UNIVERSIDAD NACIONAL

AUTONOMA DE

MEXICO

ESTRUCTURA INTERNA DEL SISTEMA OPERATIVO OS/2

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN MATEMATICAS APLICADAS Y COMPUTACION

PRESENTA

ELSA FRIAS SILVER

1993

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

I N D I C E

I N T R O D U C C I O N

Las funciones de una computadora, en cuanto a programación se refiere, son: almacenar, procesar y recuperar información, se clasifican en dos tipos de programas: del sistema y de aplicación, siendo fundamental el sistema operativo, que es el encargado de controlar los recursos de la computadora y brindar la base para poder codificar los programas de aplicación.

OS/2 conforma una nueva generación de Sistemas Operativos y en consecuencia uno de los mayores atractivos del sistema es utilizar hasta 16MB de memoria real.

En 1981 IBM estandariza tanto la arquitectura (conjunto de instrucciones, organización de memoria, entrada y salida de datos, etc.) como el sistema operativo PC DOS (MS DOS), que sirvió de base para la programación de la gran variedad de aplicaciones, sin embargo dicho sistema tenía una deficiencia, la limitación de memoria de 640K.

OS/2 resuelve problemas de MS DOS como lo es el aislamiento de programas, lo que significa que, un error de programación modificaría sin desearlo varios programas, con OS/2 si se hace una referencia accidental (es decir, no autorizada) se detecta y termina, evitando un daño mayor, garantizando de esta forma el adecuado funcionamiento del sistema.

Entre los beneficios funcionales más importantes del OS/2 se encuentra el proceso multitarea, es decir, realiza varios

procesos a través de una sóla llamada.
OS/2 es flexible y presenta a los usuarios un ambiente adecuado, administrando los recursos de tal forma que cada quien lleve a cabo sus labores de la mejor manera posible.

Este trabajo presenta un sistema operativo para computadoras personales, como solución a deficiencias de otros sistemas operativos en especial el sistema MS DOS, partiendo de la limitación de memoria de 640K, pretendiendo además se aproveche como libro de texto para un curso de Sistemas Operativos, como caso de estudio.
Se intenta a la vez cubrir las deficiencias y limitaciones de información en español, que existen sobre el tema, proporcionando de esta forma una guía básica para las personas interesadas en profrundizar en el manejo de OS/2.

En forma general el trabajo está dividido en cuatro partes: el Sistema de Archivos, el Sistema de Procesos, Técnicas de Administración de la Memoria y el Software que gira en torno a OS/2.

Algunos aspectos importantes sobre cada capítulo son:
1. Es una descripción histórica del desarrollo del sistema operativo OS/2.
2. Menciona por que el surgimiento de este sistema, comenzando p[las limitaciones que existieron en MS DOS que influyeron de

manera significativa al surgimiento de OS/2: la administración de la memoria: el control de entradas y salidas: y a grandes rasgos la arquitectura del sistema, su administración de recursos y como obtener la extensión del sistema.

3. Describe la compatibilidad con la familia de procesadores de 808X, los diferentes modos: real y protegido del 80286, el modelo de protección, la memoria real, protecciones de almacenamiento y de entrada y salida de datos: así como el entorno de compatibilidad del DOS.

4.Señala las diferentes formas de administración de la memoria, como son: paginación, carga por demanda, segmentación, entre otros.

5. Trata sobre las aplicaciones que se hacen en programación, además de las características primordiales de OS/2, la multitarea y servicios del sistema.

6. Se refiere al sistema de archivos: como crearlos, leerlos almacenarlos, administrar grandes volúmenes de ellos, particiones y servicios en OS/2.

7. La administración de los diferentes recursos del sistema.

8. Explica los tipos de dispositivos, el uso de subsistemas y la estructura de entrada y salida de datos del sistema.

9. El desarrollo de OS/2 y las interfaces que realiza con otro tipo de programación.

10.Relata el futuro de OS/2 a través de la comparación con el sistema operativo MS DOS.

Capitulo 1

HISTORIA

Antes de iniciar sobre la historia del Sistema Operativo OS/2, es necesario mencionar algunos aspectos esenciales sobre el tema.

Una computadora tiene dos elementos bàsicos: componentes electrònicos (tecnologia incorporada o hardware) y los componentes de programaciòn (tecnologia desincorporada o software). Sin embargo, esta distinciòn al paso del tiempo va no es tan notoria puesto que los adelantos en cuanto a tecnologia hacen que componentes de programaciòn simulen ser componentes electrònicos (debido al manejo tan variado de lenguajes y a la creatividad que el usuario aporta) .

La tecnologia incorporada (hardware) no cambia en forma excesiva, mientras que la programaciòn (software) es tàn versàtil que la interfaz a travès de base de datos, compiladores e interpretes, procesadores de palabra y sistemas operativos brindan una mayor oportunidad para satisfacer las innumerables peticiones del programador.

Es a travès del software que una computadora puede realizar un sin fin de actividades y precedente a cualquier lenguaje de programaciòn se encuentra el sistema operativo que es el que administra los recursos de una computadora, ademàs de ser considerado como extensiòn del hardaware lo cual hace posible

realizar innumerables combinaciones con las diferentes instrucciones, permitiendo que la interacción con el usuario sea cada vez menos compleja, y más accesible.

Si consideramos a la computadora como un conjunto de recursos asociados , debe entonces existir un proceso que lleve a cabo su administración y éste se denomina sistema operativo.

1.1 HISTORIA GENERAL DEL SISTEMA OPERATIVO

Los acontecimientos históricos de sistemas operativos no siguen una trayectoria lineal, por el contrario siguen una trayectoria reticular.

Las primeras computadoras se asignaban por fracciones de tiempo al usuario, que disponía de todos los recursos de la máquina para su uso exclusivo, por lo que en estos sistemas no había ningún tipo de cooperación ni interacción entre los usuarios, siendo esta monoprogramación estricta. Uno de los grandes inconvenientes era la ineficiencia en el uso de recursos, ya que estos permanecían buen tiempo desocupados, el usuario los utilizaba con una velocidad menor a la que éste realizaba su trabajo. Además el costo de las computadoras era muy alto, problema que afectaba en gran medida en cuestiones económicas a los propietarios.

Posteriormente surgió el desarrollo de los sistemas de multiprogramación, con ellos se buscaba utilizar el tiempo ocioso

del procesador causado por las entradas y salidas generadas en los programas, tratando de admitir varios de ellos simultáneamente en la computadora, para que al realizarse un proceso pudiera darse paso a otro, lo que requería la existencia de procesadores independientes que se encargaran de realizar las entradas y salidas al mismo tiempo con el procesamiento central, para lo que se introdujeron los llamados canales o procesadores de entrada y salida.

En este momento ya existían lenguajes de alto nivel que pretendían la interacción con el usuario, logrando aumentar considerablemente la eficiencia del sistema a través de un mejor uso de los recursos disponibles. Sin embargo, la dificultad que seguía persistiendo era la falta de interacción entre ususario y computadora.

A raíz del desarrollo tecnológico, surgen los sistemas de tiempo compartido (timesharing system), los cuales arrastran los inconvenientes de los otros sistemas, sin embargo facilitaron al usuario el acceso a la máquina. La idea central era construir sistemas que utilizaran extensivamente la multiprogramación y que permitieran a varios usuarios interactuar directa y simultáneamente con la computadora, a través de las terminales, compartiendo los recursos de la máquina.

Un aspecto importante es el hecho de utilizar una

computadora o bien otra, ya que es determinante y fundamental en la estructura y caracteristicas de un sistema operativo. en los sistemas grandes, las principales preocupaciones son la eficiencia, distribución y las politicas de asignación de los recursos, la protección, etc. En las micros se busca crear ambientes adecuados de programación para el usuario, idear mecanismos para lograr la compatibilidad entre diferentes computadoras, facilitar las comunicaciones con otras màquinas, etc.

Para el caso del sistema operativo OS\2 las caracteristicas del equipo en el que se utilizarà son esenciales.

1.2 HISTORIA DEL SISTEMA OPERATIVO OS/2

En 1981 IBM anunció una computadora la IBM PC. Seguido de esto se dió un crecimiento en el desarrollo, surgieron Microsoft (software de sistemas), Lotus (aplicaciones de negocios), Compaq (IBM PC), Borland (lenguajes), Hayes (comunicaciones) y otras màs.

Tiempo después IBM realizó la estandarización dos: la arquitectura de la microcomputadoras o PC y el sistema operativo PC DOS (MS DOS) que se convirtió en plataforma de programación en la que se basan diversas aplicaciones.

Al crecer el número de aplicaciones disponibles para las microcomputadoras o PC's, los usuarios exigían màs y pedían

aplicaciones màs sofisticadas. así como la integración de procesos. para lo cuàl era necesario que el DOS se convirtiera en un sistema operativo multitarea que pudiera ejecutar varios programas simultàneamente.

Las ideas centrales se habian cumplido: un sistema de archivos que procesaba los datos almacenados en un disco y un indicativo de petición de órdenes. un lenguaje de órdenes (batch) y una serie de utilidades.

En 1981 apareció un sistema bàsico de PC con el microprocesador 8088 a 4.77 MHz. 16K de memoria y una unidad de 160K. Al cabo de un año se desplazó a 256K de memoria y dos unidades de disco de 360K. En 1983 la PC XT aportó a las microcomputadoras la tecnología Winchester de disco duro. en 1984 apareció la PC AT con su microprocesador 80286. en este mismo año las microcomputadoras reciben el primer soporte de red de área local (LAN). conectando varias PC en un sólo sistema.

Al evolucionar el equipo de la PC. también lo hace MS DOS. La primera de las versiones soportaba las configuraciones màs simples de la PC .

A la par que crecia la tecnología también lo hacian las aplicaciones de MS DOS. Se integraran soluciones en forma de programas de aplicaciones dando al usuario soporte de decisión (hoja electrónica). procesadores de palabras. bases de datos y

comunicaciones. Sin embargo, el costo por memoria y disco era excesivo. El usuario de la PC comenzó a exigir más del hardware, como son: interfaces de gráficos, sistemas de ventana y multitarea; que de acuerdo con las primeras configuraciones de las PC, no tenía ninguna posibilidad de soportar tales aplicaciones .

En respuesta a la demanda, se determinó una integración total, una mejor utilización de la CPU (Unidad Central de Procesamiento) y de la multitarea. Microsoft anunció su Microsoft Windows, sin embargo para algunos usuarios Windows permitía utilizar datos de diversas aplicaciones simultáneamente y para otros el sistema Windows era demasiado lento. Este paquete fué de suma importancia para la industria de la computación.

Dado que el sistema operativo MS DOS no soportaba multitarea, los diseñadores de aplicaciones crearon procesos extras como soporte que se cargan en memoria y son activadas a través de ciertas teclas.

La única deficiencia del MS DOS era limitación de 640K , que no podía corregirse con el software, al crecer las aplicaciones, aumento el uso de memoria, ya que multitarea y memoria están totalmente relacionadas.

Es en 1985 cuando IBM y Microsoft Corporation anuncian un

acuerdo para hacer en conjunto un proyecto de software, que llevó al inicio una nueva generación de software para computadoras personales. Los próximos años llevarían a diseñadores y programadores de estas compañías a perseguir un fin común, la construcción de una base de programación para las aplicaciones de las PC del futuro.

Es hasta el 2 de Abril de 1987, cuando se anuncia el nuevo Sistema Operativo, en el centro de reuniones de Miami Beach, que es llamado, Operating System/ 2. El OS/2 es la culminación de los esfuerzos de un grupo de programadores de diversos laboratorios, entre otros Boca Raton, Florida (IBM); Redmon, Washington (Microsoft); Austin Texas (IBM) , y Hursley, England (IBM).
Este anuncio hacía la descripción de dos versiones del OS/2. La Edición Estándar del OS/2 y la Edición Extendida del OS/2. Este trabajo está dedicado a la versión 1.0 de la Edición Estándar del OS/2, que define el esqueleto de la arquitectura usada por otras versiones de OS/2.

OS/2 está construido en modo protegido a través del microprocesador 80286/80386, es decir, el sistema operativo ayudado por el procesador activa el equipo permitiendo obtener un mayor control sobre las aplicaciones. OS/2 tiene la característica de ser muy flexible y presentar a los usuarios un ambiente de trabajo adecuado .

Explotando las caracteristicas de modo protegido del equipo, el Sistema Operativo OS/2 resuelve las limitaciones clàsicas de MS DOS. Algo primordial es que no està limitado tanto al espacio de direcciones de 640K, como a ejecutar un sòlo programa a la vez.

Algo imprescindible para OS/2 es el microprocesador Intel 80286 (u 80386). Que incluyen muchos sistemas como son: PC/AT, el XT/286 y claro, los modelos 50, 60 y 80 del PS/2. Por estar basado en el microprocesador 8088 (y 8086) el PC/XT ,PC Transportable, PC Convertible y PS/2 modelos 25 y 30 no ejecutan el OS/2.

El Sistema Operativo OS/2 tiene una arquitectura diseñada para poder crecer. Muchas funciones del sistema son reemplazables y extendibles. Las interfaces de programaciòn del Sistema Operativo OS/2 se basan en un modelo que està optimizado para lenguajes de alto nivel (C, COBOL, FORTRAN, Pascal, entre otros).

El 2 de Abril de 1987, IBM y Microsoft anunciaron OS/2, una nueva generaciòn de sistema operativo que se levantaba donde MS DOS caia. OS/2 es la nueva generaciòn de màquinas 80286/80386 asi como el MS DOS lo fuè para las PC 8088 de 16K en 1981.

OS/2 no funciona en el IBM PC (8088) por no tener la potencia necesaria para utilizarle, ni la capacidad para poder soportar programas en modo protegido.

Como OS/2 es un sistema operativo multitarea, no sòlo

permite al usuario ejecutar varios programas simultáneamente, sino también cambiar de un programa a otro, incluso intercambiando información entre los mismos.

Capitulo 2
POR QUE UN NUEVO
S I S T E M A ?

En 1981 se anuncia una nueva computadora. IBM PC. seguido de esto la industria estandariza el sistema operativo MS DOS. que se toma como base de programación para dar inicio a una gran variedad de aplicaciones. Al ir evolucionando la tecnología en computadoras todas las aplicaciones hechas en MS DOS aumentan en complejidad. logrando corregirse de algún modo los problemas de multitarea. sin embargo lo que no podía ajustar era la limitación de memoria de 640K. ya que debido al anexo de aplicaciones el sistema requería de un aumento de memoria.

2.1 LIMITACIONES DEL MS DOS

MS DOS es un sistema operativo básico que está apoyado en el microprocesador de Intel. fué diseñado en torno al microprocesador 8088. proporciona archivos rudimentarios (discos duro y flexibles). dispositivos (impresora. teclado) y servicios de administración de memoria. La mayoría de las limitaciones del sistema operativo MS DOS se derivan de la forma en que funciona la CPU (Unidad Central de Procesamiento) de la computadora.

Como se mencionó anteriormente una de las limitaciones de MS

DOS es la direccionabilidad de memoria de este sistema. teòricamente se instala en memoria de una PC AT hasta 16MB y se utiliza con ciertos programas especiales tales como Disco Virtual (VDISK). pero esta memoria no sirve para llevar a cabo alguna aplicaciòn de MS DOS. La CPU de dicho sistema funciona de forma tal que no permite la entrada a la memoria a aquellos programas que rebasen 1MB (1 MegaByte = 1000 bytes).

Considerando esta limitaciòn de direccionamiento de un 1MB aunado a los 360K que deben restàrsele por el espacio de direcciones que se reserva para las procesos del equipo de entrada y salidad de datos (BIOS) dan un total de 640K finales del MS DOS. Donde en estos 640K se encuentran incluidos la memoria que utiliza el MS DOS y sus extensiones (50K para un DOS 3.3 solo) resultando 590K para el còdigo de datos. Si consideramos el ejemplo de una hoja electrònica de 700 * 200 fàcilmente podrìa rebasar los 590K.

Cuando se pretende utilizar alguna extensiòn de MS DOS la situaciòn se complica a pesar de que existen estructuras que pretenden habilitar màs de una aplicaciòn a la vez. proceso con frecuencia imposible por la cantidad de memoria que se ocupa en las aplicaciones. El manejo interno de direcciones en memoria se hace por medio del microprocesador 8088 causando asì que MS DOS no sea quien direccione la memoria de aplicaciòn.

En el espacio de direcciones de 640K del 8088 no existe una distinción exacta entre el código del sistema operativo y el código de aplicación. es aquí donde se encuentra cada una de las actividades que el sistema operativo realiza de acuerdo con los requerimientos del usario. debido a lo cual todo programa examina y modifica cualquier parte de él. si así se desea.

Si no exite una separación entre los programas. al haber un error al programar el sistema entero caería. En algún momento dado. esto le sucede a cualquiera. cuando por alguna causa se ha tenido que desconectar la computadora debido a que la aplicación ha quedado bloqueada. dado que internamente se ha generado un ciclo indeterminado de ejecución en la aplicación producida por el mal manejo de la misma. Al incrementarse el número de aplicaciones. la probabilidad de un bloqueo del sistema aumenta.provocando que no se obtenga la información que indique la causa que originó la caída.

La Figura 2.1 muestra la estructura del sistema operativo MS DOS que contiene un espacio total de direcciones correspondiente a 1MB (1 MegaByte) de los cuales 640K conforman las aplicaciones y el microprocesador 8088. las extensiones del sistema y el MS DOS. siendo aquí la limitante de memoria debido a que los restantes 384K son asignados para el código de datos del BIOS (Basic Input/Output System. administradores de entrada y salida y otras funciones básicas). si llegará a existir una falla en las

aplicaciones, como el programa de aplicación tiene acceso a las extensiones del sistema así como al sistema operativo, estas sufrirían modificiones que dañarían al sistema en general.

Al introducirnos en el sistema de administración de entrada/salida (E/S), PC DOS se dedica a consultar y no ha controlar la información. El modo del microproprocesador 8088 en el cual corre el sistema da a todos los programas un privilegio completo de E/S. Lo que significa que no hay necesidad de acudir directamente al software (conjunto de programas) del sistema para modificar el estado de los dispositivos del equipo dado que se hace en forma directa a través de una aplicación, tal como se observa en la Figura 2.1.

En el modo del microprocesador 8088 los programas de aplicación tienen la prioridad de deshabilitar las interrupciones del sistema por un tiempo indefinido, lo que hace que todo lo externo a ellos se pierda.

MS DOS posee ciertas necesidades de multitarea, semejante a un procesador de palabras que imprime algún archivo al mismo tiempo que se termina una edición. Sin embargo el MS DOS no provee todos los servicios que en un momento dado se requieren, por lo que los programas de aplicación hacen la labor de multitarea dando interrupciones cuando sean necesarias para que el sistema identifique el dispositivo que interumpió y llama al manejador de interrupciones apropiado, haciendo una interfaz

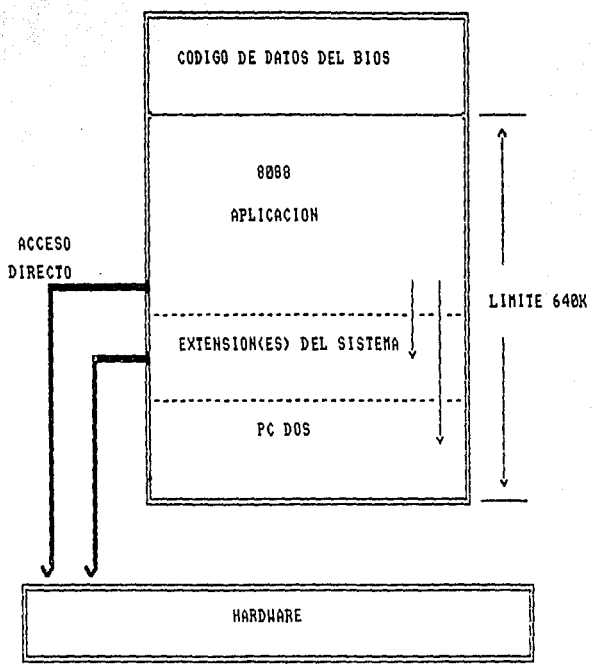


FIGURA 2.1 ESTRUCTURA DEL SISTEMA PC DOS

con el hardware; el manejador distribuye los procesos del CPU a diferentes funciones del programa. Pero MS DOS sòlo atiende una peticiòn a la vez. si se intentara pedir màs de una. el sistema se confundiria y los procesos quedarian bloqueados.

Por lo que programas que utilicen multititarea deben contener funciones de seriación de procesos, para garantizar que el sistema operativo MS DOS atenderá una sóla petición: al terminar una, comenzará la otra.

La probabilidad de que ocurra algún problema en el sistema de multitarea en aplicaciones de MS DOS es bastante alta, si se encuentran cargadas en el mismo sistema, incluso hay programas que despliegan el hecho de que no coexisten con otros programas. Algo más que interifere en el proceso de multitarea de MS DOS es el administrador de memoria del 8088, donde en la CPU las aplicaciones manipulan los apuntadores físicos de la memoria.

Por lo que MS DOS transfiere la memoria afectando de alguna forma el programa de aplicación, además de no poder optimizar la memoria dejando segmentos de aplicaciones libres de distintos tamaños.

Otro punto importante es la extensibilidad del sistema en la cual se involucran las interrupciones que el sistema operativo maneja.

La interfaz de programación de aplicaciones (API) del MS DOS está basada en interrupciones software (es una transferencia de control manejada por la CPU entre dos programas). En la Figura 2.2 se muestra una interrupción de software, del ejemplo se explica la numeración:

1) La aplicación facilita una instrucción de interrupción dirigida al manipulador de interrupciones del PC DOS (INT21).
2) El CPU busca en una tabla de vectores de interrupción residente en memoria la dirección de la rutina que manipulará INT21.
3) Utilizando la dirección resultante, el CPU transfiere el control a la rutina de interrupción.
4) Después de que PC DOS completa el tratamiento de la petición, facilita una instrucción IRET (RETorno de Interrupción) para indicar que ha completado la interrupción.
5) La CPU devuelve entonces el control a la siguiente instrucción.

Una interrupción software es un tipo de interfaz de llamada y regreso (se hace la llamada a la interrupción se analiza y regresa) entre uno y otro programa arbitrario del sistema. El programa objeto es un tipo de programación indistinto, ya que el CPU transfiere el control a cualquier dirección que esté en el vector de interrupciones. Dos o más programas pueden interceptar el mismo vector de interrupciones. En casos donde la extensión modifique o bién cambie la petición de servicio y/o extensiones

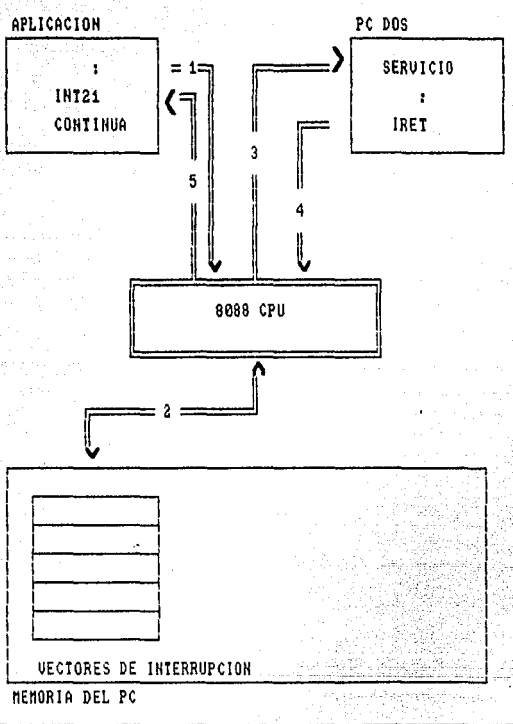


FIGURA 2.2 INTERRUPCION DE SOFTWARE

de niveles abajo tendràn fallas. La Figura 2.3 muestra las posibles combinaciones. Es responsabilidad del usuario determinar que extensiones se cargaràn juntas, aunque existen algunos programas que incluyen instrucciones de prioridad.

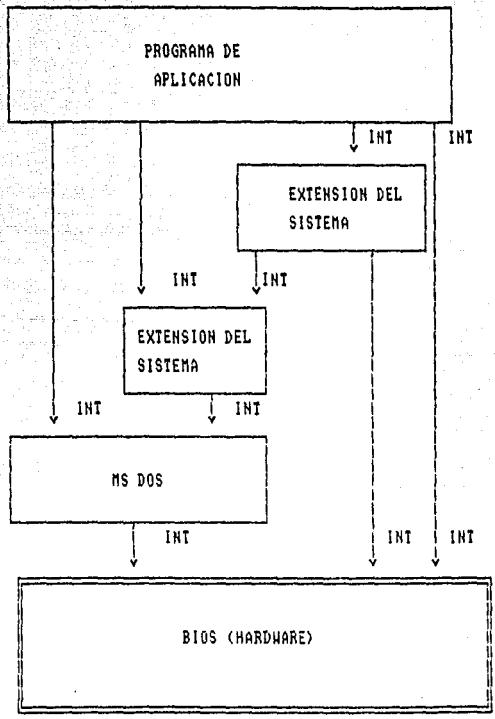


FIGURA 2.3 EXTENSION DEL MS DOS

2.2 EL ENTORNO DE OS/2

El OS/2 es un nuevo sistema operativo diseñado por Microsoft (MS) e IBM para la PC AT de IBM, que administra recursos tales como discos, impresoras y otros dispositivos periféricos (ratón, modem, trazador de gráficos,etc.). Además proporciona los medios para poder ejecutar otros programas tales como procesadores de palabras y paquetes de administración de bases de datos, en la computadora.

El OS/2 como predecesor del DOS presenta muchos aspectos, cada una de los cuales depende del usuario final (Figura 2.4).

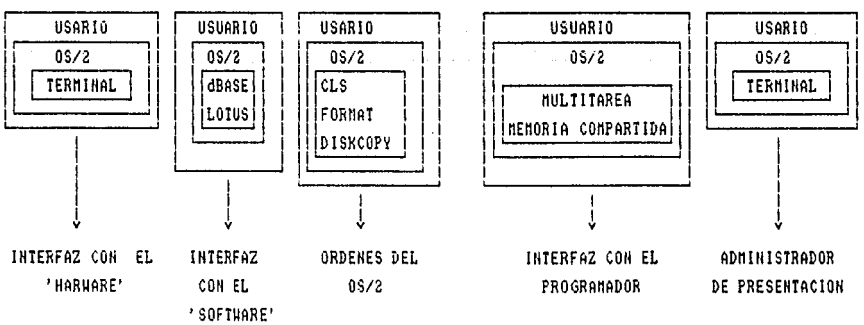


FIGURA 2.4 ASPECTOS DE OS/2

Ofreciendo posibilidades como multitarea, administraciòn virtual y soporte para aplicaciones en modo protegido. Para programadores, el OS/2 proporciona tanto programaciòn como las herramientas de desarrollo, disponibles antes exclusivas en UNIX o VAX/VMS.

Con la aparición de los procesadores 80286 y 80386, las posibilidades del hardware de la computadora exceden a las que actualmente tiene el software. Hecho por el cual se crea un nuevo sistema operativo màs potente: el OS/2 fuè diseñado como respuesta a factores tales como:

* Los 640K de espacio de programa porporcionados por el DOS son insufucientes para algunas aplicaciones sofisticadas.
* Los programas que en su dìa fueron de un sòlo usuario deben ser accesibles a sistema multiusuario en redes de àrea local.
* Las aplicaciones del usuario, requieren que la informaciòn se comparta para asegurar una integraciòn total.
* La aparición del 80286 y 80386 porporcionò procesadores lo suficientemente potentes para operar en un sistema operativo multitarea.

El OS/2 le autoriza ejecutar varios programas al mismo tiempo, permite el cambio ràpido de una aplicaciòn a la siguiente, viendo el resultado de cada una en pantalla, una

ventaja de la multitarea es que cada programa intercambia información fàcilmente, reduciendo así su carga de trabajo.

La multitarea consiste en la ejecución de varios programas simultàneamente. Por ejemplo, al ejecutarse la orden PRINT (imprimir) del DOS, èste empieza a imprimir los archivos como tareas de fondo, permitiendo que se continúe la ejecución de órdenes en modo inmediato (en el indicador del DOS), lo cual hace parecer que se ejecutan dos tareas a la vez. Por tanto, un sistema operativo multitarea da la apariencia de realizar más de una tarea a la vez, debido a la rapidez en el manejo de sus aplicaciones. El OS/2 (al igual que la orden PRINT del DOS) debe dar la apariencia de que están ocurriendo al mismo tiempo varios acontecimientos. Lo que en realidad sucede es que la computadora pasa de un programa a otro con gran velocidad.

La multitarea mejora la utilización global de las posibilidades de la computadora y la productividad de las sesiones, al minimizar el tiempo de inactividad del CPU. Al ejecutar varios programas simultàneamente, se optimiza la utilización de la capacidad del sistema.

Los programas del DOS se ejecutan en modo real, proporcionando un control completo sobre la computadora. En el modo real, los programas realizan operaciones de E/S y tienen acceso directo a cualquier posición de memoria además de

controlar los dispositivos del hardware.

La multitarea de OS/2 exige el modo protegido del 80286/80386. En este modo se mantiene un control estricto sobre cada una de las aplicaciones. Esencialmente el OS/2 protege cada uno de los programas de los demàs. Al contrario del modo real, los programas en modo protegido deben interactuar. No tienen acceso directo a algunas posiciones de la memoria, no realizan operaciones de E/S de nivel bajo y tampoco tienen acceso directo a los dispositivos del hardware. El modo protegido asegura la cooperaciòn entre las aplicaciones concurrentes.

