos tales como tiempo de procesador, espacio de almacenamiento, uso de periféricos; pero también la manera en que la gente pueda cooperar mediante el intercambio de programas o datos en la misma instalación. El compartir de una instalación de cómputo es una necesidad económica, y el propósito de un Sistema Operativo es hacer lo más tolerable esta acción de compartir.

Un Sistema Operativo deberá contar con "políticas" para el orden de selección para la atención de los usuarios y para la resolución de conflictos de requerimientos simultáneos de un mismo recurso, también deberá tener la manera de imponer esta "política" en presencia de programas de usuario incorrectos o maliciosos.

En muchas computadoras actuales se pueden ejecutar varios programas de distintos usuarios simultáneamente, permiten a los usuarios retener o almacenar datos o programas incluso por meses. La presencia de datos y programas simultáneos pertenecientes a diferentes usuarios requieren que el S. O. los proteja de los otros usuarios y así a cada uno de los demás usuarios.

Un Sistema Operativo deberá también de realizar la contabilidad del uso de recursos de todos y cada uno de los usuarios.

Muchos componentes de las actuales computadoras pueden ejecutar operaciones o transferencia de datos de uno en uno. Pero es posible tener actividades simultáneas en varios de éstos componentes.

Un Sistema Operativo define varios lenguajes en los cuales las reglas de compartir recursos y requerimientos de servicio pueden ser descritos. Uno de éstos lenguajes es el "lenguaje para control de trabajos", el cual permite a los usuarios identificar y describir por ellos mismos los requerimientos del trabajo: los tipos y la cantidad de recursos que necesitan, los nombres de los programas y archivos que usanlo.

Otro lenguaje es el "Lenguaje de Máquina Virtual": El conjunto de operaciones de máquina disponibles para un usuario durante la ejecución del programa. Para mantener el control de una computadora y aislar usuarios unos de otros, un Sistema Operativo deberá impedir a programas del usuario la ejecución de ciertas operaciones; en otro caso estos programas podrían destruir procedimientos del mismo S.O. o realizar Entradas/Salidas en periféricos asignados a otro usuario. Así que el conjunto de operaciones disponibles al usuario es normalmente un subconjunto del lenguaje de máquina virtual original.

Los usuarios deberán tener la manera de realizar las acciones de Entrada/Salida. Los Sistemas Operativos permiten al usuario hacer uso de ciertos procedimientos que le permitan manipular los periféricos de una manera bien definida. Estos procedimientos aparecen como una extensión del lenguaje de máquina. El usuario tiene la ilusión de trabajar en una máquina que puede ejecutar programas escritos en este lenguaje. Debido a que esta máquina es parcialmente simulada por programa, se le llama "máquina virtual". Así un Sistema Operativo hace la máquina virtual disponible a cada usuario y previene a estas máquinas de la interferencia destructiva de cualquier otra. La presencia simultánea de varios usuarios hacen a la máquina virtual más lenta que la máquina física.

Un Sistema Operativo puede hacer que el lenguaje de programación de la máquina virtual sea más atractivo que la máquina original. Esto puede ser dado por la disminución de detalles tecnológicos al usuario, tales como la identidad física de periféricos y aligerar las diferencias en sus operaciones. Esto hace que el usuario se concentre en conceptos lógicos tales como nombres de archivos y la transferencia de registros de datos de un origen a un destino. La máquina virtual puede también ser más atractiva ya que permite utilizar una "técnica de corrección de error"; esto hace a la máquina virtual ser de más confianza que una real.

Los recursos de periféricos autónomos pueden hacer operaciones de Entrada/Salida independiente de la acción del procesador central. Una forma de resolver el problema de la sincronización entre el procesador central y las operaciones de entrada/salida en los periféricos es usando el concepto de interrupción. Cuando una interrupción ocurre el procesador central interrumpe la ejecución del programa actual e inicia otro programa del Sistema Operativo, el cual resuelve la interrupción; ya sea que reinicie el programa interrumpido por la acción de Entrada/Salida y por lo cual se generó la interrupción al concluir su Entrada/Salida, o bien empezar otro programa más urgente.

Una interrupción es un conjunto de señales de coordinación de un periférico a un registro conectado al procesador central, éste registro es revisado por el procesador central después de que cada instrucción es terminada. Esta técnica hace la "operación concurrente" de un operador central y sus posibles recursos periféricos.

De igual modo se utiliza la misma técnica para simular "ejecución concurrente" para varios programas de usuario con un solo procesador. A cada programa le es permitido un periodo para ser procesado, y al final de este intervalo se interrumpe esta ejecución y reinicia el Sistema Operativo. Este selecciona en turno a otro programa, el cual corre hasta que se le termine su periodo de tiempo, haciendo que el S. O. seleccione a un tercer programa de usuario, y así sucesivamente. En esta forma un recurso (el procesador central) es compartido por varios usuarios.

La posibilidad de que varios programas estén en estado de ejecución a un mismo tiempo tiene considerable influencia en la organización de almacenamiento. Esto es, no es posible predecir que parte interna del programa será puesta en ejecución, si no hay correspondencia fija, al tiempo de compilación, entre los nombres usados en el programa para referir a los datos y las direcciones de sus localidades de almacenamiento durante la ejecución. Este problema de relocalización fue primeramente resuelto por medio de un cargador de programas, el cual examina los programas antes de ejecutarlos, modificando las direcciones usadas en éstos para hacerlos corresponder a las localidades de almacenamiento usado.

Posteriormente, la localización del programa fue incluida en el procesador central lógico; un registro base fue usado para modificar instrucciones de dirección automáticamente durante la ejecución por la dirección inicial del área de almacenamiento asignada al usuario.

Parte del problema de protección fue resuelto por la extensión de este concepto con la definición del registro límite, que definía la longitud del espacio de dirección disponible al usuario, con lo que se logra que cualquier intento de referir a datos o programas fuera de éste espacio sea atrapada por el procesador central y que provoca que el S. O. sea activado.

La protección ofrecida por los registros base y límite fue por supuesto, ilusoriamente tan grande como los programas de usuario que pueden modificar estos registros. El reconocimiento de esta bandera llevó a designar al procesador central con dos estados de ejecución: un estado privilegiado, en el cual no hay restricciones en la ejecución de operaciones; y un estado de usuario en el cual la ejecución de operaciones, control de interrupciones, Entrada/Salida y localización de almacenamiento sean o están prohibidas y atrapadas por el S. O. si se intentan.

Una transición al estado privilegiado es causado por interrupciones de periféricos y por violación de protección en programas de usuarios, una transición al estado usuario es causado por la ejecución de una operación privilegiada.

Un Sistema Operativo comparte un conjunto de recursos entre un número de procesos concurrentes, los recursos son asignados cuando un proceso hace un requerimiento para usar alguno de éstos recursos. Así que cuando un recurso ha sido asignado a un proceso, éste es ocupado hasta que el proceso lo libere de nuevo. Cuando procesos concurrentes comparten recursos de esta manera, hay peligro de que puedan terminar en un "deadlock", un estado en el cual dos o más procesos están esperando indefinidamente por un evento que jamás suceda.

Por ejemplo:

Supongamos que se tienen cinco unidades de cierto recurso y estamos en un estado en que dos unidades están asignadas a un proceso "P" y una unidad al proceso "Q", pero ambos procesos requieren dos unidades para concluir su corrida, "Q" podría utilizar las dos unidades y concluir su corrida y liberar las tres unidades para que el proceso "P" satisfaga sus necesidades. Pero es también posible que "P" y "Q" ambas adquieran una unidad y entonces deciden esperar hasta que alguna unidad esté disponible, cosa que jamás ocurrirá. En éste momento los procesos "P" y "Q" se encuentran en "deadlock", pues ninguno puede continuar e individualmente esperan a que el otro resuelva el conflicto.

El "deadlock" puede ser prevenido asignando todas las unidades necesarias para "P" o "Q" al mismo tiempo en vez de uno por uno. Esta filosofía forzará a que "P" y "Q" corran en tiempos diferentes, y esto a menudo es el modo más eficiente del uso de recursos.

Varios procesos son concurrentes si sus ejecuciones se traslapan en el tiempo. Para compartir recursos, los procesos concurrentes deberán ser sincronizados. La sincronización es un término general para imponer el orden de operación en el tiempo.

"Una operación A debe ser ejecutada después de la operación B" Y "una operación de prioridad P deberá ser ejecutada solo cuando todos las operaciones de prioridad mayor han sido ejecutadas" (estas acciones llevan implícitamente el concepto de exclusión mutua).

Un ejemplo de recurso que sólo puede ser usado por un proceso a la vez es una cinta magnética en una unidad de cintas magnéticas. En este caso el requerimiento de exclusión mutua es dictado por la característica física del recurso.

La comunicación entre procesos con tareas comunes es importante también. Cada proceso depende directamente del dato producido por otro miembro de la comunidad. Para la cooperación en tareas comunes los procesos deberán ser capaces de intercambiar datos. En algunos casos un proceso sólo está interesado en recibir una señal de otro proceso cuando otro evento ha ocurrido, tal sincronización es llamada "semáforo".

Un semáforo es una variable común a varios procesos para realizar la acción "recibir" y "enviar" en tal variable. Así un semáforo V será caracterizado por dos componentes enteras:

$s(V)$ el número de señales enviadas.

$r(V)$ el número de señales recibidas.

que en un principio el valor de ambas son cero. Y un tercer componente es:

$c(V)$ el número de señales iniciales.

Las reglas de sincronización de los semáforos son:

- 1).- Si la operación recibir(V) es iniciada cuando $r(V) < s(V) + c(V)$, entonces $r(V)$ es incrementada en una unidad y continua recibiendo; pero si $r(V) = s(V) + c(V)$, entonces el recibir es retrasado en un proceso de cola asociado con el semáforo V .
- 2).- La operación enviar(V) incrementa $s(V)$ en una unidad, si uno o más procesos están esperando en la cola asociada al semáforo V , entonces uno de ellos es seleccionado y habilitado para continuar y $r(V)$ es incrementado en uno.

La disponibilidad de recursos de uso frecuente pueden ser controlados por la asociación de sincronización con cada proceso y el mantenimiento explícito de colas de petición.

En forma simple, una cola de petición es una secuencia de identificadores de procesos que requieren un mismo recurso, y al asignar a un proceso el recurso se le borra de la lista. No siempre se realiza la asignación en el orden en que está en la lista, pues entran algunos factores como por ejemplo la prioridad entre los procesos.

En las técnicas usadas para compartir almacenaje interno de capacidad limitada en procesos concurrentes se debe hacer la distinción entre dos niveles principales:

a) El primer nivel realiza políticas de servicio hacia el usuario mediante la implementación de una cola de programas y datos en un almacenamiento secundario para decidir cuando es iniciada una tarea, asegurando la prioridad y la integridad de cálculos realizados durante la ejecución del programa anterior que aun no termina, y con esto poder satisfacer los requerimientos de un proceso más urgente.

b) El segundo nivel además de que se asigna el procesador se transfiere datos entre memoria primaria y memoria secundaria. Se asegura la utilización del almacenamiento interno en forma eficiente por medio de:

- 1.- Limitar la frecuencia de transferencia de datos y programas.
- 2.- Mientras otros procesos esperan por entrada/salida, el procesador atiende a los demás procesos que se encuentran en el almacenamiento interno.
- 3.- La transferencia de datos y programas al almacenamiento interno sólo cuando son requeridos.

Para hacer posible el compartir la memoria interna se hace el siguiente compromiso:

Los programas y datos serán divididos en segmentos, cada segmento consistirá de datos relativos y una dirección relativa a un origen común, los segmentos de programas y datos se dividirán en el número de páginas necesarias.

El almacenamiento interno es dividido internamente en unidades de almacenamiento de la misma longitud llamados "marcos de página", teniendo como los segmentos de programas y datos se dividirán en el número de páginas necesarias.

Durante la ejecución del programa una página puede ser almacenada en cualquier "marco de página" disponible en el almacenamiento interno.

Se manejan tablas de páginas, tablas de segmentos o lista de bloques de memoria disponible en forma paralela para chequear el estado de la memoria.

Cuando un proceso espera por la transferencia de una nueva página o de un segmento del almacenamiento secundario, es puesto en la cola de procesos que no pueden correr con una característica de espera por tener una petición de entrada/salida, los cuales son administrados por el manejador de almacenamiento interno y servidos por un manejador de almacenamiento secundario.

Cuando un nuevo proceso entra al sistema surgen dos cosas: el orden en el cual es ejecutado y la asignación de recursos. Estas dos cosas son manejadas por el despachador de alto nivel llamado "scheduler".

El despachador "scheduler" decide cual es la siguiente tarea a ejecutar dependiendo de los recursos de cada tarea. Debe iniciar un nuevo proceso cada vez que la capacidad de los recursos lo permitan.

En un sistema multi-acceso los procesos son creados cuando el usuario entra al sistema. Cada nuevo usuario incrementa la demanda de recursos, un acceso puede ser rechazado cuando el número de usuarios en el sistema sea tal que necesite aumentar el tiempo de respuesta del sistema para que sea aceptado.

El orden en el cual son ejecutados los programas se determinan por el orden en la cola del procesador o bien por el orden de selección del despachador, es decir, por la prioridad que le asigne a cada proceso.

Debido a que el comportamiento del sistema se determina por la actividad del "scheduler", éste debe tener la más alta prioridad como proceso, y debe ser siempre un proceso cuyo estado sea siempre el de ejecutable, de esta manera se asegura que el sistema reaccione rápido de acuerdo a los cambios en la demanda.

1.3 POLITICAS DE ASIGNACION Y USO DE RECURSOS

Un sistema de cómputo tiene un conjunto limitado de recursos, por lo que es necesario diseñar políticas para compartir los recursos entre varios procesos al mismo tiempo.

Cuando se presentan demandas concurrentes de todos los procesos en el sistema, las técnicas deberán de considerar el número limitado de recursos con respecto al gran número de procesos.

Los objetivos de estas técnicas son:

- 1.- Exclusión mutua de procesos para recursos no compartibles.
- 2.- Prevenir el estado "deadlock".
- 3.- Asegurar un alto nivel de utilización de los recursos.
- 4.- Permitir a todos los procesos una oportunidad de adquirir los recursos que necesite en un tiempo "razonable".

Los mecanismos muestran aspectos de cómo es hecha la asignación. Las políticas gobiernan la forma en la que son aplicados los mecanismos, y pretenden asegurar la asignación de un recurso siempre que éste se encuentre disponible.

1.3.1 "DEADLOCK"

La primera política de asignación a considerar es la que contempla el problema del "deadlock".

Con respecto al "deadlock" se pueden seguir una de las siguientes estrategias:

1^a Prevenirlo asegurando todo el tiempo que al menos una de las siguientes condiciones no se den:

- A). Los recursos involucrados no sean compartibles.
- B). Los procesos tengan asignados recursos mientras esperan por otros recursos.
- C). Los recursos no puedan ser asignados mientras estén siendo usados.
- D). Exista una lista circular de procesos tales que cada proceso tenga un recurso el cual está siendo requerido por el siguiente proceso en la lista.

La existencia de estas condiciones definen un estado de "deadlock".

2^a Detectar el "deadlock" en el momento que ocurra y entonces tratar de recuperarlo.

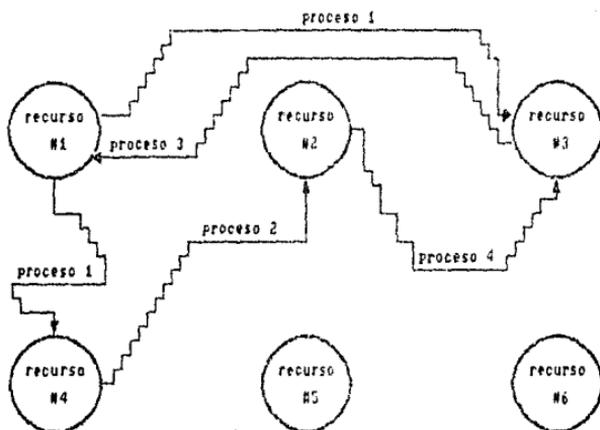
3^a Evitar el "deadlock".

PREVENIR EL "DEADLOCK".

