

**Datos generales**

**Posgrado:** Maestría

**Area :** Computación

**Asignatura:** Sistemas y ambientes operativos

**Ubicación:** primer semestre

**Horas teoría:** 3

**Horas práctica:** 2

**Créditos:** 8

**Materias antecedentes relacionadas:** Ninguna

Materias paralelas relacionadas: Estructura de datos en C, Tecnologías de información.

Materias consecutivas relacionadas: Diseño de base de datos

**OBJETIVO GENERAL.**

Que el alumno conozca los esquemas de operación de los sistemas operativos actuales y sea capaz de desarrollar un sistema operativo prototipo.

**CONTENIDO****1. Antecedentes****1.1 Definición****1.2 Evolución histórica****1.3 Conceptos básicos****1.4 Funciones de un sistema operativo****1.5 Tipos de sistemas operativos****1.4 Estructura de un Sistema Operativo****2. Estructura de un sistema computacional****2.1 Arquitectura básica****2.2 Estructura de E/S****2.3 Estructura de almacenamiento****2.4 Protección del hardware****2.5 Sistema de interrupciones**

2.6 El mapa de direcciones

2.7 Estructura de red

### 3. Procesos

3.1 Introducción a los procesos

3.2 Problemas clásicos de la comunicación entre procesos

3.3 Gestión de procesos

3.4 Planificación de procesos

3.4.1 Algoritmos de planificación

3.4.2 Evaluación del rendimiento

3.4.3 Comunicación y sincronización de procesos

### 4 Administración de la memoria

4.1 Administración de la memoria sin intercambio

4.2 Particiones e Intercambio

4.3 Memoria virtual

4.4 Algoritmos de reemplazo de paginas

4.5 Modelación de algoritmos de paginación

4.6 Segmentación y paginado

### 5. Sistemas de archivos

5.1 Archivos

5.2 Directorios

5.3 Implantación del sistema de archivos

5.4 Seguridad

5.5 Mecanismos de protección

### 6. Entrada/Salida

6.1 Hardware de entrada/salida

6.2 Software de entrada/salida

6.3 Discos

6.4 Relojes

6.5 Terminales

### 7. Bloqueos

7.1 Recursos

7.2 El algoritmo del avestruz

7.3 Evasión de bloqueos

7.4 Prevención de bloqueos

8. Sistemas operativos actuales, esquema de trabajo y operación.

8.1 Windows

8.2 9x.

8.3 NT

8.4 2000

8.5 Linux

8.6 Solaris

9. Sistemas operativos distribuidos

9.1 Conceptos básicos, Historia y objetivos de diseño.

9.2 Comunicación en los sistemas distribuidos

9.3 Sincronización

9.4 Procesos y procesadores en sistemas distribuidos

9.4.1 Hilos

9.4.2 Modelos de sistemas

9.4.3 Asignación de procesadores

9.4.4 Planificación en sistemas distribuidos

9.5 Sistemas distribuidos de archivos

9.5.1 Diseño

9.5.2 Implantación

9.5.3 Tendencias en los sistemas distribuidos de archivos

9.6 Memoria compartida distribuida.

9.7 Sistemas distribuidos: CORBA

9.7.1 Sistemas basados y orientados a objetos.

9.7.2 Componentes de CORBA.

### 9.7.3 CORBA y la distribución

## 10. Implementación de un sistema operativo

### 10.1 Entrada / Salida estrategias de programación

### 10.2 El núcleo del sistema

## LINEAMIENTOS DIDÁCTICOS

Técnicas: Exposición del maestro. Exposición en grupo e individual por parte de los alumnos, discusión dirigida y lluvia de ideas.

Herramientas: pizarrón, computadora, proyector de acetatos, practicas dirigidas y técnicas de trabajo en grupo e individual.

## LINEAMIENTOS DE EVALUACIÓN

Participación y trabajo en equipo, practicas de laboratorio, elaboración de proyectos, entrega de ensayos, trabajos de investigación y examen final.

## BIBLIOGRAFÍA BASICA

Milenkovik Milan. Sistemas operativos conceptos y diseño: Mc Graw Hill

S. Tannembaum Andrew. Sistemas operativos diseño e implementación: Prentice Hall

S. Tannembaum Andrew. Sistemas operativos modernos: Prentice Hall

S. Tannembaum Andrew. Sistemas operativos distribuidos: Prentice Hall

Patterson-Hennesey. Organización y diseño de computadoras: Mc Graw Hill

Patterson-Hennesey. Arquitectura de computadoras un enfoque cuantitativo: Mc Graw Hill

Thomas Phd. Sistema operativo UNIX Guía del usuario: Mc Graw Hill

Silvershchatz, Calvin. **Operating Systems Concepts**. Addison Wesley, 1995.

Folk, Zoellik: **File Structures**, 2nd ed. Addison Wesley, 1992.

Singhal & Shivaratri: Advanced Concepts in Operating Systems, Addison Wesley, 1994.

Wall Kurt. Programacion en Linux 2a. edicion, Prentice Hall, 2001.

## APUNTES DEL CURSO

### 1. Antecedentes

#### 1.1 Definición

Un sistema operativo es el encargado de brindar al usuario una forma amigable y sencilla de operar, interpretar, codificar y emitir las ordenes al procesador central para que este realice las tareas necesarias y específicas para completar una orden.

El sistema operativo, es el instrumento indispensable para hacer de la computadora un objeto útil. Bajo este nombre se agrupan todos aquellos programas que permiten a los usuarios la utilización de este enredo de cables y circuitos, que de otra manera serian difíciles de controlar. Un sistema operativo se define como un conjunto de procedimientos manuales y automáticos, que permiten a un grupo de usuarios compartir una instalación de computadora eficazmente.