Entre las consideraciones a las que se enfrentò OS/2 fuè la de asegurar la compatibilidad con el DOS.
Las aplicaciones del DOS se ejecutaban en modo real.

Las aplicaciones concurrentes del OS/2 se ejecutaban en modo protegido. El OS/2 ofrece la alternativa de ejecutar aplicaciones en modo real y modo protegido. Permite realizar multitarea con aplicaciones en modo protegido mientras que se mantiene la compatibilidad con las aplicaciones existentes del DOS (en modo real).

Otra forma de compatibilidad a la que tuvo que enfrentarse fuè a la de los discos. El OS/2 y el DOS utilizan formatos de discos idènticos. El OS/2 admite los discos del DOS. es un sistema operativo basado en la arquitectura del IBM PC AT, que

requiere una mayor velocidad en el manejo de datos. junto con el soporte de hardware para el procesamiento en modo protegido, del que no dispone IBM PC. El OS/2 no funciona con IBM PC. Por el contrario funciona con equipos que utilizan el procesador 80386 (aunque este no se aprovecha del todo). sucesor del 80286 (procesador del IBM PC AT). El IBM PS/2 Modelo 80, por ejemplo, emplea el 80386.

La Figura 2.5 describe las necesidades de equipo del OS/2. Necesita un procesador 80286 ò 80386 en el CPU y utiliza discos flexibles de 1.2 MB. La mayoria de las PC AT estàn configuradas con al menos una unidad de disco de 1.2 MB. Probablemente la mayor exigencia de hardware del OS/2 sea la memoria de acceso aleatorio (RAM) de 1.5 MB.

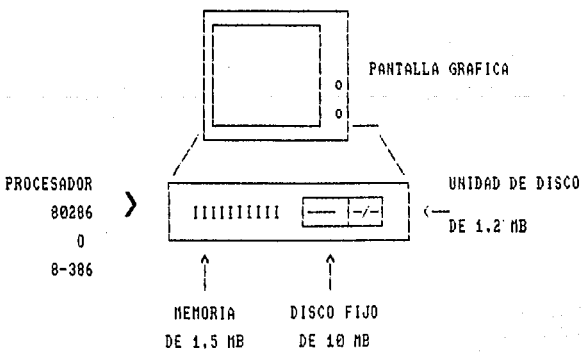


FIGURA 2.5 REQUISITOS DE EQUIPO PARA EL OS/2

El equipo mencionado anteriormente es un requisito indispensable para abrir espacios nuevos en la evolución de la programación, además tiene influencia en la complejidad del sistema operativo OS/2.

Por ejemplo si la computadora o el equipo a utilizar tiene un mecanismo de traducción de direcciones, deben existir componentes dentro del sistema para el manejo de memoria virtual.

OS/2 utiliza una técnica conocida como administración de memoria virtual, que da la impresión de que las aplicaciones disponen de una memoria ilimitada. Con el DOS, cuando un programa hacia referencia a una posición de memoria, ésta es una dirección física en la memoria RAM de la computadora (una función del procesamiento en modo real). Esta técnica se ilustra en la Figura 2.6. Cabe mencionar que dicha técnica aunque simple, imposibilita a las aplicaciones a exceder los 640K y no es apropiada para la multitarea.

Como las aplicaciones del DOS tienen acceso directo a posiciones de memoria física, no hay forma de impedir que un programa utilice posiciones de memoria, actualmente en uso por un segundo programa. Por tanto, una aplicación concurrente podría modificar fácilmente el código o los datos de una segunda aplicación, intensionada o accidentalmente, lo anterior se observa en la Figura 2.7 .

Un sistema operativo que cuenta con un entorno multitarea debe proveer los medios que protejan unas aplicaciones de otras.

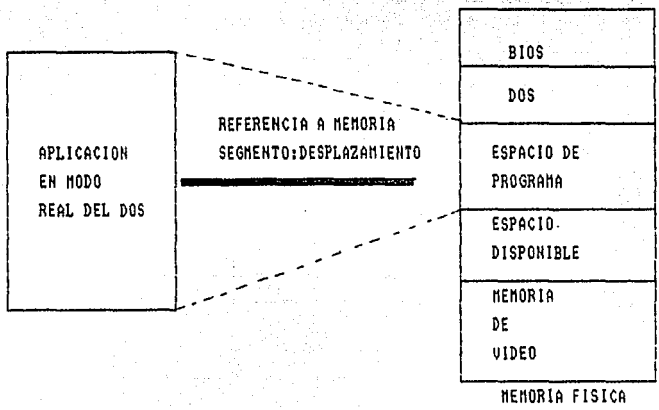


FIGURA 2.6 REFERNCIA DE MEMORIA A UNA DIRECCION FISICA DE MEMORIA RAM

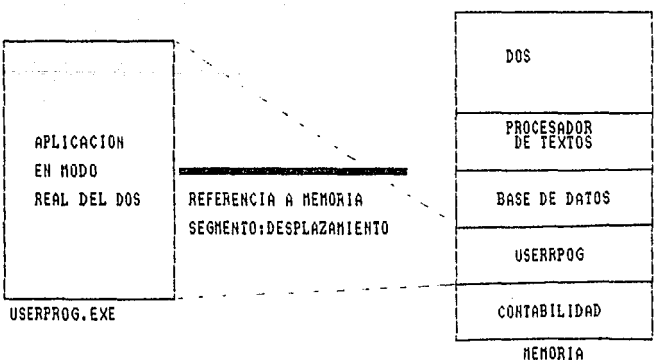


FIGURA 2.7 INCAPACIDAD DEL DOS PARA PROTEGER TODOS LOS PROGRAMAS EN MEMORIA

La administración de memoria virtual del OS/2 proporciona un nivel de direccionamiento que le permite controlar las operaciones de memoria asignadas a cada aplicación y estas tienen acceso a su propio espacio único de direcciones de memoria virtual, que es el OS/2 asociada con la memoria física (Figura 2.8).

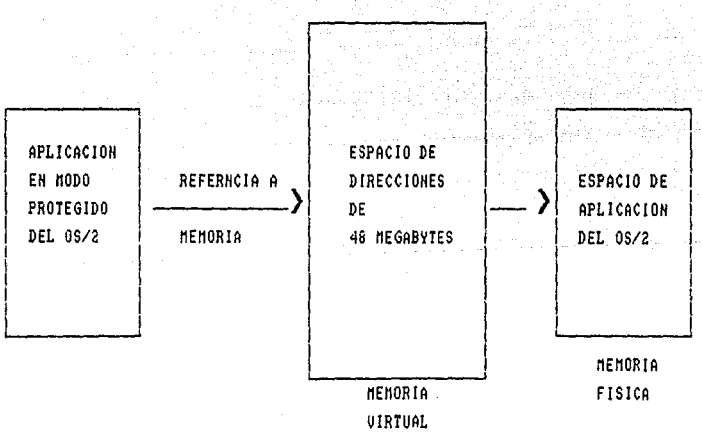


FIGURA 2.8 ESPACIO DE DIRECCIONES DE LA MEMORIA VIRTUAL CON EL OS/2

La memoria virtual se llama así porque el espacio de direcciones no existe ralmente. El OS/2 pone en marcha la memoria virtual dividiendo un programa extenso en varias partes màs pequeños. Cuando el programa hace referencia a una dirección que no recide actualmente en la memoria, el OS/2 trae el segmento referenciado a la memoria principal desde el disco según las necesidades. De tal forma que queda espacio en la memoria para el nuevo segmento. OS/2 translada un segmento desde la memoria hasta el disco para dejar espacio libre.

Despuès trae el segmento deseado a la memoria. Este proceso de transladar segmentos dentro y fuera de la memoria de esta forma, se llama intercambio de segmentos. Un programa debe resumirse en memoria para ejecutarse. Tal como se muestra en la Figura 2.9.

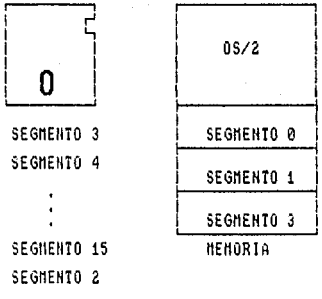


FIGURA 2.9 EL OS/2 TRAE EL SEGMENTO DESEADO A LA MEMORIA

El OS/2 trae a la memoria los segmentos de cada programa que sean requeridos para su ejecución inmediata .
Así entonces, el OS/2 utiliza la administración de memoria virtual para dar a las aplicaiones la experiencia de disponer de una memoria ilimitada. De manera similar el OS/2 usa el concepto de dispositivos virtuales, al utilizarlos, cada aplicación es manejada como si tuviera teclado, pantalla y monitor, propios.

La memoria virtual del OS/2 permite que la aplicaciones excedan los 640K de memoria y les proporciona un espacio de dirección de 48 MB. Para poder implementar la memoria virtual, se utiliza el intercambio de segmentos.

El OS/2 le da a cada aplicación la impresión de que poseè su propio teclado, monitor, impresora, ratòn y demàs, ya que cada aplicación utiliza sus propios dispositivos, de esta manera, las aplicaciones en modo protegido no interfieren entre sí. Cuando se elige una aplicación especifica para su visualización en pantalla, el OS/2 hace corresponder los dispositivos virtuales de la aplicación con dispositivos físicos reales de un modo transparente al usuario final.

Debido a que OS/2 contiene el modo protegido, el procesador ayuda a que el sistema operativo corra el hardware dàndole màs control sobre lo que los programas de aplicación tienen permitido

hacer. A través del procesador 80286/80386. OS/2 utiliza hasta 16 MB de memoria real (física) con tendencia a aumentar según los requerimientos de aplicación. Este aumento es permitido debido al modo protegido del OS/2 ya que le habilita a implementar un mayor espacio de direcciones virtuales. por lo que. OS/2 toma la responsabilidad de mover la memoria de la aplicación a y desde la memoria secundaria (disco) cuando sea necesario. a través de segmentos virtuales. cuando no están siendo utilizados por la aplicación (el proceso anterior es denominado intercambio <swapping> de segmentos. de la cual se hablará en forma más amplia posteriormente).

OS/2 aisla cada aplicación en su propio espacio de direcciones logrando que un error de programación no modifique algún otro programa de forma inadvertida. en lugar de ello. se detecta y finaliza el programa evitando así que el sistema caiga. facilitando la información del programa de aplicación. La Figura 2.10 muestra la información obtenida al fallar un programa de aplicación.

Como se mencionó anteriormente en OS/2 se corren varias aplicaciones concurrentemente si así se requiere ya que el estado del hardware se controla cuidadósamente. Debido a su modo protegido forza a las aplicaciones a aislar las E/S directas en segmentos de código especiales que deben ser identificados

adecuadamente o de lo contrario la aplicación fallará.

Para explicar con un ejemplo práctico la administración de recursos en OS/2 diremos que es semejante a un policía de tránsito que determina cuándo los recursos están disponibles para ser utilizados y cuándo no lo están. Cuando un programa manipula al hardware en forma directa, el programa pide autorización a OS/2 a través de una petición de entrada y salidad de datos.

```
SESION TITLE: OS/2 COMMAND PROMPT

     SYS1943: A PROGRAM CAUSED A PROTECTION VIOLATION

     TRAP 000D
     AS=0000 BX=00C8 CX=0000 BP=0000 SI=0000
     DI=082A DS=002F ES=002F FLAGS=2246
     CS:IP=003F:0076 SS:SP=001F:010F ERRCD=0000
     CSLIM=00DC SSLIM=01FF DSLIM=0009 ESLIM=0009
     CSACC=FB SSAC=F3 DSACC=F3 ESACC=F3

          ) END THE PROGRAM
```

FIG 2.10 TERMINACION ANORMAL DE PROGRAMA

La diferencia que existe entre el DOS y OS/2 en cuanto a estructura es muy grande. Las extensiones del sistema OS/2 se

muestran en la Figura 2.11. Los servicios del sistema que necesitan todos los programas están en el núcleo: cualquier tipo de soporte hardware específico está contenido en las rutinas de los dispositivos, y los servicios de alto nivel del sistema se encuentran en los subsistemas. Todas las funciones del sistema son accedidas por el API.

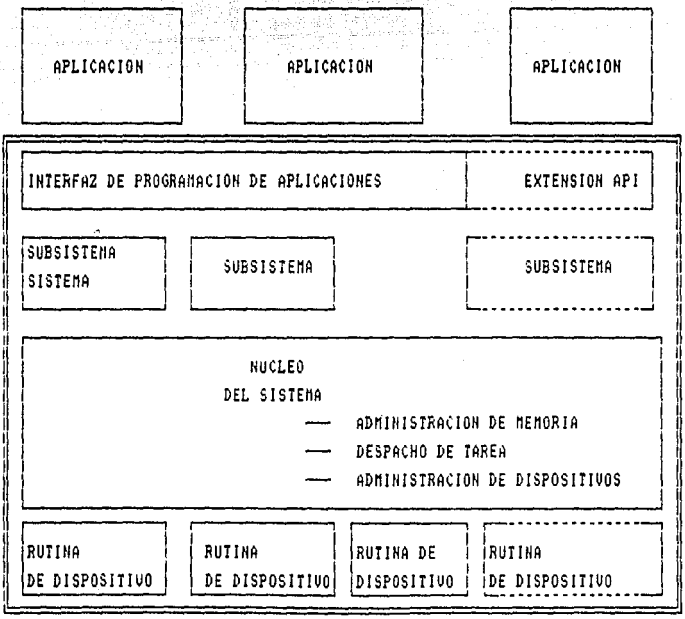


FIGURA 2.11 EXTENSIBILIDAD DEL SISTEMA OS/2

Capítulo 3

LA ARQUITECTURA DEL INTEL

Es necesario hacer algunas observaciones sobre un procesador y un microprocesador. la Figura 3.1 muestra la estructura típica de un procesador en una computadora.

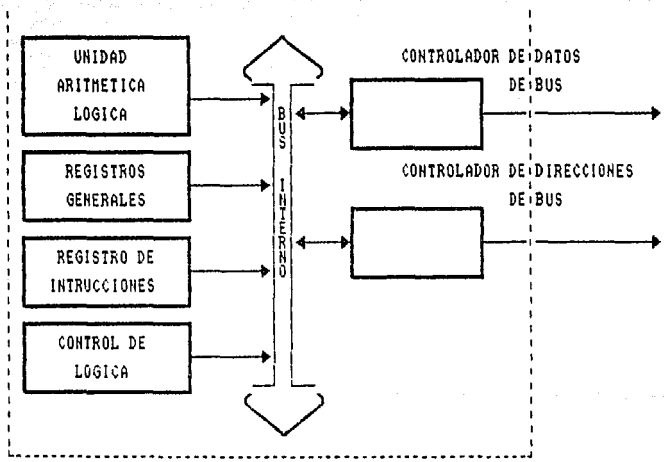


FIGURA 3.1 ESTRUCTURA TIPICA DEL PROCESADOR DE UNA COMPUTADORA

La unidad aritmètica-lògica se encarga de realizar operaciones aritmèticas y lògicas. los registros generales cumplen ciertas funciones especiales (registros de pila, contador

ordinal. etc.) o son de propósito general. y en ese caso sirven para realizar operaciones. El registro de instrucciones contiene la próxima instrucción a ejecutarse mientras ésta es decodificada (transformada a una serie de instrucciones) y ejecutada. La lógica de control se encarga de la coordinación y ejecución de las actividades que deben desarrollarse para la ejecución de las intrucciones. El bus interno comunica los distintos elementos. además de los anteriores. existen el controlador de datos del bus. que manipula la transferencia de información entre los dispositivos externos y la memoria. y el controlador de direcciones del bus que maneja el direccionamiento de los dispositivos externos.

Dependiendo de la computadora. la transferencia de información entre el bus interno y los disitintos componentes puede ser 8. 16 ó 32 bits. lo cual conduce a la clasificación de computadoras de 8. 16 ó 32 bits. la ventaja evidente de los 32 sobre los 16 u 8. o de los 16 sobre los 8. consiste en la rapidez de operaciones. dado que mientras en un caso el hardware maneja operaciones de hasta 4 bytes (32 bits). en el otro sólo se permiten de 1 byte. Esto no implica que en una computadora de 8 bits no se puedan realizar operaciones más grandes. sino que éstas deben manejarse por software. lo cual es evidentemente más lento que si se realizara por hardware.

La Figura 3.2 muestra la estructura típica de un microprocesador.

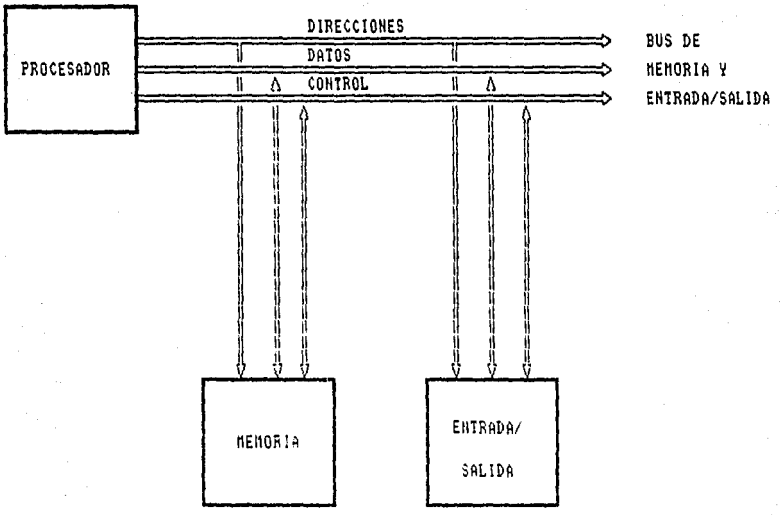


FIGURA 3.2 ESTRUCTURA TIPICA DE UN MICROPROCESADOR

Como puede verse en la Figura 3.2 el procesador está comunicado con los demàs componentes a travès de un bus. El tamaño del bus de datos es generalmente de 16 bits (8 bits en el

caso del Intel 8088). el de direcciones tiene un tamaño tipico que varia entre 20 bits (que es el caso del Intel 8086) y 24 bits (como en el Motorola 68000). El primero determina la velocidad a la que se transfieren operandos entre la memoria y el procesador. el segundo define la cantidad de memoria que debe direccionarse como màximo.

OS/2 es un sistema operativo cuyo diseño fuè hecho para operar con el microprocesador Intel 80286. Este microporcesador es compatible con la familia del 808X. Las instrucciones de la familia del 808X se ejecutan tambièn en microporcesadores 80286. Lo que hace que programas que se lleven a cabo en IBM PC correràn tambièn en IBM PC AT.

3.1 MODOS DE OPERACION

El 80286 opera en: modo de direccionamiento real (modo real) o modo virtual (modo protegido).

Mientras se encuentre en modo protegido el procesador 80286 serà ùnico y cuando estè en modo real la forma de operaciòn serà parecida a un 808X. Cuando se logra combinar los modos de operaciòn a travès de un programa. el sistema operativo podrà correr simultàneamente programas en modo real y modo protegido.

Al utilizar el modo de direccionamiento real el 80286 se vuelve compatible con sus predecesores 808X. esto en cuanto a funciones y caracterìsticas de operaciòn . pero no ayuda a los sistemas operativos en la administraciòn de recursos de la

màquina. Programas como el sistema operativo y las aplicaciones, tienen el mismo acceso a los recursos del sistema.

El 80286 es un procesador con registros internos de 16 bits que en modo real soporta hasta 1 MB de memoria real. Una dirección física corresponde a un byte, y como los programas realizados en el 80286 tan sólo reconocen a cantidades de 16 bits (esto es 64 K) no pueden direccionar directamente la memoria física, por lo que se identifican posiciones de memoria con dos valores de 16 bits: segmento y desplazamiento. Donde el primero define una dirección de comienzo en memoria y el segundo es el número de bytes desde el comienzo del segmento. La forma en que el procesador accesa a la memoria es sumando el segmento y el desplazamiento, lo cual permitirá llegar a una dirección física. Los desplazamientos pueden o estar en un registro , en una combinación de estos o bien ser una constante. El valor de un segmento es cargado en un registro especial llamado registro segmento:

seg:ofs (con seg : nombre o valor del registro segmento y
off : registro o valor de desplazamiento.)

Ejemplos:

DS:BX Segmento en registro DS. Desplazamiento en registro BX

ES:200 Segmento en regsitro ES, con desplazamiento de 200 bytes.

100:200 Segmento de 100 , con desplazamiento de 200 bytes.

Dentro de la arquitectura del Intel se encuentra el cálculo de la memoria real, donde se toman los contenidos del registro de segmento, desplazándolo cuatro bits y sumándolo al desplazamiento, lo que da como resultado una posición de memoria.

La Figura 3.3 muestra como el procesador calcula las direcciones físicas cada vez que accesa a la memoria.

Las direcciones físicas en modo real son de 20 bits con un valor máximo de 1 MB, siendo esto lo que produce la limitación de 640 K del MS DOS. Como en el modo real las direcciones de memoria son calculadas aritméticamente mucho valores seg:ofs apuntan a la misma posición de memoria física.

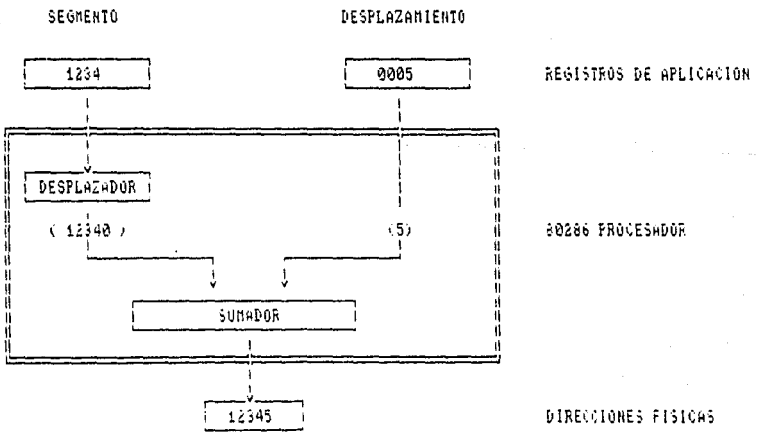


FIGURA 3.3 CALCULO DE DIRECIONES EN MODO REAL

De la misma forma se calcula el valor de contenido en el registro. Por ejemplo se puede saltar a través de àreas de datos de 16 bytes una vez incrementando el registro de segmento.

El 80286 soporta los siguiente tipos de registros:

CS	Segmento código
DS	Segmento de datos
SS	Segmento de pila
ES	Segmento extra

El modo real en el procesador 80286 no ayuda al hardware para proteger la memoria. por lo que cualquier programa puede alterar los datos y código de otros programas o iclusive al mismo sistema operativo. además no proporciona al sistema operativo control alguno sobre el tratamiento de E/S de la aplicación.

En modo protegido el procesador 80286 asiste al hardware para ayudar al sistema operativo a administrar mejor la màquina. habilitando una parte importante de soporte de la memoria real. protección de memoria y entrada y salida de datos. multitarea. niveles jerárquicos de software.

3.2 MODELO DE MEMORIA

En el modelo de memoria los registros "segmento" en modo protegido contienen indicativos lógicos llamados "hadle" o selectores y respresentan una posición de memoria.

La Unidad Central de Procesamiento (CPU) asocia selectores (es un índice para una tabla) y estructuras de datos denominándolas descriptores de segmento[1] (define los atributos de una parte de la memoria física). estos tres elementos en conjunto son utilizados por el procesador para ayudar a administrar la memoria del sistema. además de ser asignadas y mantenidas por el sistema operativo. Por otra parte mediante su actualización es como el sistema conjuntamente con una serie de reglas de acceso localiza el lugar de la memoria. su magnitud y las tareas que pueden ser ejecutadas por ella y quien en un momento dado las utiliza. Una diferencia importante en cuanto a memoria en los modos real y protegido es que en éste último las aplicaciones no tienen una relación o un apuntador que señale la posición que ocupa dentro de ésta.

Los microprocesadores Intel se basan en un modelo de memoria segmentada. lo que hace que la memoria se vea como una colección de segmentos y no como un espacio de direcciones lineales. es decir. que la información recabada será almacenada en particiones no ordenados sino colocados dispersamente.

Los segmentos individuales son identificados por selectores. donde el programa de aplicación es responsable de cargarlos en los registros segmento.

---

1 Llámese éstos a una serie de características basadas en la infromación de algún elemento.

Se utilizan selectores para referenciar segmentos de memoria en modo protegido, así mismo un selector describe una posición de memoria y es cargado en un registro segmento en el modo real.

Pero los selectores son un índice en una tabla de descriptores de segmentos, que dicen al procesador que descriptor de segmento se utilizará para calcular una posición de memoria. en la Figura 3.4 se muestra la forma en que el procesador utiliza selectores para accesar la memoria.

Al cargarse un selector no válido el procesador 80286 lo detecta y genera un fallo de protección en el sistema operativo. En los selectores no existe una manipulación aritmética realizada a través de una aplicación, ya que estos son índices y no apuntadores.

Un segmento forma parte de una memoria física y es definido por un descriptor de segmento. Un programa hace referencia a un segmento cargando su selector en un registro de segmentos.

De acuerdo al programa de la aplicación el sistema operativo crea, modifica y suprime descriptores de segmento (contienen la información que define un segmento de memoria). Para localizarlos en la tabla de descriptores la CPU utiliza un selector.

La tabla 3.5 muestra los atributos de los segmentos de la memoria del procesador 80286.

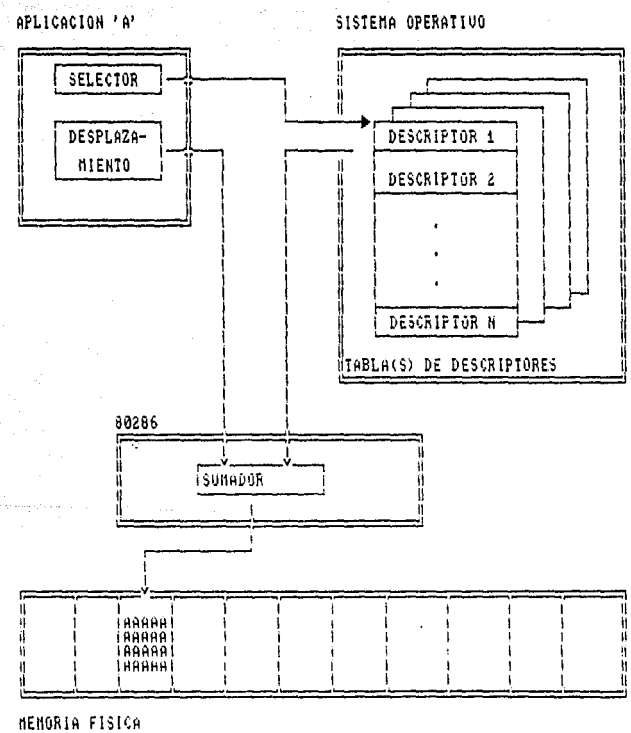


FIGURA 3.5 DIRECCIONAMIENTO DE MEMORIA VIRTUAL 80286

TABLA 3.5 DATOS DESCRIPTORES DE SEGMENTOS DEL 80286

Dirección segmento base	24 bits
Tamaño segmento	16 bits
Derechos de acceso	8 bits
Presente	Indicador presente/ no presente
DPL	Nivel de privilegio de segmento
Ejecutable	Indicador codigo/dato
Conforme	Indicador conformidad desarrollo
Lectura/Escritura	Código leible o dato escribible
Accedido	Indicador de descriptor de segmento accedido

El hecho de que puedan cargarse y ejecutarse múltiples programas a la vez es porque el sistema operativo intercambia los contenidos de memoria real para dar cabida a los nuevos segmentos.

Por lo que las opciones de administración de memoria del 80286 hacen que el sistema operativo facilite la organización de los contenidos de memoria de aplicación y de modificaciones inadvertidas.

Para poder utilizar un segmento es necesario recurrir a los derechos de acceso de este.que definen cómo y por quién podrán ser utilizados. Al establecerse las reglas de acceso. el sistema operativo carga un programa y permite que corra. en base al segmento. en el procesador 80286 se llevan a cabo las reglas de acceso establecidas por el sistema operativo. cuando se intenta usar un segmento no autorizado o bien un programa que intente usar un segmento no compatible con sus derechos de acceso. el

procesador informa al sistema operativo con un fallo de protección general.

Los usos de indicadores de derechos de acceso son:

Presente: Informa si un segmento se encuentra o no presente en memoria física.

Cuando el programa se encuentra como no presente y se intenta utilizarlo, el CPU genera una interrupción denominada de segmento no presente. Además puede ser utilizado para implementar una función de "exceso de memoria" en un sistema operativo y permite asignar a las aplicaciones más memoria virtual que memoria física tiene el sistema.

DPL El 80286 soporta múltiples niveles de privilegio. Estos niveles son utilizados por el sistema operativo para definir cuan confiados son los programas. El nivel de privilegio del descriptor (DPL) se utiliza junto con la función de autorización de un progroma del 80286 para definir què nivel de programas pueden acceder a un segmento. Este se utiliza para proteger los datos y código en un programa confiado, como el núcleo de un programa memos confiado (aplicaciones).

Si un programa en un nivel de privilegio inferior que el DPL intenta acceder al segmento, se genera un fallo de protección general.

