UNIX

Unix es uno de los sistemas operativos más ampliamente usados en computadoras que varían desde las personales hasta las macro. Existen versiones para máquinas uniprocesador hasta multiprocesadores. Debido a su historia, que evoluciona en los Laboratorios Bell de AT&T con un simulador de un viaje espacial en el sistema solar, pasando por su expansión en universidades y la creación de las versiones más importantes que son la de la Universidad de Berkeley y el Sistema V de la misma AT&T.

8.1 Estandarización de UNIX

Debido a las múltiples versiones en el mercado de UNIX, se comenzaron a publicar estándares para que todas la s versiones fuesen 'compatibles'. La primera de ellas la lanzó AT&T llamada SVID (System V Interface Definition) que defininía cómo deberían ser las llamadas al sistema, el formato de los archivos y muchas cosas más, pero la otra versión importante, la de Bekeley (Berkeley Software Distribution o BSD) simplemente la ignoró. Después la IEEE usó un algoritmo consistente en revisar las llamadas al sistema de ambas versiones (System V y BSD) y aquellas que eran iguales las definió como estándares surgiendo así la definición 'Portable Operating System for UNIX' o POSIX, que tuvo buen éxito y que varios fabricantes adoptaron rápidamente. El estándard de POSIX se llama 1003.1 Posteriormente los institutos ANSI e ISO se interesaron en estandarizar el lenguaje 'C' y conjuntamente se publicaron definiciones estándares para otras áreas del sistema operativo como la interconectividad, el intérprete de comandos y otras. En la tabla 8.1 se muestran las definiciones de POSIX. [Tan92].

Estándard Descripción

1003.0 Introducción y repaso.

1003.1 Llamadas al sistema.

1003.2 Intérprete y comandos.

1003.3 Métodos de prueba.

1003.4 Extensiones para tiempo real.

1003.5 Lenguaje Ada.

1003.6 Extensiones para la seguridad

1003.7 Administración del Sistema.

1003.8 Acceso transparente a archivos.

1003.9 Lenguaje Fortran.

1003.10 Supercómputo.

Tabla 8.1 Los Estándares de POSIX

Al momento del auge de los estándares de POSIX desgraciadamente se formó un grupo de fabricantes de computadoras (IBM, DEC y Hewlett−Packard) que lanzaron su propia versión de UNIX llamada OSF/1 (de

Open Software Fundation). Lo bueno fue que su versión tenía como objetivo cumplir con todas los estándares del IEEE, además de un sistema de ventanas (el X11), una interfaz amigable para los usuarios (MOTIF) y las definiciones para cómputo distribuido (DCE) y administración distribuida (DME). La idea de ofrecer una interfaz amigable en UNIX no fue original de OSF, ya en la versión 3.5 de SunOS de Sun Microsystems se ofrecía una interfaz amigable y un conjunto de librerías para crear aplicaciones con interfaz gráfica técnicamente eficiente y poderosa llamada SunWindows o SunVIEW. Esta interfaz junto con sus librerías estaban evolucionando desde la versión para máquinas aisladas hacia una versión en red, donde las aplicaciones podían estarse ejecutando en un nodo de la red y los resultados gráficos verlos en otro nodo de la red, pero Sun tardó tanto en liberarlo que le dio tiempo al MIT de lanzar el X11 y ganarle en popularidad.

AT&T formó, junto con Sun Microsystems y otras compañias UNIX International y su versión de UNIX, provocando así que ahora se manejen esas dos corrientes principales en UNIX.

8.2 Filosofía de UNIX

Las ideas principales de UNIX fueron derivadas del proyecto MULTICS (Multiplexed Information and Computing Service) del MIT y de General Electric. Estas ideas son:

- Todo se maneja como cadena de bytes: Los dispositivos periféricos, los archivos y los comandos pueden verse como secuencias de bytes o como entes que las producen. Por ejemplo, para usar una terminal en UNIX se hace a través de un archivo (generalmente en el directorio /dev y con nombre ttyX).
- Manejo de tres descriptores estándares: Todo comando posee tres descriptores por omisión llamados 'stdin', 'stdout' y 'stderr', los cuales son los lugares de donde se leen los datos de trabajo, donde se envían los resultados y en donde se envían los errores, respectivamente. El 'stdin' es el teclado, el 'stdout' y el 'stderr' son la pantalla por omisión (default).
- Capacidades de 'entubar' y 'redireccionar': El 'stdin', 'stdout' y el 'stderr' pueden usarse para cambiar el lugar de donde se leen los datos, donde se envían los resultados y donde se envían los errores, respectivamente. A nivel comandos, el símbolo de 'mayor que' (>) sirve para enviar los resultados de un comando a un archivo. Por ejemplo, en UNIX el comando 'ls' lista los archivos del directorio actual (es lo mismo que 'dir' en DOS). Si en vez de ver los nombres de archivos en la pantalla se quieren guardar en el archivo 'listado', el redireccionamiento es útil y el comando para hacer la tarea anterior es 'ls > listado'. Si lo que se desea es enviar a imprimir esos nombres, el 'entubamiento' es útil y el comando sería 'ls | lpr', donde el símbolo "|" (pipe) es el entubamiento y 'lpr' es el comando para imprimir en UNIX BSD.
- Crear sistemas grandes a partir de módulos: Cada instrucción en UNIX está diseñada para poderse usar con 'pipes' o 'redireccionamiento', de manera que se pueden crear sistemas complejos a través del uso de comandos simples y elegantes. Un ejemplo sencillo de esto es el siguiente. Suponga que se tienen cuatro comandos separados A,B,C y D cuyas funcionalidades son:

A: lee matrices checando tipos de datos y formato.

B: recibe matrices, las invierte y arroja el resultado en forma matricial.

C: recibe una matriz y le pone encabezados 'bonitos'

D: manda a la impresora una matriz cuidando el salto de página, etc.

Como se ve, cada módulo hace una actividad específica, si lo que se quiere es un pequeño sistema que lea un sistema de ecuaciones y como resultado se tenga un listado 'bonito', simplemente se usa el entubamiento para leer con el módulo A la matriz, que su resultado lo reciba el B para obtener la solución, luego esa solución la reciba el módulo C para que le ponga los encabezados 'bonitos' y finalmente eso lo tome el módulo D y lo

imprima, el comando completo sería ' A | B | C | D '. ø Fácil no ?

8.3 Sistema de Archivos en UNIX

El sistema de archivos de UNIX, desde el punto de vista del usuario, tiene una organización jerárquica o de árbol invertido que parte de una raíz conocida como "/" (diagonal). Es una diagonal al revés que la usada en DOS. Internamente se usa un sistema de direccionamiento de archivos de varios niveles, cuya estructura más primitiva se le llama 'information node' (i−node) cuya explicación va más allá de este trabajo. El sistema de archivos de UNIX ofreceun poderoso conjunto de comandos y llamadas al sistema. En la tabla 8.2 se muestran los comandos más útiles para el manejo de archivos en UNIX vs. VMS.

Comando en UNIX Comando en VMS Utilidad

rm delete borra archivos

cpb copy copia archivos

mv rename renombra archivos

ls dir lista directorio

mkdir create/directory crea un directorio

rmdir delete borra directorio

ln − crea una 'liga simbolica'

chmod set protection maneja los permisos

chown set uic cambia de dueño

Tabla 8.2 Manejo de Archivos en UNIX y VMS

La protección de archivos en UNIX se maneja por medio de una cadena de permisos de nueve caracteres. Los nueve caracteres se dividen en tres grupos de tres caracteres cada uno.