El sistema operativo debe poder tener un mecanismo para retomar un proceso y para sincronizarlos. El S.O. carga en memoria toda la información de cada proceso. El manejo de memoria debe mantener información de los espacios ocupados y por quien, decidir que procesos usara la memoria en la medida que esta se torne disponible y ocupar y liberar memoria. La memoria secundaria o virtual es la capacidad que tiene el sistema operativo para manejar espacios de disco como si fuera una extensión de la memoria; esto sirve para la administración del espacio libre y la ocupación de espacio. La estructura de un sistema de archivos es en forma de árbol. El sistema de archivos son los archivos y directorios. Las funciones son crear y eliminar archivos, compartir archivos para intercambiar información, agrupar archivos en forma conveniente al usuario, respaldo y recuperación, el acceso de los usuarios a la información sin la necesidad de conocer la ubicación física Interprete de comandos: Es una forma amigable de trabajar sin necesidad de usar programas de bajo nivel para comunicarse con el S.O.

#### 1.2 Evolución histórica

Los Sistemas Operativos, al igual que el Hardware de los computadores, han sufrido una serie de cambios revolucionarios llamados generaciones. En el caso del Hardware, las generaciones han sido marcadas por grandes avances en los componentes utilizados, pasando de válvulas ( primera generación ) a transistores ( segunda generación ), a circuitos integrados ( tercera generación), a circuitos integrados de gran y muy gran escala (cuarta generación). Cada generación Sucesiva de

hardware ha ido acompañada de reducciones substanciales en los costos, tamaño, emisión de calor y consumo de energía, y por incrementos notables en velocidad y capacidad.

#### Generación Cero (década de 1940)

Los primeros sistemas computacionales no poseían sistemas operativos. Los usuarios tenían completo acceso al lenguaje de la máquina. Todas las instrucciones eran codificadas a mano.

#### Primera Generación (década de 1950)

Los sistemas operativos de los años cincuenta fueron diseñados para hacer más fluida la transición entre trabajos. Antes de que los sistemas fueran diseñados, se perdía un tiempo considerable entre la terminación de un trabajo y el inicio del siguiente. Este fue el comienzo de los sistemas de procesamiento por lotes, donde los trabajos se reunían por grupos o lotes. Cuando el trabajo estaba en ejecución, este tenía control total de la máquina. Al terminar cada trabajo, el control era devuelto al sistema operativo, el cual limpiaba y leía e iniciaba el trabajo siguiente.

Al inicio de los 50's esto había mejorado un poco con la introducción de tarjetas perforadas (las cuales servían para introducir los programas de lenguajes de máquina), puesto que ya no había necesidad de utilizar los tableros enchufables.

Además el laboratorio de investigación General Motors implementó el primer sistema operativo para la IBM 701. Los sistemas de los 50's generalmente ejecutaban una sola tarea, y la transición entre tareas se suavizaba para lograr la máxima utilización del sistema. Esto se conoce como sistemas de procesamiento por lotes de un sólo flujo, ya que los programas y los datos eran sometidos en grupos o lotes.

La introducción del transistor a mediados de los 50's cambió la imagen radicalmente.

Se crearon máquinas suficientemente confiables las cuales se instalaban en lugares especialmente acondicionados, aunque sólo las grandes universidades y las grandes corporaciones o bien las oficinas del gobierno se podían dar el lujo de tenerlas.

Para poder correr un trabajo (programa), tenían que escribirlo en papel (en Fortran o en lenguaje ensamblador) y después se perforaría en tarjetas. Enseguida se llevaría la pila de tarjetas al cuarto de

introducción al sistema y la entregaría a uno de los operadores. Cuando la computadora terminara el trabajo, un operador se dirigiría a la impresora y desprendería la salida y la llevaría al cuarto de salida, para que la recogiera el programador.

### Segunda Generación (a mitad de la década de 1960)

La característica de los sistemas operativos fue el desarrollo de los sistemas compartidos con multiprogramación, y los principios del multiprocesamiento. En los sistemas de multiprogramación, varios programas de usuario se encuentran al mismo tiempo en el almacenamiento principal, y el procesador se cambia rápidamente de un trabajo a otro. En los sistemas de multiprocesamiento se utilizan varios procesadores en un solo sistema computacional, con la finalidad de incrementar el poder de procesamiento de la máquina.

La independencia de dispositivos aparece después. Un usuario que desea escribir datos en una cinta en sistemas de la primera generación tenía que hacer referencia específica a una unidad de cinta particular. En la segunda generación, el programa del usuario especificaba tan solo que un archivo iba a ser escrito en una unidad de cinta con cierto número de pistas y cierta densidad.

Se desarrollaron sistemas compartidos, en la que los usuarios podían acoplarse directamente con el computador a través de terminales. Surgieron sistemas de tiempo real, en que los computadores fueron utilizados en el control de procesos industriales. Los sistemas de tiempo real se caracterizan por proveer una respuesta inmediata.

### Tercera Generación (mitad de década 1960 a mitad década de 1970)

Se inicia en 1964, con la introducción de la familia de computadores Sistema/360 de IBM. Los computadores de esta generación fueron diseñados como sistemas para usos generales. Casi siempre eran sistemas grandes, voluminosos, con el propósito de serlo todo para toda la gente. Eran sistemas de modos múltiples, algunos de ellos soportaban simultáneamente procesos por lotes, tiempo compartido, procesamiento de tiempo real y multiprocesamiento. Eran grandes y costosos, nunca antes se había construido algo similar, y muchos de los esfuerzos de desarrollo terminaron muy por arriba del presupuesto y mucho después de lo que el planificador marcaba como fecha de terminación.

Estos sistemas introdujeron mayor complejidad a los ambientes computacionales; una complejidad a la cual, en un principio, no estaban acostumbrados los usuarios.

Cuarta Generación (mitad de década de 1970 en adelante)

Los sistemas de la cuarta generación constituyen el estado actual de la tecnología. Muchos diseñadores y usuarios se sienten aun incómodos, después de sus experiencias con los sistemas operativos de la tercera generación.