Ejecutable: Este indica si el segmento es de código o de datos. Si un programa intenta ejecutar un segmento de datos o modificar un segmento de código. la CPU genera un error de protección general.

Conforme: El 80286 introduce la noción de conformar segmentos de código. siendo este el que puede ser llamado con el DPL o cualquier nivel de privilegio mayor. Este indicador se utiliza para marcar un segmento código. como "conforme".

Lectura y Escritura: En descriptores de código ejecutable. este indicador es utilizado para marcar el segmento de sólo ejecución. Se utiliza para marcar código ya que no se tiene acceso a examinarle. Si el CPU lee este tipo de segmento. genera un fallo de protección general. Para descriptores de datos. el indicador designa el segmento como de sólo lectura. Escribir en un descriptor de datos de sólo lectura también provoca un fallo de protección general.

Accedido: El Intel lo considera como un bit que se pone a 0 cada vez que se carga el selector. Se utiliza a través de un sistema operativo para perfilar el uso de selectores como entradas a un algoritmo menos usado recientemente (LRU). Los sistemas de memoria virtual. como el sistema operativo OS/2. utilizan

extensivamente algoritmos LRU.

El sistema operativo maneja los descriptores de segmento para implemetar un modelo de administración de memoria que sea mayor a la memoria física.

Para facilitar la transferencia de control entre segmentos el procesador 80286 define puertas que son un tipo especial de descriptores.

Toda transferencia intersegmentos [como llamadas lejanas (far calls)] se hacen con puertas. El procesador 80286 define cuatro tipos de puertas:

* Puertas de interrupción
* Puertas de trampa
* Puertas de tarea
* Puertas de llamada

La Figura 3.6 muestra la estructura especial de datos de las puertas, que contienen el selector y el desplazamiento de la rutina objeto. Aunque se diese el selector y el desplazamiento de una llamada lejana, tan sólo sería necesario el selector de puerta de llamada para iniciar la rutina.

Los descriptores de puerta se encuentran contenidos en las tablas de descriptores, sin embargo son diferentes a los otros tipos de descriptores, que representan un punto de entrada, un

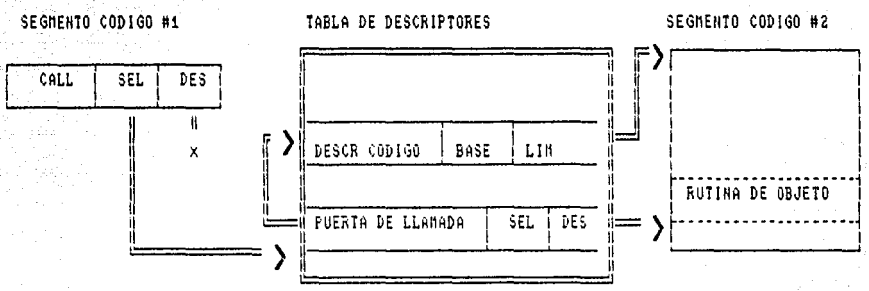


FIGURA 3.6 TRANSFERENCIA DE CONTROL INTERSEGMENTOS

lugar de una posición de memoria: si se carga un descriptor de puerta en cualquier registro segmento diferente de CS el procesador 80286 genera un fallo.

Las interrupciones y excepciones del sistema se llevan a cabo a través de las puertas de interrupción y trampa. además se encuentran en una tabla llamada de descripción de interrupciones (IDT).

Las puertas tarea transmiten control a diferentes segmentos

código, además realizan una intercambio de procesos de hardware y se encuentran en cualquier tabla de descriptores del sistema.

En OS/2 se realiza una adecuada conmutación con tareas de software de forma tal que no es necesario hacer una relación con gestiones de hardware del procesador 80286.

La "llamada" es el tipo más común de puerta. Es un selector que define una puerta y un "indicativo llamable" que define un punto de entrada fuera de su propio segmento.

Al sistema operativo le corresponde la tarea de crear puertas de llamada y colocarlas en una tabla de descriptores cuando se carga el programa. Una puerta contiene el nivel de privilegio de la rutina objeto y se utiliza además para cambiar el nivel de privilegio del código.

Generalmente los programas pasan datos en pila[1], que son diferentes para cada nivel de privilegio: también la puerta de llamada identifica el número de palabras (parámetros) a ser copiado entre las pilas.

---

1 Una pila es una forma particular de considerar los datos de que se dispone. con una lista de enlace lineal. Una pila tiene el acceso restringido a la cabeza de la lista que se denomina TOPE. y bajo éstas restricciones las operaciones se conocen como introducción (push) y extracción (pop) respectivamente. Una pila puede imaginarse como una pila de platos. al sacar un plato de la pila. siempre se hace por arriba y cuando se devuelve uno. también se hace de igual forma.

Los descriptores de segmento son agrupados en tablas, para lo que el procesador 80286 tiene tres tipos:

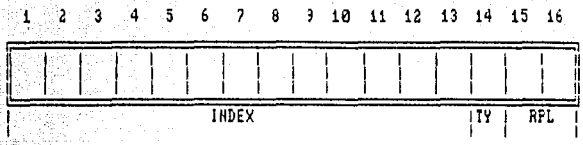
* Tabla de descriptores globales (GDT)
* Tabla de descriptores locales (LDT)
* Tabla de descriptores de interrupción (IDT)

- Tabla de descriptores globales (GDT)

  Está compuesta por los descriptores de segmento disponibles para cualquier tarea del sistema.
  Es la parte común de direcciones de cada tarea. Tiene incluidas llamadas a servicios comunes del sistema y las áreas de datos, así como el código del núcleo, se identifica por un 0 en el bit 14 del selector. (Figura 3.7). La GDT se encuentra siempre presente por lo que contiene los descriptores de datos y código de la Rutina de Servicio de Interrupción (ISR).

- Tabla de descriptores locales (LDT)

  Es la parte reservada del espacio de direcciones de cada tarea. Cuando se hace una llamada de intercambio de tareas, se guarda el estado de la tarea para que se reinicie posteriormente. Esta operación denominada "conmutar texto" también intercambia las LDT de tareas. Es considerada parte



DONDE: INDEX DESPLAZAMIENTO EN TALBA DE DESCRIPTORES
TY TIPO DE TABLA (0=GDT, 1=LDT)
RPL NIVEL DE PRIVILEGIO REQUERIDO

FIGURA 3.7 FORMATO DEL SELECTOR

del contexto mantenido a través de una tarea.

En ésta se encuentran los sectores asociados con el programa ejecutable (código, datos estáticos y dinámicos) y segmentos de memoria asignada directamente.

La identificación de selectores de LDT es de 1 en el bit 14 del selector. Esta no se utiliza para descriptores de datos y códigos de rutina de interrupción, dado que LDT es parte del contexto de tarea.

- Tabla de descriptores de interrupción (IDT)

Aquí se encuentran las interrupciones software y hardware, las puertas asociadas a cada una de las excepciones del procesador.

Al dar el procesador 80286 una interrupción utiliza la información de la puerta apropiada de la tabla de descriptores de interrupción para localizar el manipulador de interrupciones.

3.3 MODELO DE PROTECCION

Entre las caracteristicas del procesador 80286 en su modo protegido està la de aislar programas y protegerlos entre sì. El CPU tambièn permite al sistema operativo administrar el acceso de los recursos de la màquina al restringir el uso del còdigo de aplicaciòn de instrucciones de E/S. Entonces la protecciòn del procesador 80286 seria:

* Niveles de ejecuciòn de privilegio
* Protecciòn de espacio de direcciones
* Atributos de memoria
* Protecciòn de E/S

Niveles de ejecuciòn de privilegio

El procesador 80286 da cuatro niveles de privilegio. que son considerados como una jerarquia de software. èstos pueden ir desde aquèl que es el màs confiable (privilegio 0) hasta el menos confiable (privolegio 3). de tal forma que el còdigo se define para residir en un nivel u otro. Dentro de un nivel sòlo podràn ser accesados datos y còdigos por còdigo ejecutàndose en ese nivel. tal es el caso de el Privilegio 0 (PL0). Datos y còdigo residentes en el resto de los tres niveles sòlo seràn accesados por còdigo ejecutàndose en el mismo nivel o bien en

niveles inferiores. Si se intentara hacer algún otro tipo de combinación diferente a la antes mencionada ocasionaría un error. La Figura 3.8 muestra un ejemplo de como son accesados datos de los anillos por el código contenido en cada uno de los niveles.

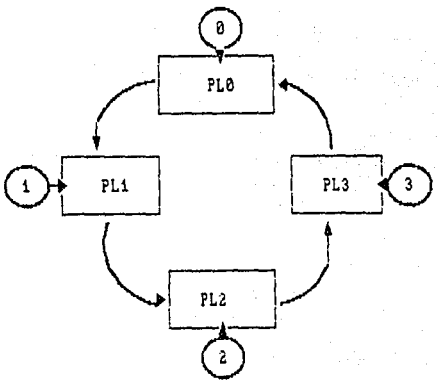


FIGURA 3.8 ACCESO DE DATOS POR CODIGO

Donde el Código en PL0 accesa datos de los anillos 0.1.2 y 3.

El Código en PL1 accesa datos de los anillos 1.2 y 3.

El Código en PL2 accesa datos de los anillos 2 y 3.

El Código en PL3 accesa datos del anillo 3.

Como puede observarse ningún nivel superior. (esto hablando en términos numéricos). accesa datos de anillos de un nivel inferior.

Las aplicaciones de memoria restringida se aislan de programas de aplicación asociando a cada tarea una LDT (Tabla de

Descriptores Locales) diferente, ya que cada tarea del sistema ocupa un espacio privado de direcciones. El código de las rutinas de servicio del sistema operativo se direcciona a través de la GDT (Tabla de Descriptores Globales) que accesa a todas la tareas. Una de las mejores formas para aislar los programas de aplicación es separando los espacios de direcciones.

Atributos de descriptores de segmentos.

Los atributos de descriptores de segmento dan protección adicional a descriptores contenidos en la LDT (Tabla de Descriptores Locales) o en la GDT (Tabla de Descriptores Globales). Atributos que hacen que el sistema operativo controle las condiciones mediante los cuales un programa accesará a un segmento.

De acuerdo con las labores del descriptor (DPL) el nivel de privilegio del mismo definirá en qué nivel debe ejecutarse un código para accesar a la memoria que el descriptor haya definido.

De la Figura 3.8 si consideramos el núcleo del sistema operativo en PL0 y las aplicaciones en PL3, toda estrucutura del sistema operativo podría proyectarse en la GDT (Tabla de Descriptores Globales) marcándose con DPL0. Por lo que una aplicación (PL3) intentará accesar el segmento marcado, el 80286 marcaría un error. Pero una rutina de servicio del sistema operativo (PL0) si podría accesar al segmento.

Una forma de proteger a los datos es definir los derechos de acceso. Aquellos descriptores de segmento que sólo podràn ser leídos se crearàn en LDT o GDT. Si se intentara escribir en alguno de estos segmentos se marcaría un error. Por medio de estos atributos un sistema operativo asigna estructuras de datos directa sin que una aplicación escriba sobre ellas, sino únicamente las leerà.

De forma anàloga los descriptores de código sòlo son ejecutados, màs no examinados.

Protección de E/S:

En la arquitectura del microprocesador 80286 existen interrupciones (CLI,STI) que hacen posible habilitar e inhabilitar interrupciones y realizar operaciones de E/S (IN,OUT). Sin embargo estas funciones resultan riesgosas en proceso multitarea.

Los procesadores 80286 y 80386 implementan la noción de nivel de protección de E/S (IOPL) que ayuda a la administración sobre què programas podràn manipular hardware .

El IOPL define el mínimo anillo de protección en el cual un programa debe ejecutarse para realizar instrucciones de E/S (CLI,STI,IN,OUT). El sistema operativo pone el IOPL para definir cuàl de los programas, de mínima confianza, podràn accesar operaciones de E/S. Si retomamos nuestra figura 3.8 y colocamos

el IOPL en el anillo 1 aquellos programas que se ejecutan en PL2 y PL3 intentaran realizar una instrucción de E/S produciría un error , ya que sólo PL0 y PL1 podrían llevarla a cabo.

3.4 IMPLEMENTACION DEL OS/2

El sistema operativo OS/2 utiliza gran parte de las características del procesador 80286. entre estas el modo protegido y el modo real. Mientras que en el modo protegido corren programas hechos para interfaces de programación de aplicaciones del OS/2 (API). en el modo real corren programas de el entorno de compatibilidad. Provocando una intercambio de modos para correr los diversos tipos de programas.

Para dar un entorno de memoria virtual a los programas OS/2 utiliza al máximo la administración de memoria obtenida a través del 80286. haciendo posible que programas conjuntos del OS/2 no excedan la memoria física del sistema dado que éstos asignan dinámicamente segmentos de memoria. determinando cuál de ellos es usado con menos frecuencia e intercambiándolos al disco cuando es utilizada la memoria física. Realizado esto. los descriptores de segmento se marcan "ausentes". por lo que si una aplicación hace referencia a un segmento intercambiando la CPU presenta un error haciendo al OS/2 leer nuevamente el segmento de memoria. Para impedir que la memoria se particione. se hace una reducción constante de segmentos. logrando mayores áreas disponibles de

memoria física, además de que el intercambio de segmentos dentro y fuera de la memoria es mínimo.

OS/2 marca descriptores de aplicación como segmentos de código de datos utilizando sus atributos. Al momento de cargar segmentos del programa CODE desde el archivo de programas ejecutables del disco se crean los descriptores de segmento de código, de la misma forma son creados los descriptores de segmento de datos sólo que aquí se cargan los segmentos del programa DATA.

La Tabla de Descriptores Locales se crea y mantiene para cada tarea por el OS/2 ya que por medio de ésta los segmentos de un programa están siempre direccionados. La idea de realizar diferentes tablas de descriptores, es proteger a la memoria de posibles modificaciones.

Capitulo 4

ADMINISTRACION DE LA MEMORIA.

Una de las partes importantes de un sistema operativo es la que se encarga de la memoria y se le denomina administrador de la memoria, es el que lleva el control de. què partes de la memoria estàn en uso y cuàles no lo estàn, asigna memoria a procesos que lo necesiten y las retira cuando terminen. ademàs administra el intercambio entre la memoria central y el disco en el momento en que la memoria central no baste para contener todos los procesos.

El algoritmo de asignaciòn de la CPU està fuertemente influenciado por las tècnicas de administraciòn de la memoria. Por lo menos parte de un proceso debe estar contenido en la memoria principal para llevarse a cabo y la CPU no la ejecuta si existe por completo en la memoria secundaria.

Sin embargo, la memoria principal es un recurso valioso que frecuentemente no contiene a todos los procesos activos del sistema. Por ejemplo, si un sistema tiene ocho Mbytes de memoria principal. no caben simultàneamente nueve procesos de 1 Mbyte. El sistema de administraciòn de memoria decide què procesos deben residir (por lo menos parcialmente) en memoria principal, y administra las partes del àrea de direccionamiento virtual de un proceso que no son residentes en memoria. Monitorea la cantidad

de memoria principal disponible y escribe periódicamente procesos a un dispositivo secundario denominado "dispositivo de intercambio" para proporcionar mayor espacio de la memoria principal.

En OS/2 la adiministración de la memoria representa los pasos que sigue el sistema operativo para determinar cuánta memoria puede utilizar un programa. si OS/2 tiene que desplazar o no por momentos. segmentos hacia el disco para dejar espacio libre a otros segmentos "swapping". además de determinar si el sistema operativo puede o no realizar la compactación de memoria (haciendo movimientos de segmentos de código para no permitir una mala administración de espacio. así como la fragmentación). Cuando OS/2 translada segmentos de memoria al disco. permite que más programas se ejecuten de los que en realidad cabrían en ésta. Es de notar que cuanto mayor sea el número de programas en ejecución. mayor será el número de intercambios que OS/2 realizará. Sin embargo. ésto produce que el rendimiento baje significativamente debido a las diversas operaciones de E/S que se realizan. para lo que OS/2 permite desactivar el intercambio. además el número de programas que se realicen concurrentemente. es limitada por OS/2.

Existe un problema en cuanto a entornos de multitarea se refiere. y éste es el de la fragmentación de la memoria. Tomemos en cuenta el siguiente ejemplo:

Supongamos que OS/2 ejecutará tres programas (A, B y C), como se observa en la Figura 4.1. Se carga uno de los programas en memoria (Figura 4.2). Tiempo después, finaliza el programa A, que es eliminado de la memoria por OS/2, vease la Figura 4.3. Seguido de ésto OS/2 necesita cargar el programa D. Pero OS/2 no tiene la suficiente memoria para cargar el programa. Al desplazar las otras dos aplicaciones hacia abajo en la memoria, OS/2 obtiene el espacio necesario. (Véase la Figura 4.4).

El modelo de memoria virtual de OS/2 realiza la compactación de la memoria.

Existe un dato "MEMMAN" del archivo CONFIG.SYS que permite activar o desactivar el intercambio y la administración de la memoria.

Como se ha mencionado anteriormente el núcleo de OS/2 asigna y mantiene tablas de descriptores correspondientes a la utilización de la memoria física, proporcionando sus servicios para administrar espacios de memoria, sin considerar sus posiciones reales. La Figura 4.5 muestra el mapa (un mapa es una lista en donde cada entrada consiste de una dirección de un recurso asignable y un número de unidades disponibles ahí) de memoria de OS/2 que indica la asignación del entorno del DOS y áreas reservadas al BIOS en la memoria inferior, una porción del

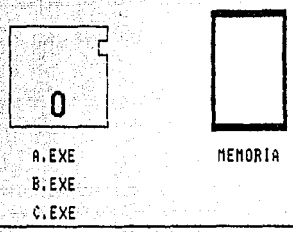


FIGURA 4.1 TRES PROGRAMAS A EJECUTAR

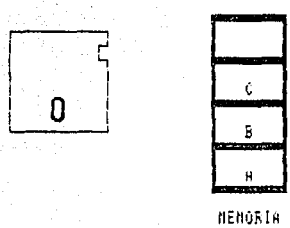


FIGURA 4.2 EL OS/2 CARGA TRES PROGRAMAS EN MEMORIA

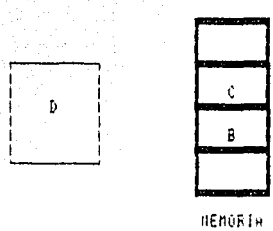


FIGURA 4.3 EL PROGRAMA A ES ELIMINADO DE LA MEMORIA DESPUES DE SU EJECUCION

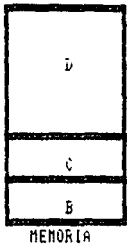


FIGURA 4.4 LOS PROGRAMAS B Y C SON DESPLAZADOS HACIA ABAJO PARA DEJAR SITIO PARA D

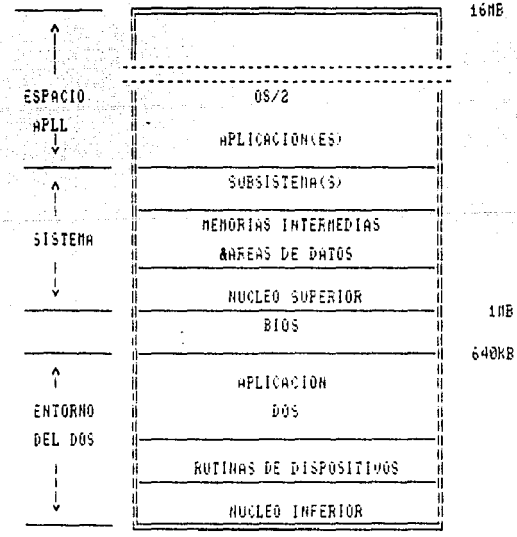


FIGURA 4.5 MAPA DE MEMORIA DEL OS/2

sistema está contenida en el entorno del DOS. el núcleo inferior. rutinas de dispositivos y algunas áreas de datos se encuentran asignados fuera de su entorno.

En OS/2 el usuario define la cantidad de memoria reservada para ejecutar programas de aplicación y también la que se dejará para las aplicaciones del OS/2. Lo anterior se aprecia en la Figura 4.6.

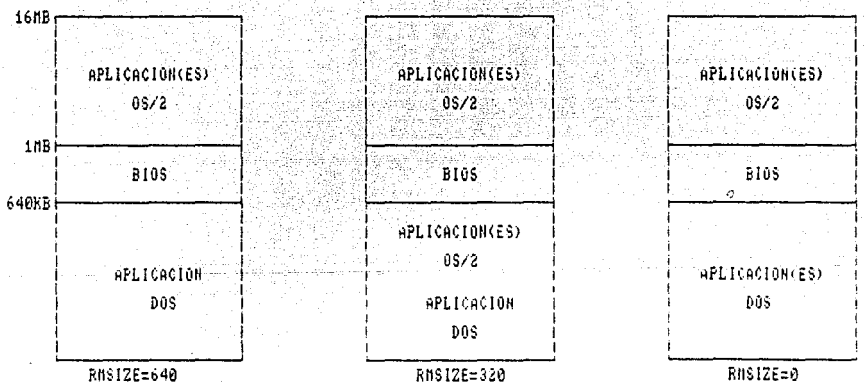


FIGURA 4.6 OPCIONES DE CONFIGURACIÓN DE MEMORIA

De el ejemplo de la Figura 4.6 en el primer caso (RMSIZE=640).el sistema se configura para que toda la memoria disponible por abajo de 1 MB (640K) sea asignada al entorno del DOS. Las aplicaciones de MS DOS tendràn completo acceso al espacio de direcciones en modo real. En el segundo caso (RMSIZE=320) se configura una divisiòn de la memoria inferior entre programas de aplicaciòn en modo real y modo protegido. obsèrvese que los segmentos de aplicaciòn en modo protegido se encuentran en la parte inferior de 1 MB del sistema de memoria y que por ser direccionables por aplicaciones del MS DOS. corriendo en su entorno podrian llegar a ser modificados de manera inadvertida. En el tercer caso (RMSIZE=0) no hay entorno de compatibilidad del DOS. Siendo la memoria utilizada por el OS/2 para correr programas en modo protegido. Es de considerarse que ésta es la configuraciòn màs segura, dado que todos los programas estàn protegidos entre sì. debido a las facilidades que el microprocesador 80286 da para la protecciòn de memoria.

El hecho de que la memoria se intercambie significa que el còdigo de aplicaciòn. datos y segmentos de pila se intercambien con el disco o bièn sean desplazados en memoria. ya sea por la frecuencia con que sean utilizados o la demanda que tengan en la memoria real por otros programas del sistema.

4.1 ESPACIO DE DIRECCIONES DEL SISTEMA

En OS/2 la tabla de descriptores globales (GDT) administra el espacio de direcciones para los programas, espacio que contiene segmentos accesibles a procesos del sistema, entre ella se encuentra: el código del núcleo y los segmentos de datos, puntos de entrada API y código de la rutina de dispositivo y segmentos de datos.

La GDT (tabla de descriptores globales) es una región fija de memoria que contiene los descriptores que definen el espacio de direcciones del sistema.

4.2 ESPACIO DE DIRECCIONES DE LA APLICACION

Entre las características de OS/2 está la de brindar un espacio de direcciones <<local>> a cada programa de aplicación, éste es asignado por una Tabla de Descriptores Locales (LDT)[1]. cada segmento que esté relacionado con una subrutina ejecutable del programa se encuentra direccionado por su Tabla de Descriptores Locales. OS/2 ajusta el tamaño de la LDT de acuerdo a las necesidades mismas del sistema. Algo importante que hay que mencionar es, que el conjunto único de segmentos para cada programa no es alterado por otros programas del sistema, dado que para esto OS/2 da una protección adecuada de memoria, la cual crea un espacio exclusivo de direcciones para cada aplicación.

---

1 La LDT es una región de memoria que contienen los descriptores que definen el espacio de direcciones de la aplicación.

4.3 CARGA DE APLICACIONES

Cuando finaliza un programa OS/2 asigna e inicializa una nueva LDT para otro programa. dado que a OS/2 le corresponde la administración de la memoria para cada aplicación. en el momento de cargar el programa crea una serie de entradas de acuerdo con cada uno de los segmentos que se hayan definido en el. cuando un programa ya ha sido utilizado frecuentemente o bién utiliza una rutina del mismo. lo que sucede entonces. es que se crea una entrada a la LDT que apuntará al segmento ya existente en la memoria física. cargando así. sólo. una copia del código. para lo cual la memoria real del sistema no necesita estar cargada físicamente.

4.4 ASIGNACION Y DESIGNACION DE MEMORIA

Un problema relacionado con el manejo de espacio virtual es el de la asignación. Si se cuenta con espacios múltiples. entonces utiliza en cada espacio el método de la asignación simple. donde se asigna a un sólo usuario todo el espacio disponible que el sistema operacional deja.

De igual forma. si se cuenta con un espacio único. se utiliza cualquiera de los métodos de asignación simple (particiones fijas. particiones variables. etc.). En el caso de un sistema con segmentación. si se usan particiones fijas. habrá fragmentación interna. tanto en el espacio virtual. como en el espacio real ya que quedan espacios de memoria desocupada.

Existen dos motivaciones para la introducción de la memoria virtual: liberarse de las restricciones físicas y hacer un mejor uso de la máquina. Las dos principales limitaciones que tiene la memoria física de una computadora, son su tamaño y su unidimensional. La primera hace que no se puedan correr programas más grandes que la memoria principal, o que para hacerlo se requiera de mecanismos sofisticados e incómodos para el usuario. La segunda forza a representar en forma unidimensional estructuras que no lo son.

La memoria virtual permite hacer mejor uso de los recursos de la máquina dado que hace posible cargar en la memoria más programas de los que se cargarían en un sistema tradicional, y con ésto la memoria y el procesador se utilizan más eficientemente. La idea consiste en no cargar todas las partes de un programa, sino únicamente aquellas que se necesitan, haciéndolo de forma automática.

Para implantar la memoria virtual es necesario contar con un mecanismo que traduzca las direcciones virtuales en direcciones reales. Una primera aproximación podría ser como se muestra en la Figura 4.7.

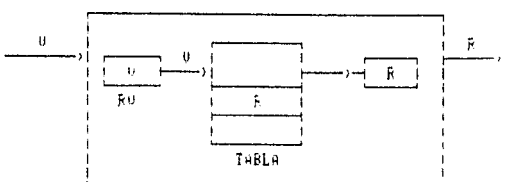


FIGURA 4.7 UNA POSIBLE IMPLANTACION DEL MECANISMO DE TRADUCCION DE DIRECCIONES

Sin embargo, para que el método anterior funcione correctamente, se requiere que la tabla tenga N entradas (en donde N es el tamaño de la memoria virtual). Dado que conduciría a un consumo muy grande de la memoria (la tabla sería más grande que la memoria), otra opción sobre el método es haciendo una búsqueda por contenido, como se observa en la Figura 4.8.

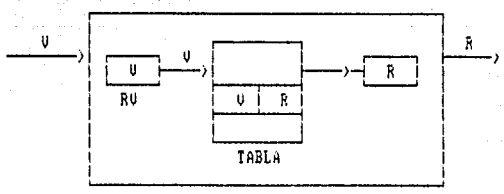


FIGURA 4.8 UNA IMPLANTACION OPTIMIZADA DEL METODO DE TRADUCCION DE DIRECCIONES

En el esquema anterior las búsquedas en la tabla se hacen por contenido: se busca en la primera parte de cada entrada la dirección virtual V, y una vez hallada, en la segunda parte de la entrada correspondiente se encuentra la dirección real R. De ésta manera, el tamaño de la tabla se reduce cuanto se quiera.

Otra forma posible de reducir el tamaño de la tabla es agrupar las posiciones de la memoria en bloques, haciendo que el mecanismo no traduzca la dirección de una posición sino de un bloque. Como los bloques se pueden definir tan grandes como se

quiera. el número de entradas también son reducidas tanto como se requiera. Una vez localizada la dirección física del bloque. se debe proceder a encontrar la dirección real dentro de éste.

Las ideas anteriores nos permiten comprender las tres formas básicas de direccionamiento utilizadas en los sistemas de memoria virtual: segmentación pura. paginación pura y segmentación-paginación.

La dirección virtual se compone de dos partes: segmento y desplazamiento. La primera sirve para indizarse dentro de la llamada de tabla de segmentos. en donde se encuentra la dirección del segmento correspondiente. A ésta se le suma el desplazamiento para obtener la dirección de la memoria real. Dentro de cada entrada. en la tabla de segmentos se encuentra también un campo que indica la longitud del segmento (L) el cuál permite verificar que no se tengan referencias fuera del segmento. En caso de que ésto ocurra se produce una interrupción (lo anterior puede apreciarse en la Figura 4.9).

Las principales ventajas del método es que evita la fragmentación interna (dado que los segmentos se definen del tamaño que se requiera haciendo que no haya desperdicios internos). y que facilita la implantación de mecanismos de protección basados en la memoria virtual. ya que los segmentos son entidades lógicas a las cuales se les asocia una protección.