RWX RWX RWX

1 2 3

El primer grupo (1) especifica los permisos del dueño del archivo. El segundo grupo especifica los permisos para aquellos usuarios que pertenecen al mismo grupo de trabajo que el dueño y finalmente el tercer grupo indica los permisos para el resto del mundo. En cada grupo de tres caracteres pueden aparecer las letras RWX en ese orden indicando permiso de leer (READ), escribir (WRITE) y ejecutar (EXECUTE). Por ejemplo, la cadena completa RWXR−XR−− indica que el dueño tiene los tres permisos (READ,,WRITE,EXECUTE), los miembros de su grupo de trabajo tienen permisos de leer y ejecutar (READ,EXECUTE) y el resto del mundo sólo tienen permiso de leer (READ). Las llamadas al sistema más útiles en UNIX son 'open', 'close' e 'ioctl'. Sirven para abrir, cerrar archivos; y establecer las características de trabajo. Por ejemplo, ya que en UNIX las terminales se accesan a través de archivos especiales, el 'ioctl' (input output control) sirve para establecer la velocidad, paridad, etc; de la terminal.

El núcleo de UNIX

El núcleo de UNIX (kernel) se clasifica como de tipo monolítico, pero en él se pueden encontrar dos partes principales [Tan92]: el núcleo dependiente de la máquina y el núcleo independiente. El núcleo dependiente se encarga de las interrupciones, los manejadores de dispositivos de bajo nivel (lower half) y parte del manejo de la memoria. El núcleo independiente es igual en todas las plataformas e incluye el manejo de llamadas del sistema, la planificación de procesos, el entubamiento, el manejo de sentilde;ales, la paginación e intercambio, el manejo de discos y del sistema de archivos.

8.5 Los procesos en UNIX

El manejo de procesos en UNIX es por prioridad y round robin. En algunas versiones se maneja también un ajuste dinámico de la prioridad de acuerdo al tiempo que los procesos han esperado y al tiempo que ya han usado el CPU. El sistema provee facilidades para crear 'pipes' entre procesos, contabilizar el uso de CPU por proceso y una pila común para todos los procesos cuando necesitan estarse ejecutando en modo privilegiado (cuando hicieron una llamada al sistema). UNIX permite que un proceso haga una copia de sí mismo por medio de la llamada 'fork', lo cual es muy útil cuando se realizan trabajos paralelos o concurrentes; también se proveen facilidades para el envío de mensajes entre procesos. Recientemente Sun Microsystems, AT&T, IBM, Hewlett Packard y otros fabricantes de computadoras llegaron a un acuerdo para usar un paquete llamado ToolTalk para crear aplicaciones que usen un mismo método de intercambio de mensajes.

8.6 El manejo de memoria en UNIX

Los primeros sistema con UNIX nacieron en máquinas cuyo espacio de direcciones era muy pequeño (por ejemplo 64 kilobytes) y tenían un manejo de memoria real algo complejo. Actualmente todos los sistemas UNIX utilizan el manejo de memoria virtual siendo el esquema más usado la paginación por demanda y combinación de segmentos paginados, en ambos casos con páginas de tamaño fijo. En todos los sistemas UNIX se usa una partición de disco duro para el área de intercambio. Esa área se reserva al tiempo de instalación del sistema operativo. Una regla muy difundida entre administradores de sistemas es asignar una partición de disco duro que sea al menos el doble de la cantidad de memoria real de la computadora. Con esta regla se permite que se puedan intercambiar flexiblemente todos los procesos que estén en memoria RAM en un momento dado por otros que estén en el disco. Todos los procesos que forman parte del kernel no pueden ser intercambiados a disco. Algunos sistemas operativos (como SunOS) permiten incrementar el espacio de intercambio incluso mientras el sistema está en uso (en el caso de SunOS con el comando 'swapon'). También es muy importante que al momento de decidirse por un sistema operativo se pregunte por esa facilidad de incrementar el espacio de intercambio, así como la facilidad de añadir módulos de memoria RAM a la computadora sin necesidad de reconfigurar el núcleo.

8.7 El manejo de entrada/salida en UNIX

Derivado de la filosofía de manejar todo como flujo de bytes, los dispositivos son considerados como archivos que se accesan mediante descriptores de archivos cuyos nombres se encuentran generalmente en el directorio '/dev'. Cada proceso en UNIX mantiene una tabla de archivos abiertos (donde el archivo puede ser cualquier dispositivo de entrada/salida). Esa tabla tiene entradas que corresponden a los descriptores, los cuales son números enteros [Deitel93] obtenidos por medio de la llamada a la llamada del sistema 'open'. En la tabla 8.3 se muestran las llamadas más usuales para realizar entrada/salida.

Llamada Función

open Obtener un descriptor entero.

close Terminar las operaciones sobre el archivo

lseek Posicionar la entrada/salida.

read,write Leer o escribir al archivo (dispositivo)

ioctl Establecer el modo de trabajo del dispositivo

Tabla 8.3 Llamadas al sistema de entrada/salida

En UNIX es posible ejecutar llamadas al sistema de entrada/salida de dos formas: síncrona y asíncrona. El modo síncrono es el modo normal de trabajo y consiste en hacer peticiones de lectura o escritura que hacen que el originador tenga que esperar a que el sistema le responda, es decir, que le de los datos deseados. A veces se requiere que un mismo proceso sea capaz de supervisar el estado de varios dispositivos y tomar ciertas decisiones dependiendo de si existen datos o no. En este caso se requiere una forma de trabajo asíncrona. Para este tipo de situaciones existen las llamadas a las rutinas 'select' y 'poll' que permiten saber el estado de un conjunto de descriptores.

OS/2

El sistema operativo OS/2 ha tenido una historia turbulenta en el seno de Microsoft e IBM, creciendo en algún tiempo bajo equipos de trabajo de ambas compañias y prosiguiendo finalmente con la última. Los objetivos para este sistema operativo eran: compatibilidad para ejecutar los programas existentes para DOS en las computadoras 80x86, ofrecer la multitarea, la facilidad de memoria virtual y servicios de red de área local [Alcal92].

10.1 Manejo de archivos en OS/2

Debido al objetivo inicial de mantener compatibilidad con DOS, las versión 1.0 de OS/2 era muy similar a la de éste sistema operativo. Posteriormente en las versiones 2.x mejoró el sistema de archivos con otras facilidades, como ofrecer dos modos de trabajo: el síncrono y el asíncrono. El modo síncrono se realiza a través del llamado a las rutinas 'DosRead' y 'DosWrite', mientras que el asíncrono se realiza por medio de 'DosReadAsync' y 'DosWriteAsync'. En el caso de que se estén ejecutando varios 'threads' de un proceso, la sincronización de las operaciones sobre archivos se puede realizar a través de semaacute;foros con la llamada a la rutina 'DosMuxSemWait'.

Respecto a los discos duros, OS/2 permite crear varias particiones en un solo disco y mantener sistemas de archivos en cada partición con su propio 'File Allocation Table' (FAT) en cada partición. A este tipo de particiones se les llama 'particiones ampliadas'. OS/2 continua usando nombres de archivos de ocho caracteres y extensiones de tres con un punto que los separa. En la tabla 10.1 se muestran algunas llamadas para la manipulacioacute;n de archivos.