Con la ampliación del uso de redes de computadores y del procesamiento en línea los usuarios obtienen acceso a computadores alejados geográficamente a través de varios tipos de terminales.

Los sistemas de seguridad se ha incrementado mucho ahora que la información pasa a través de varios tipos vulnerables de líneas de comunicación. La clave de cifrado esta recibiendo mucha atención; han sido necesario codificar los datos personales o de gran intimidad para que; aun si los datos son expuestos, no sean de utilidad a nadie mas que a los receptores adecuados.

El porcentaje de la población que tiene acceso a un computador en la década de los ochenta es mucho mayor que nunca y aumenta rápidamente.

El concepto de maquinas virtuales es utilizado. El usuario ya no se encuentra interesado en los detalles físicos de; sistema de computación que esta siendo accedida. En su lugar, el usuario ve un panorama llamado maquina virtual creado por el sistema operativo.

Los sistemas de bases de datos han adquirido gran importancia. Nuestro mundo es una sociedad orientada hacia la información, y el trabajo de las bases de datos es hacer que esta información sea conveniente accesible de una manera controlada para aquellos que tienen derechos de acceso.

### **Sistemas monousuario**

CP/M (Control Program for Microcomputers), desarrollado por Gary Kildall fue el primer sistema operativo que podía ejecutarse en PCs de diferentes fabricantes. Cuenta una anécdota que ejecutivos de IBM fueron a visitar a Kildall para ofrecerle un acuerdo para poner el CP/M en la IBM PC, pero Kildall al parecer estaba ocupado en una sesión de vuelo, y no los pudo atender. Resultado: IBM llegó a un



acuerdo con un joven llamado Bill Gates para que desarrollara un sistema operativo, que se dio en llamar MS-DOS.

Para muchos de los que hoy nos encontramos trabajando con una computadora el sistema operativo D.O.S (Disk Operating System) fue el primero que nos tocó utilizar. De la misma manera, fue también el comienzo para el "gran hombre", para la persona que muchos admiran y otros no tanto: Bill Gates y su ahora gigante Microsoft.

Fue desarrollado desde sus principios pero "de a ratos" junto con la empresa IBM, pero diferencias entre las partes hicieron que no fuera un lanzamiento en conjunto. Por el contrario, cada una de las empresas presentó "su" sistema operativo: PC-DOS (IBM) y MS-DOS (Microsoft).

D.O.S. todavía hoy utiliza antiguos comandos CP/M tales como DIR, REN y TYPE que aun hoy funcionan bajo la máquina virtual D.O.S. de Windows.

Un tercer competidor para estos sistemas operativos (y de muy buena calidad) fue el DR-D.O.S. de la empresa Digital Research que incluía comandos más prolijos y de mayor funcionalidad, pero que con el tiempo, y gracias a las campañas publicitarias de Microsoft fue quedando relegado a un segundo lugar compartido con el PC-DOS de IBM. Luego de unos años DR-D.O.S. fue adquirido por Novell que presentó una nueva versión conocida como Novell D.O.S. que realmente casi no tuvo cabida en el mercado, aunque era muy buena.

Volviendo a D.O.S., este era simplemente una pantalla de texto con una línea de comandos que nos indicaba en qué directorio nos encontrábamos "y nada más". Uno tenía que "saber" que "cosas" había que escribir para que la máquina "hiciera algo". No había "menús contextuales" ni pantallas gráficas que nos guiaran. Era lo menos intuitivo que se puedan imaginar. Pero funcionó.

Las versiones que lo hicieron famoso en el mundo entero fueron la 3.0 y la 3.3, mientras que la más utilizada fue la 5.0, que introdujo muchos cambios a sus antecesoras. La versión 4.0 de D.O.S. estuvo plagada de errores, por lo cual casi no se uso (los usuarios se mantuvieron con la versión 3.30).

La última versión del producto como tal fue la 6.22, ya que luego apareció Windows 95 que "no necesitaba de DOS", pero que incluía la versión 7.0.

### 1.3 Conceptos básicos

#### Resumen

El sistema operativo debe poder tener un mecanismo para retomar un **proceso** y para sincronizarlo, El S.O. carga en memoria toda la información de cada proceso. **El manejo de memoria** debe mantener información de los espacios ocupados y por quien, decidir que procesos usara la memoria en la medida que esta se torne disponible y ocupar y liberar memoria. **La memoria secundaria o virtual** es la capacidad que tiene el sistema operativo para manejar espacios de disco como si fuera una extensión de la memoria; esto sirve para la administración del espacio libre y la ocupación de espacio. La estructura de un sistema de archivos es en forma de árbol. **El sistema de archivos son los archivos y directorios**. Las funciones son crear y eliminar archivos, compartir archivos para intercambiar información, agrupar archivos en forma conveniente al usuario, respaldo y recuperación, el acceso de los usuarios a la información sin la necesidad de conocer la ubicación física Interprete de comandos: Es una forma amigable de trabajar sin necesidad de usar programas de bajo nivel para comunicarse con el S.O.

Desglose:

#### Llamadas al Sistema:

La interfaz entre el sistema operativo y los programas del usuario se define por medio del conjunto de “instrucciones extendidas” que el sistema operativo proporciona. Estas instrucciones extendidas se conocen como llamadas al sistema. Las llamadas al sistema varían de un sistema operativo a otro (aunque los conceptos fundamentales tienden a ser análogos).

Las llamadas al sistema se clasifican normalmente en dos categorías generales: aquellas que se relacionan con procesos y la que lo hacen con el sistema de archivo.