En OS/2 los programas pueden asignar. designar. alargar y

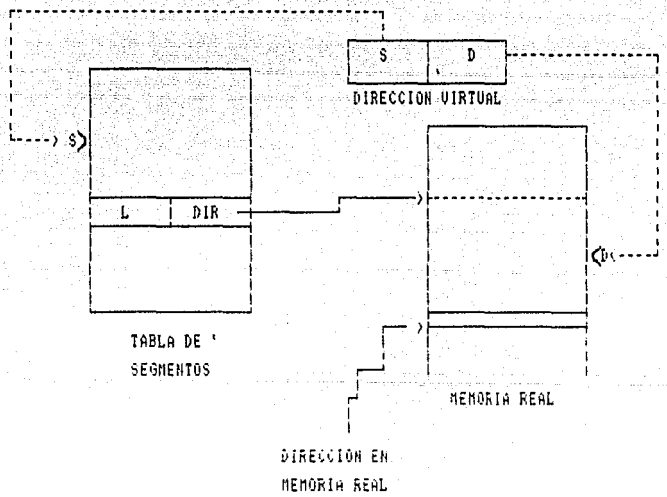


FIGURA 4.9 ESTRUCTURA DEL MECANISMO DE TRADUCCION DE DIRECCIONES EN UN SISTEMA CON SEGMENTACION PURA

estrechar segmentos de datos utilizando llamadas API del OS/2. Las llamadas usadas en el sotware de OS/2 se muestran en la tabla siguiente con una descrioción de uso.

Llamadas API	Descripción
DosCreateCSAlias	Crear subnombres ejecutables para cada segmento de datos en particular.
DosMemAvail	Devolver el tamaño del bloque mayor que se encuentre libre en la memoria
DosAllocSeg	Asigna un segmento en memoria
DosReallocSeg	Termina o alarga un segmento de memoria
DosFreeSeg	Designa un segmento de memoria
DosAllocShrSeg	Asigna un segmento llamado de memoria compartida.
DosGetShrSeg	Ganar acceso a un segmento llamado de memoria.
DosGiveSeg	Dar otro proceso y segmento compartido.
DosAllocHuge	Asignar varios segmentos de memoria.
DosGetHugeShift	Obtener un incremento del selector (en segmentos grandes).
DosReallocHuge	Termina o alarga segmento(s) grandes de memoria.

De acuerdo con las ventajas de la memoria virtual y en caso específico para OS/2 al utilizar la llamada API DosAllocSeg que de acuerdo con la tabla, asigna un segmento de memoria, lo que el administrador hace es, buscar un espacio libre en la memoria física, del tamaño que en el caso sea requerido, al encontrar el espacio, este es marcado, se inserta el dato en la LDT y es transferido al selector asociado nuevamente al programa. Cuando se designan espacios de memoria, OS/2 marca descriptores libres de LDT y libera la memoria física.

Si se considera que el tamaño máximo de un segmento es de 64K, entonces de uno depende alargarlo o acortarlo de tamaño, de acuerdo con los requerimientos del usuario y considerando el límite del segmento.

4.5 MEMORIA COMPARTIDA

La memoria compartida es la forma más simple de comunicación entre procesos (IPC). Donde se asignan un segmento y se hace disponible a otros programas del sistema, al compartir un segmento de memoria, OS/2 incrementa un contador que se encuentra relacionado con dicho segmento, al terminar un programa se liberan ciertos segmentos de memoria ocasionando que el contador se decremente, cuando el contador es igual a cero, entonces el administrador de la memoria de OS/2 libera un segmento de memoria compartida.

La memoria compartida reserva un espacio determinado, en el que dos o más procesos la accesan. lo anterior se observa en la Figura 4.10

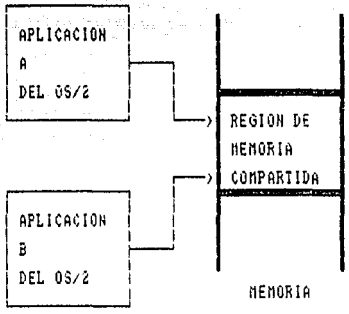


FIGURA 4.10 MEMORIA COMPARTIDA

Sin embargo, en la memoria compartida es necesario que los programas de comunicación elaboren sus propios protocolos internos, dado que no hay un control de acceso a ésta, un proceso indistinto puede accesar al espacio de la memoria compartida y modificar los segmentos en cualquier momento.

La memoria compartida la podemos dividir en dos tipos:

* Memoria global compartida

* Memoria local compartida

Al primer tipo de memoria se le llama una cadena nula terminada. ASCII (ASCIIZ). Los segmentos de memoria global compartida son accesibles a todos los programas del sistema. así como útiles para àreas de datos comunes al mismo. además de ser accesibles. los segmentos son accesados por nombre. OS/2 establece direccionabilidad para el segmento de memoria compartida en la LDT del programa que ha hecho la petición. en la segunda. la memoria compartida local. sólo dos programas se accesan al segmento de datos de la memoria. Con èsta memoria dos programas comparten memoria en privado. Aquí. la forma en que se accesan los segmentos es por indicativo (handle). el indicativo no es màs que un selector que es vàlido solamente en la LDT de un segundo programa. Al hacer uso de esta memoria dos programas pasan datos a través de un segmento de datos que tiene selectores con subnombres en ambos espacios de direcciones de programas. El selector de memoria compartida es sólo válido en la LDT del programa a la que fuè dado el segmento.

Ambas memorias. memoria global y local compartida utilizan un algoritmo de cuenta de uso. en el cual. cada vez que un programa accesa o da un segmento de memoria compartida. OS/2 incrementa el número de veces que ha sido utilizado ese segmento y èsta se decrementa al liberar los segmentos un programa.

4.6 MOVIMIENTO DE SEGMENTOS

De acuerdo con las características de la memoria virtual, y a la arquitectura del procesador 80286 que es segmentada, que da la oportinidad de tener segmentos de tamaño variable que van desde 164K bytes[1] de longitud, fragmentàndose asi màs fàcilmente la memoria. En caso de no haber actividad podrían crearse espacios de memoria sin utilizar.

Considerar el ejemplo de la Figura 4.11.

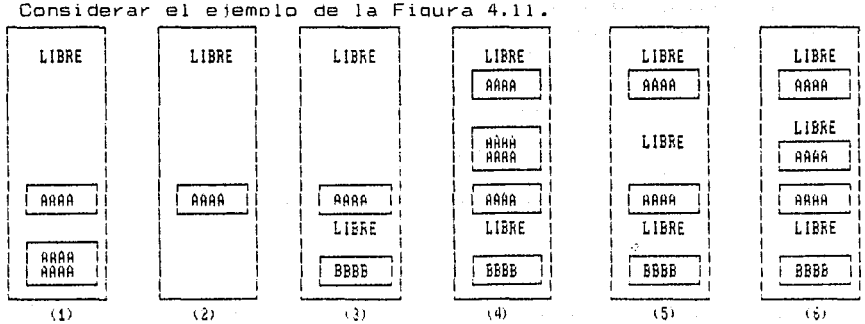


FIGURA 4.11 FRAGMENTACION DE MEMORIA

---

1 El sistema de numeración binario representa los números como una serie de unos y ceros llamados bits. Si se activa un bit, tiene el valor de 1; en caso contrario el bit es 0. Dentro de la computadora, el bit se utiliza para representar la presencia o ausencia de una señal electrónica. Si el bit está activo, la señal está presente (on), si el bit está desactivado, la señal está ausente (off). Ocho dígitos binarios (10001010) forman un byte y puede almacenar valores comprendidos entre 0 y 255. A menudo se hace referencia a los bytes en términos de "K" (164K, por ejemplo). Para simplificar se considera un "K" como el valor 1024, por tanto 164k proporcionan un espacio para :

164 * 1024 = 167.936 bytes

Un MegaByte equivale a un millón de bytes, entonces 1.5MB equivale a 1.5 millones de bytes de información.

Del ejemplo de la Figura 4.11 se explica la numeración inferior de la misma.

1) El programa A asigna dos segmentos de datos de 200 y 100 bytes de longitud.
2) Se desasigna el segmento de 200 bytes.
3) El programa B se ejecuta a la par que el A, y se asigna un nuevo segmento de 100 bytes.
4) Supongamos que el programa A asigna un segundo conjunto de segmentos de 100 y 200 bytes, sin embargo el espacio para el segmento de 200 bytes no es suficiente, ya que el sistema asigna el segmento fuera del área contigua de memoria disponible y libre.
5) Nuevamente A libera el segmento más grande de los dos asignados anteriormente.
6) El programa B (o bien otro programa) asigna un segmento de 100 bytes.

Puede observarse que la memoria disponible se dispersa entre segmentos de varios tamaños, creando una infinidad de pequeños huecos que satisfagan a las peticiones intermedias de asignación de memoria.

Un problema que surge en el ejemplo, es el de la fragmentación, para dar solución a éste, OS/2 implementa el movimiento de segmentos. En el momento en que se solicite hacer

un trabajo de memoria que no llegue a ser realizada debido al espacio de memoria libre. OS/2 vuelve a organizar segmentos para crear espacios más grandes y libres en la memoria.

Regresemos al ejemplo de la Figura 4.11 nuevamente: consideremeos los tres primeros puntos de igual forma, y modifiquemos los restantes:

1) El Programa A siona los mismos segmentos, de 100 y 200 bytes.
2) Nuevamente se desasigna un segmento.
3) Como antes el programa B asigna su segmento de 100 bytes.
4) El programa A asigna su conjunto de segmentos, el sistema detecta el fragmento extenso, reorganiza los segmentos creando espacios libres utilizados para satifacer la solicitud hecha por A.
5) Nuevamente A libera el segmento más grande de los dos.
6) El programa B (o bien otro programa) asigna un segmento de 100 bytes.

Véase la Figura 4.12

Este ejemplo muestra la eficiencia adquirida de OS/2 para una mejor utilización de los recursos de memoria del sistema. Ya que de no haber utilizado la compactación de memoria, el sistema habría mandado un mensaje que alertara sobre el uso excedido de memoria.

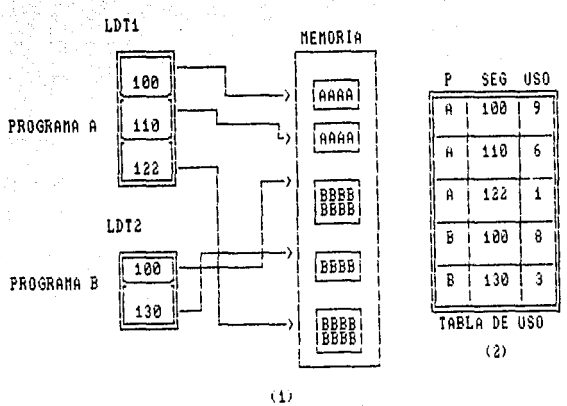


FIGURA 4.12 INTERCAMBIO DE SEGMENTOS

4.7 INTERCAMBIO DE SEGMENTOS

De acuerdo con las limitaciones de la memoria física en cuanto a tamaño y unidimensionabilidad que se han resuelto con la memoria virtual, además de la noción de segmento, que incrementa la eficiencia de los servicios del administrador de memoria de OS/2, existen programas y sus respectivos segmentos de memoria que se encuentran inactivos al no existir una interacción con el usuario. De acuerdo con la Tabla de Descriptores de Segmento,

los segmentos de memoria están definidos por un descriptor en una de las tablas de descriptores del sistema, donde cada segmento tiene atributos particulares, uno de éstos es el indicador de segmento no presente que, si es igual a 1, entonces el procesador 80286 genera una interrupción cuando algún programa le haga referencia.

En OS/2 se utiliza éste atributo para implementar un espacio de direcciones virtuales, de tal forma que al asignar un programa segmentos al administrador de memoria son creadas una serie de entradas a la LDT.

Si se diera el caso que no cupiera un segmento en la memoria contigua libre, aún cuando se realizara un intercambio de segmentos, entonces lo que hará el administrador será colocar uno de los segmentos activos fuera de la memoria y así realizar el servicio requerido. Para no cometer algún error al momento de hacer intercambios OS/2 lleva una tabla en la que se registra el número de referencias a un segmento, para que al momento de requerirse un mayor espacio de memoria física, se identifique de inmediato qué segmentos son los que tienen una utilización mínima y así transferirlo al archivo donde se realizan los intercambios del sistema; a la par, OS/2 realiza una actualización al descriptor asociado con el segmento con el fin de indicar que dicho segmento no está presente, por lo que al momento de que un programa intente hacer una referencia a este segmento, se

generará una interrupción de segmento no presente, lo cuál permite que OS/2 elimine si es necesario otros segmentos para hacer la recuperación de el segmento al disco. en èste momento el descriptor tendrá un valor de cero y el programa podrá continuar como si el segmento siempre hubiera estado presente en la memoria.

Considerar el ejemplo de la Figura 4.14. la numeración inferior de la misma se explica a continuación.

1) La memoria se encuentra asignada para los programas A y B (Vèase Figura 4.13). Es cargado un nuevo programa, C, que requiere 100 bytes de memoria, pero en este momento no se puede cargar el programa dado que OS/2 tiene asignada toda la memoria.

2) Se realiza una exploración en la tabla interna del administrador de memoria, determina que el segmento 122 corresponde al programa A, es el menos utilizado puesto que tan sòlo es referenciado una vez, además de que el administrador se percata del espacio tan grande que ocupa, espacio que puede ser utilizado para satisfacer los 100 bytes requeridos por el programa C (Figura 4.13).

3) Se realiza el intercambio del segmento A.122 al disco, marcándole en el descriptor de la LDT como segmento no presente de A, en caso de que èste segmento sea utilizado por varios programas, OS/2 actualiza cada una de las LDT de estos programas (como lo muestra la Figura 4.13)

4) Habiendo el espacio suficiente, puede cubrirse el servicio de asigación de memoria. se crea una LDT del programa C dàndole su dirección base del descriptor para asi tener la posición exacta de la memoria física (Figura 4.13).

5) El programa A hace referencia al segmento intercambiado. Sin embargo como èste tiene un descriptor con valor de 1. el procesador 80286 genera una interrupción de segmento no presente.

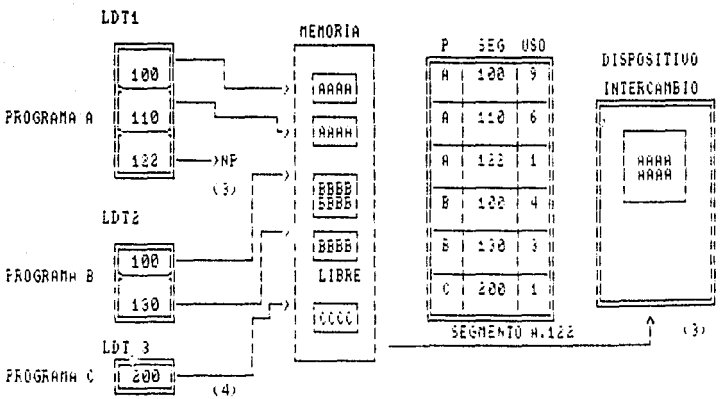


FIGURA 4.13 CARGA DEL PROGRAMA C

6) La interrupción realizada permite entonces a OS/2 recuperar el segmento del dispositivo de intercambio devolviéndolo a la memoria, para lo cual se repiten los pasos 2 y 3, pero ahora realizando el intercambio del segmento de 100 bytes de B para aumentar el espacio en la memoria (observar Figura 4.14).

El intercambio de segmentos es parecido a otro concepto llamado paginación, aunque éste tiene algunas diferencias.

Aunque el intercambio de segmentos proporciones un espacio de direcciones virtuales es diferente en varios aspectos a un concepto denominado paginación.

En una arquitectura de administración basada en paginación, el hardware que administra la memoria divide a ésta en bloques del mismo tamaño denominados páginas. Los tamaños típicos varían entre 512 bytes a 4 bytes y son definidos por el hardware. Cada localidad direccionable en memoria está contenida en una página y, consecuentemente, cada localidad se direcciona por un número de página y un byte de desplazamiento dentro de la página.

Cuando el núcleo asigna páginas físicas de memoria a una región, no requiere asignar páginas continuas o en un orden específico. El propósito de la memoria paginada es el de permitir mayor flexibilidad en la asignación de memoria física, análogo a la asignación de bloques de disco a los archivos en un sistema de archivos. De la misma manera en que el núcleo asigna bloques a un

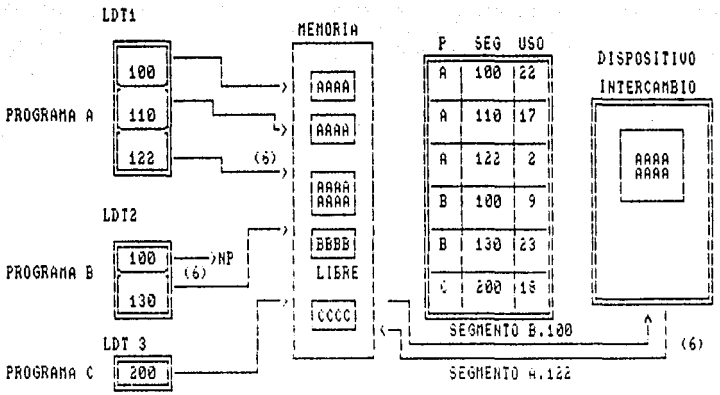


FIGURA 4.14 GENERACIÓN DE UN FALLO DE PROCESADOR DE SEGMENTO-NO-PRESENTE

archivo para incrementar la flexibilidad y reducir la cantidad de espacio no utilizado por la fragmentación de bloques. asiona páginas de memoria a una región.

El núcleo asocia las direcciones virtuales de una región a sus direcciones físicas relacionando los números lógicos de páginas en la región a números físicos en la máquina. como se

muestra en la siguiente tabla:

Número de página Lógico	Número de página Físico
0	177
1	54
2	209
3	17

Debido que una región es un rango continuo de direcciones virtuales en un programa, el número lógico de página es el índice a la lista de números físicos de páginas. La entrada de la tabla de regiones contiene un apuntador a la tabla de números físicos de páginas llamada tabla de páginas. Las entradas de la tabla de páginas también contienen información dependiente de la máquina tal como bits de permiso que autorizan la lectura o escritura de una página. El núcleo almacena las tablas de páginas en memoria y las accesa como cualquier otra estructura de datos propia del núcleo.

En el proceso de paginación pura (véase la Figura 4.15), no existe un campo de longitud en la tabla, llamada ahora tabla de páginas. Esto hace que los bloques, denominados páginas, tengan una longitud fija, e igual a la del máximo desplazamiento posible (ya que no hay verificación sobre la longitud).

La gran ventaja de este método es que evita la fragmentación externa dado que los bloques en la memoria real son de longitud fija, y por consiguiente , en cualquiera de ellos se coloca una

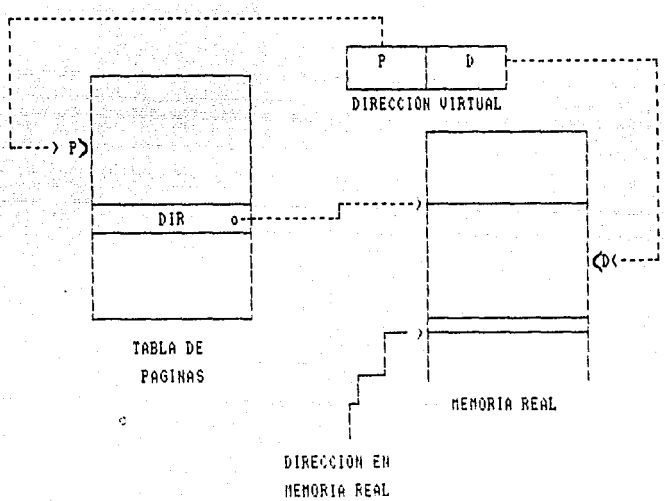


FIGURA 4.15 ESTRUCTURA DEL MECANISMO DE TRADUCCION DE DIRECCIONES EN UN SISTEMA CON PAGINACION PURA

página de la memoria virtual.

Sin embargo, lo anterior presenta dos inconvenientes: por un lado, las páginas son de longitud fija y arbitraria, no corresponden a ninguna entidad lógica y por otro lado, al tener que asignar a un programa una cantidad entera de páginas de longitud fija, se produce la fragmentación interna, la cual

consiste en que un programa no utiliza todo el espacio que le ha sido asignado.

La segmentación-paginación se implanta utilizando un mecanismo como el que muestra la Figura 4.16.

En el que las direcciones virtuales se componen de tres partes: segmento, página y desplazamiento. La primera sirve para indizarse en una tabla de segmentos, en la cuál se encuentran la longitud y dirección de la tabla de páginas, la segunda sirve para indizarse en dicha tabla, donde se encuentra la dirección de la página correspondiente; al tener el dato de ésta y el desplazamiento se llega a la dirección en memoria real.

El método de segmentación-paginación permite mantener una estructura de segmentación sin conducir a la fragmentación externa (aunque inevitablemente tendrá fragmentación interna). Una ventaja adicional es que permite mantener las tablas de páginas en la memoria auxiliar. Existe un sistema de traducción de dos niveles con bloques de tamaño fijo en cada uno de ellos. Al utilizar este nuevo esquema se pierden las ventajas asociadas con el manejo de segmentos como entidades lógicas, pero se conservan las demás características del método segmentación-paginación. Así mismo existe para cada página o segmento de la memoria real el llamado bit de referencia que se activa en el hardware cada vez que la página es referenciada. En forma similar

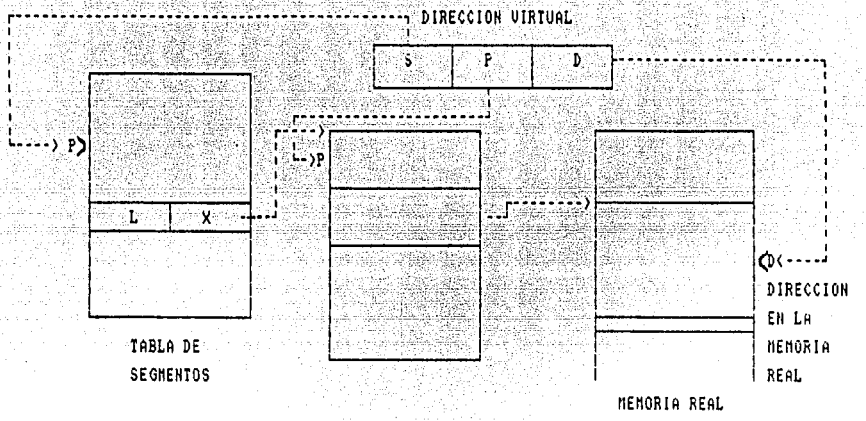


FIGURA 4.16 ESTRUCTURA DEL MECANISMO DE TRADUCCION DE DIRECCIONES EN UN SISTEMA CON SEGMENTACION-PAGINACION

existe un bit de cambio que se activa cada vez que se modifica la página.

La traducción de dirección funciona de la siguiente manera: se busca en la tabla correspondiente la dirección de la página o el segmento. si se encuentra, se calcula la dirección en la forma apropiada en cada caso; si no está, se produce una interrupción de defecto de página o defecto de segmento y el sistema operativo

se encarga de traer la página a la memoria. Este proceso ocurre dado que las páginas o segmentos no se tienen presentes conjuntamente en memoria.

La paginación se utiliza como alternativa para evitar el uso de algoritmos de compactación, puesto que en la paginación siempre se involucran <<páginas>> de memoria de igual tamaño. Sin embargo, en el intercambio de segmentos y compactación de los mismos una adecuada selección sería las opciones de configuración del sistema operativo.

4.8 CARGA POR DEMANDA

Cada programa de OS/2 contiene diversos segmentos, que se encuentran en conjunto en un archivo ejecutable. Al cargarse un programa en la memoria de OS/2 se construye un descriptor LDT para cada segmento del programa. En OS/2 la llamada "carga por demanda" (load on demand), carga los segmentos hasta el momento en que estos son referenciados y no antes. Sólo al crearse un archivo ejecutable, el segmento es marcado como <<cargado por demanda>>, creándose así entradas LDT de estos segmentos por el sistema, además de que el descriptor los marca como no presentes, por consiguiente, si un programa intentara referenciar uno de ellos, se generaría una interrupción.

Una forma en que una aplicación se inicializa rápidamente y consume menos memoria sería cargar por demanda aquellas rutinas utilizadas con menos frecuencia.

4.9 SUBASIGNACION DE MEMORIA

Hay casos en los que las aplicaciones solicitan y liberan espacios de memoria, tal es el caso de los bloques de control. para realizar èsta solicitud y liberarla. OS/2 construye la entrada LDT. actualiza el descriptor, localiza la memoria física con sus segmentos, intercambia segmentos, libera la memoria física y actualiza el descriptor. Sin embargo, OS/2 presenta una alternativa para facilitar èsto, llamada el paquete de asignaciòn de memoria (MSP), el cual permite colocar y liberar porciones de un segmento de memoria (subasignado). Las rutinas de este paquete asignan y designan memoria, fragmentando segmentos de memoria previamente asignados en partes màs pequeñas. el subsistema MSP no realiza una asignaciòn y designaciòn, sino que, busca partes de los segmentos de memoria (estas partes estàn contenidas en un segmento) e identifica que se están utilizando y cuales no lo están. Hay que tener cierto cuidado al utilizar la subasignaciòn (asignacioin y desasignaciòn) en una aplicaciòn, ya que en algunos casos es mejor la memoria asignada y en otros, los segmentos deberían ser asignados por el sistema, puesto que antes de utilizarla es necesario determinar: la frecuencia de asignaciòn y desasignaciòn de memoria, que cantidad de memoria utiliza y la protecciòn que debe dar a la memoria.

Capitulo 5

INTERFAZ DE PROGRAMACION Y MULTITAREA.

OS/2 es un sistema operativo construido para cubrir los requerimientos de una nueva generación de programas de aplicación de PC's. en los siguientes puntos se muestra cómo las partes individuales del sistema se unen para su construcción. El primero de éstos puntos es la Interfáz de Programación de Aplicaciones (API). que es un diseño que precisa las capacidades con que cuenta el sistema. así cómo las partes que lo conforman.

5.1 CARACTERISTICAS DE API (Interfaz de Programación de Aplicaciones)

API se denomina a la colección total de funciones del sistema. OS/2 fué creado a partir de una serie de conceptos viejos y nuevos que en conjunto forman un sistema.
A su vez se define como toda capacidad contenida en el sistema además de ser parte esencial de la estructura general del OS/2.

En OS/2 la API utiliza un modelo de programación de "llamada". para llamar las rutinas de servicio del sistema.

Cada API está conectada a una función que se activa de acuerdo con los requerimientos del usuario y que corresponde a un punto de entrada del sistema.
A la API se ingresa de igual manera que se hace cuando se llama a

una subrutina, a través de una instrucción de "llamada lejana" (far call). Existe una serie de llamadas que cierran un archivo abierto donde los parámetros, son parámetros en pila: la transferencia de control de dicha información se hace directamente llamadando a la función que realiza éste proceso. una vez que se ejecuta, regresa el código de error de la función al registro AX. De manera general, al llamar a una subrutina, el compilador de lenguaje de alto nivel genera un código, en OS/2 tan sólo se hace un enlace y las funciones se ejecutan de manera directa. Es decir, si se programa con lenguajes de alto nivel, las llamadas a funciones de OS/2 se hacen de manera directa cómo si se tratase de una subrutina.

La interfaz de llamada se implementa utilizando una característica denominada "enlace dinámico" , para explicarla, es necesario decir que, en todos los sistemas operativos basados en disco tienen un componente responsable de leer los programas en disco y cargarlos en memoria. ha éste componente se le llama "cargador" y forma parte del núcleo del sistema, el cuál es utilizado para arrancar todos los programas (incluyendo aplicaciones, subsistemas y rutinas de dispositivos), es la interfaz simple para todas las funciones de arranque de programas. Cómo los programas pueden tener múltiples códigos y segmentos de datos, el cargador resuelve a su vez referencias solicitadas entre los segmentos, una de las técnicas utilizadas por el cargador de OS/2 para resolver dichas referencias es el

enlace dinàmico.

OS/2 extiende la función del cargador permitiendo a los programas referenciar segmentos no incluidos en el archivo llamado EXE.[1] En èste tipo de enlace el cargador resuelve referencias EXE a un segmento incluido en bibliotecas especiales llamadas "bibliotecas de enlace dinàmico" (DLL).

Una referencia lejana (far call) a un segmento hace que el cargador traiga la DLL a memoria, còmo si fuera parte del archivo EXE del programa. Un archivo DLL tiene el mismo formato y estructura que un archivo EXE . sus rutinas son comunes y se llaman por cualquier programa de OS/2.

El enlace dinàmico permite que los programas tengan referencias externas a segmentos que se incluyen en el archivo del programa (EXE). El enlace entre el programa que llama y la subrutina llamada es inicializado por OS/2 al momento de cargar el programa. Entre algunas de las ventajas que se dan al utilizar en la API del OS/2 el enlace dinàmico se encuentran:

* Si el usuario por algún interès de programación decidiera cambiar a otra versión del OS/2. no se afectarían sus

---

1. Cualquier nombre de archivo con una extensión . COM .EXE o como FORMAT.EXE y DISKCOPY.EXE son comandos externos. dado que también son archivos. pueden crearse nuevos archivos y agregarse al sistema. Los programas que se crean con la mayoría de los lenguajes (incluyendo el lenguajes ensamblador) serán archivos ejecutables (.EXE).

programas actuales. dado que la API cambia entre versiones sin alterar la programación existente.

* Hay una serie de funciones del sistema operativo, que no son servicios del núcleo y que si se requiere se ejecutan fuera del mismo, en bibliotecas de enlace dinámico denominadas subsistemas.