Evitar la primera condición es muy difícil ya que algunos recursos son por naturaleza no compartibles, la segunda condición puede ser evitada estableciendo cuales peticiones de recursos proceden y cuales no, hasta que todos los recursos pedidos puedan ser asignados. La tercera condición es fácil de evitar imponiendo la regla de que si a un proceso se le rechaza una petición de un recurso entonces debe liberar los recursos que tenga en ese momento y volver a activar este proceso en un tiempo razonable (dándole tiempo al otro proceso para que libere el recurso). La cuarta condición puede ser evitada imponiendo un orden sobre los tipos de recursos, si a un proceso se le han asignado recursos del tipo A, entonces éste sólo puede hacer peticiones de recursos de tipo que sigan al A en orden, lo cual conlleva a poner los recursos más frecuentemente usados en orden menor.

DETECCION Y RECUPERACION DEL "DEADLOCK".

El estado del sistema en cualquier momento puede ser representado por una gráfica dirigida, en donde cada nodo representa un recurso y el arco entre dos nodos implica que existe un proceso que tiene asignado el recurso del nodo origen y está haciendo petición del recurso del nodo destino. Cuando ocurre una espera circular esto se ve representado en la gráfica como un círculo. El algoritmo trabaja sobre la gráfica detectando los ciclos, es decir detectando los "deadlock's", la inspección generalmente es hecha en intervalos fijos de tiempo, o bien puede ser hecha después de cada asignación de recursos, sin embargo como hay "simultaneidad" de asignación esta detección sería a cada momento (Figura 1).



El proceso 1 tiene asignado el recurso # 1, y hace la peticion de los recursos # 3 y # 4.

El proceso 2 tiene asignado el recurso # 4, y hace la peticion del recurso # 2.

El proceso 3 tiene asignado el recurso # 3, y hace la peticion del recurso # 1.

El proceso 4 tiene asignado el recurso # 2, y hace la peticion del recurso # 3.

El deadlock seria el ciclo:

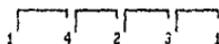


FIGURA # 1

Después de detectarla, una recuperación aceptable del "deadlock" estaría enmarcada por las siguientes técnicas:

- Abortar todo proceso implicado en el "deadlock".
- Abortos sucesivos hasta que desaparezca el "deadlock", el orden debe ser tal que minimice el costo de la pérdida de lo invertido en los recursos ya usados, el algoritmo de detección debe ser llamado para que determine que ya no existe el "deadlock".

Los recursos que estaban asignados a los procesos en "deadlock" que fueron abortados son ahora liberados.

EVITAR EL "DEADLOCK".

Para evitar el "deadlock" es necesario implementar un algoritmo que anticipe cuando va a ocurrir un "deadlock" y por lo tanto tenga la capacidad de rechazar una petición de un recurso. Esquemáticamente tal algoritmo sería como sigue: antes de asignar un recurso se debe cambiar la gráfica de estado del sistema considerando el caso en que se asigne el recurso y llamar al algoritmo que detecta el "deadlock", si el resultado de dicha llamada es que no ocurrirá, entonces se asigna la petición. Este procedimiento funciona cuando el causante del "deadlock" es el último proceso, para cuando no es así se implementara a modo que prolongara la revisión en varias instancias a futuro.

En el departamento "scheduling" las políticas son las que mantienen al sistema balanceado, es decir asegurar que ningún tipo de recurso está sobreutilizado o subutilizado además de evitar el "deadlock".

1.3.2 "SCHEDULER"

Las circunstancias en las cuales el "scheduler" puede ser activado son:

- 1.- Petición de un recurso.
- 2.- Liberación de un recurso.
- 3.- Llegada de una nueva tarea al sistema.

Los algoritmos de "scheduling" son necesarios para determinar el orden en que los recursos serán asignados a los usuarios, ya que en un ambiente de multiprogramación ocurrirá que varios usuarios deseen usar un recurso no compartible al mismo tiempo, por lo cual debe ser establecida la política de asignación de los recursos a los usuarios.

- . El primero en llegar primero en ser atendido (fcfs)

En todos los casos en que no hay razón para dar preferencia a un proceso sobre otro se usa la disciplina primero en llegar primero en servir "first come first serve". Este algoritmo es atractivo ya que las operaciones para insertar o remover un proceso son muy sencillas (También se le conoce como FIFO "First in first out").

. La tarea del menor tiempo.

La cola del procesador es ordenada de acuerdo al tiempo de ejecución requerido por cada proceso, éste algoritmo es recomendable en sistemas tipo batch (donde puede estimarse el tiempo de ejecución), en su forma más simple permite a todos los procesos chicos que su ejecución sea terminada, una modificación permite que un proceso se apropie del procesador si el tiempo asignado al proceso es menor que el que requiere para terminar su ejecución. Otra modificación asegura que las tareas largas no se retrasen indefinidamente, gradualmente incrementa la prioridad del proceso de acuerdo al tiempo que tiene en la cola.

. "Round robin".

Este algoritmo fue desarrollado para dar respuesta rápida a peticiones pequeñas de procesador, cuando no se puede estimar de antemano el tiempo de ejecución, cada proceso en el sistema recibe una cantidad fija de tiempo de procesador llamado ("quantum") y es regresado al final de la cola en caso de sobrepasar ésta cantidad, la cola es circular, y es ordenada de acuerdo a la cantidad de tiempo desde el último servicio, si el proceso requiere una gran cantidad de tiempo dará varias vueltas en la cola antes de terminar su ejecución. Si el proceso requiere menos tiempo en la primera vuelta terminará su ejecución, lo cual hace que el "overhead" sea proporcional al número de "quantums" requeridos por un proceso.

. Cola de dos niveles.

Una variación del algoritmo "round robin" el cual contrarresta la degradación en el funcionamiento del sistema, es el algoritmo de cola de dos niveles, en donde los procesos que no terminan su ejecución en un número fijo de "quantums", son sacados de la cola y puestos en otra llamada "background", la cual es servida solo si no hay procesos de terminal.

. Prioridad lineal

Un nuevo proceso incrementa su prioridad en forma lineal con respecto al tiempo y es aceptado hasta que alcance la prioridad del proceso que ha sido aceptado, o cuando el conjunto de aceptados esté vacío. Los procesos se dividen en nuevos y aceptados; los procesos nuevos no recibirán servicio hasta que se cumpla con una condición. La condición para aceptar un proceso nuevo esta determinada por las prioridades de los procesos en las colas, dicha prioridad está en función del tiempo.

. Primero el proceso de mayor prioridad (hpf)

El método más simple para calendarizar una cola de listos para ejecutar, es asignar el procesador al proceso que tenga la prioridad más alta.

Este algoritmo puede funcionar en dos casos:

- . En el caso de que sea válido la no apropiación del procesador, el proceso con prioridad mayor se ejecuta hasta terminar o bien hasta que sea bloqueado o haya una interrupción por requerimiento de entrada/salida. Si llega un proceso con mayor prioridad que la que tiene el proceso que se está ejecutando éste debe esperar hasta que termine el primero.
- . En el otro caso se vale la apropiación del procesador y si llega un proceso con prioridad mayor que el que se está ejecutando, éste se apropia del procesador y el que se estaba ejecutando tendrá que esperar.

1.3.3 POLITICAS DE CONTROL

Entre las políticas de control se encuentran las restricciones de acceso, y las de racionalización. La primera es la anulación de las facilidades de acceso que involucren gran utilización de recursos. A ciertos usuarios (programadores con poca experiencia o estudiantes) y mas aún, puede limitarse el uso de recursos a los usuarios que trabajen en tiempo compartido mientras que a los que trabajan en "batch" se le puede permitir una mayor cantidad de recursos, que serán asignados cuando éstos no tengan demanda. Las políticas de racionalización fijan tiempos de uso, por ejemplo el tiempo para usar el procesador, o la asignación de memoria que puede ser usada por un proceso. También se refieren a la cantidad de recursos que son utilizados por un usuario en una semana o un mes.

En el caso de que algún usuario quiera sobrepasar el límite de los recursos a que tienen derecho, el sistema suspenderá la ejecución del proceso o simplemente negará la asignación de dicho recurso.

1.4 SISTEMAS DE SOFTWARE

Un sistema operativo en una computadora de propósito general es desde luego un sistema de software. Algunos sistemas grandes de programación desarrollados en un lenguaje en particular, satisfacen los requisitos de ser sistemas de software; por ejemplo los sistemas de reservación en aerolíneas, manejo de información, procesos de control industrial, operaciones de tiempo real que permiten el mantenimiento de inventarios, y el control de una computadora pueden ser vistos como una instancia de sistemas de software.

Lo común en estas implementaciones es el estar desarrollados e incluidos como un componente mayor, en un computador. Son "sistemas" en el sentido que provee un número de funciones interrelacionadas.

En contraste con un programa que provee solo una función por instancia, un sistema de software usualmente implementa varias funciones de igual nivel relativo. Por ejemplo un sistema procesador de eventos en línea permite órdenes de entrar, cambiar, cancelar, recobrar para inspección puede generar facturas de cargamento, notas de producción y actualización de inventarios, y puede ser usado para generar reportes administrativos en actividades de ventas.

Las funciones que provee un sistema de software están separadas en el sentido que no todas están cargadas en el mismo programa de cómputo. Se puede borrar algunas de ellas o agregar otras y tranquilamente seguir teniendo un sistema de software. Pero si se quita un paso en un sistema, por ejemplo el sistema que calcula la función Bessel, probablemente no se podría tener un programa que sea capaz de calcular la función Bessel.

"Una propiedad fundamental de sistemas de software es que controlan o al menos influyen gradualmente el control de recursos básicos de un sistema de cómputo" (5).

Muchos sistemas de software típicamente proveen funciones de identificación de usuario. Esto es concerniente con el control de recursos del sistema.

Las características comunes que comparten los sistemas de software son:

- . Actividad concurrente
- . Recursos compartidos
- . Información compartida
- . Gran interacción con hardware
- . enlace múltiple con el mundo externo
- . Almacenamiento en gran cantidad de información
- . Almacenamiento de información en varios niveles
- . No determinístico
- . Modularidad
- . Operaciones de "multiple"

Muchos de los sistemas de hoy permiten una gran actividad concurrente a varios niveles.

Muchos usuarios pueden estar tecleando en sus terminales al mismo tiempo; transfieren información, usan los recursos de entrada y salida, ocupan el procesador, varios trabajos de usuario pueden completar parcialmente acciones diferentes para cada uno, procesos centrales múltiples pueden estar en operación, son un ejemplo típico de actividad concurrente.

Muchos sistemas comparten recursos de hardware, por razones económicas la memoria principal, archivos en disco y el procesador central pueden ser compartidos entre diferentes programas. Los mecanismos para afectar y controlar estos recursos forman una parte importante en muchos sistemas.

La información es compartida, los programas necesarios por muchos usuarios, las bases de datos usados por diferentes programas y operaciones concurrentes de datos en el sistema son compartidos. Este compartir es controlado por el sistema para prevenir el uso sin autorización y/o la destrucción.

Una gran parte del software se dedica a encauzar la acción directa del hardware (tales como transferencia de e/s) como sus resultados finales.

Muchos sistemas de software se comunican con el mundo exterior de muchas maneras, en contraste con los programas comunes que sólo se comunican con uno o dos recursos externos.

Requiere de gran almacenamiento de información y/o procesos para utilizarlos cuando sean requeridos en el sistema de software. En los sistemas tipo "batch" usualmente se tienen algunos tipos de almacenamiento permanentes, al menos para comodidad de los programas usuarios. Esto permite tener disponible la información por un periodo de tiempo que se entiende va más allá de la ejecución de un simple programa y formar las bases para ensamblar uno o más sistemas avanzados.

Dado que un sistema de software provee muchas funciones diferentes y usualmente separadas, así como la utilización de información y recursos, no es sorprendente que el sistema sea frecuentemente no determinístico. Esto no se debe entender que no se conoce el resultado de un simple conjunto secuencial de acciones; sino más bien, por la multitud de eventos no se conoce el avance del orden en el cual las acciones ocurrirán.

Modularidad. Una característica estructural de los grandes sistemas de software es que tienden a ser modulares; que está constituido por partes que están conectadas por caminos bien definidos. Estos están divididos para separar funciones en parte al factor de su longitud y por ser desarrollados por varias personas.

El "multiplex" es una técnica común a muchas partes de todos los sistemas de software. Esta técnica deriva originalmente de los sistemas de comunicación, envuelve la rotación en el uso de un recurso (tal como un canal de Entrada/salida o el procesador central) a través de varios usuarios así que cada usuario parece que usa en forma exclusiva el recurso.

Para el desarrollo de sistemas de software se tendrá que adentrar en varios aspectos importantes como:

- . La estructura de computadoras
- . Manejo de procesos
- . Manejo de memoria principal
- . Manejo de archivos y datos
- . Los sistemas de entrada/salida
- . Protección de información
- . Lenguajes de programación
- . Procesadores de lenguajes de programación
- . Métodos y herramientas para la creación de sistemas

. LA ESTRUCTURA DE COMPUTADORAS

En los sistemas de software es importante la estructura del sistema de cómputo. A través del conocimiento de estas estructuras, el conocimiento de la representación de datos internamente, tales como el número de bits necesarios para representar los caracteres y a su vez el número de éstos que constituyen una palabra; formatos utilizados en la palabra para la representación de números (tales como enteros con o sin signo, o de punto flotante).

De igual modo los estados del procesador y memoria proveen una simplificación importante para la estructura del sistema de cómputo en general. Cada vez que el sistema operativo interrumpe el proceso, el estado del procesador del viejo proceso deberá ser guardado y se carga el estado del procesador del nuevo proceso.

El estado del procesador es el conjunto de registros y banderas que contiene la información generada al terminar una instrucción y la disponibilidad de uso durante la ejecución de la siguiente instrucción. Similarmente el estado de memoria es el conjunto de palabras que constituyen la memoria principal.

. MANEJO DE PROCESOS

A la acción de ejecutar un programa o parte de un programa se le conoce como proceso.

Un proceso, es una secuencia de acciones continuas en el mismo programa, puede suspenderse. Aunque puede interactuar con otros procesos en puntos bien-definidos y no se puede asumir nada de los estados relativos del proceso. Hacer diferencia en un sistema de procesos paralelos es irracional y conduce a condiciones de competencia en el cual la salida de un proceso depende de la competencia en el tiempo con otros procesos.

Una sección crítica de un proceso es simplemente alguna secuencia de instrucciones de un proceso que no puede ser ejecutado sin la apropiada protección. En particular una sección crítica de un programa es un conjunto de instrucciones en el cual el resultado de ejecución puede variar impredeciblemente si las variables referidas en esta sección están disponibles para otro proceso paralelo y son cambiadas durante su ejecución.

Los procesos paralelos con secciones críticas (con respecto a otros) deberán ser mutuamente excluidos de la ejecución simultánea de sus secciones críticas.

Un proceso inactivo y fuera de una sección crítica no deberá impedir el avance de otros procesos independientes.

Los procesos no podrán esperar indefinidamente por recursos o señales.

Ahora bajo estos supuestos se integrarán los cuatro problemas principales del control de procesos que son: mutuaexclusión, sincronización, "deadlock" y comunicación (descritos en S. O.).

. MANEJO DE MEMORIA PRINCIPAL

Dentro del manejo de memoria principal un concepto que destaca es lo que se conoce como memoria virtual, la cual es usada para dar al programador la ilusión de que la memoria es mucho más grande de lo que es en realidad. Para hacer esto, es necesario permitir al programador el uso de un conjunto de direcciones diferentes del que provee la memoria y proveer mecanismos para la traslación de direcciones generadas por el programa dentro de la dirección correcta en memoria.

Una dirección usada por el programador es llamada "nombre" o "dirección virtual" y al conjunto de direcciones virtuales es llamado "espacio de direcciones" o "espacio de nombres". Una dirección usada por la memoria es llamada una localidad o dirección de memoria y al conjunto de tales localidades es llamado espacio de memoria.

En cuanto al mecanismo de direccionamiento se incorpora un modo de asociar nombres con localidades durante la ejecución de un programa. Para este fin se define, en todo momento una función la cual dará la dirección correspondiente en memoria de un nombre o un nulo (vacío) en el caso de que no se encuentre en memoria principal. Esta función se le conoce como la función de mapeo de dirección. Cuando el resultado es un nulo se interrumpe la ejecución del programa hasta que se transporte de memoria auxiliar al nombre requerido, y con esto la función tendrá ya una dirección a la cual referirse en memoria principal, y con esto se tiene la referencia completa y se reanuda la ejecución del programa suspendido.

Se complementa esto con dos mecanismos para cuando la memoria esté llena, un mecanismo decide que es lo que remueve de memoria principal, y un segundo mecanismo que decide cuando y en donde es cargado un requerimiento. Ninguna acción es tomada hasta que se haga presente el resultado nulo en la función de mapeo de dirección.