- **Por Procesos:** Un proceso es básicamente un programa en ejecución. Consta del programa ejecutable y la pila o stack del programa, su contador de programa, apuntador de pila y otros registros, así como la otra información que se necesita para ejecutar el programa. En si el proceso en el concepto de los sistemas operativos es como el sistema de tiempo compartido. Esto es, que en forma periódica, el sistema operativo decide suspender la ejecución de un proceso y dar inicio a la ejecución de otro, por ejemplo, porque el primero haya tomado ya más de su parte del tiempo de la CPU, en terrenos del segundo.

Cuando un proceso se suspende temporalmente, debe reiniciarse después exactamente en el mismo estado en que se encontraba cuando se detuvo. Esto significa que toda la información relativa al proceso debe guardarse en forma explícita en algún lugar durante la suspensión. En muchos sistemas operativos, toda la información referente a cada proceso, diferente del contenido de su espacio de direcciones, se almacena en una tabla de sistema operativo llamada tabla de procesos, la cual es un arreglo (lista enlazada) de estructuras, una para cada proceso en existencia. Por lo tanto, un proceso (suspendido) consta de su espacio de direcciones, generalmente denominado imagen del núcleo (en honor de las memorias de imagen de núcleo magnético que se utilizaron en tiempos antiguos) y su registro de la tabla de procesos, que contiene sus registros entre otras cosas.

- **Por Sistema de Archivo:** Una función importante del S.O. consiste en ocultar las peculiaridades de los discos y otros dispositivos de E/S y presentar al programador un modelo abstracto, limpio y agradable de archivos independientes del dispositivo. Las llamadas al sistema se necesitan con claridad para crear archivos, eliminarlos, leerlos y escribirlos. Antes de que se pueda leer un archivo, éste debe abrirse y después de que se haya leído debe cerrarse, de modo que las llamadas se dan para hacer estas cosas.

Antes de que un archivo pueda leerse o escribirse, éste debe abrirse, en cuyo instante se verifican los permisos. Si se permite el acceso, el sistema produce un entero pequeño llamado descriptor del archivo para utilizarse en operaciones subsiguientes. Si se prohíbe el acceso, se produce un código de error.

- **Shell (intérprete de comandos):** El sistema operativo es el código que realiza las llamadas al sistema. Los editores, compiladores, ensambladores, enlazadores e intérpretes de comandos definitivamente no son parte del sistema operativo, aunque son importantes y útiles. El Shell es el intérprete de comandos, a pesar de no ser parte del sistema operativo, hace un uso intenso de muchas características del sistema operativo y por tanto sirve como un buen ejemplo de la forma en que se pueden utilizar las llamadas al sistema. También es la interfaz primaria entre un usuario situado frente a su terminal y el sistema operativo.

Cuando algún usuario entra al sistema, un “shell” se inicia. El shell tiene la terminal como entrada y como salida estándar. Este da inicio al teclear solicitud de entrada, carácter como un signo de pesos, el cual indica al usuario que el shell está esperando un comando. En MS-DOS

normalmente aparece la letra de la unidad, seguida por dos puntos (:), el nombre del directorio en que se encuentra y por último el signo de “mayor que” (>). Esto es: C:\>

#### 1.4 Funciones de un sistema operativo

- Interpreta los comandos que permiten al usuario comunicarse con el ordenador.
- Coordina y manipula el hardware de la computadora, como la memoria, las impresoras, las unidades de disco, el teclado o el mouse.
- Organiza los archivos en diversos dispositivos de almacenamiento, como discos flexibles, discos duros, discos compactos o cintas magnéticas.
- Gestiona los errores de hardware y la pérdida de datos.
- Servir de base para la creación del software logrando que equipos de marcas distintas funcionen de manera análoga, salvando las diferencias existentes entre ambos

#### 1.5 Tipos de sistemas operativos

- **Sistema Operativo Multitareas:** Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitareas. La conmutación de contextos (context Switching) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que solo se esta procesando la aplicación que se encuentra en primer plano (la que ve el usuario. En la multitarea cooperativa, la que se utiliza en el sistema operativo Macintosh, las tareas en segundo plano reciben tiempo de procesamiento durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación esta esperando información del usuario), y siempre que esta aplicación lo permita. En los sistemas multitarea de tiempo compartido, como OS/2, cada tarea recibe la atención del microprocesador durante una fracción de segundo.
- **Sistema Operativo Monotareas:** Los sistemas operativos monotareas son más primitivos y, solo pueden manejar un proceso en cada momento o que solo puede ejecutar las tareas de una en una.
- **Sistema Operativo Monousuario:** Los sistemas monousuarios son aquellos que nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se este ejecutando.
- **Sistema Operativo Multiusuario:** En esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este

tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

- **Secuencia por Lotes:** La secuencia por lotes o procesamiento por lotes en microcomputadoras, es la ejecución de una lista de comandos del sistema operativo uno tras otro sin intervención del usuario. Procesamiento por lotes también puede referirse al proceso de almacenar transacciones durante un cierto lapso antes de su envío a un archivo maestro, por lo general una operación separada que se efectúa durante la noche. Los sistemas operativos por lotes (batch), en los que los programas eran tratados por grupos (lote) en vez de individualmente. La función de estos sistemas operativos consistía en cargar en memoria un programa de la cinta y ejecutarlo. Al final este, se realizaba el salto a una dirección de memoria desde donde reasumía el control del sistema operativo que cargaba el siguiente programa y lo ejecutaba. De esta manera el tiempo entre un trabajo y el otro disminuía considerablemente.
- **Tiempo Real:** Procesa las instrucciones recibidas al instante, y una vez que han sido procesadas muestra el resultado.
- **Tiempo Compartido:** Consiste en el uso de un sistema por más de una persona al mismo tiempo. El tiempo compartido ejecuta programas separados de forma concurrente, intercambiando porciones de tiempo asignadas a cada programa (usuario)

## 1.6 Estructura de un Sistema Operativo

Internamente los sistemas operativos estructuralmente se clasifican según como se hayan organizado internamente en su diseño, por esto la clasificación más común de los S.O. son:

- **Sistemas monolíticos:**

En estos sistemas operativos se escriben como un conjunto de procedimientos, cada uno de los cuales puede llamar a cualquiera de los otros siempre que lo necesite. Cuando se emplea esta técnica, cada procedimiento del sistema tiene una interfaz bien definida en términos de parámetros y resultados, y cada una tiene la libertad de llamar a cualquiera otra, si la última ofrece algún cálculo útil que la primera necesite.