Cuando un programa se carga en memoria también se une, uno de aplicación a las rutinas de servicio del sistema. Con la API se obtiene un acceso directo de funciones, en el momento en que una aplicación le llama, la transferencia de control es administrada por la Unidad Central de Procesamiento (CPU), sin que exista la necesidad de que el sistema intervenga de manera directa.

Para saber cuales son las bases de la API es necesario considerar la arquitectura del Intel diseñada para OS/2. El microprocesador 80286 ofrece a un sistema, una serie de bloques de construcción, siendo uno de éstos la "puerta de llamada 80286"[1]. Algunas funciones importantes de OS/2, cómo administración de memoria y tareas se ejecutan en el núcleo, con el fin de separalas lo más posible de los programas de aplicación. El núcleo corre a un nivel de privilegio del 80286 (cero), aquellas funciones de OS/2

---

1. Una puerta de llamada da la oportunidad a un programa de llamar a otro con un nivel de "aislamiento" asignado por la Unidad Central de Procesamiento.

que son subrutinas de aplicación común son accedidas por la API pero su código corre a un nivel similar que el del programa de aplicación, así ésta llama en forma directa al núcleo porque la API va a través de una "puerta de llamada".

5.2 PROCESO DE TAREAS

Para entender tanto la estructura como el funcionamiento básicos de un sistema operacional, especialmente las actividades concurrentes en él, es fundamental introducir el concepto de proceso. Existen muchas formas de definirlo: una de ellas será: es una actividad asincrónica, o bien, el espíritu animado de un procedimiento[1], etc. Sin embargo, la definición más utilizada es la que dice que un proceso es la ejecución de un programa en la computadora. Para aclarar un poco esto, se expone el siguiente ejemplo: si suponemos que un usuario quiere correr la nómina de su empresa en computadora, para llevarlo a cabo el usuario debe realizar un programa (que puede incluir ordenamiento y manejo de archivos, impresión, etc.) en el que posiblemente llama a varios procedimientos (liquidaciones, cuota del seguro, aumento o disminución de horas, etc.). Al correrse se convierte en un trabajo (asociado a un usuario), que al materializarse en la computadora debe correr por cuenta de un proceso[2], al que a su vez se le asignará el procesador para que sea ejecutado.

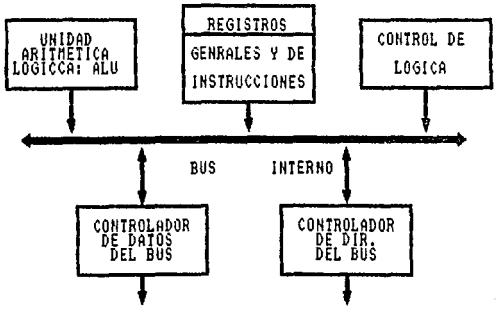
---

1 Deitel, Harvey M., An Introduction To Operating System. Capítulos 3,4,5,6,10 y 19. Ed. Addison-Wesley,1983.

2 Un procesador está compuesto de una serie de elementos utilizados para el manejo general de una computadora. Su estructura típica es:

Aunados a la noción de proceso existen otros términos semejantes a éste, como el de multitarea o multiprogramación (multitasking o multiprogramming), multiusuario (multiuser) y multiproceso (multiprocessing). El primero se refiere a aquellos sistemas que permiten la coexistencia de varios procesos dentro de una computadora, el segundo a los que permiten la interacción simultánea de varios usarios, y el tercero a los que manejan varios procesadores.

Entre las actividades de un sistema multitarea, en un sistema operativo está la de administrar recursos, siendo la CPU (Unidad Central de Procesamiento) de mayor importancia puesto que comparte diversos programas dando a cada uno su periodo de ejecución, denominado "intervalo de tiempo". una de las características de un sistema multitarea es que los programas se



ESTRUCTURA DE UN PROCESADOR

asocian con unidades repartibles que identifican los programas que el sistema operativo puede correr concurrentemente, algo similar a lo que ocurre en multiprogramación, donde la base es partir la memoria, de forma tal que cada partición realice una tarea diferente, por ejemplo, mientras que una tarea espera a que se complete la entrada y salidad de datos, otra tarea puede utilizar la CPU. Si se mantienen suficientes tareas en la memoria central a un mismo tiempo, la CPU se mantendrá ocupada casi el 100% del tiempo. Cada que se da fin a una tarea, el sistema carga una nueva del disco en la partición no vacia y la lleva a cabo.
Al hecho de controlar múltiples programas, distribuyendo los recursos proporcionados por el sistema y el tiempo del procesador que hay entre ellos, se le nombra "administración de tareas". Para ello OS/2 introduce un modelo de tareas que determina la norma utilizada para el manejo de programas que corren concurrentemente, su objetivo es proporcionar una serie de elementos para que los programas identifiquen unidades de trabajo repartibles, además de los recursos que estén asociados a èstas.

En el sistema multitarea hay ciertas unidades que se reparten y que identifican los programas que el sistema operativo puede correr al mismo tiempo.

OS/2 introduce algunos conceptos para su modelo de tarea.

Thread (hilo): Identifica una unidad de trabajo repartible o tareas independientes.

Proceso: Conjunto de uno o mas thread (hilos) y los recursos

asociados al sistema (memoria, archivo y dispositivos).

Sesión: Conjunto de uno o mas procesos asociados con una consola virtual (teclado, monitor, mouse).

En OS/2 thread (hilo) es la unidad básica de ejecución, cada proceso tiene cuando menos uno, proporciona códigos de programas que contiene valores de registros, stack (pila) y modo de CPU. Toda ejecución que gira a su alrededor es referenciada mediante el contexto de thread. OS/2 da procesos adecuados a cada thread (hilo). Cuando a un thread se le asigna un dato en su registro y stack, y este es nuevamente utilizado, conservando el dato que tenia antes de ser solicitado. Lo que marca la diferencia entre un hilo y otro es su código de ejecución, el mismo código de ejecución hace posible que corran diversos hilos al mismo tiempo.

Una aplicación utiliza uno o màs hilos. Por ejemplo cuando deseamos imprimir, un hilo (thread) debe leer desde el disco, que entrarà al buffer y a su vez llega a otro hilo (thread) que pasa la información a la impresora, en un momento dado, disco e impresora trabajan al mismo tiempo. Siendo ésta otra de las ventajas de programación multiple. Los hilos no poseen recursos del sistema, en lugar de ellos comparten recursos del proceso al que estàn referenciados. Sòlo cuando finalizan todos los hilos de un proceso, OS/2 termina también el suyo. Como los hilos forman

parte del mismo programa y no preceden el orden de ejecución. se hace una seriación de acceso a los recursos. cuando un hilo actualiza un campo. activa un señalizador que indica que el dato fué actualizado. Entre hilos no existe organización jerárquica. no hay relación padre-hijo.

En OS/2 un proceso es una unidad lógica que contiene los recursos de los programas. es un conjunto de recursos del sistema que se encuentran en un programa determinado.

Cuando OS/2 da inicio a un programa se genera un proceso para adquirir los recursos (segmentos de datos. archivos. colas de información. semáforo[1] de hilos) que necesita para su desarrollo. carga el programa y un hilo se encarga de ejecutar su código.

Cuando un proceso libera sus recursos (cerrar archivos o liberar segmentos de memoria) OS/2 los cancela de los procesos a los que se relaciona. Si por alguna razón un programa termina en forma tal que entorpece el sistema. los servicios de mantenimiento del OS/2 liberan los recursos conectados a él.

Los procesos son creados con una estructura jerárquica. cuando un proceso ejecuta uno segundo. se denomina padre y al proceso a

---

1. Un mecanismo que se utiliza para el manejo de la exclusión mutua es el de los semáforos. el cual constituye una extensión del de Reserve y Libere. y tiene la ventaja de que permite resolver también los problemas de sincronización entre procesos. Un semáforo consta de dos partes: un entero y una cola de procesos en espera.En OS/2 un semáforo es un indicador o contador que las aplicaciones utilizan para sincronizar o restringir el acceso a los recursos. La

ejecutar se le llama hijo. la Figura 5.1 muestra la relación padre-hijo . El proceso A es padre de B y C. El proceso D y E son hijos de C.

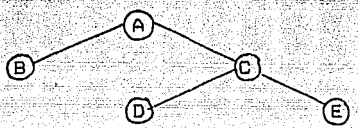
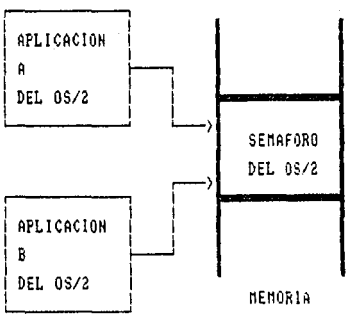


Figura 5.1 Estructura Jerárquica de procesos

siguiente muestra como en la memoria compartida. un semáforo reside en una zona a la que puede accesar cada proceso. OS/2 implementa dos tipos de semáforos: semáforos del sistema y semáforos RAM.



ZONA DE MEMORIA PARA SEMAFOROS

A los procesos B,C,D y E se les denomina descendientes de A. Generalmente los procesos hijos adquieren los recursos de los padres salvo que èstos sean adquiridos sin opciones de herencia . Al inicio de un programa el proceso padre determina que parámetros recibirán los procesos hijos, ya sean de entrada o de entorno, de tal forma que un proceso es un conjunto de recursos del sistema incluyendo: hilos, segmentos de memoria, dispositivos de archivo, colas y stack. Estos recursos, unidos, ejecutan un programa. El padre controla a los hijos y facilita llamadas API.

Para marcar una diferencia de hilos con ciertas características opcionales OS/2 utiliza ciertas clases de prioridad, como son:

TIEMPO CRITICO

PRIMER PLANO

REGULAR

DESOCUPADA

Los hilos tienen cierto tiempo de ejecución en la CPU. Debido a que existen varios hilos haciendo èsta solicitud se hace necesario marcar una diferencia entre aquellos que tienen tareas de mayor prioridad, por ejemplo, un hilo que controla una línea de comunicaciones tiene mayor prioridad que aquèl que corriendo en una impresora. En OS/2 existe un esquema de prioridades de niveles múltiples hecho a través de una variación

dinâmica.

En la tabla 5.1 se muestran los niveles de prioridad para cada clase, así cómo los elementos que corren en cada uno de ellos.

CLASE DE PRIORIDAD	NIVELES DE PRIORIDAD	ELEMENTOS QUE CORREN DENTRO DE LA CLASE
TIEMPO CRITICO	31 (DISTINTOS DE PRIORIDAD	AQUELLOS HILOS DE ATENCION INMEDIATA , COMO COMUNICACIONES O APLICACIONES EN TIEMPO REAL.
PRIMER PLANO	1	EL PROGRAMA DE APLICACION EN LA PANTALLA. UTILIZANDO SI ES NECESARIO LAS CLASES, REGULAR Y DESOCUPADA.
REGULAR	31 (CON VARIACION DINAMICA)	AQUELLOS HILOS CONSIDERADOS POR EL SISTEMA, DE ACUERDO CON SUS CARACTERISTICAS OCUPACIONALES, CONSIDERANDO LA ENTRADA Y SALIDA ASI CÓMO EL USO DE LA CPU.
DESOCUPADA	31	AQUELLOS HILOS QUE OCUPAN CON MENOS FRECUENCIA EL SISTEMA.

TABLA 5.1 MANEJO DE CLASES DE PRIORIDAD.

5.3 COMUNICACION ENTRE PROCESOS

El problema que debe resolverse con respecto al manejo de los procesos concurrentes es el de la comunicación , que consiste en proporcionar a los procesos ciertos elementos que le permitan intercambiar información. Se han ideado diversos mecanismos para la comunicación entre procesos. Una forma sencilla de representarlos consiste en definir dos grupos de procesos llamados: productores y consumidores que, como lo indica su nombre, "producen" y "consumen" infromación, respectivamente. La información producida es almacenada en una zona, cuyo tamaño depende de las características del problema, y de ahí es retirada por los consumidores.

Un ejemplo claro de cómo trabajan es èl de cola por impresión: la orden PRINT (imprimir) del OS/2 utiliza un almacenamiento llamado "cola", que controla los archivos que desean imprimirse, siendo una cola, una lista de espera. OS/2 reserva una área llamada creador de colas de impresión. Cuando se ejecuta la instrucción PRINT del OS/2 coloca los nombres de los archivos en esta área tal como se muestra en la Figura 5.2

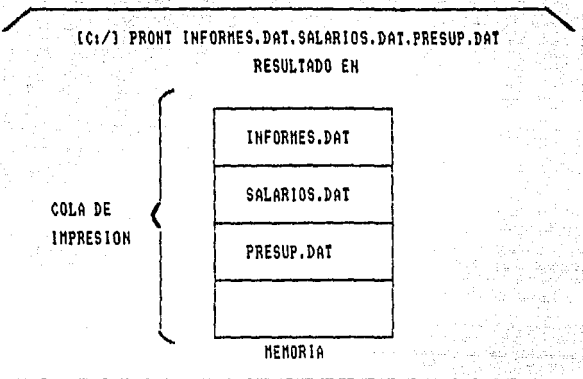


FIGURA 5.2 ALMACENAMIENTO DE ARCHIVOS EN LA COLA

Las actividades de una cola son:

* La cola puede encontrarse vacia (Figura 5.3)
* Cierto trabajo puede entrar y recibir atención inmediata (Figura 5.4).
* Al encontrarse en servicio un trabajo el inmediato debe esperar (Figura 5.5).
* Cuando existen varios trabajos, estos deben esperar (Ver Figura 5.6).

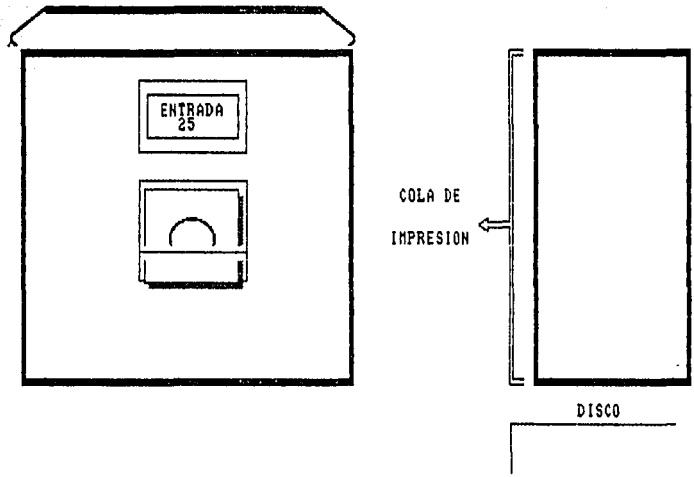


FIGURA 5.3 COLA VACIA

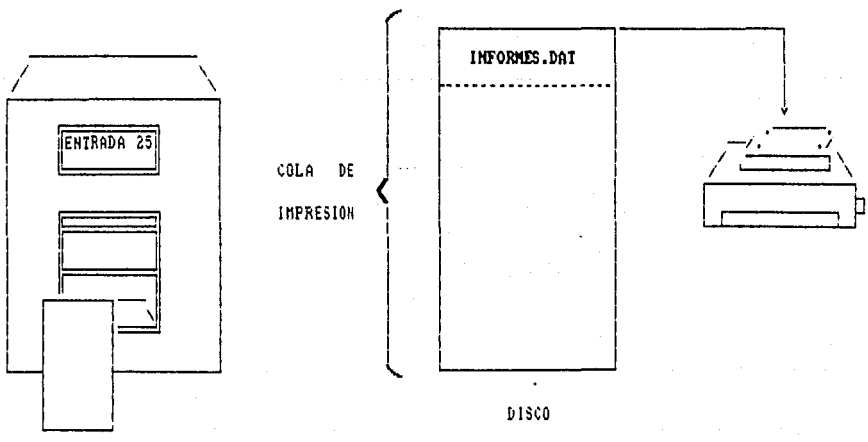


FIGURA 5 . ACCION INMEDIATA EN LA COLA.

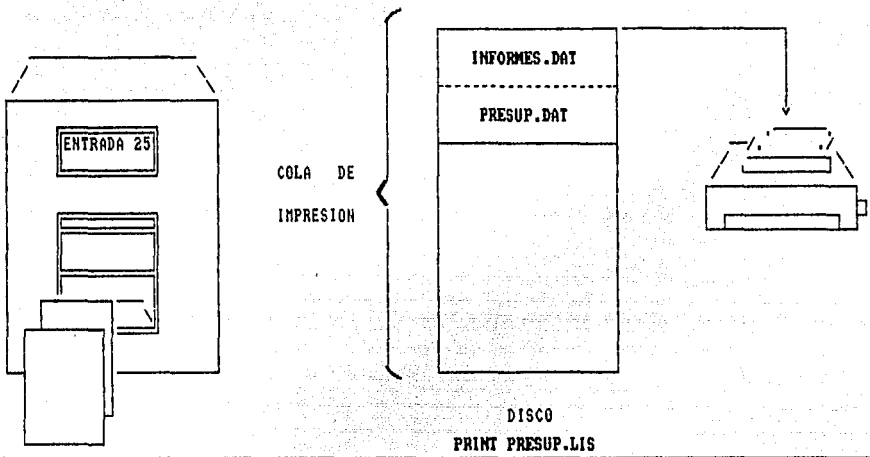


FIGURA 5.5 ESPERA EN LA COLA

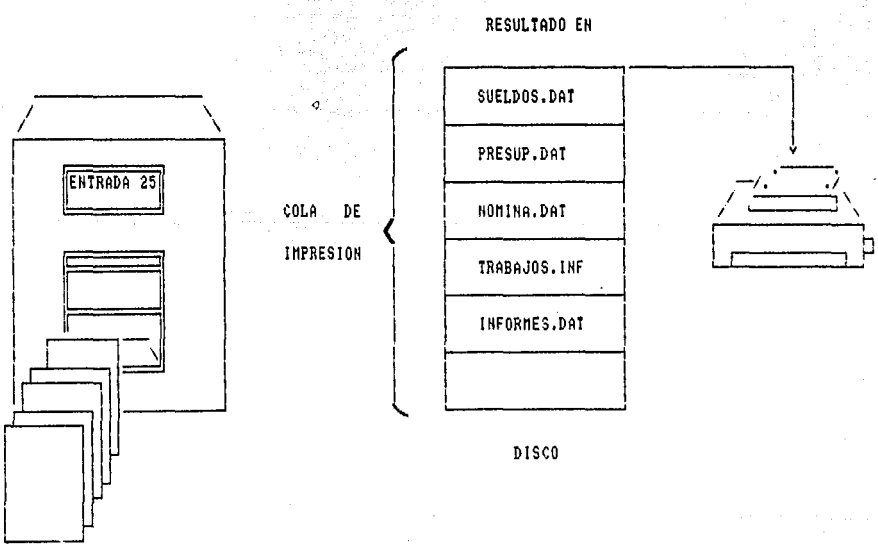


FIGURA 5. TRABAJOS QUE DEJAN EN COLA

* Cuando la cola se encuentra saturada se niega el servicio, tal como se muestra en la Figura 5.7, debido a que el espacio en disco está lleno. si un trabajo se desactiva deja la cola antes de que se le de el servicio (Ver Figura 5.8).

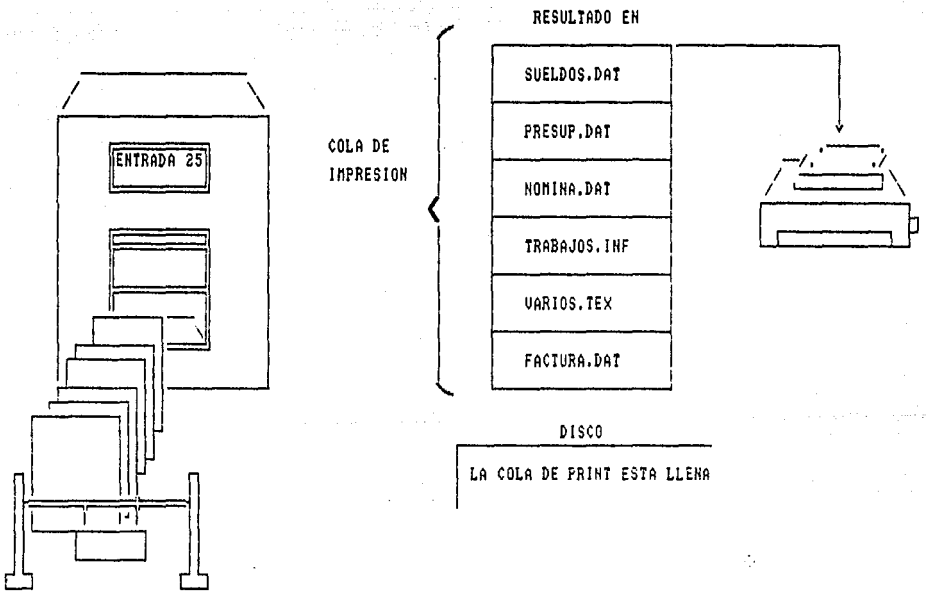


FIGURA 5.7 DENEGACION DE ACCESO A LA COLA DE TRABAJOS

Con esto se observa de forma sencilla la ejecución simultanea de programas . con los que se aprovechan las posibilidades de cada proceso. Ahora bien, la comunicación entre

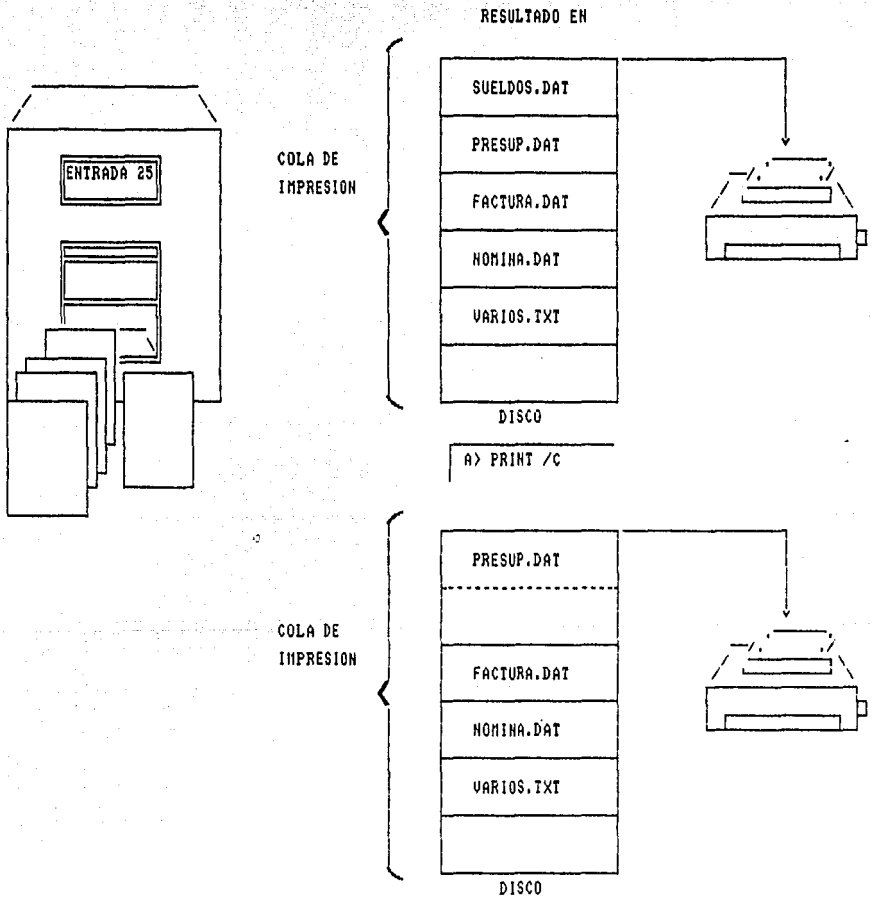


FIGURA 5.8 UN TRABAJO DEJA LA COLA ANTES DE SER ATENDIDO.

procesos (CEP) consiste en el intercambio de información entre dos o màs programas concurrentemente. OS/2 brinda algunas posibilidades que proporcionan soporte a CEP:

- Memoria compartida
- Semàforos
- Conductos
- Colas

Capítulo 6

SISTEMA DE ARCHIVOS

Cada sistema operativo tiene una forma particular de almacenamiento de información, permitiendo al usuario definir objetos llamados archivos, siendo un archivo la unidad básica de almacenamiento de información a largo plazo y está formada por un conjunto de elementos o registros de importancia para el usario. Los archivos pueden dividirse en dos niveles: lógico y físico, el primero se refiere a la forma como el archivo se encuentra almacenado en la memoria secundaria. En consecuencia debe contarse con una interfaz que permita al usuario la manipulación de sus archivos independientemente de la forma como se almacenen en la memoria secundaria. ésto se consigue con varios programas llamados "sistema de archivos" y a través de los cuales se establece la correspondencia entre archivos lógicos y físicos. La Figura 6.1 muestra la manera en que son llevadas a cabo las referencias de ambos por medio del sistema de archivos.

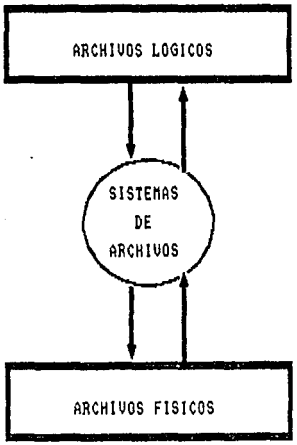


FIGURA 6.1 INTERFAZ ENTRE LOS ARCHIVOS LOGICOS Y LOS ARCHIVOS FISICOS

Las operaciones básicas que se llevan a cabo sobre un archivo son: lectura y escritura de registros. transmiten información de la memoria secundaria a la memoria primaria para la lectura. y de la memoria primaria y la secundaria en la escritura. Dicha transmición de información se realiza en unidades de tamaño fijo denominadas "bloques".

En OS/2 el sistema de archivos se encuentra en el núcleo. es el encargado de organizar y mantener los datos en los programas de aplicación y los dispositivos externos. es también la parte del sistema operativo que proporciona aplicaciones a través de los medios de almacenamiento. como es el caso de los discos.

Un disco es un conjunto de registros. compuesto de pistas. que empiezan en el borde exterior y van hacia el centro (Véase Figura

6.2). el número de pistas existentes dependen del tipo de disco, los flexibles estàndares (340K) tiene 40 pistas por cara. los de alta densidad utilizan 80. OS/2 hace una división de las pistas del disco en unidades de tamaño similar denominadas sectores (Véase Figura 6.3). el número de sectores al igual que las pistas dependen también del tipo de disco.

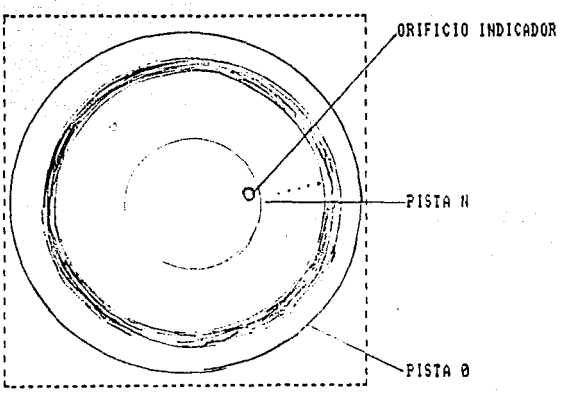


FIGURA 6.2 PISTAS QUE COMPONEN UN DISCO

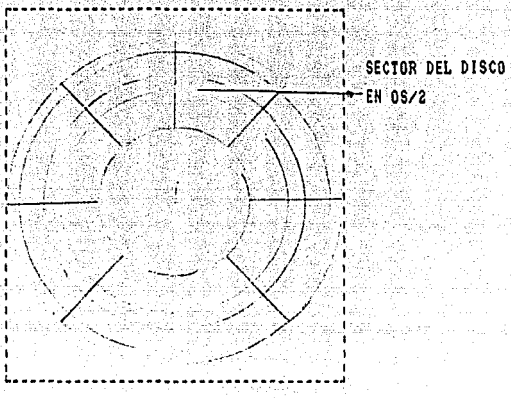


FIGURA 6.3 SECTORES QUE COMPONEN LAS PISTAS

La siguiente tabla 6.1 nos muestra las pistas y sectores que existen de acuerdo con un tipo determinado de disco:

Tipo de disco	Pistas por cara	Sectores en total
Una cara. 8 sectores por pista	40	320
Una cara. 9 sectores por pista	40	360
Doble cara. 8 sectores por pista	40	640
Doble cara. 9 sectores por pista	40	720
Alta densidad. 9 sectores por pista	80	1440
Alta densidad. 15 sectores por pista	80	2400

Tabla 6.1 Pistas y sectores de un disco

Cada sector contiene la misma longitud en bytes. El espacio de un disco (1.2 MB) de alta densidad no es unicamente para almacenar datos. al formatearle OS/2 hace ciertas reservaciones para: el registro de arranque. las tablas de asiginaciòn de archivos (FAT) y las entradas al directorio raiz. El primero se encuentra en el sector uno y es el que permite que el proceso inicie. en caso de no tenerlo. el sistema manda un mensaje de error. en el segundo. se administran los espacios del disco. contiene las entradas para los sectores que se utilizaràn para almacenar y asignar archivos. ademàs de aquellos que no se encuentran disponibles debido a desperfectos del soporte. Para evitar cualquier contratiempo OS/2 realiza una copia de la FAT. còmo repuesto en caso de daño. es la encargada de localizar bloques (dos o màs sectores de un disco. dependiendo del tipo del disco) del disco.