Llamada Descripción

DosBufReset Graba al disco los buffers del archivo

DosClose Cierra el archivo

DosDelete Borra el archivo

DosDevIOCtl Establece parámetros de trabajo

DosMkDir Crea un directorio

DosNewSize Cambia el tamaño de archivo

DosFileInfo Obtiene información sobre el archivo

DosSetFileInfo Establece información del archivo

DosOpen Abre un archivo

DosSetFileMode Establece el modo de operación

DosRmDir Borra un directorio vacío

DosSelectDisk Selecciona un disco para trabajar

Tabla 10.1 Algunas llamadas de OS/2 para archivos

Como en UNIX y algunos otros sistemas operativos, OS/2 permite ser instalado en una partición de disco duro y dejar otras intactas para instalar otros sistemas operativos, dando así la facilidad de poder usar una misma computadora con diferentes sistemas operativos. OS/2 ofrece una interfaz gráfica para que el usuario trabaje, en particular ofrece un ícono para representar los archivos y una barra de menús para realizar operaciones sobre ellos como abrirlos, cerralos, copiarlos, etc. Si el usuario está acostumbrado a teclear comandos, entonces puede pedir una sesión de DOS para usar los comandos habituales de ese sistema operativo. En particular, en el ambiente de ventanas se tiene un ícono denominado 'Sistema OS/2' que contiene otro ícono llamado 'Unidades' y ahí existen íconos que representan el disco duro, unidades de disco flexible, etc. Para realizar copias de archivos, borrados, etc; basta con arrastrar los íconos correspondientes de/hacia el origen/destino deseado. La versión inicial de OS/2 tenía incluído el sistema Windows, pero debido a las regalías que debía pagar a Microsoft, éste fue eliminado y el usuario debe adquirirlo por separado, y configurarlo al momento de instalación.

10.2 Manejo de procesos en OS/2

OS/2 utiliza un esquema de planificación apropiativa, es decir, los procesos pueden ser suspendidos para darle su turno de ejecución a otro diferente. Los procesos pueden estar divididos en 'threads' que cuentan con sus propios registros, pila y contador de programa y todos los 'threads' de un mismo proceso comparten la memoria. Esto facilita la comunicación entre ellos y la sincronización. También es posible que un proceso genere un proceso hijo, en tal caso el hijo hereda todos los atributos del padre como son los descriptores de archivos abiertos, los valores en memoria, etc; prácticamente igual que el sistema operativo UNIX.

Otra facilidad de OS/2 es la facilidad de crear 'conductos' lo cual también es una función heredada de UNIX.

La calendarización de procesos o 'threads' se hace por prioridad y dándoles una intervalo de ejecución a cada proceso o 'thread'. Se manejan tres niveles de prioridades: procesos preferentes, procesos preferentes interactivos y procesos normales. OS/2 eleva a la categoría de prefentes a aquellos procesos que hacen mucha E/S.

Otra facilidad notable de OS/2 es la carga dinámica de librerías, que consiste en la generación de aplicaciones cuyas librerías no forman parte del coacute;digo compilado, sino que son cargadas cuando el programa es ejecutado. Esto sirve bastante sobre todo cuando las librerías son de uso común. Como se ve, esta facilidad es parecida a las del sistema operativo UNIX SunOS.

10.3 Manejo de memoria en OS/2

La versión inicial de OS/2 usaba segmentación pura debido sobre todos a las restricciones de los procesadores. Pero ya que el 80386 soportaba segmentación y paginación, IBM prometió un manejo de

memoria virtual más sofisticado. El algoritmo de sustitución de segmentos era el 'Menos Recientemente Usado'. Con el 80386 se rompió la barrera de segmentos de 64 kilobytes para ofrecer los llamados 'segmentos gigantes' que podian estar formados de varios segmentos de 64k. Debido a que OS/2 debe hacer uso del modo protegido, no se permiten algunos manejadores de extensión de memoria que violan este modo de trabajo. En particular, la versión 2.0 soporta aplicaciones que usan el modo protegido de DOS 'DOS Protect−Mode Interface', el 'Expanded Memory Specification' (EMS), o el 'Extended Memory Specification' (XMS). Los programas que usan WINMEM32.DLL no eran soportados, ni los que accesan directamente los sectores fisicos del disco duro.

Para estas fechas, es posible contar con una versión de OS/2 que maneje la memoria con paginación.

10.4 Manejo de entrada/salida en OS/2

En OS/2 se tuvo un gran problema de diseño en este aspecto, ya que se deseaba dar compatibilidad a los programas existentes para DOS. En este aspecto, existen gran cantidad de programas de DOS que accesaban directamente algunos periféricos, incluso interceptando los vectores de interrupciones para realizar un manejo propio en la entrada/salida. Todos esos programas no son soportados en forma nativa en OS/2, sino que deben ser recreados usando una facilidad llamada 'supervisor de dispositivos'.

OS/2 sigue soportando la idea de 'device drivers' en una forma parecida que en DOS. De hecho, algunos estudiosos de los sistemas operativos afirman que DOS se puede considerar como un sistema 'microkernel' por esta característica.

Para que un proceso sea candidato a manejar un dispositivo, debe informarlo a través de una llamada a 'DosMonOpen' y 'DosMonReg'. El supervisor de dispositivos usará un modelo de productor−consumidor para enviar y recibir datos con el proceso candidato. Tambieacute;n es factible que para un mismo dispositivo el supervisor envíe los datos a varios procesos interesados en leer de él. Los dispositivos en OS/2 se clasifican en aquellos orientados a bloques y aquellos orientados a caracteres. Los dispositivos orientados a caracteres se manejan de manera síncrona.

Los procesos también pueden indicar los permisos de los archivos y dispositivos para indicar quiénes pueden accesarlos al mismo tiempo. De este modo se consigue que los datos estén íntegros.

También existe el servicio de reloj, lo cual permite sincronizar algunos eventos, por medio del reloj del sistema que oscila 32 veces por segundo y otro que oscila millones de veces. Dependiendo de la precisión deseada se usa el reloj adecuado. Las llamadas para el reloj de mayor precisión se hacen en un área llamada 'segmento de información global' por medio de la rutina 'DosGetInfoSeg'.

VMS

El sistema operativo VMS (Virtual Memory System) es uno de los más robustos en el mercado, aunque es propietario de la compañia Digital Equipment Corporation. Actualmente con su versión OpenVMS 5.x existe para los procesadores de las máquinas VAX (CISC) y con el Alpha−chip (RISC). Ofrece un amplio conjunto de comandos a través de su intérprete Digital Command Language (DCL), utilidades de red (DECnet), formación de 'clusters' de computadoras para compartir recursos, correo electrónico y otras facilidades. Es un sistema operativo multiusuario/multitarea monolítico.

9.1 El manejo de archivos en VMS

El sistema de archivos de VMS es jerárquico aunque la descripción de sus senderos tiene una sintaxis propia. En la figura 9.1 se muestra un ejemplo.