. MANEJO DE ARCHIVOS Y DATOS

Un archivo es una colección de información relacionada; los datos son información acerca de entradas que queremos compartir en la computadora: direcciones, valores numéricos, instrucciones,...etc.

Un sistema manejador de datos provee la facilidad para compartir los archivos de información. Un sistema de archivos es un sistema manejador de datos rudimentario, permite al usuario describir jerarquías de información, nombrar y usar relaciones entre conjuntos de datos. La mejor manera de caracterizar las relaciones entre archivos y sistemas manejadores de datos es observando que el hardware almacena o procesa información conocida como archivos.

Los conceptos relativos a los recursos físicos de almacenamiento son: campos, registros y archivos; donde la unidad básica para la transferencia entre memoria principal y auxiliar es el bloque, y la unidad de almacenamiento de memoria auxiliar es el volumen.

Un directorio como una estructura de datos es guardado como un archivo en memoria auxiliar. La función de un directorio es la de establecer y almacenar la asociación entre los nombres simbólicos y la localidad física donde son almacenados. Otra función es la de dar protección de información concerniente a que archivos se pueden acceder y cuáles no por otros usuarios. Otra función es proporcionar información de la organización de los archivos (por ejemplo si es secuencial, aleatoria o indexado).

. PROTECCION DE INFORMACION

Un buen sistema de archivo provee mecanismos para realizar un respaldo de la información del archivo de usuario y respaldo de procedimientos en eventos de error.

. SISTEMA DE ENTRADA/SALIDA

Históricamente la complejidad de los recursos de Entrada/Salida conjuntamente con el desarrollo de sistema de programas que fueron los precursores de los sistemas operativos actuales. Programas similares son una parte esencial de los sistemas operativos, compiladores, grandes programas de aplicación y otros sistemas que realizan la distribución de una amplia variedad de recursos de entrada/salida.

"El propósito principal de un sistema de entrada/salida es transferir información, ayudar a los usuarios a mover la información de apoyo, entre un proceso y la memoria principal o a otras partes del sistema de cómputo" [5].

Un computador tiene instrucciones las cuales tienen que efectuarse para provocar que la información sea transferida entre memoria principal y recursos externos, estas instrucciones de un computador son a nivel físico, y deberán ser de acuerdo a las características especificadas por el usuario.

Las instrucciones de un computador que provocan transferencia de información típica tiene dos partes: una parte es el dato y la otra es la parte de control. La parte de control selecciona el recurso particular implicado en la transferencia y manda información al recurso para activarlo o prepararlo para la transferencia.

Mientras la transferencia de información es llevada por mecanismos de hardware, el software deberá ser usado para el control de la transferencia.

La transferencia de información envuelve varios pasos, una posible secuencia sería:

Un programa produce alguna información en un arreglo de almacenamiento local. Entonces manda una señal al sistema de entrada/salida para la ejecución de una instrucción lo cual activa un módulo particular en el sistema operativo. Esta señal manda al sistema la transferencia de información a un recurso en particular. La información puede ser menor o mayor a un bloque estandard para transferir al recurso, y si, el sistema debe bloquear la información. Si hay un bloque parcial, el sistema deberá decidir si manda la información o espera por mas. El rebloqueo puede ser visto como encontrar una vía dentro de un "buffer" de longitud adecuada. Entonces el sistema deberá encontrar un canal que pueda ser usado para transferir la información al usuario.

. PROTECCION DE INFORMACION

La necesidad de compartir recursos en un moderno sistema de cómputo implica la necesidad de proteger la información. En muchos sistemas la mayoría de los recursos están en constante demanda por un número grande de usuarios. Muchos programas ocupan la memoria principal. Los datos pertenecientes a ciertos usuarios son almacenados en algún disco. Muchos programas tienen usuarios concurrentes entre sí. Para mantener una comunidad así es necesario que la integridad de la información se mantenga.

La protección de información engloba dos aspectos: el primero son las consideraciones técnicas y el otro son las consideraciones administrativas externas; En los mecanismos de protección interna incluyen programas que preguntan por códigos y contraseñas autorizados, bits de protección de memoria y estados de ejecución restringida. Las consideraciones externas incluyen todo el medio ambiente de condiciones administrativas que circundan en una instalación de cómputo y su uso.

A un usuario no se le debe permitir interferir con el monitor por el uso impropio de comandos indebidos de entrada/salida, de halt,..etc. Mediante la negación de ejecución de instrucciones privilegiadas, las cuales sólo pueden ser ejecutadas por programas privilegiados tales como los del sistema operativo.

Agrandar la privacidad de datos en sistemas compartidos, mediante la liga de una contraseña a su propio directorio de archivos y la referencia de archivos a su propio directorio. Cada archivo tiene control de acceso asociado a la contraseña.

Dado un conjunto de estados definidos para el sistema, cada estado tiene diferente información y diferente acceso a recursos. Un proceso será controlado permitiéndole ejecutar solo lo que le permite su estado definido, limitando y controlando los accesos del proceso.

. LENGUAJES DE PROGRAMACION

Un lenguaje es un conjunto de símbolos con significado usado para comunicación. En particular un lenguaje de computadora es usado para comunicarse con una computadora.

La interrelación entre sistemas de software y los lenguajes de computación se presenta en tres aspectos:

- 1.- los lenguajes son usados para construir sistemas de software.
- 2.- Los lenguajes de programación son una función importante prevista por muchos sistemas de software.
- 3.- Los lenguajes proveen la interfaz entre el usuario y otras funciones del sistema.

Los aspectos que envuelven a un lenguaje de computación son: sintaxis, semántica y pragmática; la sintaxis trata con la estructura de declaraciones en el lenguaje; la semántica es relacionada con su significado; y la pragmática incluye todo el demás comportamiento del lenguaje, particularmente su utilidad y factibilidad.

. PROCESADORES DE LENGUAJES DE PROGRAMACION

La mayoría de los lenguajes de programación permiten al programador crear programas usando una notación o formato que no puede ser ejecutado directamente, esto es, se requiere que éste texto sea procesado. Se cuenta con cuatro clases de procesadores de lenguajes de programación que realizan esta tarea, estos programas procesadores de lenguaje de programación se les conoce como:

- . Ensambladores
- . Compiladores
- . Intérpretes
- . Ligadores-Cargadores

Los ensambladores traducen las operaciones simbólicas a códigos de operación numéricas entendibles por el hardware, asignando espacio para interrupciones, operadores y manejo de transferencia simbólica de referencia a la dirección de memoria. Muchos ensambladores producen programas relocalizables en los cuáles todas las direcciones pueden ser cambiadas por el cargador antes de la ejecución, pudiéndose cargar y ejecutar en cualquier área arbitraria en memoria principal.

El compilador toma el programa en un lenguaje de alto nivel y produce un conjunto de instrucciones de máquina. Algunos deben ser ligados. Algunos producen lenguaje simbólico ensamblable que deben ser ensamblados posteriormente. Hay compiladores que producen instrucciones listas para su ejecución inmediata.

Los intérpretes producen instrucciones para una máquina hipotética que es implementada como un programa y ejecuta la instrucción para provocar un comportamiento específico. El código generado por el intérprete es en forma inmediata entre el lenguaje fuente y el código de máquina el cual es usado como entrada al ciclo de ejecución principal.

Los cargadores-ligadores, al cargar el programa dentro del almacenaje computacional mediante la adquisición del espacio en memoria principal y leer el programa dentro del área de memoria y notificar al "scheduler" cuando el programa es localizado. Resolver las referencias simbólicas entre programas independientes y el traslape de módulos dentro de la referencia de direcciones para su ejecución es la acción del ligador.

. METODOS Y HERRAMIENTAS PARA LA CREACION DE SISTEMAS

Después de haber escrito un programa que realiza alguna función, determinaremos cuando si y cuando no se realizará la función. Las pruebas pueden ayudarnos a determinar cuando el programa opera óptimamente, pero en forma general no se puede llegar a una prueba completa para un programa por la infinidad de situaciones del medio ambiente que se puede presentar en un tiempo determinado.

Dada la identificación y redefinición de las necesidades de sistema procesador de información presentadas en algún área de una organización se definirá claramente y sin ambigüedad que es lo que el sistema deberá hacer. El comportamiento deseado del sistema será establecido explícitamente en términos funcionales.

Una vez que se tienen las especificaciones funcionales del sistema se definirá la lógica del sistema que generan las funciones requeridas. El diseño lógico incluye el diseño del flujo de control, el flujo de datos, y la representación de datos que se usarán.

El siguiente paso será la designación y codificación del programa en módulos individuales en términos de sus entradas, salidas y funciones propuestas anteriormente definidas.

Después de que cada módulo se codificó, se harán las pruebas individuales. Una vez que sean correctos se ensamblarán los módulos en subsistemas y se probarán para detectar errores entre la llamada interfaz entre módulos. Eventualmente después de que todos los subsistemas han sido chequeados, una integración puede ser hecha. El sistema resultante deberá ser probado en situaciones operacionales.

En esta parte se diseñarán los cambios para readaptar al sistema en base a las pruebas tempranas en sistemas operacionales.

Una vez en operación el sistema deberá tener mantenimiento. En el caso del mantenimiento de software se refiere a los cambios y agregados de funciones dentro del sistema y también la tarea sin fin de encontrar errores (bugs) y errores operacionales en el sistema.

1.5 VENTAJAS Y LIMITACIONES DE LAS POLITICAS PROVISTAS POR EL SISTEMA

Uno de los componentes principales de los sistemas de cómputo es sin duda la memoria, ya que éste es el elemento que permite almacenar datos para poder procesarlos. La memoria se divide en memoria principal y memoria secundaria, de éstas la segunda permite almacenar grandes cantidades de datos por periodos grandes de tiempo. La memoria secundaria puede estar formada por varios dispositivos como son: cintas magnéticas, tarjetas perforadas, discos magnéticos, etc. "De estos dispositivos, los más adecuados para trabajar en línea son los discos magnéticos ya que son los que permiten tener un acceso más rápido a los datos" [7].

Como se mencionó anteriormente, los discos son dispositivos de almacenamiento para procesos en línea, por lo tanto se requerirá que la información que deberán contener estos discos sea la que se esté procesando continuamente.

La información que no se utilizará en procesos a corto plazo debe ser almacenada en otros dispositivos como son, por ejemplo, las cintas magnéticas. Tener almacenada información estática en dispositivos para trabajo en línea, como son los discos, representa una forma de hacer uso ineficiente de los recursos de cómputo.

En los sistemas UNISYS el subsistema de entrada/salida es el responsable de manejar los recursos en los dispositivos de almacenamiento secundario. Es decir, establece el medio de comunicación entre el procesador central y los datos almacenados en memoria secundaria.

La política de trabajo de los equipos UNISYS mantiene la siguiente filosofía: el sistema operativo no restringe a los usuarios a una cantidad de recursos, sino que al contrario, permite que un usuario pueda disponer de todos los recursos disponibles en el sistema si en un momento dado, así lo solicita el proceso que ejecuta.

Esta filosofía de trabajo es muy adecuada, ya que permite, en un momento dado, que un usuario pueda usar los recursos necesarios en procesos temporales, sin que para ello sea necesario que se le asignan todos esos recursos a dicho usuario permanentemente, sin embargo, lo anterior ocasiona que los usuarios no se preocupen por mantener en línea solo aquella información que necesitan procesar a corto plazo. Los usuarios conservan en disco, archivos que ya no usan y que podrían ser removidos después de ser respaldados en cinta magnética en previsión de un posible uso futuro.

1.6 EL DESARROLLO DE SOFTWARE EN FORMA CORRECTA Y EFICIENTE

En el proceso de creación de software se encuentra inmersa la solución de problemas. Tal proceso se puede resumir en seis pasos(1):

- . Formular el problema.
- . Análisis del problema.
- . Investigación.
- . Decisión.
- . Especificación.
- . Implementación.

1 Ver [7] en su presentación.

La formulación del problema es el primer paso. Para contemplar el problema. Uno de los mas desastrosos modos de formular un problema es el tratar inmediatamente de adaptarlo a una solución existente o por adoptar una solución general a un entendimiento pobre del problema.

El análisis del problema es el paso en el cual se especifica una solución. La mejor solución será seleccionada mediante la incorporación de parametros de costo, mantenimiento, exactitud, precisión, eficiencia y necesidades del usuario final.

El análisis del problema se llevará a cabo mediante el método de descomposición. La descomposición puede ser llevada a cabo mediante la partición funcional. Esto es un proceso en el cual el problema es tratado como una colección de partes intelectualmente manejables con interfaces entre las partes que sean tan simples y funcionales como sea posible.

La investigación es el paso en el cual se explorarán todas las posibles vías de solución al problema. Es en la cual se incluye la experiencia, textos, técnicas, prácticas existentes y sobre todo creatividad.

La fase de decisión es la de seleccionar entre la lista de soluciones potenciales, una de manera objetiva. El establecer el criterio de selección para una solución es una tarea tan difícil como la formulación del problema por si misma.

El proceso de decisión variará de problema en problema, pero generalmente sigue una serie de cuatro pasos:

- 1.- Definición del criterio de selección.
- 2.- El cumplimiento de soluciones alternas con respecto al criterio deberán ser previstas tan exacta como posible.
- 3.- El cumplimiento de soluciones alternas deberán ser consideradas en base a su cumplimiento previsto.
- 4.- La selección de la solución preferida debe ser hecha.

La fase de especificación es un proceso de ordenación y claridad, que debe ser completamente entendido por otras personas, mediante un documento formal que describa la solución; mediante la realización de una asignación y presentación de la evaluación a ésta. Es a través de este reporte que la solución es dada a conocer, y mucho más significativo en el reporte, es la habilidad de comunicación efectiva. También se presentará un modelo, el cual provee una manera eficiente de ilustrar y describir la solución final.

La etapa final es la implementación y entrega del producto terminado.

Cada proyecto de software tiene sus propias características; duración, extensión, contenido técnico, condiciones de trabajo y distribución de requerimientos que afectan significativamente la naturaleza de la estructura del proyecto de software.

El desarrollo de sistemas de software implica una serie de etapas donde la oportunidad de intervención de equivocaciones humanas es enorme. Los errores pueden ocurrir durante la asignación y desarrollo.

El factor de calidad básica para el software es que éste realice sus funciones de una manera congruente a objetivos específicos.

El producto final deberá de contener un mínimo de errores en la implementación de sus intenciones y equivocaciones acerca de las intenciones mismas.

Para tal efecto se cuenta con la verificación y validación para que la calidad del software se asegure.

La exactitud de un producto software es una medida específica de su calidad. La comprobación de una gran exactitud es el objetivo principal de la verificación y validación que es la ambición del desarrollo del software.

La prueba de un sistema es definida como el ejercicio controlado del código del programa para exponer errores. Mediante la entrada de todos los posibles casos de entrada para ver si se generan las salidas correctas.

"De acuerdo al objetivo de la prueba se puede obtener el tipo de error que se desea exponer.

Los errores que se pueden lograr exponer son:

- lógicos.
- de documentación.
- de sobrecarga.
- de cronometraje y coordinación.
- de capacidad y salida directa.
- de recuperación y retirada.
- de hardware

" [11].

La manera prudente de prueba de software es la que se realiza de manera jerárquica progresiva; comenzando con el módulo del nivel más bajo.

Después de las pruebas de los distintos módulos, las unidades de nivel se conectarán para determinar su funcionamiento en conjunto. Las pruebas en la integración enfatizan en la integración entre los componentes del software y sus interfaces.

CAPITULO 2

PRESENTACION DEL PROBLEMA

2.1 PROBLEMA A RESOLVER

Los sistemas de cómputo UNISYS serie "A", son sistemas multiusuario, en los cuales, como en todo sistema multiusuario, el compartir los recursos es una actividad primordial. Entre varios recursos que se deben de compartir se encuentra la "memoria secundaria" que físicamente constituyen los discos magnéticos.

La Dirección General de Servicios de Computo para la Administración asigna una serie de recursos a cada usuario, para que puedan procesar adecuadamente su información. Entre los recursos que se asignan a cada usuario está una cantidad de segmentos o sectores en disco magnético.

En los sistemas UNISYS serie "A", a un "disk pack" (disco apilado) se le llama "familia". En una "familia" se tendrán almacenados los archivos de varios usuarios.

Una "familia", además de contener los archivos, es el medio de almacenamiento temporal necesario para que los procesos en ejecución realicen las actividades de los usuarios asignados a ésta. Por lo que el espacio asignado a cada usuario tendrá que ser hecho de acuerdo a sus proyectos, pero también considerando el número de usuarios que comparten la "familia" y el espacio a usarse temporalmente por los procesos en ejecución.