OS/2 asiona 1.024 bytes a un archivo que es equivalente a dos sectores o un bloque. cuando un archivo crece rebasando los 1.024 bytes se asiona otro bloque al archivo. y el tercero es para los archivos del directorio raíz. OS/2 reserva un espacio limitado. ya que cada uno de ellos requiere de una entrada de 32 bytes.

Para una mejor administraciòn de archivos es preciso utilizar subdirectorios.[1]

---

1. Los directorios permiten agrupar archivos en diferentes categorias. pueden cargar a otros directorios. denominados subdirectorios. que son una organizaciòn de la estructura de archivos.

6.1 MODELO DEL SISTEMA DE ARCHIVOS

Las funciones de un sistema de archivos son, proporcionar operaciones lògicas de alto nivel, independientes a la implantaciòn del nùcleo.

Las divisiones del mismo son: una externa (para el usuario) y una interna (para el sistema).

En la primera, està organizado de manera jeràrquica. Su raiz se encuentra en un dispositivo premeditado del sistema.
Su jerarquia es de tres tipos: de directorio: compuesto de varias entradas, cada una de las cuales contiene el nombre de un archivo y un apuntador a su representaciòn interna. Cada directorio tiene dos entradas como mínimo: la primera de nombre, se refiere a si mismo, y la segunda, de padre .. se refiere al directorio padre en la jerarquia, el segundo tipo es de archivos corrientes: que no tienen ningún tipo de estructuraciòn para el sistema, se encuentran constituidos por una secuencia de bytes, y el tercero es de archivos especiales: los cuales estàn asociados a los dispositivos de entrada/salida. Como los archivos de diferentes tipos, tienen primitivas comunes para su manejo, su existencia permite tener acceso a los archivos corrientes y a los dispositivos con un mismo mecanismo.
Para la visiòn exrterna, en la jerarquia descrita anteriormente.

las referencias a los archivos se componen de su nombre (externo) y de un apuntador a su descripción. Este último apunta a una entrada en una tabla de descripción de archivos o tabla-i . que contiene:

- Identificación del creador
- Identificación del grupo del creador
- Identificadores de protección
- Dirección física
- Tamaño
- Fecha de creación. del último uso y de la última modificación
- Número de apuntadores al archivo
- Tipo
- Mapa de asignación

Ahora bien. los archivos son una vista lógica de un conjunto de sectores en un disco y están representados como un flujo en serie de caracteres (tienen nombre ASCII). estructurados como un nombre de ocho caracteres seguidos de una extensión de tres más de ellos. la extensión es utilizada para designar el tipo de datos que contiene el archivo (como puede observarse en la Figura 6.4).

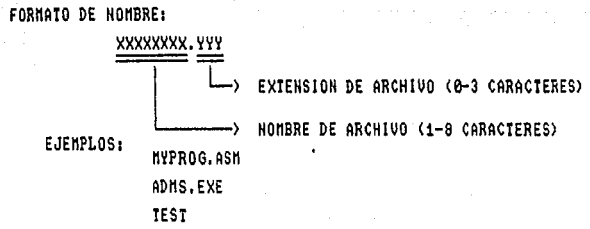


FIGURA 6.4 NOMBRES DE LOS ARCHIVOS

Los archivos generalmente residen en discos (en algunos casos son discos virtuales). Cada uno de ellos tiene un directorio (colección de archivos) que almacena toda la información asociada a los archivos guardados en él.

Todas las entradas al directorio tienen una marca que indica si está disponible.

Cada archivo contiene de 1 a 16 entradas en el directorio. Cada una de ellas define una extensión. que comprende hasta 32K caracteres. El tamaño máximo de un archivo es de 512K de caracteres (limitados por las características físicas del disco).

El directorio posee 64 entradas (en algunas versiones del sistema este número varia). en cada una de ellas existe la siguiente información:

- Nombre del archivo
- Mapa de asignación

- Número de extensión (de 0 a 15)

- Número de registros en esta extensión

El mapa de asignación describe el espacio designado a la asignación del archivo. Está compuesto por 16 entradas de un byte, donde cada una es la dirección lógica de un bloque de tamaño de 2K.

El número de extensiones representa la cantidad de entradas del directorio, y el número de registros en la extensión se refiere a los registros (o sectores de 128 caracteres) que contiene la extensión .

Cada vez que es iniciado el sistema, se construye en la memoria real, a partir del directorio, una tabla de indicadores que representa el estado de ocupación de la memoria auxiliar.

Cuando se abre un archivo, se trae de la memoria principal la entrada del directorio correspondiente a su primera extensión, y se construye con base en ella el llamado FCB (bloque de control de archivo), que contiene la misma información que la entrada, mas el número del próximo registro a leer o escribir, que el usuario debe dar al procedimiento de apertura del archivo.

Cada disco tiene al menos un directorio, llamado raíz (es un archivo especial, que de acuerdo con las necesidades del usario comprende otros directorios). La Figura 6.5 muestra una estructura hipotética del directorio.

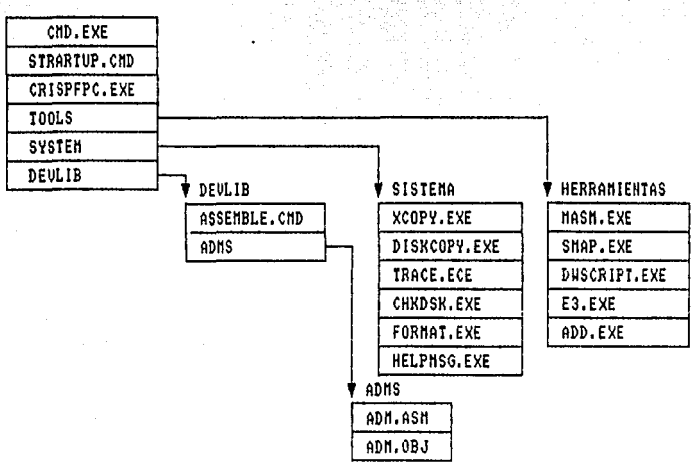


FIGURA 6.5 ESTRUCTURA JERARQUICA DE DIRECTORIO

6.2 FUNCIONES DE MANIPULACION DE ARCHIVOS

Es a través del manejo de los archivos cómo se da dirección a los datos almacenados en la memoria secundaria, además de permite organizar la información que posteriormente se guardará en la memoria de manera permanente. Así mismo en ellos se almacena la información, por bastante tiempo, en medios de almacenamiento masivo, de forma tal que cuando se requiere la

información. se copia en la memoria primaria.

La información que se transfiere a la memoria primaria es almacenada en un área temporal llamada. buffer. en la que se encuentran los datos. En primera instancia se depositan en el buffer y posteriormente pasan a la memoria secundaria (ver Figura 6.6).

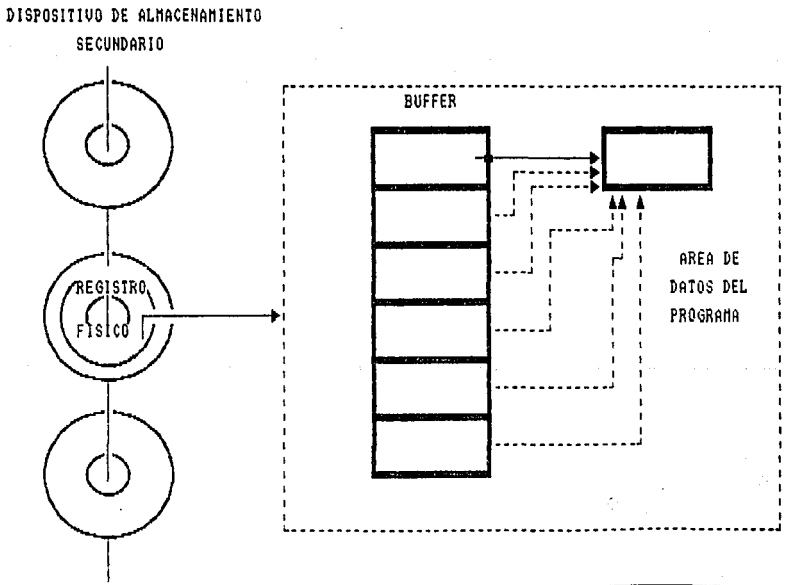


FIGURA 6.6 USO DEL BUFFER EN LA LECTURA DE DATOS.

OS/2 cuenta con una serie de funciones API para crear, leer, escribir y destruir archivos.

Existe una petición (DosOpen) que pide permiso de acceso a los archivos, contiene un indicador de opción que le dice al sistema como procesar la petición de apertura, al usar èste indicador, se define el tipo de derechos de acceso asignados al programa. Al va existir un archivo se crea un identificador de archivos (2123) llamado indicativo de archivo, que es la base para cualquier referencia hecha al archivo. En la Figura 6.7 se muestra como el programa A pide el uso del achivo TEST.ASM.

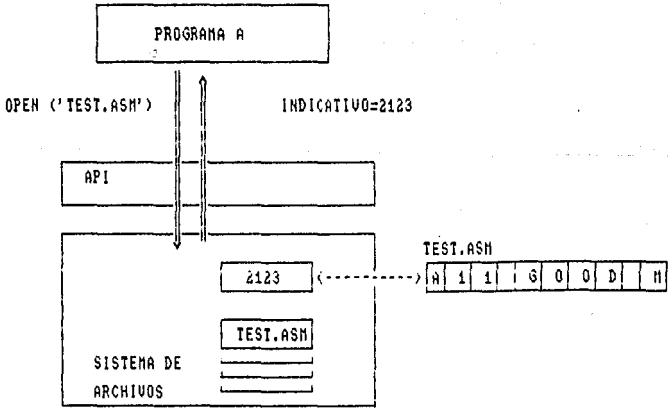


FIGURA 6.7 APERTURA DE UN ARCHIVO

Los archivos son respresentados como un flujo de caracteres en serie. La siguiente posición del flujo a ser leída o escrita se determina por un apuntador especial llamado "apuntador lógico del archivo" (Ver Figura 6.8).

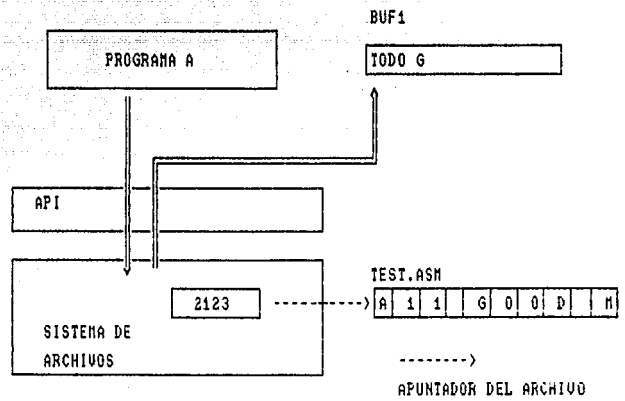


FIGURA 6.8 LECTURA DE DATOS DE UN ARCHIVO

De la Figura 6.8:

Suponqamos que el programa A lee datos de TEST.ASM, a través de una petición de lectura de archivo, que requiere de tres elementos de información:

1) El indicativo de archivo, que identifica el archivo que se está leyendo.
2) La memoria intermedia (buffer) de aplicación que será utilizada al recibir los datos.
3) La cantidad de bytes a transferirse.

El programa A hace la petición de lectura de datos para cinco bytes en un buffer llamado BUF1, utilizando el indicativo 2123, para localizar TEST.ASM y transfiere los cinco primeros bytes a la memoria secundaria.

Para realizar la escritura de datos en un archivo abierto se requiere de la siguiente información:

1) Un indicativo de archivo que indica el archivo que se va a escribir.
2) Una memoria intermedia de aplicación que contenga los datos.
3) La cantidad de bytes a transferir.

De igual manera que un programa crea archivos, también produce subdirectorios, debido a que un subdirectorio es un tipo especial de archivo que contiene enlaces con otros archivos.

6.3 IMPLEMENTACION

La implementación recae en la forma en que actua el sistema, para llevar a cabo las funciones de los archivos.

Las órdenes del OS/2 son internas y externas. Las internas residen en la memoria del OS/2 y las externas residen en disco.

Antes de que OS/2 ejecute una orden externa, debe cargar la orden en memoria.

En cada uno de los niveles del sistema existen ciertas operaciones a realizar, y es en el mas alto de estos que, los programas de aplicación manipulan los datos del disco en forma lógica, por medio de archivos y subdirectorios.

De acuerdo con lo visto en la sección anterior , la FAT (tabla de asignación) es mantenida por el sistema de archivos, dicha tabla hace un enlace de sectores de un archivo formando una cadena, donde cada sector de la cadena contiene parte del archivo. En el momento enque un programa lee o escribe datos, el sistema de archivos auxiliado por la FAT determina que sector el es que contiene el dato ya sea de lectura o de escritura.

El sistema de archivos considera al disco como una serie de sectores lógicos, que va de 1....n donde n está determinado por la geometria del disco, de la manera siguiente:

Sectores totales (n) = (sectores/pista) * (pistas totales) * (número de cabezas)

El sistema de archivos requiere números de sectores lógicos de la rutina del dispositivo, donde hace una conversión de números de los sectores a peticiones cabeza/pista/sector comprendidas por el hardware.

OS/2 1.0 al igual que MS DOS tiene una limitación, la de que

sus discos se restringen a 32MB por lo que el máximo de tamaño en disco es:

(65536 sectores) * (512 bytes/sector) = 32767KB
= 32MB

6.4 ARCHIVOS COMPARTIDOS

Cuando varios usuarios requieren trabajar juntos para la elaboración de un proyecto, por tal motivo, es necesario que compartan archivos, por lo cuál es necesario que un archivo compartido[1] se encuentre de forma simultánea en distintos directorios que a su vez pertenecen a diferentes usuarios.

Al trabajar diferentes usuarios juntos en algún proyecto, en ocasiones necesitan compartir archivos, por lo que conviene que archivos compartidos figuren en distintos directorios, en la Figura 6.9 se observa como el archivo C se encuentra tanto en el mismo archivo, como en el archivo B. La conexión entre el directorio de B y el archivo compartido se llama enlace.
De acuerdo con lo anterior, se dan casos en que dos programas compartan un mismo archivo a un mismo tiempo. Sin embargo el hecho de compartir archivos resulta conveniente, pero también genera problemas, dado que si los directorios contienen direcciones al disco, entonces tendrá que hacerse una copia de

---

1.Un archivo compartido es aquel que utilizan más de una persona.

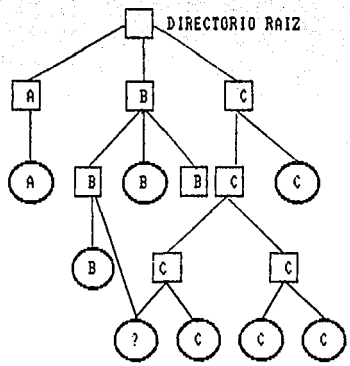


FIGURA 6.9 SISTEMA DE ARCHIVO QUE CONTIENE UN ARCHIVO COMPARTIDO

las direcciones del disco en otro directorio. al enlazar un archivo. Si dos archivos se anexan después a uno tercero. los nuevos bloques se listarán sólo en el directorio del usuario que efectúa la anexión. provocando que el usario no se percate de los cambios realizados. lo que. repercutirá al momento de compartir archivos.

El problema anterior tiene dos soluciones. la primera de ellas es que los bloques del disco no sean listado en un directorio. sino en una estructura de datos asociada con el mismo archivo.los directorios apuntarían después a la estructura de datos. La segunda solución es que ambos archivos se enlacen y que el sistema cree un nuevo archivo. y se coloque ese archivo en

uno de los directorios de los archivos enlazados. El nuevo archivo contendrá tan sólo el nombre de la ruta al cual se enlazará. A este método se le llama enlace dinámico.

En OS/2 se dan facilidades para que los programas controlen el acceso de los archivos y para èsto es necesario que primero sea abierto un archivo para posteriormente ser accesado.

Al abrirse se define que procesos podrá utilizar. así cuando un proceso bloquee una actividad que, en un momento dado cause algún problema, podrán continuar otras actividades. En la Figura 6.10 un archivo SYSTEM.LOG es compartido por varios programas del sistema, al tiempo que un programa llamado SORT clasifica la salida del archivo por fechas y lo copia a otro, cuidando que no se realicen modificaciones su archivo, un segundo programa PRINTIT también abre SYSTEM.LOG con un còdigo de acceso READ_ONLY (petición que sòlo acepta lectura de datos), comienza un tercer programa UPDATE que cambiará el contenido de SYSTEM.LOG, para lo que necesita de un acceso READ_WRITE (permite aceptar solicitudes de escritura y lectura) al archivo, asegurándose de que no interferirá algún otro programa, requiere de un modo compartido DENY_ALL, dado que SORT va está corriendo no permitirá correr a ningún programa con acceso de escritura, por lo que el sistema detecta la petición hecha por UPDATE y la rechaza.

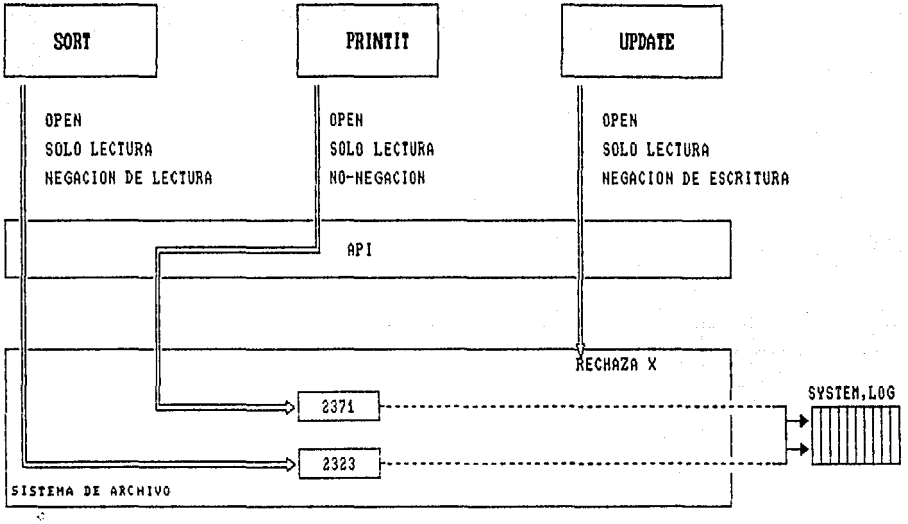


FIGURA 6.10 COMPARTIMIENTO DE ARCHIVOS ENTRE SORT, PRINTIT Y UPDATE

Sin embargo esto no es efectivo en todos los casos. si un programa administra una base de datos que tiene un archivo grande que contenga datos. al utilizar estos mecanismos el archivo de datos sería abierto y cerrado cada vez que se le referencie. ocasionando un bloqueo por tiempo prolongado al archivo. Lo que resolvería el problema es el cierre de archivos (file locking). que permite tener procesos que protejan partes pequeñas de un archivo accesible a otros procesos. El cierre de archivos proporciona una llamada del sistema de archivos del OS/2 para

cerrar rangos de bytes de un archivo abierto. provocando que otros procesos que intenten accesar estos rangos sean bloqueados hasta que sean liberados del cierre. Los rangos de los bytes deben ser cerrados antes de que un dato sea leído ya que cualquier proceso puede ser apropiado por el repartidor. Si un programa es apropiado después de que lee el dato y antes de que lo cierre. otro proceso puede colarse y actualizar el registro. lo que provocaría invalidar el dato de la memoria intermedia del primer programa.

6.5 ALMACENAMIENTO INTERMEDIO DE SECTORES

Cualquier aplicación que necesite realizarse se hará a través del sistema de archivos. sin embargo. sus operaciones se vuelven lentas debido a la velocidad de la computadora. Para hacer rendir más al sistema es necesario cambiar los accesos de datos del disco por copias rápidas de memoria a memoria. al mantener un almacén de memoria de sectores de discos. el sistema de archivos sabrá el contenido de archivo.

Al escribir un sector[1] fuera del disco. se modificará primero la copia de memoria. de forma tal que. al hacerse una petición al sistema de archivos lo primero será. examinar el almacén de memoria. si el caso fuese de lectura ésta sería satisfecha sin

---

1. Sector es la unidad más pequeña de transferencia al disco.

tener que regresar al disco, dado que el dato sería copiado desde el sistema de memoria, en el buffer (memoria intermedia) de la aplicación.

En la Figura 6.11 está la técnica, llamada Bufferino del Sector.

Al leer o grabar información en disco, son utilizadas zonas de almacenamiento (buffers de disco) en su memoria para guardar información transferida. Cuando un programa de aplicación lee o graba un registro de datos de tamaño distinto al del sector del disco (por lo general el sector es de 512 bytes), OS/2 almacena provisionalmente el registro, cuando el registro en un disco no sea múltiplo del tamaño del sector del disco, se colocará en un buffer del mismo. Al realizar operaciones de lectura, primero se comprueban los buffers para ver si los datos requeridos se encuentran en memoria, lo que elimina una entrada y salida de disco.

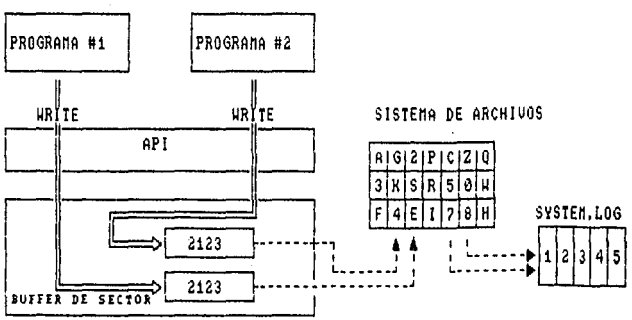


FIGURA 6.11 BUFFER DE SECTORES

Si una aplicación utiliza registros de 64 bytes, un sólo sector almacena ocho de ellos. (vèase la Figura 6.12).

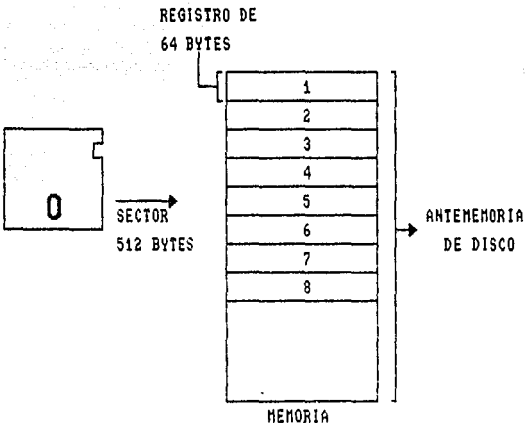


FIGURA 6.12 ALMACENAMIENTO DE OCHO REGISTROS EN UN SECTOR DE 64 BYTES

Si un programa necesitara leer el registro 1 del disco, el sistema lee todo el sector que contiene el registro en un buffer. lo anterior se muestra en la Figura 6.13.

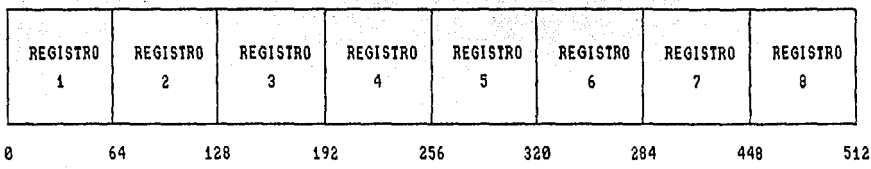


FIGURA 6.13 REGISTROS DENTRO DE UNA APLICACION

Lo cual quiere decir que se dispone de los registros 2 al 8 en memoria. Cuando el programa solicite el registro 2, OS/2 buscará el registro en sus buffers de memoria (Ver Figura 6.14).

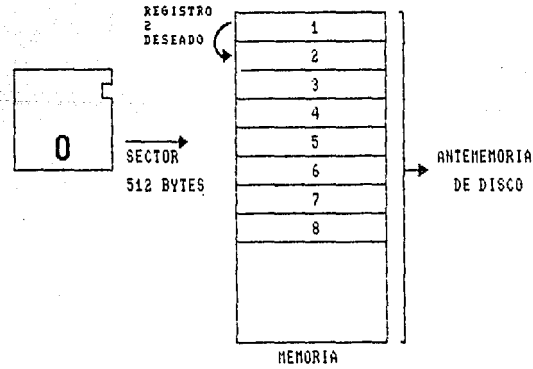


FIGURA 6.14 BUSQUEDA DE UN REGISTRO EN LOS BUFFERS DE MEMORIA

Si el registro es localizado, no habrá necesidad de leer un nuevo sector del disco y únicamente se utilizarán los buffers del disco. Si se incrementa el número de buffers de archivo se mejorará el rendimiento de aquellas aplicaciones que realizan operaciones de entrada y salida, ya que de ésta forma se lleva a cabo un adecuado manejo de peticiones hechas a dichas aplicaciones.

Debido a que el acceso al disco es mucho más lento que el de la memoria, el sistema de archivo fué diseñado para reducir el número de accesos al disco, utilizando una técnica llamada de reserva de bloque o reserva del buffer. Existen diversos algoritmos para menejar la reserva, en éste caso el utilizado es el algoritmo LRU (la paginación usada menos recientemente), en el que al ocurrir una falla de bloque, se desecha la página que halla estado sin uso por el periodo de tiempo más largo.

6.6 PARTICIONES DEL DISCO

OS/2 utiliza una técnica llamada "particionamiento del disco" que se implementa fundamentalmente en la rutina del dispositivo de disco. Para hablar de las particiones del disco es necesario saber como se utiliza: no todo el espacio de 1.2 MB de un disco de alta densidad se utiliza para almacenamiento de datos, al momento de dar formato a un disco OS/2 reserva ciertos sectores del disco, con datos específicos como: registros de arranque,

tablas de asignación de archivos (FAT) y entradas del directorio raiz. Cada disco del sistema contiene el registro de arranque en el primer sector. si en un momento dado no existiera dicho registro, se enviaría un mensaje de error del disco desde el sistema. En la tabla de asignación de archivos (FAT) se encuentra la información del estudio general del disco como: entradas a sectores utilizados para almacenar archivos, sectores que están disponibles y cuales no, para ser asignados a archivos, además de los sectores que no se encuentran disponibles debido a desperfectos de soporte. En el momento de asignar un espacio, el sistema lo hace en bloques[1] y la FAT marca estados del disco de la siguiente forma:

0	Bloque disponible
FFFF	Fin de archivo
FF7	Archivo defectuoso
nnn	"n" es el siguiente bloque asignado al archivo

Por ejemplo considerando la estructura de la Figura 6.15 las entradas a la FAT son las mostradas en la tabla 6.16.

---

1. Un bloque es la unidad de asignación de espacio tanto para un disco como para un archivo. Un bloque son dos o más sectores cercanos del disco (considerando el disco que se está manejando).

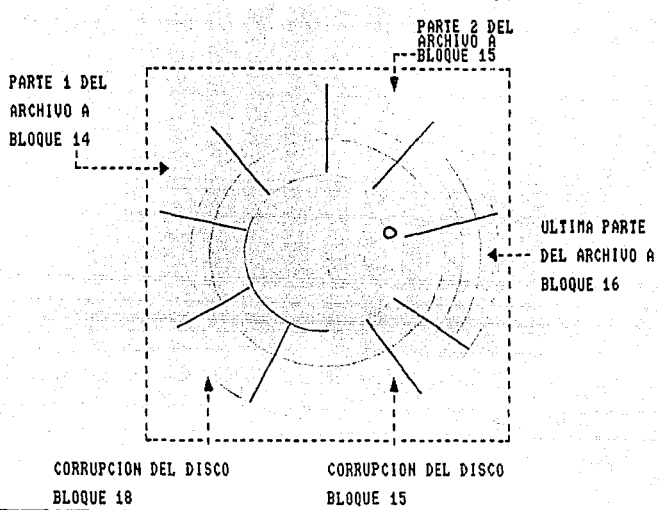


FIGURA 6.15 ESTRUCTURA DE DISCO

Número de bloque	Desplazamiento
0	Reservado
1	Reservado
2	n
3	n
4	n
5	FF7
6	n
.	.
.	.
.	.
14	16
15	17
16	FFF
.	.
.	.
.	.
355	n

6.16 Tabla de asignación de archivos.

OS/2 encuentra el bloque inicial de un archivo por medio de la entrada a su directorio y recorre la lista de bloques hasta hallar la entrada de. fin de archivo (FFFF).

La fragmentación de un disco se hace cuando los archivos del disco se encuentran en sectores físicamente esparcidos por el disco. tal es el caso de la Figura 6.17. donde las cuatro partes del archivo (bloques) están dispersas.

Los archivos fragmentados aumentan la cantidad de tiempo necesaria para que el disco tenga acceso a los sectores. en caso de necesitarse rotaciones adicionales del disco para leer el

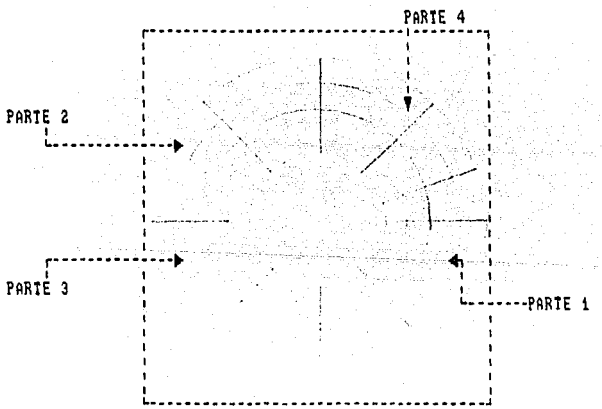


FIGURA 6.16 EJEMPLO DE FRAGMENTACION DE DISCO

contenido de un archivo. lo que provoca que las aplicaciones se efectuen más lento.