Para construir el programa objeto real del sistema operativo cuando se usa este método, se compilan todos los procedimientos individuales a archivos que contienen los procedimientos y después se combinan todos en un solo archivo objeto con el enlazador.

En términos de ocultamiento de información, esencialmente no existe ninguno; todo procedimiento es visible para todos (al contrario de una estructura que contiene módulos o paquetes, en los cuales mucha información es local a un módulo y sólo pueden llamar puntos de registro designados oficialmente del exterior del módulo)

Esta organización sugiere una estructura básica del sistema operativo:

- 1.- Un programa central que invoque el procedimiento de servicio solicitado (Shell o Kernel)
- 2.- Un conjunto de procedimientos de servicios que realice las llamadas al sistema.
- 3.- Un conjunto de procedimientos de uso general que ayude a los procedimientos de servicio

- **Sistemas con capas:**

Estos sistemas operativos se organizan como una jerarquía de capas, cada una construida arriba de la que está debajo. El primer sistema construido en esta forma fue el sistema THE que se fabricó en Technische Hogeschool Eindhoven de Holanda por E. W Dijkstra (1968) y sus alumnos. El sistema THE era un sistema de lote para una computadora alemana, la Electrológica X8, que tenía 32K de palabras de 27 bits ( los bits eran costosos en aquellos días)

El sistema tenía 6 estratos, estos se muestran en la siguiente tabla:

|   |  |
|---|--|
| 5 | Operador de THE                                    |
| 4 | Programas del usuario                              |
| 3 | Administración de Entrada/Salida                   |
| 2 | Comunicación entre el operador y el proceso        |
| 1 | Administración de la memoria y el tambor magnético |
| 0 | Distribución del procesador y multiprogramación    |

- La capa 0 trabajaba con la distribución del procesador, cambiando entre procesos cuando ocurrían interrupciones o los relojes expiraban. Sobre el estrato 0, el sistema constaba de procesos

secuenciales, cada uno de los cuales podía programarse sin tener que preocuparse por el hecho de que múltiples procesos estuvieran corriendo en un solo procesador. En otras palabras, el estrato 0 ofrecía la multiprogramación básica de la CPU.

La capa 1 realizaba el manejo de memoria. Esta distribuía espacio para procesos contenidos en la memoria central y en un tambor de 512K palabras que se usaba para contener partes de procesos (páginas) para las cuales no había espacio en la memoria central. Sobre el estrato 1, los procesos no tenían que preocuparse de si estaban en la memoria o en el tambor; el software del estrato 1 se hacía cargo de asegurar que las páginas se trajeran a la memoria siempre que se necesitaran.

La capa 2 manejaba la comunicación entre cada proceso y la consola de operador.

La capa 3 se hacía cargo de manejar los dispositivos de E/S y de separar la información en flujo que entraba y salía de ellos. Sobre la capa 3 cada proceso podía trabajar con dispositivos de E/S abstractos con propiedades agradables, en vez de dispositivos reales con muchas peculiaridades.

La capa 4 era donde se encontraban los programas de los usuarios. No tenían que preocuparse por el manejo de los procesos, memoria, consola o E/S. El proceso operador del sistema se localizaba en la capa 5.

### Maquinas virtuales

El concepto de maquina virtual surge a fines de la década de los años 70's cuando se desarrolla el sistema CP/CMS al cual después se le llamó VM/370 el cual contiene lo que se denominó el monitor de la maquina virtual, la cual se ejecuta en el hardware simple y realiza la multiprogramación proporcionando varias maquinas virtuales a la siguiente capa superior, las distintas maquinas virtuales son copias exactas del hardware real y pueden ejecutar distintos sistemas operativos.

Cuando un sistema CMS ( Conversational Monitor System) ejecuta una llamada al sistema, la llamada se captura para el sistema operativo en su propia maquina virtual, no para VM/370, tal y como sucedería si corriera en una maquina real en vez de una virtual, después CMS emite las instrucciones de E/S normales para leer su disco virtual, dichas instrucciones las ejecuta el sistema operativo como parte de su simulación del hardware real.

### Modelo cliente - servidor

Este modelo aplica en esencia para los modernos sistemas operativos y su funcionamiento es a partir de desplazar código a los niveles o capas superiores dejando un kernel mínimo. El método general

consiste en implementar la mayoría de las funciones del sistema operativo en procesos del usuario. Para solicitar un servicio, como la lectura de un bloque de un archivo, un proceso del usuario (al cual se le identificara como proceso del cliente), envía la solicitud a un proceso servidor, que realiza el trabajo y devuelve la respuesta.

Todo lo que hace el kernel es manejar la comunicación entre clientes y servidores. Al dividir el sistema en partes, cada parte se vuelve pequeña y difícil de manejar y lo que termina de complicar el asunto es que el sistema operativo corre en como procesos en modo de usuario y no en modo de kernel, por lo que existen servicios de archivo, servicios de proceso, servicios de la terminal o servicios de manejo de la memoria.

Una ventaja de este modelo es su adaptabilidad para adaptarlos a sistemas distribuidos.



## 2. Estructura de un sistema computacional

### Introducción:

La primera PC desarrollada en torno a los microprocesadores de Intel 80X86 fue la IBM PC. Nacida en 1981, IBM adoptó para la PC el microprocesador 8088 como unidad central de proceso. La característica más destacable de estos equipos fue su concepción "abierta" que permitía ampliar o agregar nuevas prestaciones con gran facilidad. Esta característica se conserva aún hoy en los nuevos modelos, generándose una norma no escrita, conocida como "compatibilidad IBM".