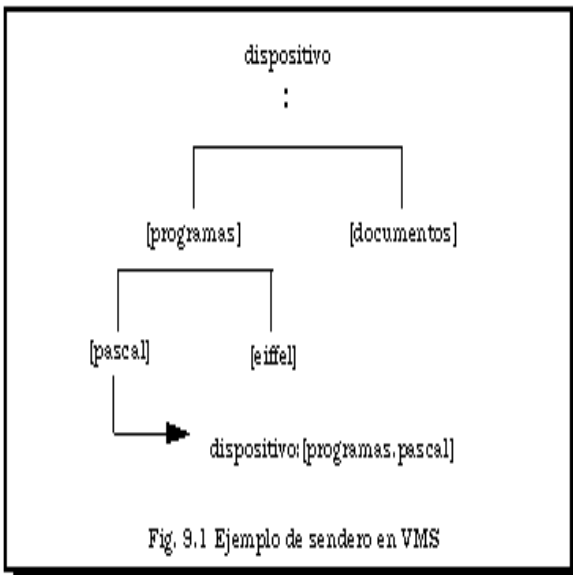

Fig. 9.1 Ejemplo de sendero en VMS

Los archivos en VMS se referencían con la sintaxis 'nombre.tipo;versión', donde 'nombre' es una cadena de caracteres alfanuméricos, 'tipo' es la extensión del archivo y se usa generalmente para describir a qué aplicación pertenece ('pas'=pascal, 'for' fortran, etc.) y 'versión' es un número entero que el sistema se encarga de asignar de acuerdo al número de veces que el archivo ha sido modificado. Por ejemplo, si se ha editado tres veces el archivo 'lee.pas', seguro que existirán las versiones 'lee.pas;1', 'lee.pas;2' y 'lee.pas;3'. De esta forma el usuario obtiene automáticamente una 'historia' de sus archivos.

La protección de los archivos se realiza mediante listas de control de acceso (Access Control Lists). Se pueden establecer protecciones hacia el dueño del archivo, hacia los usuarios privilegiados (system), hacia los usuarios que pertenecen al mismo grupo de trabajo que el dueño y hacia el resto del mundo. Para cada uno de los anteriores usuarios se manejan cuatro permisos: lectura, escritura, ejecución y borrado. Por ejemplo, el siguiente comando:

$ set protection=(S:rwed,O:rwed,G:d:W:e) lee.pas

establece que el archivo 'lee.pas' dará todos los permisos al sistema (S:rwed) y al dueño (O:rwed), mientras que a los miembros del grupo de trabajo le da permiso de borrar (G:d) y al resto del mundo permiso de ejecución (W:e). [VMS89].

Una lista de los comandos sobre archivos más útiles en VMS se mostró en la tabla 8.2, que son bastante mnemónicos en contraste con los comandos crípticos de UNIX.

En VMS, a través de su 'Record Management System' (RMS) se obtienen las facilidades para la manipulación de archivos tanto locales como en red. En el RMS, se provenn facilidades tales como: múltiples modos de acceso a archivos para lograr accesarlos en forma concurrente y permitiendo su consistencia e integridad, establecimiento de candados automáticos al momento de apertura para evitar actualizaciones erróneas y optimización interna en las operaciones de entrada/salida al accesar los archivos. En el caso de que los archivos no son locales, sino remotos, se utiliza internamente el protocolo llamado 'Data Access Protocol' (DAP).

9.2 Manejo de procesos en VMS

Soporta muchos ambientes de usuario tales como : Tiempo crítico, desarrollo de programas interactivos, batch, ya sea de manera concurrente, independiente o combinado.

El calendarizador VAX/VMS realiza calendarización de procesos normales y de tiempo real, basados en la prioridad de los procesos ejecutables en el Balance Set.Un proceso normal es referido a como un proceso de tiempo compartido o proceso background mientras que los procesos en tiempo real se refieren a los de tiempo crítico.

En VMS los procesos se manejan por prioridades y de manera apropiativa. Los procesos se clasifican de la prioridad 1 a la 31, siendo las primeras quince prioridades para procesos normales y trabajos en lote, y de la 16 a la 31 para procesos privilegiados y del sistema. Las prioridades no permanecen fijas todo el tiempo sino que se varían de acuerdo a algunos eventos del sistema. Las prioridades de los procesos normales pueden sufrir variaciones de hasta 6 puntos, por ejemplo, cuando un proceso está esperando un dispositivo y éste fue liberado. Un proceso no suelta la unidad central de procesamiento hasta que exista un proceso con mayor prioridad.

El proceso residente de mayor prioridad a ser ejecutado siempre se selecciona para su ejecución.Los procesos en tiempo crítico son establecidos por el usuario y no pueden ser alterados por el sistema. La prioridad de los procesos normales puede ser alterada por el sistema para optimizar overlap de computación y otras actividades I/O.

Un aspecto importante del planificador de procesos en VMS es la existencia de proceso 'monitor' o 'supervisor', el cual se ejecuta periódicamente para actualizar algunas variables de desempeño y para re−calendarizar los procesos en ejecución.

Existen versiones de VMS que corren en varios procesadores, y se ofrece librerías para crear programas con múltiples 'threads'. En específico se proveen las interfaces 'cma', 'pthread' y 'pthread−exception−returning'. Todas estas librerías se conocen como DECthreads e incluyen librerías tales como semáforos y colas atómicas para la comunicación y sincronización entre threads. El uso de threads sirve para enviar porciones de un programa a ejecutar en diferentes procesadores aprovechando así el multiproceso.

Servicios del Sistema para el Control de Procesos

- **Crear un proceso:**

El servicio de creado de sistema permite a un proceso crear otro. El proceso creado puede ser un subproceso o un proceso completamente independiente. (se necesitan privilegios para hacer esto).

- **Suspender un proceso:**

Esto es que le permite a un proceso suspenderse a sí mismo o a otro (también necesita tener privilegios).

- **Reanudar un proceso:**

Permite a un proceso reanudar a otro si es que este tiene privilegios para hacerlo.

- **Borrar un proceso:**

Permite que se borre el proceso mismo o a otro si es que es un subproceso, o si no tiene que tener privilegios de borrado.

- **Dar Prioridad:**

Permite que el proceso mismo se ponga prioridad o a otros, para el calendarizador.

- **Dar el modo de espera:**

Permite que el proceso escoja de dos modos: el modo por default es cuando un proceso requiere un recurso y está ocupado y espera a que esté desocupado, y el otro modo es cuando está ocupado el recurso, el proceso no espera y notifica al usuario que el recurso no se encuentra disponible en ese momento en lugar de esperar.

- **Hibernar:**

Es cuando un proceso se hace inactivo pero está presente en el sistema. Para que el proceso continue necesita de un evento para despertar.

- **Wake:**

Esto activa a los procesos que estan hibernando.

- **Exit:**

Es cuando se aborta un proceso.

- **Dar nombre al proceso:**

Este puede dar un nombre al proceso mismo o cambiarlo (el PCB contiene el nombre).

9.3 Manejo de memoria en VMS

El sistema operativo VMS utiliza un esquema de manejo de memoria virtual combinado de segmentación paginada que se describe exactamente como se vió en el capítulo de administración de memoria de este trabajo. Lo novedoso en VMS es que usa un doble esquema de paginación cuando las páginas se van a intercambiar de memoria RAM hacia disco duro. En primer lugar, cuando una página necesita cargarse a RAM ésta se carga junto con varias páginas que están adyacentes, justificando esto por medio de la teoría del conjunto de trabajo que especifica que es muy probable que las referencias a memoria en el futuro inmediato caerán precisamente en esas páginas. De este modo, se tiene un doble algoritmo: al hecho de cargarse las páginas cuando se necesitan se le llama 'paginación por demanda' y al hecho de traerse las otras páginas del conjunto de trabajo por anticipado se le llama 'paginación anticipada'.