En los discos fijos no se tiene acceso al soporte de almacenamiento, el disco es menos elástico y gira más rápidamente que un disco flexible, proporcionando una mayor capacidad de almacenamiento y un tiempo de acceso más rápido. se componen de varias secciones llamadas platos. en su mayoría utilizan 2 de éstos, proporcionando cuatro caras en las que se graba información, y se combinan varias pistas para formar cilindros.

OS/2 soporta particiones de, hasta 32 MB en longitud, cuando el disco fijo es demasiado grande, se le debe dividir en distintas particiones. El particionamiento del disco se implementa principalmente en la rutina de dispositivo del disco, donde cada partición es tratada como un volumen separado, el sistema de archivos hace referencia a cada una de ellas con un identificador de unidad diferente, como si fuese un disco distinto.

El particionamiento de un disco permite a OS/2 soportar discos duros de hasta 768 MB de tamaño o designaciones de 24 unidades (de la letra C a la Z con 32 MB de partición).

6.7 SERVICIOS DE DISPOSITIVO DE E/S

En OS/2 las aplicaciones no accesan directamente al hardware, por lo que el sistema da soporte a los dispositivos utilizando un modelo de entrada y salida de datos con flujo orientado, que administra cada aplicación de acceso a los

mismos.

Cuando el nombre de un dispositivo finaliza con <<$>>, implica que no será accesada directamente por un programa de aplicación. Cada dispositivo del sistema está identificado con nombre ASCII.

Un programa de aplicación accesa al mismo. En general, se crea un identificador, denominado "indicativo de dispositivo", y se devuelve a la aplicación, utilizándose para operaciones posteriores de entrada y salida. Los indicativos de dispositivo se utilizan con peticiones de lectura y escritura, para el paso y recepción de datos.

Capitulo 7

ADMINISTRACION DE

R E C U R S O S.

La administración de recursos comprende el conjunto de sistema hardware. estrechamente relacionado al entorno de monousuario con facilidades de multitarea . que a diferencia de MS DOS. corre. concurrentemente diversos programas. Si existe un programa de aplicación que necesite utilizar recursos. este lo hará a través de servicios del OS/2.

7.1 ADMINISTRACION DE RECURSOS Y LA API

Entre las estrategias que emplea OS/2 para la administración de recursos. se encuentran:

* Las diferentes funciones de API. que brindan la oportunidad al usuario de averiguar las funciones y la capacidad de respuesta. eligiendo una API adecuada.
* Dar servicios del sistema de altas prestaciones para funciones generalmente implementadas por orden directa del hardware. como por ejemplo. el monitor y teclado.
* Cuando los servicios al sistema no son los apropiados. autoriza aplicaciones pidiendo acceso directo al

hardware e identificando estos programas como de <<entrada y salida directa>>, y permitiendo al usuario final la decisión, de no ejecutarlos.

El sistema maneja todas y cada una de las funciones que intervienen en los recursos de la PC.

Existe una variación en cuanto al control de un dispositivo dado por OS/2. En la Figura 7.1 los escalones son el número de funciones proporcionadas por el sistema para el manejo de los recursos. Un escalón alto representa varias funciones del sistema operativo, un escalón bajo representa pocas o bien ninguna función del sistema. Específicamente, en OS/2 las funciones de subsistema de video son menores, a diferencia de otras como lo es el sitema de archivos.

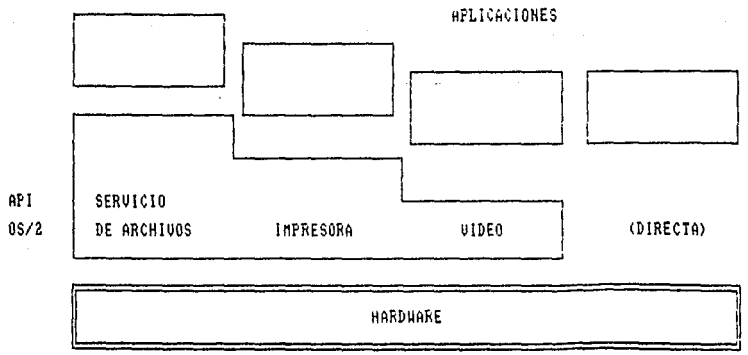


FIGURA 7.1 API DEL SISTEMA

7.2 ADMINISTRACION DE DISPOSITIVOS

Entre las responsabilidades de la rutina de servicio se encuentra la de compartir y cumplir ciertas peticiones, comunicando al núcleo de OS/2, que información acepta y cual rechaza, para la toma de esta decisión se requiere de cierto número de programas que hayan abierto el dispositivo con una serie de características específicas.

Observemos la Figura 7.2, tomemos dos dispositivos: una impresora (LPT1:) y una puerta en serie (COM1:). Para la E/S a la impresora, dos programas distintos, pueden accesarla, la primera rutina del dispositivo acepta múltiples peticiones OPEN (abrir programas con cierto contenido de información) o de distintos procesos, y con la PID proporcionada en las peticiones divide la salida en diferentes flujos de datos (aunque el núcleo genera todos los indicativos del dispositvo para los programas A y B, la rutina del dispositivo decide si la petición OPEN se acepta o no).

Para el caso del dispositivo COM1:, puede llegar a suceder que el programa de aplicación esté ocupado en un paso de comunicación con otro programa en una máquina diferente, por lo tanto no tendría sentido alguno abrir la puerta de comunicación ahora, por lo que el dispositivo COM1: está únicamente dedicado

de forma lógica al programa que inicialmente los abrió, hasta terminar la conexión para después cerrar el dispositivo.

Al accesar la rutina de dispositivo por medio de una aplicación, el resto de estas queda excluida de él.

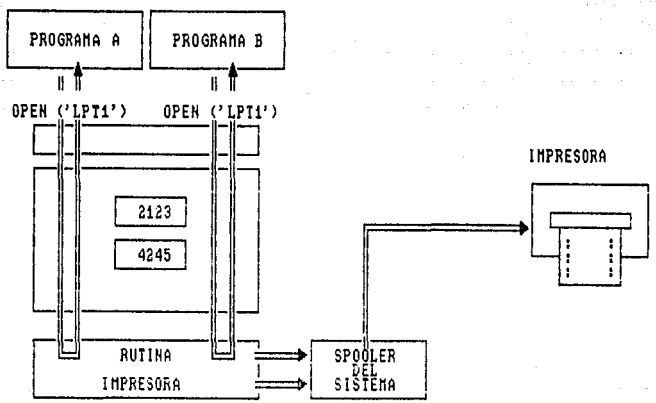


FIGURA 7.2 DISPOSITIVO DE ACCESO COMPARTIDO

El núcleo de OS/2 niega la autorización de acceso al dispositivo y a las rutinas individuales.

7.3 SISTEMA DE ARCHIVOS

El sistema de archivos es parte también de la administración de recursos. controla asuntos de distintos programas. que se encuentren asignados con acceso a dispositivos y archivos. existen dos niveles de la administración de recursos. que son implementados por el sistema de archivos:

* Administración de acceso y nombres generales.
* Asignación y control de bloques de dispositivos (disco fijos y flexibles).

La Figura 7.3 muestra la relación entre el sistema de archivos. rutinas de dispositivos y programas de aplicación.

Los dispositivos y archivos de OS/2 se manejan con nombres ASCII. El sistema de archivos mantiene tablas lógicas correspondientes a los directorios de los archivos. el nombre que les sea asignado no debe por ningún motivo duplicarse. si así se hiciera. en el momento de solicitar una petición de ABRIR se mandaría un mensaje de error. indicando la existencia de un archivo con dicho nombre. Para accesar archivos existen dos niveles que son: atributos de archivo y modos decompartimiento.

El sistema de archivos establece la correspondencia entre los archivos y estructuras del directorio con los sectores lógicos del disco. lo anterior se observa en la Figura 7.4.

Como se mencionó en secciones anteriores. por medio de la FAT (tabla de asignación de archivos) se asegura que dos archivos

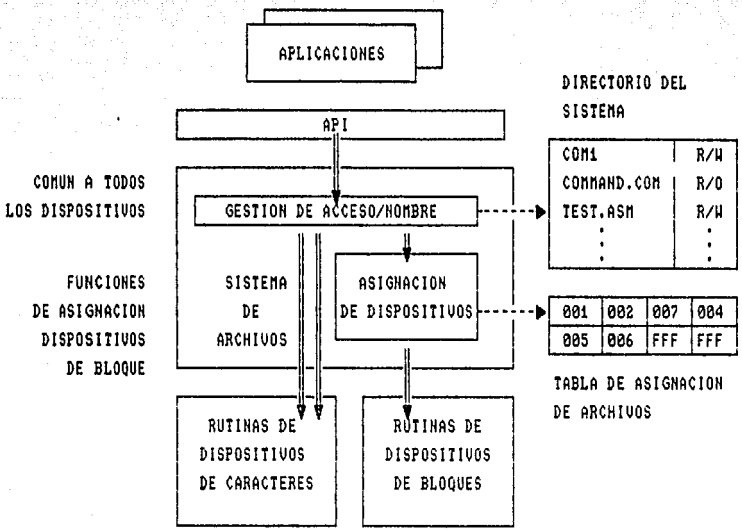


FIGURA 7.3 ADMINISTRACION DE LOS RECURSOS DE LOS ARCHIVOS DEL SISTEMA

no ocupen el mismo sector en disco. ya que provocaría una serie de anomalias como es. la translapación de información. Las unidades de asignación de la FAT son un grupo generalmente de cuatro. que son manipuladas por el sistema de archivos. Para reducir al minimo el tamaño de la FAT se utilizan unidades de asignación[1] en lugar de sectores lógicos.

---

1. Una unidad de asignación de cuatro sectores implica que todos los archivos del disco ocupan múltiplos de 2048 (4 * 512) bytes.

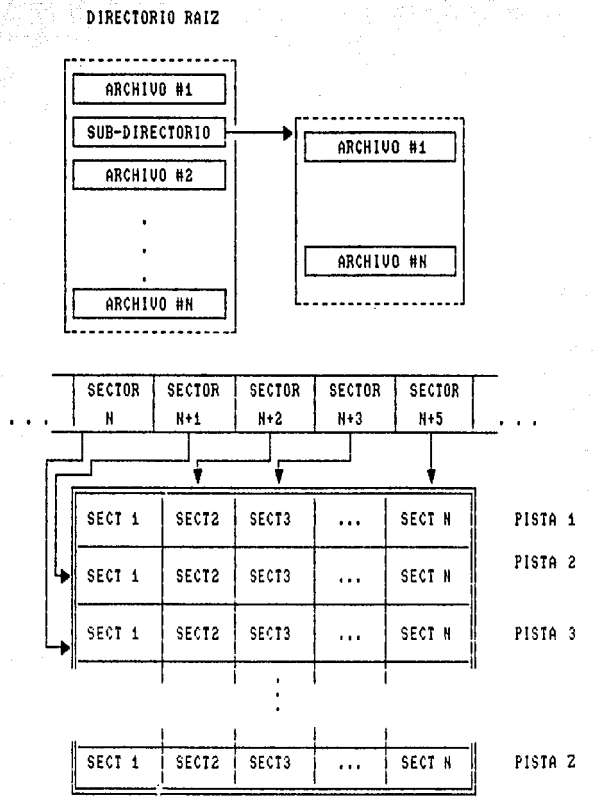


FIGURA 7.4 MODELO LOGICO DE ARCHIVO DEL DOS

Cada una de las posiciones del arreglo FAT corresponden a una posición del disco. Un elemento del arreglo contiene el índice

del siguiente elemento del archivo. El valor de la última unidad de asignación del archivo es: 65535. El sistema de archivos administra el disco de forma tal que cada elemento de la FAT esté presente en una sóla cadena de archivos.

7.4 ADMINISTRADOR DEL PROCESADOR

El procesador o CPU de OS/2 corre diversos programas pero por intervalos de tiempo específico (Véase la Figura 7.5).

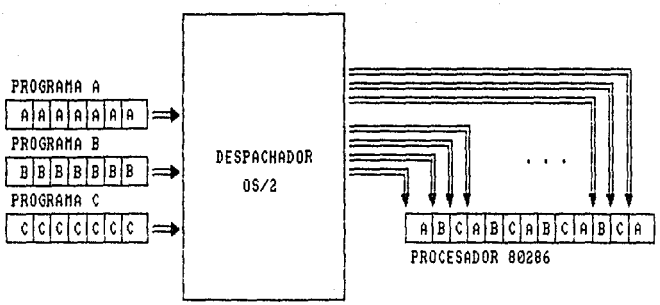


FIGURA 7.5 COMPARTIMIENTO DEL TIEMPO DEL 80286 CON INTERVALOS DE TIEMPO

En OS/2 se comparte el tiempo del procesador ejecutando un repartidor de intervalos de tiempo, que divide el tiempo total en

períodos más pequeños. dando a cada proceso un intervalo de tiempo para su ejecución.

El repartidor del sistema tiene la labor de interrumpir cualquier información mandada al procesador por ciertos intervalos específicos de tiempo. por ejemplo. si un programa de aplicación tiene cierta posesión sobre el procesador. estas <<apropiaciones>> periódicas se llevan a cabo por el reloj del sistema. de acuerdo con cada señal el repartidor de OS/2 checa entre los threads (hilos) actuales. si alguno ha sido utilizado por un intervalo de tiempo completo. se repartirá entonces uno nuevo.

En OS/2 los programas de aplicación no pueden inhabilitar interrupciones. ya que estos son ejecutados en un nivel de privilegio donde no se realiza esta inhabilitación. Lo que si manipulan en forma directa es el hardware. a través de segmentos privilegiados de Entradas/Salidas (IOPS).

7.5 ADMINISTRACION DE MEMORIA.

OS/2 utiliza las grandes ventajas que surgen del procesador 80286. creando así un modelo de memoria virtual. donde el sistema permite accesar las aplicaciones en segmentos virtuales. debido a que la memoria física no se manipula directamente. Por qué se le llama segmentos de memoria virtual?. Porque las aplicaciones "desconocen" la posición física real de los segmentos. lo que

permite al sistema administrar el espacio de memoria física.

Si alguna aplicación llegara a necesitar un espacio en memoria, se crean entonces, entradas (que representan sectores) a una tabla de descriptores, posteriormente se asignan sectores para cubrir su petición: sin embargo los programas no podrán accesar ninguna de las tablas de descriptores (GDT o LDT), las estructuras de datos sólo son accesadas por el código del núcleo, corriendo en PL0 (nivel de privilegio cero), siendo el núcleo el que manipula las tablas de descriptores, al solicitarse para satisfacer las peticiones de administración de memoria de la aplicación.

De acuerdo con las características del modo protegido del procesador 80286, cada aplicación es aislada de otra y del sistema en general.

7.6 ADMINISTRACION DE INTERRUPCIONES

Existe una gran diferencia en el manejo de interrupciones en MS DOS y OS/2. En MS DOS se manejan directamente interrupciones, mientras que en OS/2 sólo se permite hacerlo a través de rutinas de servicios que son controladoras de dispositvos para recibir interrupciones. La Figura 7.6 muestra una interrupción realizada por el OS/2.

Las interrupciones son hechas por el manipulador de

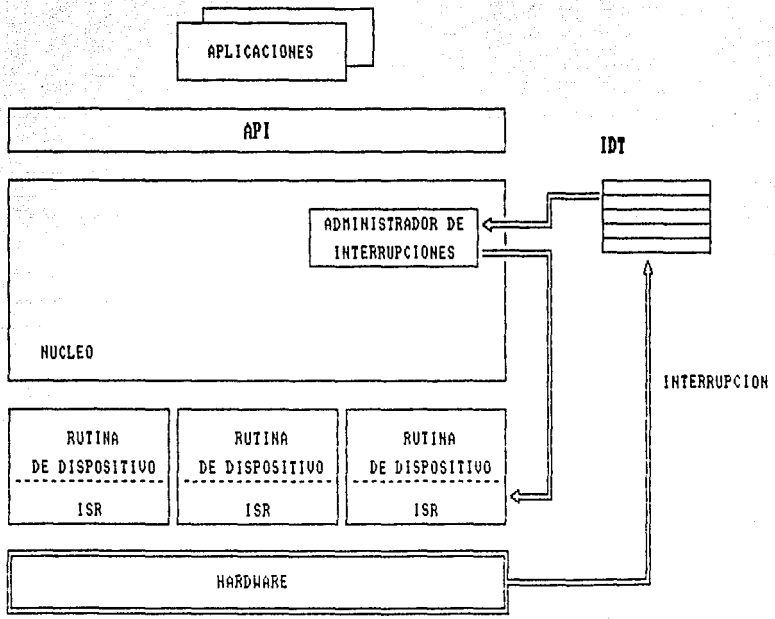


FIGURA 7.6 ANDMINISTRACION DE INTERRUPCION DEL OS/2

interrupciones del primer nivel del núcleo (FLIH). Algunas interrupciones como: señales de temporizador y fallos de procesador son controladas directamente por el núcleo, mientras que las de dispositivos de E/S son tratadas primero por el núcleo

y pasadas después a rutinas de dispositivo según sea el caso (ISR).

Capítulo 8

DISPOSITIVOS Y SUBSISTEMAS DE OS/2.

Previo al tema es necesario hacer notar que las rutinas de dispositivos, son los componentes de OS/2 que tienen como función el aislar los núcleos de los subsistemas, y aplicaciones del hardware, son también programas autorizados por OS/2, que corren en el mismo nivel de privilegio de ejecución del núcleo (PL0), poseen derechos de entrada y salida de información, siempre habilitado (se ejecuta en el contexto del núcleo, dado que no accesan a la API del OS/2), recibe servicios de un conjunto especial de interfaces del núcleo denominadas rutinas del ayudador de dispositivo (DevHlp). Cada uno de los dispositivos del hardware, tal como teclado, discos, impresora, etcétera, tienen asociada una rutina.

Todo aquel dispositivo que genere una interrupción deberá ser soportado por una rutina de dispositivo, de acuerdo con la Figura 8.1 las interrupciones hardware son enviadas al núcleo de OS/2, y el núcleo a su vez las translada a la rutina de dispositivo adecuada. En OS/2 la manipulación de interrupciones sólo se podrá lograr, a través de los códigos: núcleo, o bien, a través de rutinas de dispositivos.

8.1 TIPOS DE DISPOSITIVOS

De acuerdo con la arquitectura de las rutinas de

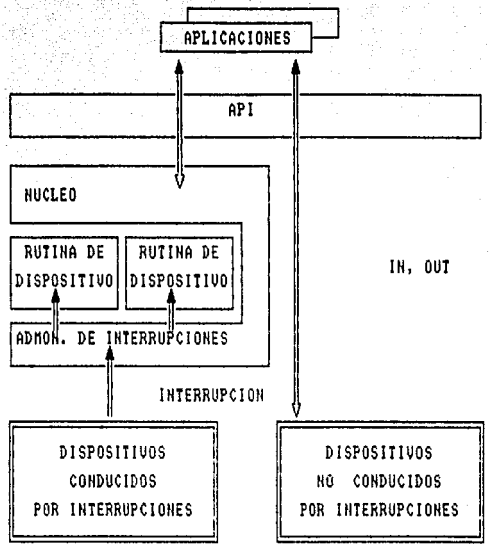


FIGURA 8.1 FLUJOS DE E/S DEL OS/2

dispositivos de OS/2. se dividen en: dispositivos de caracteres y dispositivos de bloques. Los primeros soportan dispositivos orientados al flujo.[1] La E/S de dispositivos de caracter se

1. Un dispositivo orientado al flujo espera datos de una secuencia particular.

realiza en forma de serie. es decir. el programa de aplicación debe leer todos los bytes en el orden en que fueron enviados por el dispositivo. sin permitir ninguna E/S aleatoria como sería el caso de los discos.

Cuando un dispositivo de caracter requiere de E/S de caracteres multiples. son utilizadas las aplicaciones de la API del OS/2.

Las rutinas de dispositivos de caracteres asignan nombres ASCII (de hasta ocho caracteres de longitud) a los dispositivos y se encargan directamente de la E/S de flujo de caracteres. controlando un carácter a la vez. Cuando se activa el sistema en forma general. el sistema de archivos de OS/2 añade los nombres de los dispositivos a su directorio interno. haciéndolos disponibles a las aplicaciones del OS/2. de forma tal que. cuando una aplicación requiera accesar a un dispositivo. solicita autorización al sistema de archivos a través de la función DosOpen (abrir archivos). utilizando el nombre del dispositivo para identificar la rutina adecuada. una vez logrado esto. el sistema de archivos establece un camino entre el programa y la rutina de dispositivo. El sistema de archivos es el encargado de pasar tanto las peticiones de entrada (con una función de LECTURA). como las de salida (a través de la función de ESCRITURA).

La Figura 8.2 muestra el funcionamiento de una rutina de un dispositivo de caracteres de OS/2 . la rutina (IMPRESION)

administra un adaptador de impresora. al iniciar el sistema OS/2 carga la rutina de dispositivo y coloca una entrada que contiene el nombre del dispositivo (IMPRESION) en el espacio de nombres del sistema de archivos, para que después sea arrancado un programa de aplicación que utilice el dispositivo impresor.

En los dispositivos de caracteres, los flujos de datos de caracteres de entrada y salida son procesados en serie. De acuerdo con el modelo de dispositivos de caracteres, estos dispositivos no soportan E/S de acceso aleatorio.

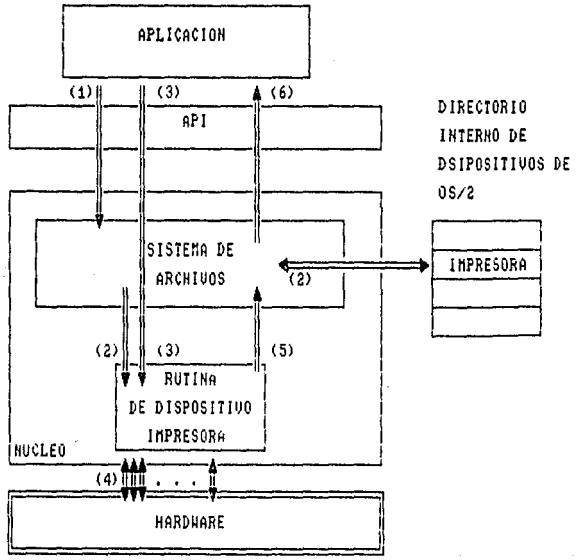


FIGURA 8.3 OPERACION DE UN DISPOSITIVO DE CARACTERES

De acuerdo con el ejemplo de la Figura 8.2 los números escritos entre paréntesis indican:

1) La petición hecha (de imprimir) por la aplicación, accesa al dispositivo de impresión, utilizando una llamada de ABRIR ARCHIVO.
2) El sistema de archivos verifica en su tabla el nombre del dispositivo solicitado y pide la autorización para accesar a la rutina.
3) La aplicación ofrece la petición de ESCRITURA multibyte al dispositivo.
4) La rutina de dispositivo de caracteres acepta la petición, e inicia la transferencia de datos al dispositivo, hasta finalizar el flujo de datos.
5) La rutina de dispositivo comunica al archivo del sistema la finalización de la petición hecha por la aplicación.
6) El sistema de archivos regresa un código completo al programa de aplicación.

En los dispositivos de bloque a diferencia de los dispositivos de caracteres, contiene generalmente, un extenso volumen de datos que se accesan en forma aleatoria, además de que el paso de sus datos es en bloques, que en la mayoría de los casos están ocultos en el sistema de archivos.

En aplicaciones de OS/2 , las operaciones de E/S de bloques son

representadas como E/S de dispositivo serie. Los datos de los dispositivos de bloque están representados como paquetes discretos recuperables en orden indistinto. los dispositivos de bloque de OS/2 son nombrados por un identificador de unidad de dispositivo. al primer dipositivo de bloque se le llama A:. al segundo B:. al tercero C:. y así sucesivamente.

Considerando el número de datos que manejan los dispositivos de bloque. frecuentemente es utilizado el hardware para la transferencia de datos directa a la memoria intermedia (buffer) de aplicación. A este hardware que es un dispositivo especial se le denomina controlador de acceso dinámico de la memoria (DMA)[1]. La Figura 8.3 muestra una rutina de dispositivo de disco instalada en el sistema. a la vez que se carga. OS/2 asigna al dispositivo lógico el identificador de dispositivo F: (para una mejor comprensión de la Figura 8.3 los números escritos entre paréntesis se muestran a continuación). Una aplicación puede leer datos en la forma siguiente:

1) Las peticiones de la aplicación. para el acceso al archivo F:TEST.ASM se hacen a través de la función ABRIR ARCHIVO del

---

[1] DMA es una CPU dedicada. que pasa datos entre dispositivo y memoria. sus transferencias son más eficientes ya que dejan al procesador principal para tareas más importantes.

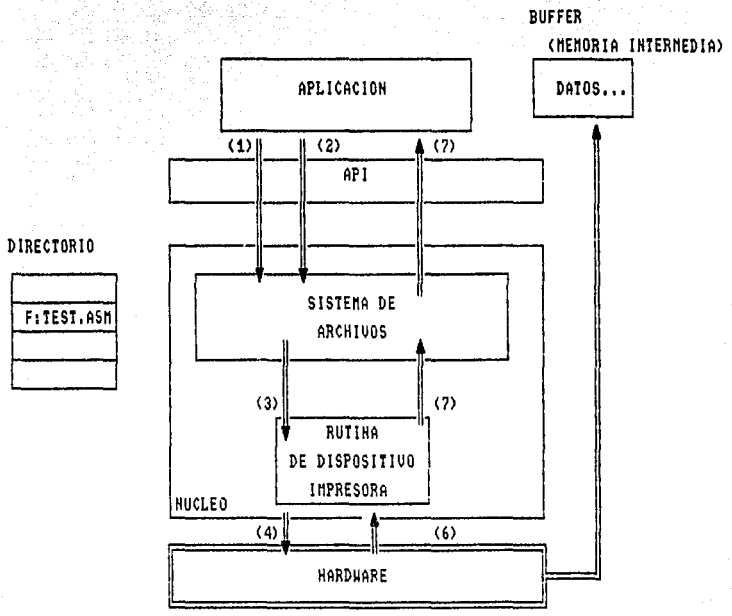


FIGURA 8.3 OPERACION DE UN DISPOSITIVO DE BLOQUE

sistema de archivos. Se verifica el nombre y devuelve un indicativo.

2) El programa facilita la función de LECTURA DE DATOS para los primeros 512 bytes de TEST.ASM.

3) Se procesa la petición en el sistema de archivos y de acuerdo a su información interna, determina que el dato pedido esté en un sector lógico del dispositivo F:. Se construye una petición para el sector, hecha por el sistema de archivos, y la transfiere a la rutina específica del dispositivo de bloque.
4) La rutina de bloque determina la posición precisa del sector lógico (pista, cabeza, etc.) y envía la petición al hardware.
5) El hardware mueve el sector requerido directamente al buffer de datos de aplicación.
6) El hardware activa la rutina de dispositivo generando su interrupción.
7) El controlador de disco notifica al sistema de archivos, que a su vez informa a la aplicación que la petición de E/S se ha completado y el dato se encuentra en el buffer (memoria intermedia).

Las rutinas de dispositivos de OS/2 tiene el papel de proporcionar al núcleo una interfaz estándar con los dispositivos de E/S a el sistema. A diferencia de los programas de aplicación, las rutinas de los dispositivos de OS/2 tienen una estructura de ordenes con interfaces y subcomponentes bien definidos. Al conjunto de reglas que definen como se construye una rutina de dispositivo se le llama modelo de la rutina de dispositivo.

Los sistemas de multitarea arrancan operaciones de E/S y corren otros programas, mientras el dispositivo (relativamente lento) está realizando operaciones. Este concepto, llamado E/S solapadas, es implementado en OS/2 con la cooperación de las rutinas de dispositivos.

Una rutina de dispositivo tiene los siguientes componentes:

- Rutina de inicialización
- Rutina de estrategia
- Rutina de interrupción

Rutina de inicialización: se ejecuta al cargar por primera vez la rutina del dispositivo (sólo corre una vez), lo inicializa, estableciendo cualquier estructura de datos que sea necesaria en la unidad de dispositivo al correr el sistema. Esta rutina corre con un thread (hilo) del sistema en modo protegido con privilegio de E/S.

Rutina de estrategia: recibe las peticiones de E/S del núcleo, interpreta los paquetes de petición del mismo y comienza la operación de E/S. Para el caso de dispositivos de E/S con cola, la rutina de estrategia (actua como una <<subrutina de E/S>>) añade en ocaciones al paquete de petición la cola de trabajo del dispositivo, si esta ya está sirviendo una petición previa de E/S. Es responsable de convertir direcciones lógicas del paquete (apuntadores del buffer) en direcciones físicas, que más tarde pueden ser utilizadas con la interrupción. Las rutinas de

estrategias corren en PL0 en modo del núcleo.

Rutinas de servicio de interrupciones (ISR): es el único programa de OS/2 autorizado a recibir interrupciones hardware. Al controlar la rutina de interrupción, se tiene acceso al espacio de direcciones del sistema (GDT), pero no hacia direcciones de una aplicación específica (LDT). La mayoría de los sistema tienen un dispositivo hardware especial, denominado controlador de interrupciones.[1] Las rutinas de interrupción corren en PL0 en modo del núcleo. Al habilitarlas, una ISR se suspende por otra interrupción.