A la IBM PC sucedió en 1983 la XT, que conserva el microprocesador 8088, pero con el agregado de algunas mejoras, como la aparición del disco rígido.

El fenómeno de la "compatibilidad" no solo se hizo visible en la aparición de una gran cantidad de adaptadores y una variedad de software de aplicación, sino también en la aparición de réplicas semejantes a los modelos de IBM denominadas "compatibles" o "clones".

### 2.1 Arquitectura básica

Si observamos el diagrama en bloque de la PC XT, identificaremos varios de los componentes vistos a lo largo del curso. Podemos ver el microprocesador 8088, el coprocesador matemático 8087, el controlador de interrupciones 8259, el de DMA 8237, etc.

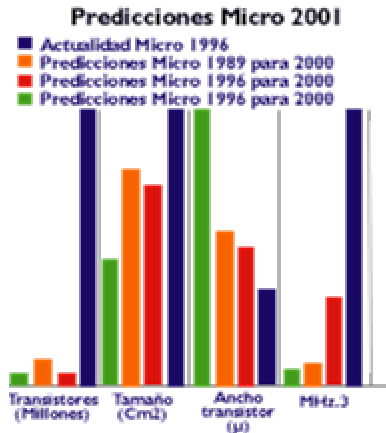
También se pueden observar los conectores de expansión del BUS, que le dan a la PC la característica de arquitectura "abierta", ya que permite el diseño de cualquier tipo de módulo de entrada salida, conectable al mencionado BUS.

A continuación describiremos cómo se interconectan los mencionados componentes y las funciones principales de cada uno de ellos.

### **Microprocesadores.**

Han pasado más de 25 años desde que Intel diseñara el primer microprocesador, siendo la compañía pionera en el campo de la fabricación de estos productos, y que actualmente cuenta con más del 90 por ciento del mercado. Un tiempo en el que todo ha cambiado enormemente, y en el que desde aquel 4004 hasta el actual Pentium II hemos visto pasar varias generaciones de máquinas que nos han entretenido y nos han ayudado en el trabajo diario.

Dicen que es natural en el ser humano querer mirar constantemente hacia el futuro, buscando información de hacia dónde vamos, en lugar de en dónde hemos estado. Por ello, no podemos menos que asombrarnos de las previsiones que los científicos barajan para dentro de unos quince años. Según el Dr.

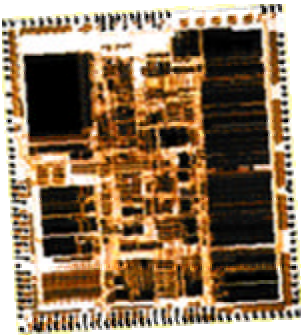


Albert Yu, vicepresidente de Intel y responsable del desarrollo de los procesadores desde el año 1984, para el año 2011 utilizaremos procesadores cuyo reloj irá a una velocidad de 10 GHz (10.000 MHz), contendrán mil millones de transistores y será capaz de procesar cerca de 100 mil millones de instrucciones por segundo. Un futuro prometedor, que permitirá realizar tareas nunca antes pensadas.

### Los inicios

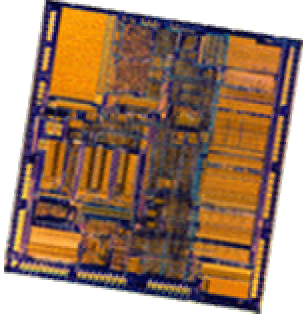
Sin embargo, para que esto llegue, la historia de los procesadores ha pasado por diferentes situaciones, siguiendo la lógica evolución de este mundo. Desde aquel primer procesador 4004 del año 1971 hasta el actual Pentium II del presente año ha llovido mucho en el campo de los procesadores. Tanto, que no estamos seguros si las cifras que se barajan en Intel se pueden, incluso, quedar cortas. Aquel primer procesador 4004, presentado en el mercado el día 15 de noviembre de 1971, poseía unas características únicas para su tiempo. Para empezar, la velocidad de reloj sobrepasaba por poco los 100 KHz (sí, habéis leído bien, kilohertzios), disponía de un ancho de bus de 4 bits y podía manejar un máximo de 640 bytes de memoria. Realmente una auténtica joya que para entonces podía realizar gran cantidad de tareas, pero que por desgracia no tiene punto de comparación con los actuales micros. Entre sus aplicaciones, podemos destacar su presencia en la calculadora Busicom, así como dotar de los primeros tintes de inteligencia a objetos inanimados.

Poco tiempo después, sin embargo, el 1 de abril de 1972, Intel anunciaba una versión mejorada de su procesador. Se trataba del 8008, que contaba como principal novedad con un bus de 8 bits, y la memoria direccionable se ampliaba a los 16 Kb. Además, llegaba a la cifra de los 3500 transistores, casi el doble que su predecesor, y se le puede considerar como el antecedente del procesador que serviría de corazón al primer ordenador personal. Justo dos años después, Intel anunciaba ese tan esperado primer ordenador personal, de nombre Altair, cuyo nombre proviene de un destino de la nave Enterprise en uno de los capítulos de la popular serie de televisión Star Trek la semana en la que se creó el ordenador. Este ordenador tenía un coste de entorno a los 400 dólares de la época, y el procesador suponía multiplicar por 10 el rendimiento del anterior, gracias a sus 2 MHz de velocidad (por primera vez se utiliza esta medida), con una memoria de 64 Kb. En unos meses, logró vender decenas de miles de unidades, en lo que suponía la aparición del primer ordenador que la gente podía comprar, y no ya simplemente utilizar.