Simultáneamente a la asignación de una cantidad de segmentos, se fija a los usuarios una "familia" de trabajo en la cual deberán almacenar sus archivos, ningún usuario (exceptuando claves privilegiadas) debería cambiarse de "familia" de trabajo.

El uso inadecuado de éste recurso nos puede llevar a serios problemas, uno de estos es la saturación de los discos magnéticos, que provoca que procesos que se deben ejecutar estén en un estado de espera, por falta del espacio suficiente en el disco para realizar sus actividades.

El problema de la saturación en disco magnético es provocado principalmente por el poco cuidado que se tiene en el uso (por parte del usuario) de su espacio asignado por la conservación de archivos en disco que no se utilizan en forma inmediata, y que provocan que el espacio asignado a cada clave usuarla sea rebasado muy fácilmente.

Otra causa de la saturación es la presencia de archivos de usuario en otras familias distintas a la familia de trabajo que le fue asignada.

Otro problema es la acción de remover los archivos de los usuarios cuando es necesario. Se tienen sistemas que llevan a cabo la remoción de archivos, pero se considera que dicha acción se realiza en forma arbitraria en el sentido de que no se toma en cuenta la opinión del usuario en la determinación de cuáles archivos conviene remover de acuerdo a sus objetivos inmediatos o futuros.

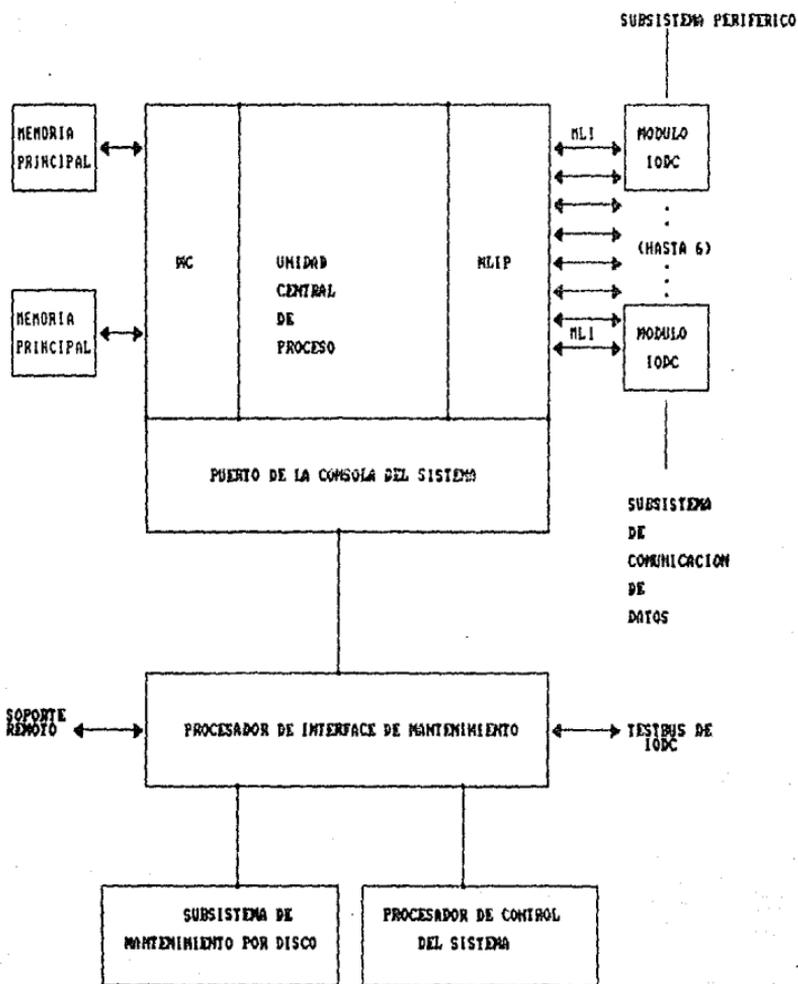
Por lo anterior es necesario contar con un sistema que lleve el control del uso de espacio en disco, de una manera eficiente a modo que:

- . Desaparezcan los procesos en espera de espacio en disco por la saturación causada por los archivos de algunos usuarios.
- . Detectar a usuarios con archivos que rebazan al espacio asignado en disco.
- . Notificar al usuario correspondiente del mal uso de éste recurso.
- . Si se encuentra en una situación que amerite remover archivos, que se tome en cuenta la opinión del usuario para la realización de ésta acción.

2.2 DESCRIPCION EL EQUIPO

La descripción de un sistema central "A9" se puede resumir definiendo sus cuatro módulos principales (Figura 2) que son:

- . El procesador central.
- . El subsistema de memoria.
- . El subsistema de mantenimiento.
- . El subsistema de entrada/salida y comunicación de datos.



MODULOS PRINCIPALES DEL SISTEMA UNISYS "A9".

FIGURA # 2

EL PROCESADOR CENTRAL

Al procesador central del sistema "A9" se le conoce como el Procesador de Lógica Múltiple (PLM), que basándose en el Sistema Operativo de los sistemas grandes de UNISYS llamado Programa Maestro de Control (PMC), que puede ejecutar tareas concurrentes. Este procesador central está constituido por cinco módulos los cuales son:

- . Controlador de Programas (CP).
- . Controlador de Tareas (CT).
- . Trayectoria de Datos (TD).
- . Unidad de Direcciones y Entradas (UDE).
- . Controlador Lógico de Almacenamiento (CLA).

El "CP" examina el código objeto, determina las tareas a ejecutar y establece los requerimientos y prioridades de las tareas. El "CP" entrega las tareas al controlador de tareas.

El "CT" localiza y controla el sistema de recursos para llevar a cabo las tareas entregadas por el "CP". Mantiene el estado de todas las tareas (las que están en espera, las que están listas para ejecución), selecciona la siguiente tarea para su ejecución, suspende a una tarea cuando es necesario y sincroniza la ejecución de tareas. También asigna el par de registros Topo-de-la-pila para las tareas que realizan operaciones con datos.

El "TD" almacena el conjunto de datos para realizar las operaciones lógicas y aritméticas. El "TD" contiene un arreglo para el almacenamiento de datos (como los registros Cima-de-la-pila), y contiene la unidad lógica-aritmética.

El "UDE" convierte las direcciones de memoria virtual, que se calculan en tiempo de compilación, a direcciones de memoria absoluta en tiempo de ejecución.

El "CLA" es un microcódigo el cual contiene los algoritmos usados para controlar la ejecución de otras partes del procesador. Cuando el "CP" recibe un operador y el número de pasos localiza el algoritmo para el operador y controla la ejecución de los pasos del requerimiento. Por tener el sistema central "A9" el controlador lógico de almacenamiento se le denomina "sistema E-mode" (Figura 3).

Al nivel más bajo hay cuatro pasos en la ejecución de una instrucción, que son:

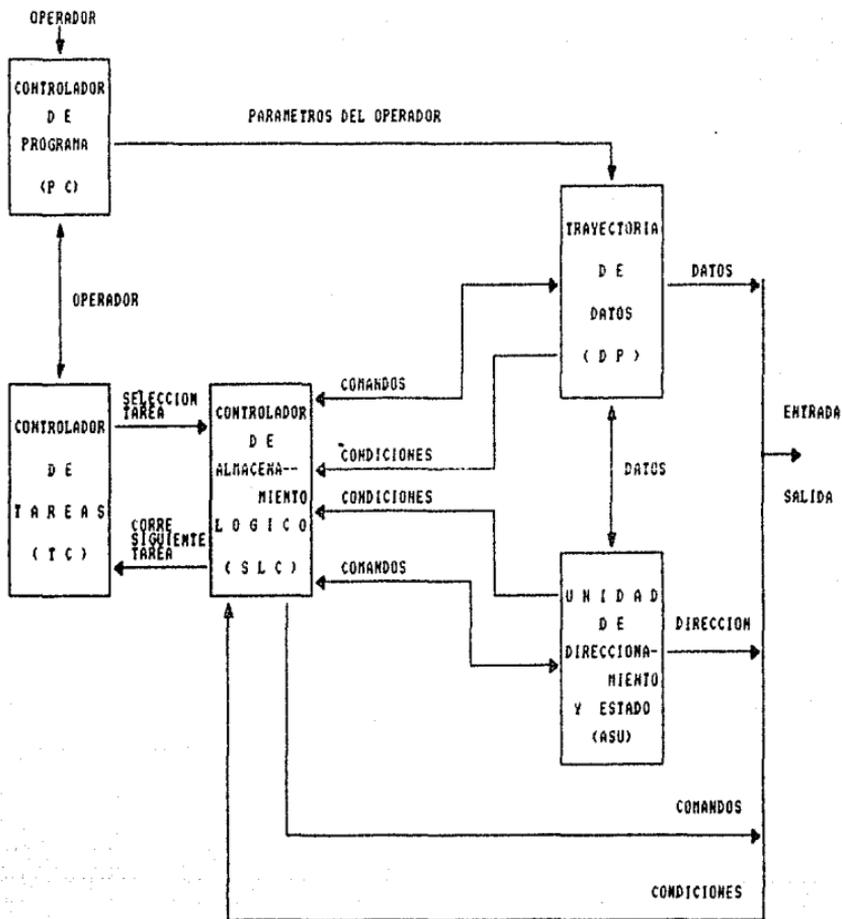
- 1º Búsqueda de instrucción en memoria.
- 2º Decodificar la instrucción.
- 3º Ejecutar la instrucción.
- 4º Almacenar el resultado.

La técnica "Pipelining" permite que varios submódulos dentro de la unidad de proceso central ("UPC") realicen en paralelo varios pasos de ejecución de una instrucción.

Como resultado se tiene que el "UPC" puede empezar la siguiente instrucción mientras se completa el proceso de la anterior.

EL SUBSISTEMA DE MEMORIA

El subsistema de memoria del sistema central "A9" es capaz de direccionar 1, 2, 3 o 4 millones de palabras (6, 12, 18 o 24 millones de bytes) de memoria.



EL PROCESADOR DE LOGICA MULTIPLE

FIGURA # 3

El subsistema de memoria interactúa con el procesador a través del controlador de memoria. El controlador de memoria también controla gran parte de la memoria cache, la cual contiene copias de bloques recientemente cargados en memoria. Si un bloque es requerido se verificará que este presente en la memoria cache antes de ir directamente a memoria principal.

La memoria principal es un recurso de almacenamiento de acceso al azar, contiene un número variable de palabras. Esta memoria es accesible solo al PLM al cual está conectada.

. EL SUBSISTEMA DE MANTENIMIENTO

"El subsistema de mantenimiento incluye:

- 2 consolas de operación
- 1 diskete removible 5 1/4
- 2 discos winchester fijos
- 1 procesador de interfase de mantenimiento" [16].

Las consolas de operación son ET2000 las cuales funcionan como sistema de Terminales de Despliegue de Operación (TDO) y como sistema de control de proceso, con el cual se puede iniciar el sistema, desplegar el estado del sistema o provocar la ejecución de programas mantenimiento y diagnóstico.

El procesador de inter-fase de mantenimiento contiene un subsistema de mantenimiento y un microprocesador que puede ejecutar programas de mantenimiento que prueban y controlan todos los otros recursos físicos del sistema central "A9".

• EL SUBSISTEMA DE ENTRADA/SALIDA Y COMUNICACION DE DATOS

El sistema central "A9" usa el manejador de comunicación de datos y periféricos de los equipos "Burroughs Universal I/O". Cada subsistema periférico es conectado a un Procesador Ligador de Datos (PLD), el cual es designado para controlar un tipo específico de periférico. Algunos PLD's pueden tener solo un periférico (impresoras en línea), pero otros periféricos (disk pack, cintas) pueden ser conectados a un simple PLD a través de un controlador y/o intercambiador.

El subsistema de comunicación de datos incluye dos tipos especiales de PLD's el "network support processors" (NSP) y el "line support processors" (LSP).

Los PLD's están agrupados dentro de un módulo "I/O-data comm"; cada módulo puede contener hasta ocho PLD's dependiendo del número de circuitos cargados en los diferentes tipos de PLD's usados. El sistema central "A9" puede incluir de uno a tres módulos "I/O-data comm". Estos módulos son conectados al Procesador de Interfaz a Nivel de Mensaje (PINM) que es un procesador de Entrada/Salida que hace contacto directo con el subsistema "Universal de I/O". Cuando una operación de E/S es realizada, el controlador de programa genera una operación PINM para ser ejecutada. Otras operaciones se continúan ejecutando concurrentemente con la operación PINM así que la ejecución de instrucciones continúa mientras una E/S está siendo ejecutada.

El dato de salida a un recurso es transferido desde el sistema central al PINM, a través de los cables llamados "Message Level Interface", a la base "I/O-data comm", y al procesador ligador de datos y finalmente al periférico destino. Un dato de entrada desde un periférico sigue la ruta inversa.

El PINM cuenta con procesadores ligadores de datos especiales para cada tipo de periférico. Cada procesador incorpora un complemento de memoria local que es usado de "buffer". La memoria está dividida internamente en muchos bloques los cuales son estructurados de acuerdo al tipo de periférico, hay que tener en cuenta que la transferencia de datos entre el procesador de nivel y el procesador ligador de datos es realizada en bloque. Cada módulo base de entrada/salida puede estar físicamente conectado hasta a cuatro interfaces de nivel de mensaje, permitiendo una gran flexibilidad en la configuración de la entrada/salida.

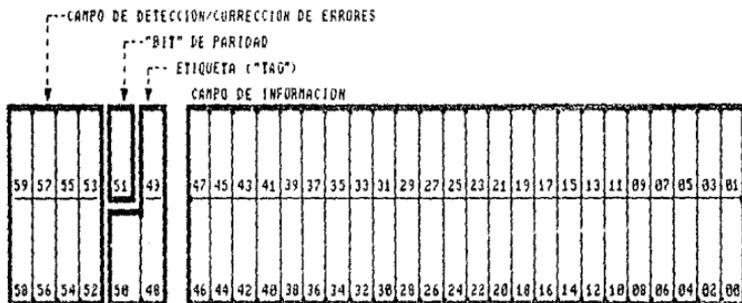
Para la representación de datos, se utiliza la unidad básica que es la palabra, la cual consiste de 50 "bits" consecutivos que se dividen en un campo de 8 "bits" para el código de detección y corrección de errores, un "bit" de paridad, un campo de tres "bits" llamado etiqueta y otro campo de 48 "bits" que es el que contiene la información (Figura 4).

El campo de detección y corrección de errores no está disponible al usuario; sólo lo utiliza el sistema para detectar y corregir errores de un solo "bit", se detectan errores que involucren un número mayor de "bits" pero no se corrigen.

El "bit" de paridad tampoco está disponible al usuario, el sistema sólo lo usa cuando las palabras son transferidas de un módulo a otro; checar la paridad en una palabra es checar si el número binario es par o impar y que coincida con el "bit" de paridad de la palabra transferida.

El campo etiqueta se utiliza para determinar que tipo de palabras es (de doble precisión, de precisión simple, de control, descriptor de datos, de dirección indirecta, etc.).

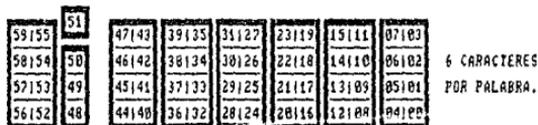
LA ESTRUCTURA DE LA PALABRA EN LOS EQUIPOS UNISYS SERIE "A".



PARA REPRESENTAR VALORES NUMERICOS DE PRECISION SENCILLA, EL CAMPO DE INFORMACION TOMA LA SIGUIENTE ESTRUCTURA:



PARA REPRESENTAR CADENAS DE CARACTERES EBCDIC, EL CAMPO DE INFORMACION TOMA LA SIGUIENTE ESTRUCTURA:



PARA REPRESENTAR CARACTERES EN EXADECIMAL EL CAMPO DE INFORMACION TOMA LA SIGUIENTE ESTRUCTURA:



FIGURA # 4

Las palabras tipo caracter contendrán cadenas de caracteres y tienen en el campo de "tag" el valor cero, y contiene hasta seis caracteres.

En el caso de que contenga un valor numérico o información lógica el tipo es operando u operandos lógicos respectivamente.

En los operandos de precisión simple o sencilla el campo de información se divide en subcampos:

el bit 46 es usado para el signo de la mantisa.
el bit 45 es usado para el signo del exponente.
los seis siguientes bits del 44 al 39 es el campo del exponente.
los 39 bits restantes serán el campo de la mantisa.