8.2 DISPOSITIVOS DEL OS/2

Las rutinas de dispositivo de OS/2 están incluidas en su sistema, procesan paquetes de petición desde el núcleo, imitan servicios del BIOS y soportan nuevas aplicaciones de OS/2, así como programas de compatibilidad del DOS.

8.3 POR QUE SON NECESARIOS LOS SUBSISTEMAS?

Hasta aquí los servicios de entrada y salida de datos de OS/2 se han visto en un contexto de modelo de archivo en serie, conformados como un subconjunto de las capacidades generales del sistema de entrada y salida. OS/2 incluye un conjunto de dichos servicios para los dispositivos, que son necesarios en la

---

1. El controlador de interrupciones asocia interrupciones de dispositivos con niveles de petición de interrupción (IRQ). Las interrupciones en un IRQ alto tiene prioridad sobre las que tengan un IRQ bajo.

implementación de una interfaz de usuario de OS/2 (tal como el teclado, presentación por viedo y mouse). En estos dispositivos el sistema de entrada y salida de bajo nivel se ejecuta a través de componentes especiales llamados subsistemas.

Podemos preguntarnos, por qué existen estos servicios, a caso el modelo de archivos no proporciona todas las funciones necesarias para el acceso a estos dispositivos? La respuesta está en relación a la capacidad de respuesta frente a una función sencilla. En ocasiones las interfaces avanzadas del usuario necesitan que se trate a la pantalla como un espacio de presentación con acceso directo a sus distintas coordenadas. Las porciones aleatorias de la pantalla se actualizan conforme el usuario se mueve de un área a otra. La mayoría de las aplicaciones de pantalla completa necesitan un rendimiento óptimo de video, ya que la capacidad de respuesta de la interfaz es crítica con respecto a la utilidad de la aplicación. Al bajar la jerarquía funcional se gana en rendimineto pero se pierde en funcionabilidad. La Figura 8.4 muestra las aplicaciones de entrada y salida por video en PC DOS. En el nivel más alto, se encuentran los programas de aplicación que accesan a la consola por medio del sistema de archivos. En tal caso, la aplicación obtiene los servicios de direccionamiento, soporte ANSI y una representación de alto nivel de la pantalla, como si la consola

fuera un terminal pasivo[1]. Si el redireccionamiento de entradas y salidas del archivo es importante. la aplicación accesa directamente a los servicios del BIOS escribiendo aleatoriamente los caracteres y sus atributos.

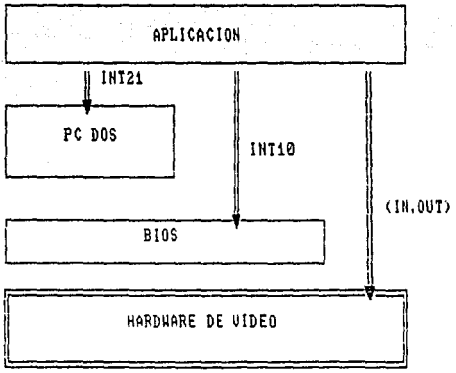


FIGURA 8.4 APLICACIONES DE E/S POR VIDEO EN PC DOS

1. Un terminal pasivo es un término usado para describir los dispositivos de presentación sin inteligencia local. Este tipo de dispositivo frecuentemente se diseña para soportar un flujo de datos particular (tal como ANSI. 3270. etc.)

8.4 ESTRUCTURA DE E/S DEL OS/2

El susbsistema de E/S de OS/2 aumenta el rendimiento. cuando un programa de aplicación utiliza los atributos genéricos de la E/S estàndar.

Cada uno de los subsistemas suministra un conjunto de API de enlace dinàmico a través de los cuales una aplicaciòn accesa a sus servicios. Además cada subistema tiene su archivo de tiempo de ejecuciòn DLL y una unidad de dispositivo.

La Figura 8.5 muestra el flujo global del sistema de E/S.

En el nivel superior una aplicaciòn hace llamadas directas al sistema de archivos o al subsistema de E/S. Cuando se requiere de E/S a través de un dispositivo manual, la llamada va primero al sistema de archivos para determinar si el redireccionamiento es efectivo y se manda la salida al dispositivo adecuado o al archivo. En caso de no estar redireccionado, la petición pasa a un subsistema de E/S para su determinación. Cuando un programa de aplicaciòn no requiere de servicios de redireccionamiento, se llama directamente al subsistema de E/S.

Las API del subsistema de E/S, forman parte de la API del OS/2 total. Al utilizarse estas API directamente, el programa de aplicaciòn no utiliza el modelo de flujos de datos en serie y las capacidades de redireccionamiento del modelo, sin embargo el programa adquiere el control sobre el dispositivo.

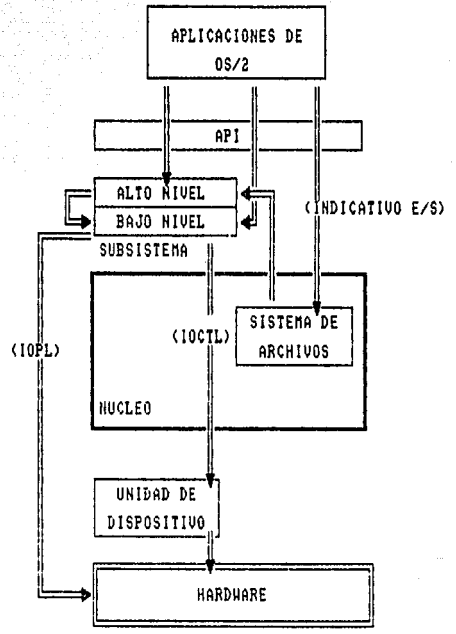


FIGURA 8.5 FLUJOS DE E/S EN OS/2

Los subsistemas de E/S del OS/2 son reemplazables por programas de aplicación o extensiones del sistema. La Figura 8.6 muestra la estructura de un subsistema, que contiene un componente para la ruta y un conjunto de rutinas de servicio.

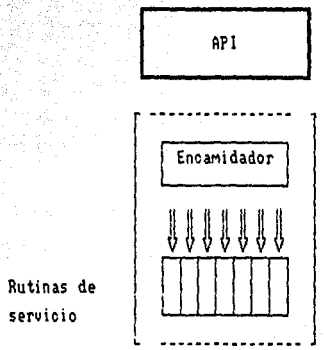


FIGURA 8.6 ESTRUCTURA DEL SUBSISTEMA DE E/S.

Cuando un programa lee o escribe desde uno de los subsistemas de E/S, la petición es procesada por el encaminador o marcador de ruta, y generalmente pasada a la rutina de servicio de E/S suministrada por el sistema.

OS/2 implementa múltiples sesiones[1] de usuario, cada sesión tiene su propio conjunto de área de datos que define el estado del teclado, presentación por video o ratón.

En el caso de los servicios por video, el sistema mantiene un buffer de video lógico (LVB: Logical Video Buffer) para cada

1. Una sesión es uno o más procesos ejecutándose juntos y compartiendo todos el teclado, presentación por video y ratón. Generalmente una sesión corresponde con un programa de aplicación.

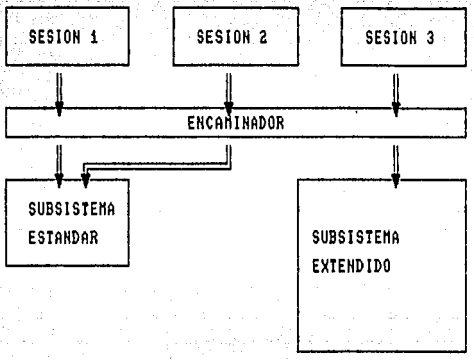


FIGURA 8.7 SESIONES EN OS/2

sesión. El usuario de OS/2 puede arrancar hasta 12 sesiones concurrentemente. La Figura 8.7 muestra cómo el sistema soporta múltiples sesiones.

Capítulo 9

DESARROLLO DE SOFTWARE EN OS/2

Si bien el software existente proporciona grandes alternativas, el desarrollo de este es bastante considerable. Uno de los aspectos importantes de OS/2 en el manejo de información es el acceso a la misma, tomando como punto básico que su obtención sea rápida, fácil y adecuada. Actualmente existen programas de propósito general, que se conectan a diferentes subsistemas, emulando terminales, transfiriendo archivos y automatizando operaciones. En seguida se presentan algunos elementos de programación que brindan la oportunidad de facilitar y transferir información, de manera compatible con OS/2.

9.1 REXX

REXX es una estructura de programación de lenguajes, asignada a leer y escribir datos en forma sencilla. Fué concebido e implementado considerando las necesidades de los usuario entre 1979 y 1982 por Mike Cowlishaw de IBM.
La compañia lo describe como un proceso que permite realizar programas basados en algoritmos claros y en forma estructurada, dando la oportunidad al programador de actualizarlos en forma eficaz y sencilla, facilita la manipulación de objetos simbólicos

transformàndolos en palabras, números y nombres. Fuè diseñado para hacer màs accesible el manejo de datos, por lo que la información, se procesa en: forma dinàmica, por cadena de caracteres, con almacenaje directo, además de que se cuenta con acceso directo al sistema de comandos. Entre las características de REXX se encuentra la de, ser una herramienta eficiente en el manejo de aplicaciones de OS/2.

Es útil para programadores que necesitan de lenguajes de programación ràpidos.

El aspecto de REXX no difiere demasiado de PASCAL, C, o de otros lenguajes que tienen como ancestros a ALGOL. En consecuencia REXX tiene mucho en común con ALGOL que contiene, procedimientos, variables, expresiones, estructuras de control, subrutinas, etc.

En seguida se muestra un programa escrito en REXX que consta de una lista de ordenes (menú), y la pregunta de, la selección deseada de estas, para llevar a cabo la ejecución de ella.

```
/* execute file utilities */
say 'Enter file name:
pull file_name
say 'Choose a file operation by number:'
say '    1 - Edit'
say '    2 - Print'
say '    3 - Delete'
pull response
select
when response = 1 then 'edit' file_name
when response = 2 then 'print' file_name
when response = 3 then 'erase' file_name
otherwise
say response 'is an incorret choice.'
end
exit
```

Considerando las ventajas de OS/2. como son las de: multitarea y directorios de espacios extensos: las cuales forman un ambiente ideal para REXX. Todo lo que OS/2 necesita es aumentar la interface de comunicación entre REXX y sus aplicaciones individuales.

9.2 PM (Page Maker)

Después de más de dos años de desarrollo la empresa ALDUS PAGE-MAKER para OS/2 empezó a distribuir dicho producto en Septiembre de 1989.

Los Ingenieros de ALDUS han trabajado con Microsoft en el desarrollo de varios experimentos de Software. La compañía creó y puso a la venta nuevas versiones del PM para Macintosh y Windows. dando a ALDUS la oportunidad de entrar al mundo de OS/2.

Su intención principal era usar todo lo aprendido con Windows, preservando el dasarrollo del PM, aprovechando las cualidades de OS/2 para incrementar la productividad cuanto fuera posible.

PM es una convinación de dos niveles API : administración de ventas y manejo de gràficas.

Para crear sesiones de comandos múltiples, se ofrece la oportunidad de utilizar la capacidad de OS/2, cualidad adecuada para compilar o transferir información en redes ràpidamente.

PM para OS/2 trae muchos beneficios para sus usarios, su arquitectura da una coordinación muy rígida. También es posible trabajar con múltiples redes a varios niveles de visibilidad.

9.3 QUE HAY CON WINDOWS 3.0?

Hace algunos años se tenía un sistema operativo dominante (MS-DOS), con características de monotarea que permite correr sus programas, en el entonces enorme espacio de memoria de 640 k, definido por el procesador 8088 y la arquitectura de IBM PC.

Después IBM introduce el primer PC AT en 1984, anunciando cambios significativos. La AT se construyó sobre el procesador INTEL 80286. A diferencia del 8088 en las máquinas anteriores, el 286 podía operar en modo protegido, dando a los programas acceso hasta 16 MB de memoria y se presta para implementar

operaciones reales de multitarea en la PC.

Windows 1.0 al correr en máquinas con procesador 8088 realizaba un manejo de memoria sofisticado. movía segmentos de programa y datos en memoria. permitía que multiples instancias de un programa compartieran código y datos localizados en bibliotecas de enlace dinámico. así como sacar segmentos de código de programas de la memoria al realizarse una nueva ejecución. El manejo de memoria emplementado en Windows es realmente uno de sus mayores logros.

Muchos programadores de Windows con manejo de memoria en modo protegido. escribieron programas para Windows que seguian cuidadósamente sus reglas de programación y que no intentaban hacer nada que pudiera causar problemas si los programas fueran a correr algún día en ese modo.

Windows ha sido un producto muy exitoso de Microsoft . Si sus ingenieros de software encuentran cómo mejorar a Windows corriendo programas para Windows en modo protegido. el producto resultante competiría con OS/2.

Windows y PM son similares en cuanto a su estructura general y en la interfaz con el usuario. sin embargo. son muy distintos en los detalles de la interfaz con el programa de aplicación. El convertir un programa de Windows al PM (o viceversa) no es tarea fácil.

Windows 3.0 da a los programadores la memoria necesaria para aplicaciones grandes y sofisticadas y Windows tiene la base de

usarios va establecida como para proveer utilidades inmediatas.

9.4 ALGUNAS MEJORAS DE LA VERSION OS/2

El modelo de direccionamiento y organizaciòn usada por el procesador de 32 bits representa un paso adelante en la evoluciòn del software en los procesadores 80 y 86.

Ya no se usan los segmentos y selectores que eran comunes e indispensables para toda operaciòn de la memoria en las mejoras hechas a OS/2. En cambio permite casi completamente implementar una protecciòn a la memoria, tanto compartida como virtual, en la unidad de pàgina del procesador 80386.

Al iniciarse en la programaciòn con OS/2 mejorado en el modelo de 32 bit, automàticamente se olvida el pequeño, mediano, compacto, modelo de memoria debajo de DOS y OS/2.

La aplicaciòn del 32 bit es muy parecido a MS DOS. COM, excepto en el segmento en el que estarà 4GB, en lugar de 64K.

Sin el segmento de 4GB, la aplicaciòn del còdigo de datos residiria al final. Por lo tanto para la perspectiva de aplicaciòn, todos los saltos y llamadas estàn cerca, incluyendo la llamada a funciones del sistema operativo.

Dentro de las mejoras a OS/2, la aplicaciòn en el espacio de direcciòn es limitado a 512MB. Esta es una divisiòn de tres àreas: el còdigo de aplicaciòn estàtica, datos y el còdigo de enlace (incluyendo los puntos de entrada de todas las funciones

del sistema operativo), están en niveles superiores, y la memoria no grabada entre los anteriores.

Lo más importante acerca de las mejoras, es la liberación y asignación de bytes, el segundo punto es la distinción entre memoria de asignación y memoria directiva.

Las facilidades de las mejoras para la protección de la memoria son el tercer punto importante. Esto se basa en la captabilidad de la unidad 80386.

Para la perspectiva de los programas el gran cambio de OS/2 sobre previas traducciones será que ya no habrá más segmentos.

OS/2 cuenta con 32 bits registrados , dirige una memoria virtual y posee la habilidad para correr múltiples programas.

La versión mejorada de OS/2 continua usando segmentos, utilizando la expansión o dirección de los mismos.

Al cargar OS/2 se da forma a cuanto se desarrolla, utilizando los servicios y aplicaciones que brindan las mejoras de OS/2.

9.5 SOFTWARE EXISTENTE PARA OS/2

Actualmente existe una gran variedad de elementos adicionales para elevar el nivel del sistema operativo OS/2.

Se crean un sin fin de productos, lo único que se necesita es tener un objetivo y creatividad.

Existen algunas alternativas de programación. entre ellos las redes de los noventas.

La mayoría de los servidores de redes funcionan como administradores de periféricos compartidos. Permiten a las PC's compartir impresoras y discos duros.

Como LAN Manager es una aplicación que se procesa bajo OS/2. puede ejecutar programas de hojas de cálculo. procesamiento de textos y otros. mientras trasmite o mueve información a otras computadoras de la red. Lo cual significa que cada cliente en la red ejecutando OS/2 actua como un servidor LAN Manager de impresoras. archivos o comunicaciones. y como una estación de trabajo al mismo tiempo.

Capitulo 10

FUTURO DE OS/2

La continua innovación en interfaces, manejo de memoria, procesamiento, multitarea y conectividad hacen que OS/2 se mantenga en un punto óptimo.

10.1 COMPARACIONES CON OTROS SITEMA OPERATIVOS

El sistema operativo MS DOS subsiste, aún cuando se enfrenta a un OS/2 tecnológicamente superior.

En la actualidad aproximadamente el 12% de usuarios utilizan computadoras con procesador 386, siendo esta característica, un requisito suficiente para correr eficientemente el sistema operativo OS/2. Las ventas de procesadores 386 aumentan, aunque se requerirá también que los precios de la memoria disminuyan considerablemente.

Hay cierta similitud en el manejo de interfaces de Windows con respecto de OS/2. Presentation Manager, del OS/2, ofrece una interface gráfica parecida a la de Macintosh que proporciona no sólo una perspectiva más secilla y más visual de la computación personal, sino también un enfoque mucho más consistente.

OS/2 es capaz de direccionar varios gigabytes de memoria.

Para los usarios, el recurso del OS/2 de ejecutar dos o màs aplicaciones simultàneamente es quizà su beneficio funcional màs importante.

Aunque, hay otros productos que le permiten realizar algunas funciones de multiplicaciòn de tareas con MS DOS. VM/386 Multitasker (IGC), por ejemplo, no genera ventanas, pero permite a los usuarios de màquinas con procesador 386 con dos o màs megabytes de memoria, dividir sus recursos en un nùmero de sesiones con MS DOS simultàneos o independientes.

Los sistemas operativos multiusuario compatibles con MS DOS, como Concurrent DOS y PC-MOS, ofrecen tambièn una ruta hacia el proceso multitarea, aunque no son tan adecuados para el uso individual como lo son Windows y Desqview.

Tècnicamente, la multitarea es el proceso de obtener dos o màs programas en operaciòn (càlculo, bùsqueda, etc.) al mismo tiempo. Pero en tèrminos orientados a soluciones, la multitarea ofrece la posibilidad de cambiar ràpidamente entre varios programas sin ningùn procesamiento a fondo. Esta conmutaciòn, le permite moverse ràpidamente de una aplicaciòn a otra.

OS/2 es un sistema operativo multitarea concebido y diseñado con necesidades de enlace en redes en mente, permitiendo a grupos de trabajo compartir aplicaciones y datos de manera eficiente y sin mayores complicaciones.

OS/2 se enfrenta a una fuerte competencia de sistemas operativos de LAN existentes, entre ellos Netware 386, de Novell y Vines, de Banyan. Estas redes de diseño propio están hechas para servidores de trabajo con MS DOS y OS/2.

10.2 COMPARACIONES CON WINDOWS 3.0

La parte gráfica de Windows 3.0, la hizo más fácil de usar y virtualmente idéntica a la interface del Presentation Manager de OS/2. Program Manager permite agrupar aplicaciones en ventanas separadas y utilizar iconos en color para marcar programas de y no de Windows. Los colores y patrones de fondo son más agradables a la vista, y existe un efecto de tercera dimensión para muchos de los elementos visuales. También se agregó un programa de dibujo más óptimo y un nuevo medio de registros de macros al repertorio de accesorios de escritorio de Windows.

Sin embargo, más importante que cualquiera de las modificaciones de tipo cosmético es la posibilidad de Windows 3.0 de funcionar en modo protegido en computadoras con procesador 286 y 386, y con lo cual se direccionan hasta 16 MB de memoria. Hasta ahora, las aplicaciones de Windows y MS DOS operaban sólo en modo real, lo cual limita la memoria directamente accesible a 640 K. Pero funcionando en el modo protegido, Windows 3.0 permite designar memoria extendida (cerca del nivel de 1 MB), semejante a OS/2. En sistemas 386, Windows utilizará también técnicas de

memoria virtual (almacenando temporalmente cualquier elemento que no quepa en la memoria RAM de su disco duro) para extender el alcance de su memoria.

Un impacto inmediato en el manejo de la memoria serán los recursos multitarea más sencillos y rápidos.

Con 2 MB o más RAM en la máquina, se podrá cargar y ejecutar una combinación de aplicaciones de Windows y MS DOS estándar en forma multitarea. Windows no soporta procesos simultáneos dentro de la misma aplicación (conexión múltiple) como OS/2, pero el recurso multitarea que lo permite debe ser suficiente para cubrir las necesidades de usuarios de MS DOS. Muchas aplicaciones de Windows no funcionan con Windows 3.0 sin una reparación menor, para que funcione con el nuevo esquema de manejo de la memoria. Windows no es una plataforma de servicio de redes, de manera que no es solución definitiva para grupos de trabajo. Sistemas operativos de LAN con Netware de Novell y OS/2 LAN Manager de Microsoft soportan fácilmente estaciones de trabajo con Windows instalado.

C O N C L U S I O N E S

Después de haber indagado en el interior del Sistema Operativo OS/2, cabe mencionar los diversos servicios que brinda para el usuario.

* Control en el manejo de procesos, permitiendo la creación, suspensión, comunicación y terminación de éstos.
* Adecuada administración de la memoria, en la ejecución y asociación de procesos y sobre los dispositivos periféricos.
* Creación de interfases utilizadas para la construcción de programas de aplicación que escriban en archivos y dispositivos de entrada y salida de manera particular: a partir de un modelo de archivo en serie.

Lo anterior es importante para aquellas personas interesadas en el desarrollo de software de apoyo: hojas de cálculo, manejadores de bases de datos, etcétera; ya que utilizan comandos tales como comunicación y creación entre procesos, manejo de dispositivos y diferentes tipos de terminales.

OS/2 es un sistema operativo con medio ambiente sobresaliente. Provee varias características, comenzando con operaciones de 32 bits e incluyendo multitarea con prioridad.

que permite que corran múltiples aplicaciones a la vez.además de un control y coordinación de una o más de ellas. multiples hilos de ejecución. apoyo para diferentes procesadores así como para redes. protección de datos y seguridad.

OS/2 se basa en sistemas de 32 bits. diseñado completamente para las PCs basadas en Intel.

Entre las capacidades de contar con 32 bits está la velocidad o capacidad para manejar grandes bloques de datos. permitiendo así mejorar su rendimiento. Los primeros sistemas de 16 bits ; podían tener acceso hasta 64k de memoria. pero no podían acomodar matrices mayores en ella.

Para rebasar el límite de 640k era necesario insertar un nuevo proceso de acceso a la memoria. que combina una dirección de 4 (bits) segmentos con un desplazamiento de 16 bits elevando así los límites de 640k a 1MB.

OS/2 programa operaciones entre aplicaciones. dando una porción de tiempo para cada una con los recursos del sistema y la CPU. Cuando se le acaba el tiempo a una aplicación se suspenden las operaciones de la aplicación pero se guarda su estado y se restaura cuando le vuelve a tocar el turno de aplicación.

Dado que cuenta con modo protegido. cada aplicación recibe su propia área de memoria en las que otras aplicaciones no entran. también apoya los múltiples hilos de ejecución separando operaciones de CPU; los hilos se ejecutan simultaneamente .

A nivel de estudiantes. es necesario conocer los componentes del

mismo, ya que el anàlisis de los conceptos fundamentales, les permitirà obtener mayores alternativas para crear gran diversidad de software y una mejor visualizaciòn de los problemas al momento de programar debido a que hay màs conocimientos en el manejo de informaciòn a nivel interno.

OS/2 es un Sistema Operativo extenso, que se encuentra en constante evoluciòn, por tal motivo esta tesis abarca conocimientos generales de la estructura interna del sistema.

Por lo tanto, a continuaciòn se enuncian una serie de trabajos que pueden surgir de esta tesis tales como:

- Implementaciòn de una red de equipos con sistema operativo OS/2, manejado por la estructura interna y que sea transparente a los procesos del usario.
- Estudio de algoritmos en el manejo de la informaciòn de: el sistema de archivos, memoria general y memoria secundaria.
- Implementaciòn practica de un sistema de archivos.
- Implementaciòn del control de procesos cuando se tienen varios procesadores en una computadora.
- Implementaciòn de OS/2 en sistemas distribuidos.
- Creaciòn y modificaciòn de manejadores de dispositivos.

Este trabajo expone los valores que el Sistema Operativo proporciona al usuario para el adecuado manejo de la informaciòn y muestra las ventajas que, el mismo presenta, sobre MS-DOS.

Para poder profundizar en la comprensiòn de OS/2 es necesario

tener un nivel de conocimientos y experiencia en lenguajes de alto nivel, bastante avanzado.

Glosario Bilingüe

Administrador de entradas/salidas:	Driver
Afinamiento:	Tunning
Arquitectura con marcas:	Tagged architecture
Biestable:	Flip-flop
Bloque común:	Common Block
Bloque de control de proceso. BCP:	Process Control Block. PCB
Buffer de video lògico:	Logical Video Buffer: LVB
Canal. procesador de entradas/salidas:	Channel, I/O processor
Capacidades:	Capabilitves
Carga de prueba tipica:	Benchmark
Carga por demanda:	Load on demand
Ciclo infinito:	Loop
Cierre de archivos:	File locking
Componentes de programaciòn (tecnologia desincorporada):	Software
Componentes electrònicos (tecnolooia incorporada):	Hardware
Computadoras con arquitectura de pilas:	Stack machines
Con retiro de procesador:	With preemption
Conducto:	Pipe
Depuraciòn. Depuradores:	Debugging. Debuggers
Desarrollo ascendente:	Battom up
Desarrollo descendente (por refinamientos sucesivos):	Top down
Desbordamiento:	Overflow
Desempeño:	Performance
Despachador (asignador de procesos al procesador):	Dispathcer.Low-level scheduler
Desvio:	Trap
El primero que se ajuste:	First fit
El que màs se ajuste:	Best fit
En linea:	On line
Ensamblador:	Assembler
Espacio de trabajo:	Working set

Estudios de desempeño:	Performance evaluation
Etapa:	Step
Expresiones de camino:	Path expressions
Fuentes de caracteres:	Fonts
Fuera de línea:	Off line
Graficador:	Plotter
Hilo:	Thread
Indicador:	Flag
Indicativo:	Handle
Iniciación:	Boot
Iniciación en caliente:	Warm boot
Iniciación en frio:	Cold boot
Intensivo en uso de procesador:	CPU bound
Intensivo en entradas/salidas:	I/O bound
Interbloqueo:	Deadlock
Intercambio global:	Swapping
Llamada lejana:	For call
Memoria de lectura exclusiva:	Read only memory (ROM)
Método de los compañeros:	Buddy system
Método de los tanques:	Quickcell
Multiproceso:	Multiprocessing
Multiprogramación:	Multiprogramming
Ordenamiento:	Sorting
Palabra de estado de programa. PEP:	Program Status Word.PSW
Parche:	Patch
Pila:	Stack
Pista:	Track
Planificador (asignador de trabajos al sistema):	Job Scheduler,High-level scheduler
Planificador de nivel intermedio:	Intermediate-level shceduler
Procedimiento. rutina. subrutina:	Procedure. routine. subroutine
Procesador:	Processor. CPU
Procesador. Tarea:	Process. Task
Programa:	Program
Punto de interrupción:	Breakpoint
Reciclaje:	Round Robin
Recubrimiento parcial:	Overlay
Recubrimiento total:	Chaining
Registros asociativos para traducción de direcciones:	Translation Lookaside Buffer o T.L.B
Reloj:	Timer
Respaldo:	Back up
Retorno de carro:	Carriage return
Sección de encadenamiento:	Linkage section
Seguimiento:	Trace

Señal de listo a recibir:	Promt
Sistemas de lectores e impresoras virtuales:	Spooling system
Sistema de tiempo compartido:	Timesharing system
Sistema grande:	Mainframe
Sobrecarga:	Overhead
Sondeo:	Polling
Tiempo de residencia en el sistema:	Elapsed time
Trabajo:	Job
Tratamiento por lotes o tandas:	Batch processing
Vaciado de memoria:	Dump
Varios conductos:	Pipeline
Vaya a:	Go to
Zona intermedia o amortiguadora:	Buffer

I N D I C E

B I B L I O G R A F I A

Andraws. S. Tanenbaum
"SISTEMAS OPERATIVOS / Diseño e Implementación"
TRADUCCION: Juan Carlos Vega Fogoaga
Traductor Técnico. IAMC.
Traducido de la primera edición:
OPERATING SYSTEMS: DESING AND IMPLEMENTATION
Lugar de Origen: united States
Ed. Prentice-Hall Software Serie. 1988
No. de pag.: 274

Allport Crooks
"User Guide to OS/2"
United States of America. 1989
Primera Edición
No de pag.: 429

Kathleen and Daniel Paquette
TW
"Understanding OS/2 With Presentation Manager"
Ed. Howard W.Sams & Company
Indianapolis. Indiana. 1989
Primera Edición
No. de pag.: 279

Dick Coklin
"OS/2 A Busines Perspective"
Ed. Jonh Wilev & Sons
Canada. 1989
Primera Edición
No. de pag.: 258

Jeffrey I.Krantz. Ann M. Mizell. Robert L. Willians
TM
"OS/2 Features Functions. and Aplications"
Ed. Jonh Wiley & Sons. Inc.
United States of America. 1988
Primera Edición
No. de pag.: 282