9.4 El manejo de entrada/salida en VMS

En VMS, se usan nombres 'lógicos' para describir a los dispositivos existentes en el sistema. Un concepto importante tanto en archivos como en dispositivos es el 'User Identification Code' (UIC) que permite establecer protecciones adicionales a los ACL. En los dispositivos se manejan cinco tipos de permisos: leer, escribir, ejecutar, borrar y controlar. No todos los permisos se aplican a todos los dispositivos. El permiso de 'control' no se maneja explícitamente sino que se otorga por omisioacute;n al dueño y al sistema. Los permisos de los discos, unidades de cinta y otros dispositivos son establecidos por el administrador del sistema.

Los dispositivos reciben nombres 'lógicos', por ejemplo, para una unidad de cinta el nombre puede ser 'MTA0'.

System Interface" (SCSI) que son ampliamente usados en diversas plataformas. El intercambio de datos entre la unidad central de proceso y los periféricos se lleva a cabo a través de los 'buses' normalizados UNIBUS y MASSBUS.

WINDOWS NT

Windows NT es el nuevo sistema operativo de Microsoft. Fue diseñado para tomar ventaja de todo el poder que ofrecen los procesadores más avanzados de Intel, así como algunos de los procesadores RISC. Windows NT es la respuesta de Microsoft a UNIX. NT ofrece los mismos servicios que UNIX, interopera con redes UNIX pero remplaza los comandos criacute;pticos de UNIX, su estructura de archivos ARCANE y la mezcla de GUIs con una simple y estandarizada interfaz para el usuario como lo es Windows. Además, NT tiene las características que originalmente iba a tener el OS/2: un avanzado sistema operativo de 32 bits y compatibilidad con Windows GUI, además de soportar las aplicaciones hechas en DOS pero liberándose de las limitaciones de éste. Las características de diseño que hacen de Windows NT un sistema operativo avanzado son [MJS Jul−Ago92]:

- Extensibilidad: El código podrá ser alterado (crecer o cambiar) de manera sencilla según cambien las necesidades del mercado.
- Portabilidad: El código podrá utilizar cualquier procesador sin que esto afecte su desempeño de manera negativa.
- Confiabilidad y robustez: El sistema deberá auto−protegerse tanto de los malos funcionamientos internos como de los externos. Así mismo se deberá comportar de manera predecible en cualquier momento y las aplicaciones no deberán afectar su funcionamiento en forma negativa.
- Compatibilidad: El sistema se extendará hacia la tecnología existente pero al mismo tiempo sus API y sus UI serán compatibles con los sistemas ya existentes de Microsoft.
- Multiprocesamiento y escalabilidad: Las aplicaciones podrán tomar ventaja de cualquier computadora y los usuarios podrán correr las mismas aplicaciones tanto en una computadora de un procesador como en una multiprocesador.
- Cómputo distribuído: NT será capaz de repartir sus tareas computacionales a otras computadoras en la red para dar a los usuarios más poder que el que tenga cualquier computadora por sí misma en la red. Podrá usar computadoras tanto local como remotamente de manera transparente al usuario (efecto de sinergia en red).
- Desempeño: El sistema debe responder y ser lo más rápido posible en cada plataforma HW.
- Compatibilidad con POSIX: POSIX (Portable Operating System based on UNIX) es un estándar especificado por el gobierno de los EU, el cual deberán de cumplir todos los contratos en el área computacional que sean vendidos a ese gobierno. NT puede proporcionar un ambiente opcional para la ejecución de aplicaciones POSIX.
- Seguridad certificable por el gobierno de EU: El gobierno de EU estableció niveles de seguridad computacional como guías a cumplir para todas las aplicaciones gubernamentales. El rango de estos niveles va desde la D (menor) hasta la A (mayor), en donde la C y B tienen varios subniveles. NT puede soportar el C2 (el dueño del sistema tiene el derecho de decidir quién tiene permiso de acceso y el sistema operativo puede detectar cuándo los datos son accesados y por quién) pero en futuras versiones puede ser mejorada para alcanzar niveles de seguridad más altos.

11.1 Características de WindowsNT

Un sistema operativo es un programa complejo que necesita un modelo unificado para asegurarse que el sistema puede acomodar sus características propias sin que éstas alteren el diseño. El diseño de Windows NT fue guiado por una combinación de diversos modelos que fueron unidos en Windows NT. Los razgos característicos de NT son [LenF93]:

- Direccionamiento de 32−bits.
- Soporte de memoria virtual.
- Preemptive multitasking.
- Soporte para multiprocesador.
- Arquitectura cliente/servidor.
- Seguridad e integridad del sistema.
- Compatibilidad con otros Sistemas Operativos.

- Independencia de plataformas.
- Networking (Interoperatividad).

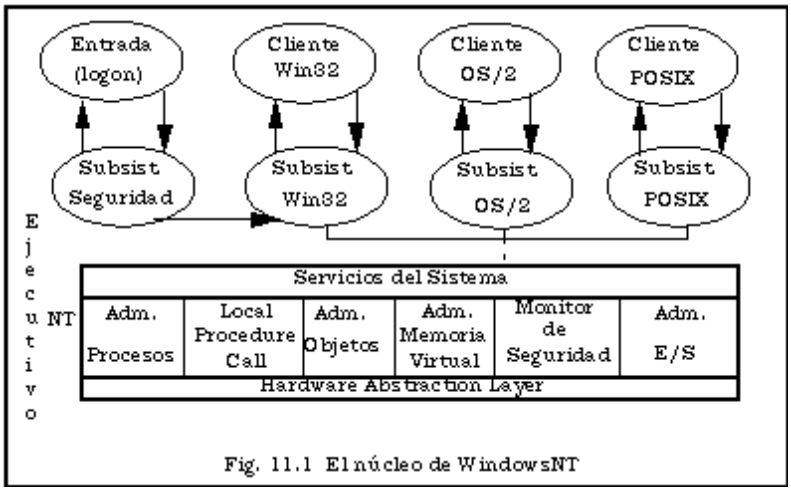

Fig. 11.1 El núcleo de WindowsNT

11.2 El núcleo de WindowNT

El núcleo es la base del sistema operativo, en donde reside el ejecutivo del NT por medio del cual se realizan las siguientes operaciones:

- Entradas y salidas de tareas al sistema.
- Proceso de interrupciones y excepciones.
- Sincronización de los multiprocesadores.
- Recuperación del sistema después de una caída.

11.2.1 Entradas y salidas de tareas al sistema

Cada objeto de tipo tarea es creado como una respuesta a una requisición de la aplicación que contenga una mini−tarea consistente en una llamada al kernel que es usada para iniciar la ejecución de una tarea más larga, cada una de las tareas puede encontrarse en los estados de ejecución, espera en cola, espera por recursos, lista para ejecución o finalizada. El kernel cuenta con un módulo llamado despachador que se encarga de permitir la entrada de los procesos y de darlos por terminados. El despachador igualmente examina la prioridad de los procesos para determinar en qué orden van a ser ejecutados; suspendiendo y activando los procesos.

11.2.2 Proceso de interrupciones y excepciones

En Windows NT se manejan las interrupciones como en cualquier sistema operativo. La llegada de señales por el bus debido a fallas de los programas o por peticiones de entrada/salida de los periféricos son atrapadas por el núcleo. En la Figura 11.1 se pueden observar las partes del núcleo de WindowsNT. El paso de los subsistemas de OS/2, POSIX y Win32 hacia los servicios del sistema se hace a través de mensajes y de atrapado de interrupciones.