La mantisa se representa internamente en base octal.

Los operadores de precisión doble utilizan dos palabras concatenadas de tal modo que forman un valor numérico simple, con punto octal entre las dos palabras.

Los operandos lógicos son palabras que resultan de operaciones lógicas o de una operación relacional. Una operación relacional es determinada cuando se comparan dos valores, y se tiene una operación lógica cuando se comparan "bits" correspondientes entre dos palabras.

2.2.1 EL SISTEMA OPERATIVO UNISYS

El Programa Maestro de Control (PMC) es el sistema operativo del sistema central "A9". El cual está escrito en un lenguaje llamado "NEWP".

El lenguaje "NEWP" permite usar el concepto modular. Cada módulo es una estructura independiente, con muchos procedimientos definidos en forma local al módulo global para usarse por otros módulos.

El PMC ocupa de 50 a 60 k-palabras y debe estar residente en memoria.

A través del PMC un usuario puede mantener o crear un catálogo de archivos.

El PMC se divide en 16 secciones que son:

- . Sistema manejador de interrupciones.
- . Manejador de memoria.
- . Programa controlador de entrada/salida.
- . Controlador de procesos.
- . Control de periféricos.
- . Manejador de discos.
- . Utilerías básicas.
- . Operador de interfaz.
- . Diagnósticos.
- . Entrada/salida de usuarios.
- . Comunicación de datos.
- . Pruebas de periféricos.
- . Flujo de trabajo "workflow".
- . Ordenamiento.
- . Ejecución independiente.

El sistema manejador de interrupciones maneja el uso de módulos del PMC para interactuar con los distintos módulos manejadores de recursos. Se dan tres tipos de interrupciones:

- 1.- Las interrupciones provocadas por el procesador de datos.
 - a). El intervalo de tiempo (Quantum); provoca al PCM cambiar entre diferentes tareas.
 - b). "Prencense bit" causa peticiones para el módulo manejador de memoria.
 - c). Operador inválido; usualmente debido a códigos objeto inválidos.
 - d). Índice inválido; debido a programas de usuarios con error en índices.
 - e). "Stack overflow"; debido a programas de usuarios con necesidad de una área de trabajo mayor.

- 2.- Las interrupciones provocadas por entrada/salida.
 - a). Cambio de estado; cuando una unidad periférica cambia su estado.
 - b). Fin de una entrada/salida; un requerimiento anterior de entrada/salida es completado.

- 3.- Las interrupciones provocadas por el software.
 - a). Siempre que una tarea interrumpe a otra tarea del mismo "job".

En el manejo de memoria se contemplan dos métodos. Estos métodos son parte del concepto de memoria virtual y son:

1.- Manejo de memoria virtual en demanda.

La memoria es asignada a los procesos hasta que la ocupen y no se le asignará toda la memoria requerida cuando empieza una tarea.

2.- Manejo de memoria virtual por conjunto de trabajo.

Mediante la determinación del porcentaje de memoria que siempre debe estar disponible y el porcentaje de memoria que por medio de "overlays" será pasado a disco, para que el sistema tenga memoria disponible.

El subsistema de entrada/salida y comunicación de datos realiza las funciones de entrada/salida, como la inicialización física de E/S para el usuario. Construye las palabras de control necesarias para las operaciones físicas de entrada/salida, comunica las necesidades de entrada/salida al MLIP, conduce las señales de terminación de una entrada/salida a la tarea del usuario, los errores físicos de entrada/salida también los maneja.

El control de procesos determina cuáles tareas correrán y cuando, chequea los cambios de estado de los periféricos, así como la localización de nuevos archivos, nuevas tareas y nuevos trabajos.

El control de periféricos se responsabiliza de todos los recursos periféricos, asigna los recursos a los programas mediante el uso de tablas de las unidades listas.

El manejador de discos inicializa los discos, hace la asignación de espacio a los nuevos archivos y lleva la seguridad mediante el control de lectura y escritura de datos en los archivos en disco.

Las utilerías básicas del MCP son: "autobackup" y los procedimientos de mantenimiento de la biblioteca de archivos.

Los procedimientos de comunicación con el operador son responsables de la entrada y salida de mensajes via el sistema TDD.

La responsabilidad del procesamiento de diagnóstico provee la habilidad de obtener "dumps" (vaciado de memoria) de programas cuando éstos los soliciten o cuando fallen.

Las salidas o entradas de los usuarios se mejoran mediante procedimientos que manejan datos; como apertura o cerradura de archivos, checar y modificar atributos de archivo, etc.

El módulo de comunicación de datos es la conexión entre programas del usuario y el sistema comunicador de mensajes y el "data comm".

El módulo manejador de datos es la conexión entre programas de usuario y bases de datos de los usuarios.

El manejador de pruebas de periféricos cuenta con rutinas que pueden ser usadas para mantenimiento y prueba confidencial del subsistema de entrada/salida.

La sección de flujo de trabajo "workflow" cuenta con tres submódulos; el primero es el compilador que genera código objeto después de checar su sintaxis, el módulo controlador que da el control después que el compilador "WFL" ha terminado y el tercero es el "Jobformatter", el cual ve las entradas en el trabajo e imprime la información en el "Jobsummary" y al final imprime el "backup".

El módulo de ordenamiento ordena los archivos del usuario en disco, en cinta o memoria, bajo el control del programador.

Se tiene que tres procedimientos están siempre corriendo en forma independiente, uno es el "ETERNALIR" cuya responsabilidad es la de buscar tareas para cargar y también checa los cambios de estado de periféricos disponibles; otro es el "anabolism" cuya responsabilidad es cargar tareas que son ejecutadas independientemente; el tercero es el "CONTROLLER" que es el principal manejador del sistema de comunicación.

2.2.2 LA INICIALIZACION

La inicialización del sistema se puede llevar a cabo a tres niveles que son:

- a) "cool/start".
- b) "halt/load".
- c) "cold/start".

El "cold/start" es requerido cuando el sistema es prendido por primera vez, o si el sistema en "disk pack" presenta alguna corrupción, el sistema deberá tener acceso a una cinta para copiar el MCP o un "disk pack" que contenga una copia del MCP.

Los requerimientos de un "cold start" son establecidos por las entradas para el programa cargador "system/loader".

El "halt/load" se puede llevar a cabo cuando se desee mientras el sistema esta corriendo, el operador dará un "halt" y después un "load", cuando esto ocurre todas las tareas son abortadas y reiniciadas al principio. Se realizará un "halt/load" si se agrega un nuevo módulo, cuando se agrega y reconfigura un nuevo periférico o cuando el sistema está caído.

El "cool/start" cargará el código objeto del MCP a "disk" pero dejará cualquier otro archivo en "disk" intacto (en el "cold/start" si hay archivos en "disk" se pierden todos), la misma función del "cold/start" es el "cool/start" excepto los mensajes "OLAY ROW" del sistema "system/loader".

2.3 HERRAMIENTAS A USAR

En las instalaciones de la Dirección General de Servicios de Cómputo para la Administración, se cuenta con tres sistema centrales UNISYS serie "A"; dos A9 y una A3, para los cuales se desarrolló el sistema Controlador de Recursos en Disco (SCR).

El desarrollo de dicho sistema se realizó en el equipo central A3 de esta dependencia, la fase de prueba se realizará en los tres sistemas centrales para que su desempeño sea óptimo en las tres máquinas.

Se utilizó el lenguaje de alto nivel DCALGOL que es una extensión del lenguaje ALGOL. ALGOL es el lenguaje nativo de los equipos UNISYS; muchos de sus sistemas y software ambiental incluyendo el compilador de ALGOL y sus extensiones están escritos en ALGOL.

DCALGOL es un lenguaje para programas especializados. Este lenguaje en los equipos UNISYS es el óptimo para el desarrollo de programas cuya finalidad específica gira alrededor del control de seguridad, control del sistema y el manejo de colas. DCALGOL provee el acceso a algunas funciones del sistema operativo, funciones de estado del sistema y es usado por programadores de sistemas que necesitan usar ciertas funciones privilegiadas.

CANDE provee funciones de edición de archivos en línea, proporciona comunicación entre programa y terminal mediante la compilación y ejecución de programas; también controla la red incluyendo recuperación de errores y seguridad.

Se realizó una modificación a CANDE para que se genere una coordinación entre CANDE y el programa CRDISK (ver apéndice C) para que CANDE se encargue de activar y desactivar dicho programa cuando sea necesario.

SOLUCION PROPUESTA

3.1 OBJETIVOS QUE CUMPLE

Los objetivos que cumple el sistema SCR son:

- Evitar el uso inadecuado de los recursos en disco (principalmente cuando se rebasa el número de segmentos asignados).
- Restringir el uso del sistema central a las claves usuarias que presenten un exceso de segmentos ocupados en disco.
- Respetar las políticas hacia algunas claves usuarias que manejan procesos importantes de la administración de la Universidad.
- Adecuarse al Sistema de Administración Y Control de Usuarios SAYCUS (ver apéndice c) a modo que se integren en forma óptima.

Para poder cumplir con los objetivos del sistema se efectúan las siguientes funciones:

- 1.- Monitorear los directorios del sistema central para detectar entre todas las claves usuarias las que rebasan sus segmentos asignados en disco.
- 2.- Evitar que una clave pueda usar libremente los comandos de CANDE, si se detecta que alguna clave tiene asignado un código de segmentos excedidos.

- 3.- Proporcionar las facilidades necesarias para que el usuario con el código de segmentos excedidos pueda realizar las acciones pertinentes para que pueda limpiar su clave del código mencionado, mediante la activación (por parte de CANDE) del programa CRDISK.
- 4.- Aplicar las políticas vigentes a los usuarios de procesos importantes.
- 5.- Mantener una comunicación eficaz con el módulo de segmentos del sistema SAYCUS.

3.2 PROBLEMAS QUE RESUELVE

La acción de remover archivos en forma arbitraria puede llevar a originar problemas si no se hace adecuadamente.

Mediante la acción de proporcionar la facilidad de que sea el usuario el que se encargue de remover sus propios archivos, cuando se encuentre excedido desaparece la posibilidad de que se presente algún problema. El usuario de acuerdo a sus intereses u objetivos tendrá que decidir cuáles archivos de su directorio se remueven y cuáles no.

Al restringir el uso del sistema central a un usuario por el exceso de segmentos ocupados en disco, hasta que regularice su situación, se destaca la importancia del buen uso de éste recurso.

Al mantener cada usuario su directorio respectivo sin sobrepasar el número de segmentos asignados en disco, el problema de saturación de disco será menos frecuente. Así que se desvanecerá el problema de procesos en espera de espacio disponible en disco.

3.3 DESCRIPCION GENERAL DEL SISTEMA

En una primera fase se revisarán todas y cada una de las claves usuarias, para determinar que claves utilizan un espacio en disco mayor al asignado. Las claves que sean detectadas se les caracterizarán con un código que el sistema central "A9" lo reconocerá como "clave excedida en uso de segmentos en disco".

Al entrar al sistema a través de CANDE con una clave de código de excedida, activará el programa CRDISK.

Esta primera fase se realiza por el programa "cred" que forma parte del módulo de segmentos del sistema SAYCUS, el cual se modificó de modo que además de revisar a todas las claves usuarias en el uso de segmentos en disco, se le agregó la rutina que realiza la caracterización de las claves con el código de excedido.

Una vez activado el programa CRDISK tiene dos maneras legales de concluir: una a través del módulo "liberación" el cual lleva a cabo la acción de quitarle a la clave el código "excedida en segmentos" y retomando el uso de CANDE libremente. La otra manera es a través del módulo "terminación" el cual termina el programa pero sin liberar a la clave de dicho código.

La primera pantalla que despliega el programa presenta los objetivos del mismo. Una vez que el usuario leyó el despliegue, pasará a la siguiente pantalla (pantalla "home") con un simple "transmit" la cual despliega los comandos que puede activar. Esta pantalla "home" se hará presente cada vez que termine cualquier comando activado.

El programa CRDISK está compuesto por seis módulos (Figura 5) los cuales son:

- El módulo ayuda.
- El módulo copia.
- El módulo remover.
- El módulo liberación.
- El módulo directorio.
- El módulo terminar.

El módulo "ayuda" está compuesto por una serie de pantallas que despliegan la información de todos y cada uno de los comandos implementados, una vez que el usuario se encuentre en éste módulo, cada pantalla de información le va mostrando los comandos disponibles que puede activar para obtener más información.

El módulo "copia", es el que se activará cuando se desee hacer una copia del total o una parte de su directorio de archivos hacia una cinta magnética; éste módulo va guiando al usuario para ensamblar un job que copiará sus archivos a la cinta magnética.

El módulo "remover" es el módulo que se debe activar para que el usuario remueva sus archivos. Este módulo tiene un proceso el cual al darle los nombres de los archivos los va poniendo bajo el directorio REMOVE/(NOMBRE), que tiene como finalidad que se verifique si está residente dicho archivo o si no, para que corrija algún error en el nombre del archivo, antes de realizar la acción de remover.

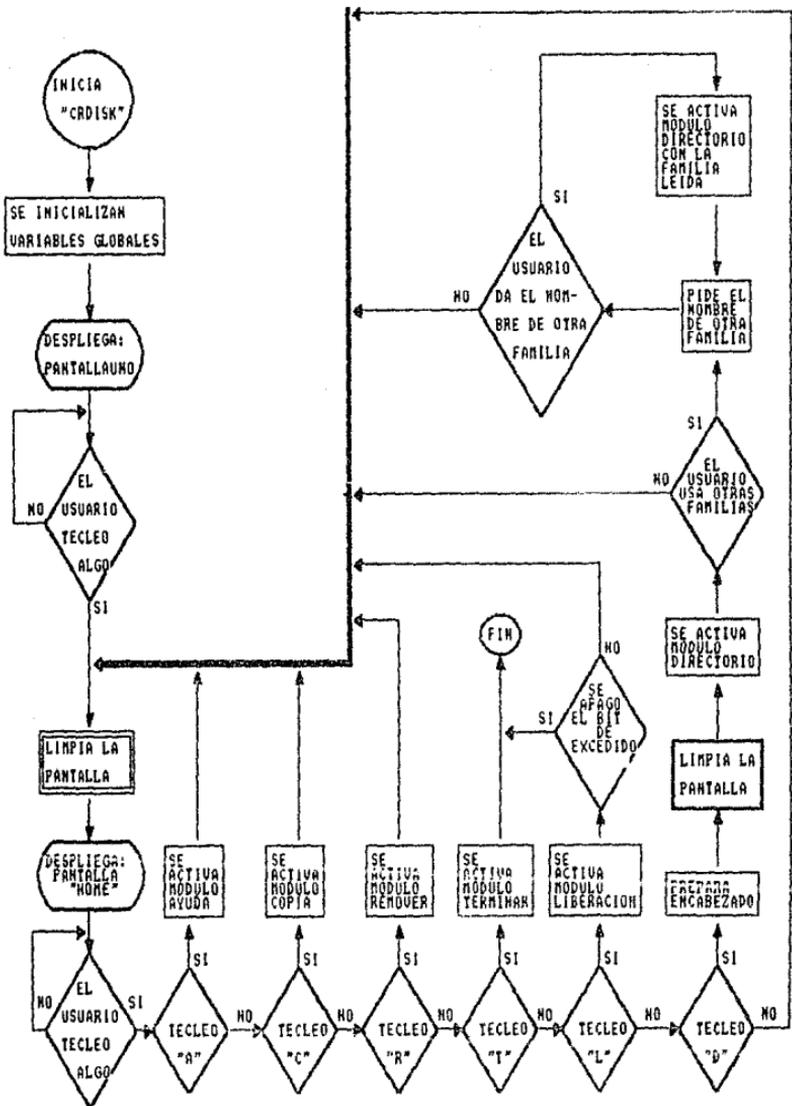


DIAGRAMA DE FLUJO "CRDISK"

FIGURA 8 5

El módulo "liberación" es el módulo que el usuario activará después de haber removido los archivos de exceso; teniéndose que éste módulo chequeará si el uso de espacio en disco es el adecuado de acuerdo al espacio asignado anteriormente a la clave, si éste espacio no es menor o igual al asignado, le avisará al usuario que es necesario que remueva más de sus archivos para que pueda liberarlo, de otra manera no podrá terminar óptimamente el programa CRDISK.

El módulo "directorío" se deberá activar para poder ver el directorío de los archivos del usuario, el usuario podrá ver su directorío y con esto decidirá que archivos borrará de su directorío de acuerdo a sus intereses.