### La introducción de IBM

Sin embargo, como todos sabemos, el ordenador personal no pasó a ser tal hasta la aparición de IBM, el gigante azul, en el mercado. Algo que sucedió en dos ocasiones en los meses de junio de 1978 y de 1979. Fechas en las que respectivamente, hacían su aparición los microprocesadores 8086 y 8088, que pasaron a formar el denominado IBM PC, que vendió millones de unidades de ordenadores de sobremesa a lo largo y ancho del mundo. El éxito fue tal, que Intel fue nombrada por la revista "Fortune" como uno de los mejores negocios de los años setenta. De los dos procesadores, el más potente era el 8086, con un bus de 16 bits (por fin), velocidades de reloj de 5, 8 y 10 MHz, 29000 transistores usando la tecnología de 3 micras y hasta un máximo de 1 Mega de memoria direccionable. El rendimiento se había vuelto a multiplicar por 10 con respecto a su antecesor, lo que suponía un auténtico avance en lo que al mundo de



la informática se refiere. En cuanto al procesador 8088, era exactamente igual a éste, salvo la diferencia de que poseía un bus de 8 bits en lugar de uno de 16, siendo más barato y obteniendo mejor respaldo en el mercado.

En el año 1982, concretamente el 1 de febrero, Intel daba un nuevo vuelco a la industria con la aparición de los primeros 80286. Como principal novedad, cabe destacar el hecho de que por fin se podía utilizar la denominada memoria virtual, que en el caso del 286 podía llegar hasta 1 Giga. También hay que contar con el hecho de que el tiempo pasado había permitido a los ingenieros de Intel investigar más a fondo en este campo, movidos sin duda por el gran éxito de ventas de los anteriores micros. Ello se tradujo en un bus de 16 bits, 134000 transistores usando una tecnología de 1.5 micras, un máximo de memoria direccionable de 16 Megas y unas velocidades de reloj de 8, 10 y 12 MHz. En términos de rendimiento, podíamos decir que se había multiplicado entre tres y seis veces la capacidad del 8086, y suponía el primer ordenador que no fabricaba IBM en exclusiva, sino que otras muchas compañías, alentadas por los éxitos del pasado, se decidieron a crear sus propias máquinas. Como dato curioso, baste mencionar el hecho de que en torno a los seis años que se le concede de vida útil, hay una estimación que apunta a que se colocaron en torno a los 15 millones de ordenadores en todo el mundo.

### **Microsoft también juega**

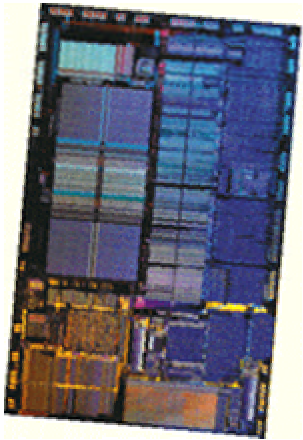
El año de 1985 es clave en la historia de los procesadores. El 17 de octubre Intel anunciaba la aparición del procesador 80386DX, el primero en poseer una arquitectura de 32 bits, lo que suponía una velocidad a la hora de procesar las instrucciones realmente importante con respecto a su antecesor. Dicho procesador contenía en su interior en torno a los 275000 transistores, más de 100 veces los que tenía el primer 4004 después de tan sólo 14 años. El reloj llegaba ya hasta un máximo de 33 MHz, y era capaz de direccionar 4 Gigas de memoria, tamaño que todavía no se ha superado por otro procesador de Intel dedicado al mercado doméstico. En 1988, Intel desarrollaba un poco tarde un sistema sencillo de actualizar los antiguos 286 gracias a la aparición del 80386SX, que sacrificaba el bus de datos para dejarlo en uno de 16 bits, pero a menor coste. Estos procesadores irrumpieron con la explosión del entorno gráfico Windows, desarrollado por Microsoft unos años antes, pero que no había tenido la suficiente aceptación por parte de los usuarios. También había habido algunos entornos que no habían funcionado mal del todo, como por ejemplo el Gem 3, pero no es hasta este momento cuando este tipo de entornos de trabajo se popularizan, facilitando la tarea de enfrentarse a un ordenador, que por aquel entonces sólo conocíamos unos pocos. Windows vino a ser un soplo de aire fresco para la industria, pues permitió que personas de cualquier condición pudiera manejar un ordenador con unos requerimientos mínimos de informática.

Y si esto parecía la revolución, no tuvimos que esperar mucho para que el 10 de abril de 1989 apareciera el Intel 80486DX, de nuevo con tecnología de 32 bits y como novedades principales, la incorporación del caché de nivel 1 (L1) en el propio chip, lo que aceleraba enormemente la transferencia de datos de este caché al procesador, así como la aparición del co-procesador matemático, también integrado en el procesador, dejando por tanto de ser una opción como lo era en los anteriores 80386. Dos cambios que unido al hecho de que por primera vez se sobrepasaban el millón de transistores usando la tecnología de una micra (aunque en la versión de este procesador que iba a 50 MHz se usó ya la tecnología .8 micras), hacía posible la aparición de programas de calidad sorprendente, entre los que los juegos ocupan un lugar destacado. Se había pasado de unos ordenadores en los que prácticamente cualquier tarea compleja requería del intérprete de comandos de MS-DOS para poder ser realizada, a otros en los que con mover el cursor y pinchar en la opción deseada simplificaba en buena medida las tareas más comunes. Por su



parte, Intel volvió a realizar, por última vez hasta el momento, una versión de este procesador dos años después. Se trataba del 80486SX, idéntico a su hermano mayor salvo que no disponía del famoso co-procesador matemático incorporado, lo que suponía una reducción del coste para aquellas personas que desearan introducirse en el segmento sin necesidad de pagar una suma elevada.