11.2.3 Sincronización de los multiprocesadores.

Esta característica asegura que sólo una tarea puede accesar un mismo recurso a la vez. En un sistema basado en multiprocesadores con memoria compartida, dos o más procesadores pueden estar ejecutando tareas que necesitan accesar la misma página de memoria o realizar operaciones sobre un mismo objeto. El núcleo y el ejecutivo de NT proveen mecanismos para asegurar la integridad del sistema a través de la sincronización; en el caso del kernel la sincronización es manejada a través de candados colocados en puntos críticos de las instrucciones del nivel despachador, de esta manera, ningún otro procesador puede ejecutar código o accesar

datos protegidos por uno de los candados de tipo spin hasta que éste es liberado. El ejecutivo del NT realiza la sincronización a través de la familia de los objetos de sincronización.

11.2.4 Recuperación del sistema

La última función del kernel consiste en la recuperación del sistema en caso de una caída. Cuando existe una falla de alimentación en un sistema NT se dispara una interrupción de alta prioridad la cual dispara a su vez una serie de tareas diseñadas para preservar la integridad del sistema operativo y de los datos tan rápido como sea posible.

El mcro−núcleo de WindowsNT contiene una capa de abstracción del hw que es el límite entre el ejecutivo del NT y el hw específico de la computadora. NT fue diseñado de tal manera que los cambios de código son mínimos para ser acoplados a las diferentes plataformas de hw tomando como ejemplo los sistemas UNIX.

11.2.5 Arquitectura cliente/servidor

Windows NT tiene dos modos de operar, modo usuario y modo privilegiado (kernel). Programas de aplicaciones como una base de datos, una hoja de cálculo, o un sistema de reservaciones de un hotel, siempre son ejecutadas en modo usuario. El ejecutivo de NT es el corazón del sistema. El ejecutivo de NT realiza tareas como el manejo de entradas y salidas, la memoria virtual, y todos lo procesos, además de controlar las ligas entre NT y el hardware de la computadora. El ejecutivo de NT es ejecutado en modo kernel, el cual es una modo de alta seguridad libre de interferencias de los procesos de los usuarios. Consúltese la figura 11.1.

El modo usuario, hay también los llamados subsistemas protegidos. Un ejemplo de estos es el Win32 API. Usando esta API los programadores no tienen porque preocuparse acerca del hardware donde el programa va a ser ejecutado y por otro lado protege al sistema de aquellos programadores que traten de modificar su memoria y para hacer que falle el sistema. Adicionalmente el API tiene reglas de seguridad que protegen a los otros subsistemas de interferencias entre ellos.

En el ambiente de NT los programas de aplicación de los usuarios son los clientes y los subsistemas protegidos son los servidores. Las aplicaciones (clientes) mandan mensajes a los subsistemas protegidos a través del ejecutivo de NT, el cual provee un conjunto de servicios compartidos para todos los servidores. Y a su vez los servidores contestan a los clientes de la misma forma.

En NT, los servidores ejecutándose en un procesador local pueden mandar mensajes de sus clientes a otros servidores que estén siendo ejecutados en procesadores remotos sin que se necesite que el cliente sepa algo de los servidores remotos.

El modelo cliente/servidor hizo que el sistema operativo fuera más eficiente eliminando recursos duplicados y elevó el soporte que ofrece el sistema operativo para multiproceso y redes. Esta arquitectura permite que otros API's sean añadidos sin tener que aumentar un nuevo ejecutivo de NT para su manejo. Por otro lado cada subsistema es un proceso separado en su propias memorias protegidas, así, si uno de los subsistemas falla no hace que todo el sistema falle también.

El ejecutivo NT (Ver figura 11.1) es un sistema operativo completo que no cuenta con interfaz y está compuesto de cuatro capas, siendo éstas las siguientes:

- Servicios del sistema: son las llamadas al sistema que sirven como medio de comunicación entre los modos de los procesos y los componentes del ejecutivo. La manera en que interactúan los dos componentes anteriormente mencionados es a través de llamadas al sistema; en otras palabras los servicios del sistema son el API para el modo de usuario.
- Componentes del ejecutivo: el ejecutivo de NT tiene seis componentes primarios cada uno de los

cuales realiza el siguiente conjunto de operaciones críticas del sistema: manejador de objetos, monitoreo de la seguridad del sistema, manejador de procesos, facilidad para la llamada de procesos locales, manejador de la memoria virtual y manejador de las entradas y salidas.

11.2.6 Manejador de Objetos

Este módulo es el responsable de crear, manejar y borrar los objetos del ejecutivo de NT, siendo este tipo de objetos procesos y datos, así como objetos propios de los niveles del sistema.

Existen dos tipos principales de objetos: los objetos ejecutivos que son creados dentro del ejecutivo y que son accesibles para el ejecutivo y los subsistemas protegidos, y la otra clase se objetos que son sólo accesibles por el ejecutivo y que se llaman objetos del kernel y que sólo pueden ser modificados dentro del mismo. El manejador de objetos tiene las siguientes funciones:

- Asignar memoria.
- Asigna un descriptor de seguridad del objeto el cual permite o prohíbe el acceso a dicho objeto.
- Coloca el nombre del objeto dentro de la posición adecuada en el directorio de objetos.
- Crea y regresa un "manejador" o apuntador al objeto el cual elimina la necesidad de llamar al objeto por su ubicación.

11.2.7 Monitor de la seguridad del sistema

El monitor de la seguridad del sistema trabaja en conjunción con el manejador de objetos para proveer un mecanismo de control de acceso a los objetos mismos.

La información de control de acceso esta atada a cada objeto, dentro de esta información cada objeto maneja una lista de control de accesos (ACL) en esta lista cada objeto registra los permisos de acceso con los que cuenta su creador pero siempre manteniendo la peculiaridad de que el dueño de dicho objeto puede cambiar los permisos.

11.3 Manejo de archivos en WindowsNT

En lo relativo al sistema de archivos de NT tiene compatibilidad con los siguientes sistemas de archivos:

- FAT (DOS)
- HPFS (OS/2)

La migración de archivos desde DOS o Windows 16−bits al sistema manejador de archivos de Windows NT (NTFS) puede dar como resultado que el sistema y los usuarios confundan la seguridad de estos archivos pero esto tiene una fácil solución con la intervención del administrador.

La facilidad de soportar diferentes tipos de archivos ayuda a lograr una característica llamada 'personalidad del sistema operativo'. Esta característica consiste en la facilidad de que un sistema operativo soporte la ejecución de aplicaciones creadas para un sistema operativo diferente. Como se puede observar en la figura 11.1, los susbsistemas de Win32, de POSIX y de OS/2 complementan el logro de diferentes personalidades.

11.4 Manejo de procesos en WindowsNT

En la arquitectura de NT los procesos son segmentados en componentes más pequeños llamados 'threads'. WindowsNT soporta varias tareas al mismo tiempo. Existen dos tipos de multitarea, el apropiativo (preemptive) y el no apropiativo (no preemptive). Con la multitarea apropiativa la ejecución de un 'thread' puede ser suspendida después de un tiempo determinado (time slice) por el sistema operativo para permitir

que otro thread sea ejecutado. Mientras que con la multitarea no apropiativa, es el thread el que determina cuándo le regresará el control al sistema operativo para permitir que otro thread sea ejecutado. NT así como OS/2 y UNIX usan preemptive multitasking para soportar la ejecución "simultánea" de varios procesos.