El módulo "terminar" se activará para concluir el programa CRDISK sin liberar a la clave usuario; al finalizar el programa de este modo, la clave seguirá con el atributo de excedido, con lo cual al tratar de entrar al sistema a través de CANDE se activará de nuevo el programa CRDISK. Es recomendable activar este módulo cuando no se puede o no se quiere seguir el proceso al momento, y se desea dejarlo para después. En este módulo, se generarán los comandos para que se termine el programa y CANDE termine la sesión.

3.4 DESCRIPCION DETALLADA DEL SISTEMA

El módulo "ayuda" (Figura 6), está integrado por los siguientes procedimientos:

- . Pantallauno.
- . Ayuda.
- . Directo.
- . Copi.
- . Rem.
- . Lib.
- . Termina.

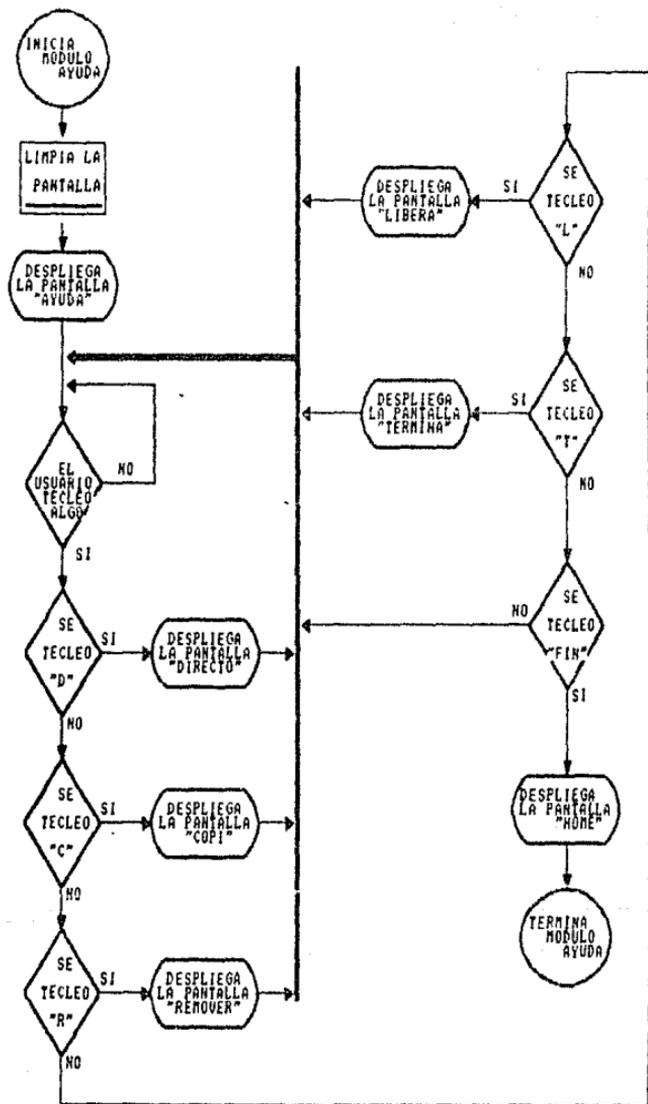


DIAGRAMA DE FLUJO MODULO "AYUDA"

FIGURA # 6

El procedimiento "Pantallauno" despliega la primera pantalla, ésta es la presentación del programa, la cual explica los objetivos del sistema. La pantalla permanecerá desplegada hasta que el usuario teclee cualquier tecla, haciéndose presente la pantalla "home".

El procedimiento "Ayuda" es el que controla los despliegues de información referente a los comandos que se pueden activar, mediante la opción de elección, por parte del usuario de cuál información desea; el usuario elegirá una tecla de acuerdo a la letra inicial del comando que desee la información, hasta que teclee la palabra: 'fin' para salirse del módulo ayuda.

El procedimiento "Directo" despliega una pantalla con información acerca del comando "directorio".

El procedimiento "Copi" despliega la pantalla con la información del comando "copia".

El procedimiento "Rem" despliega la pantalla con información del comando "remover".

El procedimiento "Lib" despliega la información relativa al comando "liberación".

El procedimiento "Termina" despliega la información del comando "terminar".

El módulo "copia" (Figura 7), construye un trabajo (Job), mediante una serie de datos que solicita al usuario; éstos son:

- . El nombre de la etiqueta para la cinta magnética.
- . Un mensaje dirigido al operador para que monte la cinta.
- . La protección de su directorio total o parcial.
- . Si es parcial, los nombres de los archivos.

Todas estas respuestas son guiadas mediante una interrogación directa hacia el usuario por parte del programa.

Cuando el "job", está completo el programa lo libera para su ejecución; el usuario se percatará del desarrollo de ejecución del "job" a través de los mensajes desplegados en pantalla. Estos mensajes son:

- . El programa esta suspendido.
- . Los nombres de los archivos copiados a cinta.

El programa se suspende en su ejecución para que el operador monte la cinta y haga la asignación de la cinta al proceso.

Cuando el "job" termina el programa le avisa al usuario, para que éste continúe mediante la transmisión de cualquier tecla; con esto se hará presente la pantalla "home".

Este módulo está constituido por un solo procedimiento llamado copy.

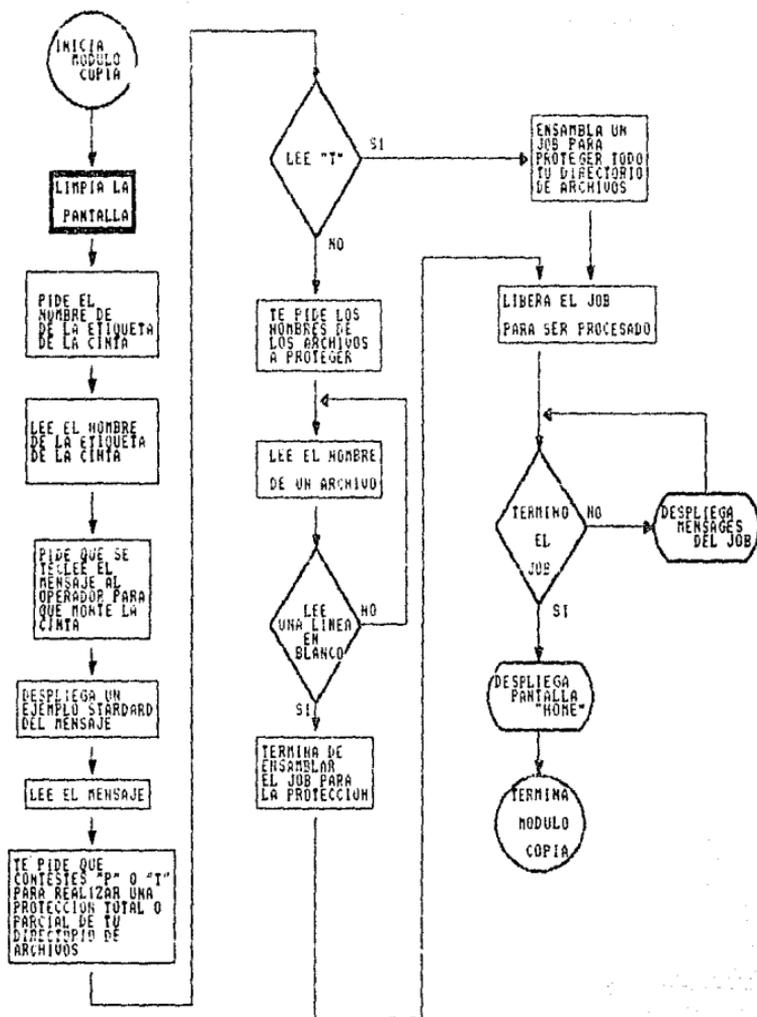


DIAGRAMA DE FLUJO MODULO "COPIA"

FIGURA # 7

El módulo "remove" (Figura 8), cuenta con un solo procedimiento llamado remove1; en el cual se verifican los nombres de los archivos a remove mediante una agrupación de estos nombres de archivos en un solo directorio. Esta verificación se lleva a cabo mediante el chequeo si el archivo está residente en el directorio del usuario.

Si el nombre de archivo corresponde a un archivo residente pasará a formar parte del directorio REMOVE/(nombre), en caso contrario se le avisará al usuario que el archivo correspondiente al nombre dado no existe, con esto el usuario corregirá algún error existente en el nombre.

Una vez que se tienen todos los nombres de los archivos a remove bajo el directorio REMOVE/=, se activará la acción de remoción; como respuesta a esta acción el sistema responderá con el número de archivos removidos.

Al terminar el módulo el programa le avisará al usuario para que éste continúe con el proceso; al oprimir cualquier tecla y transmitir se presentará la pantalla "home".

El módulo "liberación" (Figura 9), cuenta con un solo procedimiento booleano llamado "liberame".

En éste procedimiento se lleva a cabo la verificación del uso de segmentos. A través del módulo directorio se contabilizan los segmentos ocupados por los archivos de la clave correspondiente.

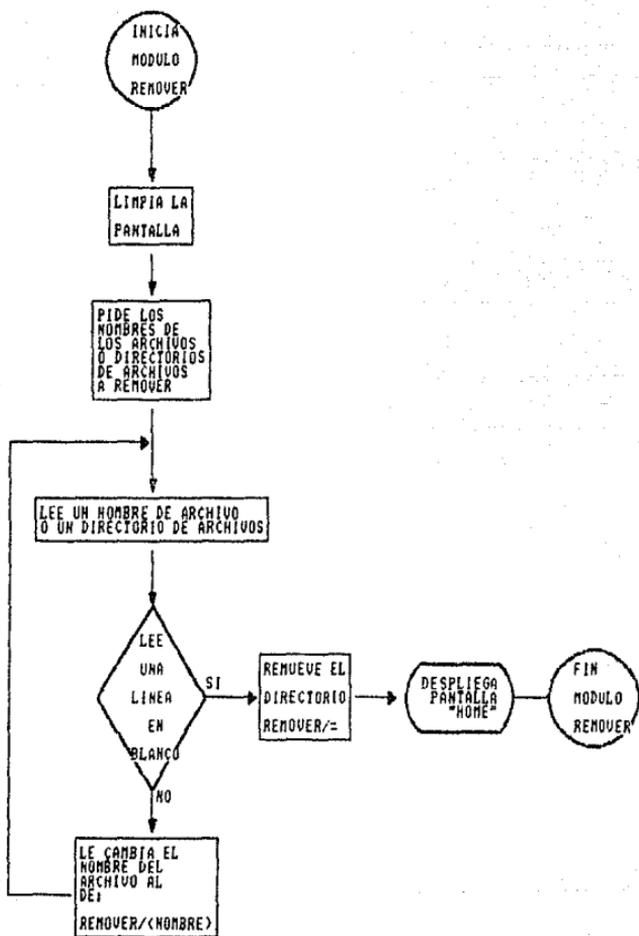


DIAGRAMA DE FLUJO MODULO "REMOVER"

FIGURA # 8

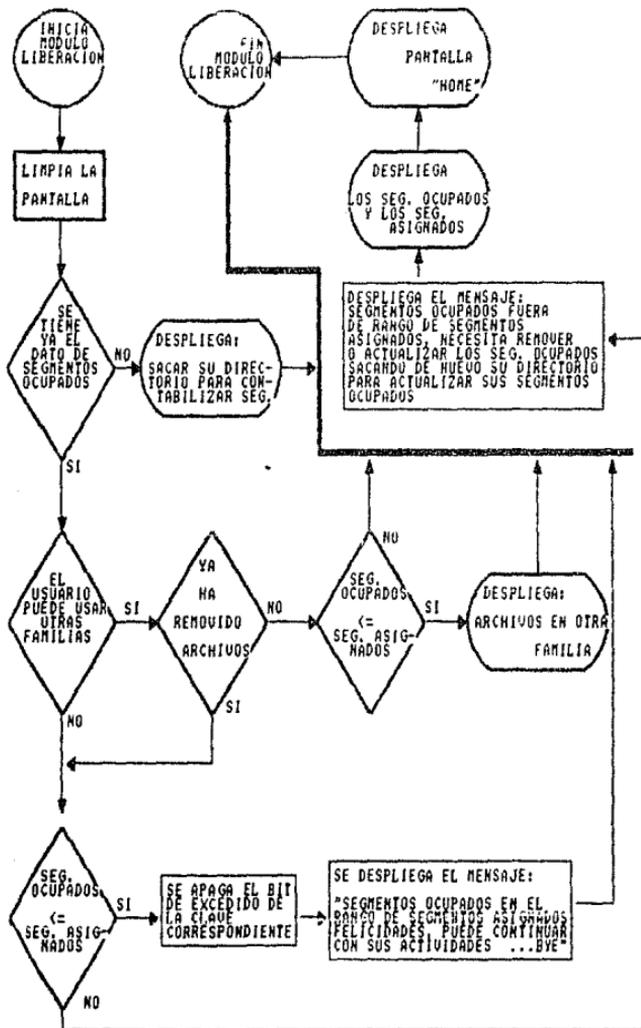


DIAGRAMA DE FLUJO MODULO "LIBERACION"

FIGURA # 9

Después de la acción de remove, si el número de segmentos ocupados en disco son menores o iguales a los segmentos asignados, se le quitará a la clave el código de "excedido en segmentos" y terminará en forma óptima el programa. En caso contrario, el programa le pedirá al usuario que haga otra remoción de archivos y le mostrará tanto el número de segmentos ocupados como el número de segmentos asignados.

Al terminar el procedimiento el programa avisará al usuario para que éste continúe con el proceso; al oprimir cualquier tecla y transmitir se presentará la pantalla "home".

El módulo "directorío" (Figura 10), cuenta con doce procedimientos cuyos nombres son:

- . Limpiatty.
- . Preparaencabezado.
- . Escribencabezado.
- . Familyonly.
- . Prodir.
- . Newkind.
- . Usoeficiente.
- . Ultimossegmentos.
- . Llamagestatus.
- . Tomaclaves.
- . Fechasjulianas.
- . Procesapack.

El procedimiento "Limpiatty" es el encargado de limpiar la pantalla de la terminal donde esté corriendo la aplicación. Este procedimiento se usa en toda la aplicación, no solo en el módulo directorío.

El procedimiento "Preparaencabezado" es el encargado de tener listo el encabezado de la pantalla utilizada en la presentación de los archivos residentes en el directorío de las claves usuarias.

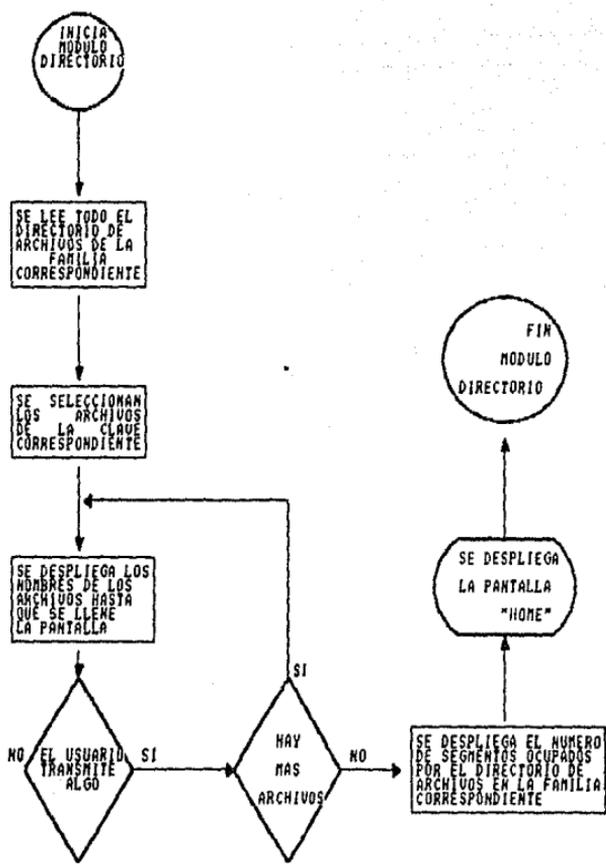


DIAGRAMA DE FLUJO MODULO "DIRECTORIO"

FIGURA # 18

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

El procedimiento "Escribencabezado" realiza la acción de desplegar en pantalla lo que el procedimiento anterior preparó.

El procedimiento "Famillionly" checa si el pack donde se procesará el directorio es el llamado "disk", si éste es el caso se le asociará al programa una característica necesaria referente a la familia para optimizar la acción.

El procedimiento "Prodir" es el encargado de activar al procedimiento extraedirectorios y si la clave utiliza varias familias (pack's), totaliza los segmentos ocupados por los correspondientes archivos.

El procedimiento "Newkind" es el encargado de determinar de que clase de archivo son los elementos del directorio de la clave usuaria. La clase puede ser: archivos de código, archivos job, archivos tipo data, archivos tipo backup, ..., etc.

El procedimiento "Usoeficiente" es el encargado de determinar si los archivos están "crunch" (esto es necesario para determinar correctamente el número de segmentos ocupados de cada archivo).

El procedimiento "Ultimossegmentos" es el encargado de calcular la longitud en segmentos del último renglón de los archivos, aquí se utiliza el resultado del procedimiento anterior.

El procedimiento "Llamagestatus" es el encargado de invocar a la función "gestatus" (función existente en el lenguaje DCALGOL) para obtener información de los archivos en las distintas familias en el sistema central; también realiza la acción de despliegue de errores en dicha invocación de la función.

El procedimiento "Tomaclaves" es el encargado de desplegar en pantalla los nombres de los archivos, el número de segmentos ocupados, junto con sus fechas de creación y último acceso. Este despliegue lo hace por una serie de pantallas las cuales son controladas por el usuario, teniéndose que una pantalla cuenta con un encabezado un número fijo de nombres de archivo, y la siguiente pantalla se hará presente hasta que el usuario así lo requiera.

El procedimiento "Fechasjulianas" es el encargado de la transformación de las fechas de creación y último acceso de los archivos, ya que la función "getstatus" los emite en el formato Juliana DDDAA y la transforma al formato DD/MM/AA.

El procedimiento "Procesapack" es el puente entre los procedimientos tomaclaves y llamagestatus; además de invocar a llamagestatus para que tomaclaves tome la información correspondiente a la clave usuaria, se realiza la acción de determinar cuál pack es el implicado para tal transacción.

El módulo "terminar" (Figura 11), esta constituido por un solo procedimiento el cual es el encargado de generar el código requerido para que el programa pueda terminar.



DIAGRAMA DE FLUJO

MODULO "TERMINAR"

FIGURA # 11

CONCLUSIONES

Para la realización del proyecto, fue necesario conocer el medio ambiente prevaleciente en el centro de cómputo, evaluar los requerimientos necesarios para que la aptitud de los usuarios, cambiara a favor de una mejora en el uso de recursos.

A la necesidad de poder tener un ambiente propicio para poder proporcionar una mejor atención a los usuarios, surge el proyecto de creación de un proceso que albergue un mayor control sobre las condiciones que pueden afectar a dicho ambiente. Al detectar el uso permanente y en constante crecimiento de los procesos que remueven archivos por exceso de segmentos, surge la necesidad de aumentar el control en el uso de la memoria secundaria, pero sin descuidar la atención a los usuarios y más aun mejorarla.

Con los cambios hechos a CANDE se obtiene una gran coordinación automática entre éste y el programa CRDISK, proporcionando una gran confiabilidad, exactitud y eficiencia. Mediante esta coordinación se logra que unos usuarios tengan una gran facilidad para utilizar el sistema central, y por otro lado se garantiza la comunicación directa entre el programa CRDISK y los usuarios que presenten irregularidades en el uso del recurso disco magnético, para que el propio usuario inmediatamente corrija su situación.

El mantener al sistema central atendiendo solo a usuarios que presentan en ese momento una situación óptima en el uso de recursos, ayudará a que el medio ambiente prevaleciente en el sistema de cómputo sea el más adecuado.

Como la restricción es una acción inmersa en el concepto de control, y el control de recursos en un sistema de cómputo es parte esencial del mismo sistema; la restricción de recursos es una acción indispensable y válida dentro del ámbito del control de recursos en el sistema de cómputo.

La corrección obligatoria de alguna irregularidad en el uso del recurso, antes de poder utilizar el sistema central es una política que ayudará a mantener bajo el nivel del mal uso de los recursos en discos magnéticos, pues ésto resalta la importancia del buen uso de dicho recurso para el usuario, que en última instancia es lo que se pretende.

Aunque restringir el uso del sistema central puede llevar a conclusiones de que se está realizando una incompatibilidad con una de las principales funciones de la dependencia, se debe tomar en cuenta que esto sólo se aplicará a los usuarios que presenten irregularidades en el uso de espacio en disco magnético y no a todo el conjunto de usuarios, y también que esta restricción terminará inmediatamente después de que el usuario corrija su situación.

Como es la primera vez que se implementa tal política, y también la incorporación del nuevo aspecto de realizar varias acciones en línea, tales como la acción de proteger y remover sus propios archivos, la acción de notificación del mal uso de los segmentos en disco magnético se integrará al ambiente del centro de cómputo como una medida para garantizar la optimización en el uso del recurso.

Con la implementación de éste proyecto se logra avanzar en el área de la administración de los sistemas de cómputo de la dependencia, principalmente por la integración del propio usuario en acciones encaminadas tanto a la administración de los equipos como a lograr la optimización en el uso de los discos magnéticos. Además el conocimiento adquirido al implementar dicho sistema, ayudará a abrir nuevos caminos para continuar con el desarrollo de proyectos con nuevos objetivos.

4.1 MEJORAS A FUTURO

Se implementarán formas de hacer al programa CRDISK más amigable, de manera que cuando el usuario se encuentre dentro del módulo remove y el programa le pida los nombres de los archivos a remove, el usuario sólo tenga que dar un número en lugar de dicho nombre. Con éste número el programa sabrá a cuál archivo se refiere. Para esto se cambiará la presentación del directorio de archivos a modo que en la pantalla se presente un campo mas que será el número de archivo, con esto el usuario sólo tendrá que escoger el número correspondiente al archivo que desea remove.

Otra mejora que se implementará es la de tener la opción de que el directorio se pueda imprimir cuando así se requiera.

APENDICE A

CAMBIOS NECESARIOS A CANDE

Una de las acciones que el sistema controlador de recursos en disco debe realizar es la de coordinarse con CANDE y que sea éste el que se encargue de activar al programa "crdisk".

La primera fase del sistema es el chequeo de todas y cada una de las claves usuarias en su situación en cuanto al uso de espacio en disco; mediante la exploración del "header" de todas la familias definidas en el sistema central (ver apéndice B).

En esta primera fase se clasificarán a las claves usuarias en: claves con exceso de segmentos en disco y claves con uso de espacio en disco en forma óptima; prendiéndole el "bit" de excedido a las primeras. En el archivo de control de claves del sistema (UDF).

La segunda fase del sistema es la que hará que CANDE detecte a las claves con exceso de segmentos en disco, y al tratar de hacer uso del sistema central implemente una liga de estas claves con el programa "crdisk".

La tercera fase del sistema es el proceso de limpieza de las claves del código de excedido en segmentos, éste proceso está implementado en el programa "crdisk".

23000	(SCRATCHPT), 0, LOGARRAY).SIGNBIT);	87958120
24000	IF BEXCII THEN	87958130
25000	BEGIN	87958140
26000	BEXCED100 := TRUE;	87958150
27000	SENDIT("EXCED100 EN SEGMENTUS.");	87958160
28000	END;	87958180
29000	NEXCED100 := * + 1;	87958190
30000	NLSN(NEXCED100) := LSN;	87958200
31000	TOTALSNEXC := * + 1;	87958210
32000	BEXCII := FALSE;	87958220
33000	END;	87958230
34000	IF BEXCED100 THEN	99364510
35000	BEGIN	99364525
36000	LSNI := TOTALSNEXC;	99364530
37000	DO BEGIN	99364540
38000	IF LSN EQL NLSN(LSNI) THEN	99364570
39000	IF BEXCED100 THEN	99364580
40000	BEGIN	99364590
41000	WELKND := SCRBLK; GETNEWELK;	99364600
42000	REPLACE P(TBC1) BY "RUN *DGSCA/A";	99364610
43000	TBC1 := 0 & 12 SIZEF & STAX WHEREF;	99364620
44000	RETURNLAY;	99364630
45000	GRINDIT(TASK0, 0, 0);	99364640
46000	END;	99364650
47000	LSNI := * - 1;	99364670
48000	END;	99364680
49000	UNTIL (LSNI LEQ 0);	99364690
50000	END;	99364700

APENDICE E:

EL ALMACENAMIENTO DE INFORMACION EN DISCOS MAGNETICOS

Unidades básicas de almacenamiento.

Las unidades básicas de almacenamiento en disco magnético son:

- a) El segmento.
- b) La palabra.
- c) El "byte".
- d) El "bit".
- e) El área.
- f) La longitud del área.
- g) La longitud del bloque.
- h) La longitud de registro.

El segmento es una medida fija de almacenamiento que es de 30 palabras (180 "bytes" o 1440 "bits"), es la unidad para medir el espacio ocupado por los archivos en el disco.

La palabra en los equipos UNISYS es la unidad básica de representación de datos, que tiene una longitud de 60 "bits" consecutivos, de los cuales 48 "bits" son utilizados para la representación de los datos del usuario y los "bits" restantes son utilizados por el sistema.

El "byte" es el conjunto de "bits" para representar un caracter. En particular el código ASCII Y EBCDIC necesitan ocho "bits" para representar un caracter.

El "bit" es la unidad lógica que sólo presenta dos estados: prendido o apagado.

El Area es una unidad de almacenamiento que agrupa bloques de registros físicos de un archivo.

La longitud del Area nos especifica el número de registros que contendrá una Area de un archivo.

Un bloque es un grupo de registros físicos que son empaquetados juntos para ser transferidos para o de archivos físicos.

La longitud del bloque es el número de registros lógicos del archivo dentro del bloque.

Un registro es un conjunto de datos conteniendo cadenas de caracteres, grupo de palabras binarias o ambas.

La longitud del registro nos especifica el número de palabras o "bytes" que contiene el registro.

Descripción del almacenamiento del "header" de los archivos residentes en una familia.

En los equipos UNISYS serie "A" a una colección de discos magnéticos con un nombre en común se les llama familia. Cada familia tiene un archivo llamado "flat directory" el cual contiene el "header" de los archivos de todos y cada uno de los archivos que residen en dicha familia.

Cuando una familia está constituida por varias unidades de disco se genera un "familyindex" que contendrá de uno hasta el total de unidades que constituyen la familia. El archivo llamado "flat directory" contiene el "header" para cada archivo residente en la familia, el cual contiene toda la información acerca del archivo.

"El "header" de cada archivo tiene una longitud variable de hasta 2048 palabras y presenta la siguiente división:

- a) Área fija de datos "header".
- b) Palabras de direccionamiento lineal (row).
- c) Palabras de atributos opcionales.
- d) Área de datos "header".
- e) La palabra "checksum" " [15].

El área fija de datos tiene una longitud de 18 palabras en las cuales se alojan los siguientes datos:

```
%-----%  
% PALABRA 0 (Validación, dirección y longitud) %  
%-----%
```

```
MARKERF = [47:16] % marca de validación  
HORBLOCKLENGTH = [31:11] % longitud del "header"  
HDRLOCATION = [20:21] % localización
```

------%
 % PALABRA 1 (Información "header") %
 ------%

OPENCOUNTF = [47:12] % numeración de stacks usando HDR
 FILEINDF = [35:12] % tipo de archivo
 PERMANENCYF = [23:01] % archivo temporal o no
 WRITTENONF = [22:01] % tiene información o no
 WROTELASTROWF = [21:01] % tiene el fin de archivo o no
 UPDATEBITF = [20:01] % se ha actualizado o no
 HORFIXEDSIZEF = [19:03] % la longitud de la parte fija del "header"
 HORAVAILSPACEF = [11:12] % número de palabras disponibles en el área de datos del "header"

------%
 % PALABRA 2 (Información) %
 ------%

IADBIT = [47:01] % 1 si es "IAD"
 PROTECTIONF = [46:01] % es archivo protegido o no
 CHECKEOF = [45:01] % deberá encontrar fin de archivo
 SYSTEMFILEF = [44:01] % es un archivo removible o no
 APLMAF = [43:01] % es un archivo control APL
 PHYSICALMODEF = [42:03] % "exmode" al crearlo
 RCDUNITS = [39:01] % atributo "units"
 RCROTYPE = [38:04] % atributo "filetype"
 SIZEMODE = [34:03] % atributo "sizemode"
 SIZEOFF = [31:16] % atributo "sizeoffset"
 SIZE2 = [15:16] % atributo "size2"

------%
 % PALABRA 3 ("Diskblocking") %
 ------%

BLOCKSIZEF = [47:16] % la longitud del bloque físico
 MINRECSIZEF = [31:16] % la longitud mínima del registro fís.
 MAXRECSIZEF = [15:16] % la longitud máxima del registro fís.

------%
 % PALABRA 4 ("Timestamp" en formato time(6)) %
 ------%

------%
 % PALABRA 5 (Información "row") %
 ------%

VERSIONF = [47:04] % identificadores de formato header
 % 3 para: 2.7 --> 3.2
 % 4 para: 3.3 --> 3.5
 % 5 para: 3.6 (solo DISK)
 % 6 para: 3.6 --> ?

```

PRIVUSERF      = [49:01] % de un usuario privado el "header"
SENSITIVEDATAF = [42:01] % una área cancelada o no
SECURITYCODEF  = [41:02] % tipo de seguridad
                  % 0 = pública
                  % 1 = escoltada
                  % 2 = controlada
                  % 3 = privada
READWRITEF     = [37:02] % seguridad en lectura y escritura
                  % 0 = lectura y escritura
                  % 1 = escritura
                  % 2 = lectura
                  % 3 = asegurada
REBUILDFF      = [37:01] % "IAD" escritura del "header"
MODEF          = [36:03] % exclusivo, compartido,... etc
NUMROWSF       = [33:10] % número de palabras de
                  % direccionamiento de áreas
ROWSSIZEF      = [23:24] % segmento en cada área

%-----%
%          PALABRA 6 ( Información geneológica)          %
%-----%
*
SAVEFACTORF    = [47:10] % factor de resguardo
GENEALOGYF     = [37:22] % información geneológica
AVAIL0F        = [15:08] % sin uso actualmente
FILEORGANIZATIONF = [07:03] % atributo "fileorganization"
DUPLICATEDBIT  = [04:01] % el archivo es duplicado o no
                  = [03:04] % número de copias

%-----%
%          PALABRA 7 ( Información "disk pack")          %
%-----%
DISKPACKF      = [47:01] % es disk (IAD) o pack
HDRCATF        = [46:01] % el archivo es catalogado
WILLCRUNCHF    = [45:01] % si el archivo se hará "crunch"
CRUNCHEDF      = [44:01] % si el archivo está "crunch"
TIMESTAMPSYNCF = [43:01] % cuando es cambiado el archivo si
                  % será "timestam catalog bit"
NUMOPTIONALTISF = [42:11] % el número de palabras de atributos
                  % opcionales
AVAIL1F        = [31:01] % no usado actualmente
FLATHDRRLENGTHF = [30:11] % longitud en palabras del "header"
                  % para archivos permanentes
JOBORDERF      = [19:20] % solo archivos "job"

%-----%
%          PALABRA 8 ( Información del nombre del "header")%
%-----%
INTERCHANGEFF  = [47:01] % pack intercambiable
SINGLEF         = [46:01] % si todas las áreas están en el mismo
                  % pack

```

AVAIL2F = [45:01] % no usado actualmente
 CYLMODEHORF = [44:01] % localización del "cylinder" límite
 HDRTITLESIZEF = [43:08] % longitud del nombre en "bytes"
 HDRTITLEINDEXF = [35:12] % palabra índice donde empieza el
 % nombre de "header"
 BASEUNITF = [23:12] % número de unidades de la unidad base
 % de la familia
 LASTF = [11:12] % usado por "getuserdisk"

%------%
 % PALABRA 9 (Información de "EOF") %
 %------%

EOFU = [47:20] % el número de "bits" que excede
 % del último segmento completo
 EOFV = [27:28] % el número de segmentos ocupados

%------%
 % PALABRA 10 (Creación formato time(6)) %
 %------%

CREATIONDATEF = [47:16] % fecha de creación
 CREATIONTIMEF = [31:32] % hora de creación

%------%
 % PALABRA 11 (Modificación formato time(6)) %
 %------%

ALTERDATEF = [47:16] % fecha de modificación
 ALTERNIMEF = [31:32] % hora de modificación

%------%
 % PALABRA 12 (Acceso formato time(6)) %
 %------%

ACCESSDATEF = [47:16] % fecha de acceso
 ACCESSTIMEF = [31:32] % hora de acceso

%------%
 % PALABRA 13 (Palabra multiuso) %
 %------%

BOSTKNUMF = [11:12] % número de stack con el cual el
 % archivo BD sera asociado
 ALLOCATINGF = [47:01] % si "getarow" corre en este
 % directorio HDR

%------%
 % PALABRA 14 (Palabra de índice central) %
 %------%

AVAIL3F = [47:04] % sin uso actualmente
 ORIGHEADERVERSIONF = [43:04] % versión original del "header"

CONTENDORSI = [39:12] % número de "waitors" para archivo
 % exclusivo
 DPENERF = [27:12] % número de "stack" para marcas de
 % apertura
 COREINDEXF = [15:12] % índice de encabezamiento en arreglos
 % "diskfileheaders" en memoria

------%
 % PALABRA 15 (Reservada) %
 ------%

------%
 % PALABRA 16 (Sin uso actualmente) %
 ------%

------%
 % PALABRA 17 (Sin uso actualmente) %
 ------%

Las palabras de direccionamiento lineal después de el área fija de datos usa una palabra por línea con un máximo de 1000 líneas.