11.4.1 Manejador de Procesos.

El manejador de procesos es un componente ambiental que crea y destruye procesos y tareas, como el manejador de objetos, el manejador de procesos ve los procesos como si fueran objetos en efecto el manejador de procesos puede ser considerado como un instancia específica del manejador de objetos porque dicho manejador crea, maneja y destruye un sólo tipo de objetos.

Se puede únicamente distinguir una funcionalidad adicional al manejador de objetos con la que cuenta el manejador de procesos que consiste en el manejo del estadío de cada uno de los procesos (ejecutar, suspender, reiniciar, terminar una tarea).

Las llamadas a procedimientos locales (LPC, ver figura 11.1) son usadas para pasar mensajes entre dos diferentes procesos corriendo dentro de un mismo sistema NT, estos sistemas fueron modelados utilizando como modelo las llamadas a procedimientos remotos (RPC); los RPC consisten en una manera estandarizada de pasar mensajes entre un cliente y un servidor a través de una red. Similarmente los LPC's pasan mensajes de un procedimiento cliente a un procedimiento servidor en un mismo sistema NT.

Cada proceso cliente en un sistema NT que tiene capacidad de comunicación por medio de LPC's debe tener por lo menos un objeto de tipo puerto asignado a él, este objeto tipo puerto es el equivalente a un puerto de TCP/IP en un sistema UNIX.

11.4.2 Soporte para multiprocesador

Existen dos tipos de multiproceso, el asimétrico y el simétrico. En el asimétrico hay un procesador (maestro) en el cual se ejecuta el sistema operativo y los demás (esclavos) donde se ejecutan las demás tareas. La ventaja de éste es que al aumentar más procesadores se tiene que hacer un cambio mínimo y fácil para el manejo de éstos y en general se eliminan muchos problemas de integridad de datos. La gran desventaja es que al haber sólo una copia del sistema operativo en un sólo procesador (maestro) cuando este procesador falla todo el sistema falla porque todos los recursos que son manejados por el sistema operativo no pueden ser accesados.

En el simétrico se ejecuta el sistema operativo − o una gran parte de él − en cualquiera de los procesadores disponibles y todos ellos tienen acceso a los recursos a menos que cada recurso sea asignado a un procesador específico. Aunque es mas difícil de implementar tiene muchas más ventajas. Primero, este tipo de sistemas tienden a ser más eficientes porque las tareas tanto del sistema operativo como de los usuarios pueden ser distribuídas en forma balanceada a todos los procesadores. Debido a que las demandas del sistema operativo pueden ser repartidas a todos los procesadores, el tiempo de inactividad de un procesador mientras otro está sobretrabajando es mínimo. Segunda, si un procesador falla, es posible que sus tareas sean repartidas entre los demás y no es necesario que todo el sistema sea parado o que falle el sistema. Y finalmente, la portabilidad del sistema es mayor debido a que no sigue la arquitectura de mastrer/slave. NT implementa este modelo de multiproceso.

11.5 Seguridad e integridad del sistema

Seguridad en relación a Windows NT se refiere a dos cosas básicamente:

- El control total en el acceso al sistema y a los archivos o subdirectorios que hay en el sistema. (Control de acceso y seguridad del sistema)

- La protección individual de los procesos y del sistema operativo, para que en caso de un bug o de un programa destructivo no pueda hacer que el sistema se caiga o afecte a otros programas o aplicaciones. (Integridad del sistema)

En el primer punto, el control sobre el acceso al sistema se refiere al manejo de user names y passwords para poder accesar al sistema operativo, de esta manera se mantienen a los usuarios sin autorización fuera del sistema. El siguiente nivel de seguridad en cuanto a este punto se refiere, son los privilegios que tiene un usuario, todos los usuarios o grupos de usuarios a los directorios y archivos del sistema, p.e. el acceso a los archivos del sistema de NT está estrictamente limitado al administrador del sistema, mientras que las aplicaciones comunes como lo son hojas de cálculo o procesadores de palabras pueden ser accesados por todos los usuarios.

El segundo punto trata acerca de la integridad del sistema, la pérdida de información en sistemas operativos para un sólo usuario no es tan grave comparada con la de los sistemas operativos para redes, en los cuales se pudo haber perdido información que tardará horas en ser recuperada. NT tiene amplias facilidades para asegurar la integridad del sistema para hacer correr a NT bajo condiciones difíciles, así como para recuperar el sistema de manera rápida y sencilla.

11.5.1 Control de Acceso y Seguridad del sistema.

Windows NT cuenta con un extenso sistema de control de seguridad para el acceso a archivos. El propósito de la seguridad en Windows NT es brindarle el acceso sólo a aquellos usuarios que están autorizados, controlar el acceso concurrente a archivos, a los directorios y a los recursos del sistema.

La seguridad en los sistemas Windows NT debe ser configurada por el administrador del sistema siendo necesario para todos los sistemas un administrador (incluyendo los sistemas monousuarios). El administrador establece los nombres de usuario, crea grupos de usuarios, asigna los usuarios a los grupos, controla los passwords, permite los niveles de acceso a las funcionalidades del sistema; en pocas palabras el administrados controla todos los puntos de acceso al sistema.

El administrador puede controlar el acceso específico a ciertas funciones del sistema, especialmente aquellas que afectan el funcionamiento del mismo, este sistema de control es llamado la política de derechos del usuario. De esta manera el administrador a través de esta política puede controlar las labores que efectúa un usuario tanto local como remotamente.

11.5.2 Integridad del sistema

Entendemos por integridad del sistema a la habilidad del mismo de permanecer activo cuando una de sus aplicaciones falla. Windows NT está diseñado para prevenir la caída catastrófica del sistema en caso de que algunas de sus aplicaciones falle y para esto establece los siguientes cuatro mecanismos de protección de memoria:

- Espacio de direcciones separado: cada proceso maneja sus propias direcciones virtuales y el sistema prohíbe el acceso a espacios de memoria de otros procesos.
- Modos de Kernel y usuarios separados: todas las aplicaciones corren en modo de usuario pro lo tanto está prohibido el acceso o modificación del código o datos del sistema que residan en el kernel.
- Banderas de páginas: cada página de la memoria virtual tiene una bandera la cual determina cómo puede ser accesada en modo usuario y en modo kernel.
- Seguridad de los Objetos: el manejador virtual de la memoria crea un tipo especial de objeto llamado objeto−sección el cual funciona como una ventana hacia la memoria virtual, por lo tanto cada vez que un proceso accesa un objeto−sección el sistema determina si el proceso tiene los permisos de lectura y/o escritura sobre éste.

Dentro de la integridad del sistema Windows NT establece políticas y procedimientos de protección el acceso a recursos de esta manera protege a los procesos de caer en estados muertos cuando compiten por recursos.

11.6 Manejo de memoria en WindowNT

Como se mencionó al comienzo de este capítulo, WindowsNT es un sistema operativo de 32 bits con la facilidad del manejo de memoria virtual. A continuación se verán a detalle las caracteristicas ofrecidas en este S.O.

11.6.1 Direccionamiento de 32 bits

Este tipo de direccionamiento tiene varias ventajas. Primera, eliminando la memoria segmentada, el desarrollo de software es mas fácil y rápido. Los programadores no necesitarán estar familiarizados con los requerimientos de memoria de sus aplicaciones. Además, el direccionamiento de 32−bits mejora el desempeño del sistema eliminando parte del 'overhead' del software para el manejo de la memoria. Quitando los manejadores de memoria elimina también las incompatibilidades en hw y sw, lo que significa que la instalación y configuración de NT es tan simple y fácil como la de DOS o la de 16−bit Windows.

La ventaja final del direccionamiento de 32−bits es un incremento considerable en el tamaño disponible para los programas y los datos. NT soporta un máximo de 4 Gigas de programas y sistema, lo que es n veces más grande de lo que soporta el DOS o el mismo 16−bit Windows, ésta es una gran ventaja si se van a manejar aplicaciones complejas que procesan archivos muy grandes (como los de procesamiento de imágenes) o a aplicaciones orientadas a transacciones críticas, las cuales serían imposibles de implementar en DOS y Windows.

11.6.2 Soporte de memoria virtual

El direccionamiento de 32−bits le da a las aplicaciones acceso a 4 Gigabytes de memoria, de los cuales 2 Gigas están reservados para uso del sistema operativo, y que son más que suficientes para casi cualquier aplicacion concebible.

Cuando el usuario o el administrador instala por primera vez NT, el NT setup program checa cuánto espacio en RAM y en DD está disponible. Basándose en esto NT crea un swap file, el cual debe de ser al menos del mismo tamaño del RAM. El manejador de memoria virtual de NT realiza dos tareas básicas. Primero, maneja los datos guardados en disco y mapea las direcciones de los datos que están en disco al espacio de direcciones en 32−bits lineales. Las aplicaciones pueden hacer operaciones con los datos sin importar la localización física de ellos (disco o RAM).

Segundo, el manejador de memoria virtual mueve algunas porciones del RAM al swap file cuando los procesos tratan usar más RAM del que está disponible. En este caso, las partes inactivas de RAM son movidas temporalmente al swap file hasta que son necesitadas en RAM, el tamaño de página con que se hace el swap de RAM a disco es de 4 K. Es decir, se usa paginación por demanda.

11.6.3 Manejador de memoria virtual

El manejador de memoria virtual (MMV) de los sistemas NT realiza tres funciones escenciales: el manejo del espacio virtual de cada uno de los procesos, el espacio de memoria compartida entre los procesos, la protección de la memoria virtual de cada proceso. Dentro del manejo de la memoria virtual de cada proceso se realizan las siguientes tareas :

- Reservar y liberar la memoria virtual
- La lectura y escritura de páginas de memoria virtual

- El establecimiento de candados en las páginas seleccionadas de la memoria virtual lo cual significa, el mantiene unas páginas de la memoria real sin ser intercambiadas a disco (swap).
- El encadenamiento de la información dentro de las páginas de memoria virtual protegida
- El vaciado de las páginas virtuales a disco

El manejador de memoria virtual permite que uno o varios procesos compartan las mismas páginas de memoria virtual, de tal manera que dos o más procesos puedan tener manejadores a la misma área de memoria virtual. El MMV tiene una característica singular que consiste en el poder direccionar una pequeña área del espacio de memoria virtual de otro proceso, esta ventana del espacio total de memoria virtual de procesos es llamada vista y ésta permite que un proceso trabaje con muchas porciones pequeñas de largos espacios de memoria virtual para crear su propio espacio de memoria virtual.

11.6.4 Memoria protegida

El manejador de memoria de Windows NT permite proteger ciertas regiones de memoria de accesos inadvertidos o deliberados realizados por otros procesos. El MMV es responsable de hacer el mapeo entre las direcciones de memoria virtual y las direcciones de hw específicas asegurando de esta manera que dos procesos no puedan accesar una misma página de memoria. El MMV utiliza técnicas de manejo de memoria en hw que están disponibles en la computadora host y de esta manera establece la protección a cada una de las páginas. Todas las protecciones de las páginas no están provistas por el hw por lo que Windows NT tuvo que hacerlo a través del sw definiendo páginas individuales de memoria como de lectura y escritura, sólo lectura, sólo escritura, de ejecución o sin acceso.

Para aplicaciones que utilizan largos sectores de memoria Windows NT introduce un concepto llamado " bookend " el cual consiste en un página que marca el final del código o de datos; cuando el proceso llega a una de estas páginas llamadas páginas guardia sabe que se encuentra en un estado fuera de memoria y solicita memoria adicional al MMV protegiendo de esta manera la caída de la aplicación.

En situaciones donde dos o más procesos necesitan accesar la misma región de memoria, el MMV realiza una copia de la página para que el segundo proceso lo utilice estableciendo de esta manera el mecanismo de protección de páginas y a su vez estableciendo la memoria compartida.

Cuando un proceso quiere modificar ciertos datos en la memoria compartida debe primero modificarlo en su copia de las páginas de memoria y después notificar al MMV que necesita actualizar los cambios en las páginas de los demás procesos, previniendo de esta manera que el proceso modifique directamente las páginas de memoria que no le pertenecen.

11.7 Manejo de entrada/salida en WindowsNT

En Windows NT el manejador de las entradas y salidas debe ser considerado más bien como un despachador de las entradas y salidas al sistema, puesto que este módulo establece la comunicación entre los subsistemas protegidos y los controladores de dispositivos por otro lado.

Cuando cualquier aplicación solicita un servicio de entrada/salida, el manejador de entradas/salidas convierte la solicitud en un IRP (I/O request packet) e identifica el manejador de dispositivos adecuado para llevar acabo la requisición hecha por el proceso. Cada uno de los manejadores de dispositivos recibe el paquete de datos y lo procesa mandando el resultado hacia el manejador de entradas y salidas o si es necesario mandando su resultado al siguiente manejador de dispositivos para que procese su resultado, teniendo como destino final, el paquete de datos, el manejador de entradas y salidas. Después de que una requisición ha sido pasada a un manejador de dispositivos éste es responsable del control de las mismas a través de sistemas de colas.

11.8 Compatiblidad con otros Sistemas Operativos

Una de los más grandes cualidades dentro de Windows NT es la capacidad de soportar múltiples sistemas operativos. Un sistema NT puede simultáneamente correr la mayoría de los programas de DOS, Windows 16−bits, y la mayoría de las aplicaciones orientadas a caracteres de OS/2 versión 1.x y las que cumplan con el estándar POSIX

11.8.1 Independencia de plataformas

El propósito de Windows NT es el de ser un sistema operativo diseñado para correr en distintas plataformas soportando los siguientes procesadores:

- La familia Intel x86
- De motorola 680x0
- El MIPS 400
- El ALFA de Dec.
- El HP−PA de Hewlett Packard
- Los SPARC RISC processors de Sun Microsystems.
- El RS/6000 de IBM
- Una futuras versiónes del Powerpc (Apple, IBM y Motorola)

La independencia de plataforma está basada en el concepto de el desarrollar un kernel específico para cada uno de los distintos procesadores que sirva de interfaz entre el hardware específico y las llamadas al sistema de NT.

11.9 Interoperatividad (Networking)

Windows NT ofrece cuatro tipos diferentes de soporte de redes:

- Punto a punto: En las conexiones punto a punto con otros sistemas Windows NT y Windows para grupos.
- Interoperabilidad: con otros sistemas operativos orientados a red como lo son : DEC Pathworks, Novell Network, BanyanVINES a través de la arquitectura de sistemas abiertos de Windows (WOSA) , al igual que sistemas UNIX basados en TCP/IP.
- SNA: Conexiones a host basados en redes SNA a través de una propia versión de los servidores de comunicaciones de Microsoft DCA.
- Soporte para redes Microsoft basadas en sistemas operativo de red LAN Manager.