------%
 % PALABRAS DE DIRECCIONAMIENTO LINEAL %
 ------%

DMLOCKBITSF = [47:02] % DMS de uso
 DKIADORWLOF = [45:01] % "IAD o WRITE LOCK OUT"
 DKDELETEOF = [44:01] % si ha sido borrado
 ALLOCATEDROWF = [43:01] % el espacio es localizado por líneas
 INDEXWASSETF = [42:01] % el índice de familia fue prendido
 FAMILYINDEXF = [41:00] % índice de familia
 ACTIVERROWF = [33:01] % el miembro familiar está en una
 % línea
 seqdressf = [32:33] % sectores direccionados

Las palabras de atributos opcionales en seguida de las palabras de direccionamiento lineal. Dependiendo de la longitud del valor del atributo, una palabra contendrá el valor o la dirección donde se encuentra éste valor.

------%
 % PALABRAS DE ATRIBUTOS OPCIONALES %
 ------%

HAAIILF = [47:01] % si la palabra esta disponible
 HAHWSTOREOF = [46:05] % como están los atributos almacenados
 HANUMF = [41:10] % número de atributos
 HASIZEF = [3:16] % número de entradas
 HAINDEXF = [15:16] % palabra índice del inicio de la
 % lista en el área de datos "header"
 % para los tipos de atributos

El Área de datos "header" contiene los valores de los atributos opcionales cuyos valores no es posible que estén dentro de la palabra atributo por sí misma. Si el título del atributo está presente es almacenado en esta Área. Entre otros se encuentran los siguientes:

```

%-----%
%          PALABRA [Inicia-información + 0]          %
%-----%

```

KINDARCH = [46:08] % tipo de archivo

```

%-----%
%          PALABRA [Inicia-información + 1]          %
%-----%

```

CREATIONDATE = [47:48] % fecha de creación del archivo

```

%-----%
%          PALABRA [Inicia-información + 2]          %
%-----%

```

BLKSIZE = [47:16] % longitud del "block" del archivo
 MINRESIZEF = [31:16] % longitud mínima del reg. del archivo
 MAXRECSIZEF = [15:15] % longitud máxima del reg. del archivo

```

%-----%
%          PALABRA [Inicia-información + 3]          %
%-----%

```

SAVEFACTOR = [47:48] % factor de resguardo del archivo

```

%-----%
%          PALABRA [Inicia-información + 5]          %
%-----%

```

ROWSIZE = [47:48] % longitud del área del archivo

```

%-----%
%          PALABRA [Inicia-información + 6]          %
%-----%

```

CRUNCHED = [01:01] % está "crunch" el archivo?

```

%-----%
%          PALABRA [Inicia-información + 10]         %
%-----%

```

NUMROWS = [47:48] % número de áreas usadas por el arch.

```

%-----%
%          PALABRA [Inicia-información + 11]         %
%-----%

```

```

DSKEOFV      = [47:48] % segmentos ocupados por el archivo
%-----%
%           PALABRA [Inicio-información + 12]           %
%-----%

DSKEOFU      = [47:48] % "bits" ocupados después del último
                % segmento completo
%-----%
%           PALABRA [Inicio-información + 14]           %
%-----%

RCUNITS      = [39:01] % palabras o caracteres la unidad
                % de almacenamiento del archivo
%-----%
%           PALABRA [Inicio-información + 15]           %
%-----%

LASTACCESDATE = [47:48] % fecha del último acceso al arch.
%-----%
%           PALABRA [Inicio-información + 34]           %
%-----%

FILEORGANITATION = [47:48] % el tipo de organización del arch.

La última palabra del "header" es el "CHECKSUM", que es el fin
del "header".
%-----%
%           CHECKSUM                                     %
%-----%

```

EL SISTEMA CONTROLADOR DE RECURSOS EN DISCO PARA SISTEMAS
UNISYS SERIE "A" COMO UN MODULO DEL SISTEMA "SAYCUS"

Para los sistemas centrales con que cuenta la DGSCA se ha desarrollado un sistema para la administración de recursos y control de usuarios llamado (SAYCUS) [14]; éste sistema (Figura 12) cuenta con tres módulos principales, ALTAS, RECURSOS y SEGMENTOS.

El módulo ALTAS genera claves homogéneas y constantes, de tal manera que los otros módulos pueden hacer uso de la información asociada a cada clave de acuerdo a una estructura y características predefinidas.

El módulo RECURSOS tiene como función principal llevar una contabilidad de los recursos que se consumen por clave de usuario.

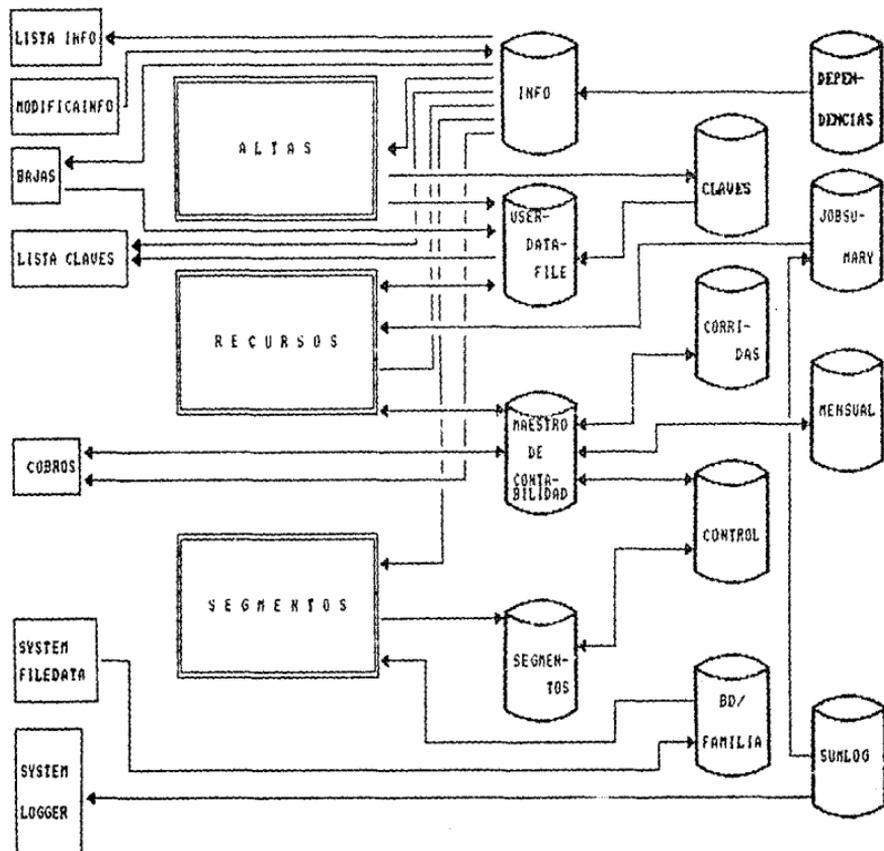
El módulo SEGMENTOS tiene como función principal obtener muestras de la situación prevaliente en los discos del sistema. Con esta información en primera instancia se alimenta al módulo de recursos; por otro lado se maneja una serie de estadísticas, reportes y procedimientos de remoción de archivos cuando se detecte alguna irregularidad.

SUBSISTEMAS Y
UTILIDADES DE SOPORTE

SISTEMAS

ARCHIVOS

ARCHIVOS TEMPORALES



SISTEMA DE ADMINISTRACION Y CONTROL DE USUARIOS "SAVCUS"

FIGURA # 12

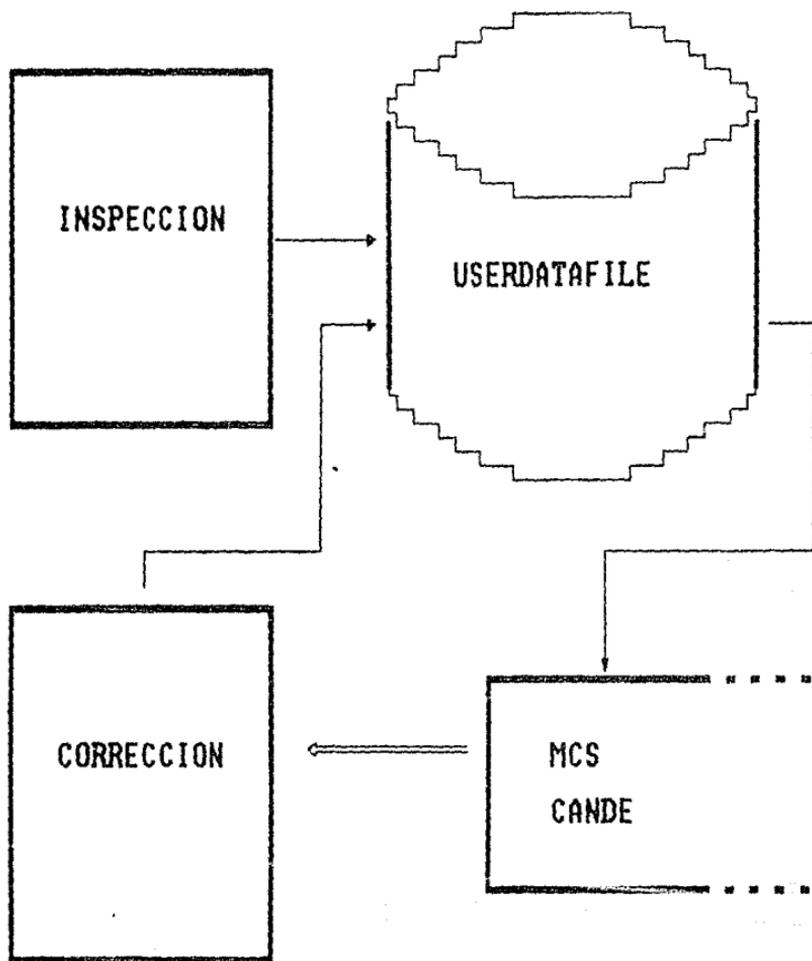
Además existen los siguientes módulos auxiliares:

El módulo LISTACLAVES que emite reportes de las claves registradas en el sistema bajo diferentes formatos.

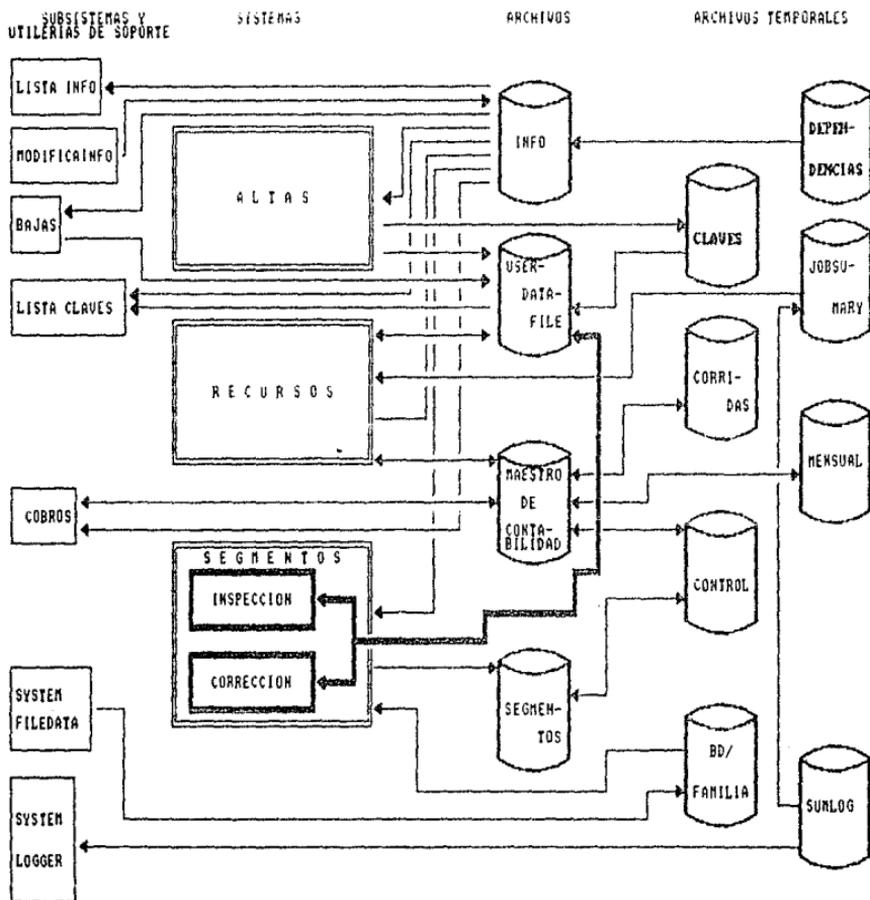
El módulo COBROS que emite reportes dándole valor económico a los diferentes campos existentes en el archivo MAESTRO de contabilidad.

Para el cumplimiento del objetivo de esta tesis se modifica al módulo del sistema SAYCUS llamado SEGMENTOS y se desarrollo el programa CRDISK que como módulo se le llamó módulo CORRECCION. Al módulo SEGMENTOS modificado se le llamó módulo INSPECCION (figura 13).

Al sistema controlador de recursos en disco para sistemas UNISYS serie "A", se le puede ver como un módulo del sistema SAYCUS (figura 14).



MODELO DEL SISTEMA "SCR"



EL SISTEMA "SCR" COMO UN MODULO DEL SISTEMA "SAYCUS"

FIGURA N 14

BIBLIOGRAFIA.

- 1) BEIZER BORIS, "Software Testing Techniques", Von Nostrand Reinhold, 290 pags.
- 2) BRINCH HENSEN PER, "Operating System Principles", Prentice Hall Clieffs N. J., 366 pags.
- 3) CLIFFORD JONES, "Software Development", Prentice - Hall, 1980, 382 pags.
- 4) COMER DOUGLAS, "Operating System Desing", Prentice - Hall inc., 1987, 567 pags.
- 5) FRELMAN PETER, "Software Systems Principles", Science Research Associates, 1975, 665 pags.
- 6) GALINDO RODRIGUEZ GUSTAVO MANUEL, "La Teoría General de Sistemas, Aplicadas a Generación, Procesamiento y Análisis de Información", Tesis (ACTUARIA), UNAM, Facultad de Ciencias - 1975.
- 7) GUTIERREZ CARDOSO PEDRO, "Sistema para la Administración y Control de Recursos en Disco en un Sistema Mayor BURROUGS", Tesis, Facultad de Ingeniería - 1987.
- 8) LUNA LOPEZ MARTHA PATRICIA, "Patología Organizacional Un Enfoque metodológico", Tesis (ACTUARIA), UNAM, Facultad de Ciencias - 1985.
- 9) MONROY ALVARADO GERMAN SERGIO, "Evolución y Perspectivas de la Metodología de Sistemas", Academia Mexicana de Ingeniería A. C., Abril/1985.
- 10) PLAUGER KERNINGHAN & PLAUSER P. J., "Software Tools", Addison Wesley Publishing co., 338 pags.
- 11) RANDALL W. JENSEN y CHARLES C. TONIES, "Software Engineering", Prentice-Hall, 1979, 580 pags.
- 12) RUSSELL LINCOLN ACKOFF, "Rediseñando el Futuro", New York: J. Wiler, 1974, 460 pags.
- 13) TURNER RAYMOND W., "Operating Systems Desing & Implementation", Macmillan Publishing co., 1980, 258 pags.
- 14) "SAYCUS" Sistema para la Administración Y Control de Usuarios, (Reporte interno OGSCA - 1981).

16) "Introduction to Large Systems", (UNISYS CORPORATION, March-1988).