### **Llega el Pentium**



Sin embargo, Intel no se quedó contemplando la gran obra que había creado, y rápidamente anunció que en breve estaría en la calle una nueva gama de procesadores que multiplicaría de forma general por cinco los rendimientos medios de los 80486. Se trataba de los Pentium, conocidos por P5 en el mundillo de la informática mientras se estaban desarrollando, y de los que la prensa de medio mundo auguraba un gran futuro, tal y como así ha sido. Estos procesadores pasarán a la historia por ser los primeros a los que Intel no los bautizó con un número, y sí con una palabra. Esto era debido a que otras compañías dedicadas a la producción de procesadores estaban utilizando los mismos nombres puesto que no se podía registrar una cadena de ellos como marca, y por lo tanto, eran de dominio público. De modo que a Intel no le quedó más remedio que ponerle una palabra a su familia de procesadores, que además, con el paso del tiempo, se popularizó en los Estados Unidos de tal

forma, que era identificada con velocidad y potencia en numerosos cómics y programas de televisión. Estos procesadores que partían de una velocidad inicial de 60 MHz, han llegado hasta los 200 MHz, algo que nadie había sido capaz de augurar unos años antes.

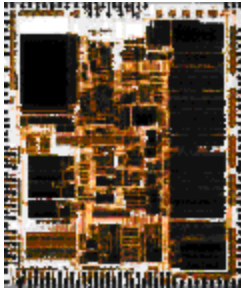
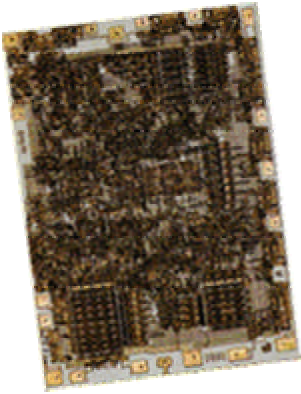
Con una arquitectura real de 32 bits, se usaba de nuevo la tecnología de .8 micras, con lo que se lograba realizar más unidades en menos espacio (ver recuadro explicativo). Los resultados no se hicieron esperar, y las compañías empezaron aunque de forma tímida a lanzar programas y juegos exclusivamente para el Pentium, hasta el punto que en este momento quien no posea un procesador de este tipo, está seriamente atrasado y no puede trabajar con garantías con los programas que actualmente hay en el mercado. Algo que ha venido a demostrar la aparición del nuevo sistema operativo de Microsoft Windows 95, que aunque funciona en equipos dotados de un procesador 486, lo hace sin sacar el máximo partido de sus funciones.

### **Pentium Pro y Pentium II**

La aparición, el 27 de marzo de 1995, del procesador Pentium Pro supuso para los servidores de red y las estaciones de trabajo un aire nuevo, tal y como ocurriera con el Pentium en el ámbito doméstico. La potencia de este nuevo procesador no tenía comparación hasta entonces, gracias a la arquitectura de 64 bits y el empleo de una tecnología revolucionaria como es la de .32 micras, lo que permitía la inclusión de cinco millones y medio de transistores en su interior. El procesador contaba con un segundo chip en el mismo encapsulado, que se encargaba de mejorar la velocidad de la memoria caché, lo que resultaba en un incremento del rendimiento sustancioso. Las frecuencias de reloj se mantenían como límite por arriba en 200 MHz, partiendo de un mínimo de 150 MHz. Un procesador que en principio no tiene muchos visos de saltar al mercado doméstico, puesto que los procesadores Pentium MMX parecen cubrir de momento todas las necesidades en este campo. No podemos asegurar que en un futuro cercano esto no acabe

ocurriendo, pues en el mundo de la informática han sucedido las cosas más extrañas, y nunca se sabe por dónde puede tirar un mercado en constante evolución.

Una evolución que  
denominado  
suma las



demostró Intel hace muy poco con un nuevo procesador, Pentium II, que viene a ser simplemente un nuevo ingenio que tecnologías del Pentium Pro con el MMX. Como resultado, el Pentium II es el procesador más rápido de cuantos ha comercializado Intel. Por el momento únicamente se dispone de las versiones a 233 y 266 MHz, pero después de este verano podremos disfrutar de la versión de 300 MHz, que supondrá un nuevo récord de velocidad de reloj. El Pentium II, cuyas características fueron tratadas con detalle en el artículo de portada del pasado mes de la revista, es hoy (por poco tiempo) el extremo de la cadena evolutiva de Intel.

### **El futuro de los microprocesadores**

La evolución que están sufriendo los procesadores es algo que no parece escapar a la atención de millones de personas, cuyo trabajo depende de hasta dónde sean capaces de llegar los ingenieros de Intel a la hora de desarrollar nuevos chips. El último paso conocido ha sido la implementación de la nueva arquitectura de 0.25 micras, que viene a sustituir de forma rotunda la empleada hasta el momento, de 0.35 micras en los últimos modelos de procesador. Esto va a significar varias cosas en un futuro no muy lejano. Para empezar, la velocidad se incrementará una media del 33 por ciento con respecto a la generación de anterior. Es decir, el mismo procesador usando esta nueva tecnología puede ir un 33 por ciento más rápido que el anterior. Para que os podáis hacer una idea del tamaño de esta tecnología, deciros que el valor de 0.25 micras es unas 400 veces más pequeño que un pelo de cualquier persona. Y este tamaño es el que tienen los transistores que componen el procesador. El transistor, como muchos sabréis, permite el paso de la corriente eléctrica, de modo que en función de en qué transistores haya corriente, el ordenador realiza las cosas (esto es una simplificación de la realidad, pero se ajusta a ella más o menos). Dicha corriente eléctrica circula entre dos puntos, de modo que cuanto menor sea esta distancia, más cantidad de veces podrá pasar pues el tiempo de paso es menor. Aunque estamos hablando de millonésimas de segundo, tened en cuenta que un procesador está trabajando continuamente, de modo que ese tiempo que parece insignificante cuando es sumado a lo largo de las miles de millones de instrucciones que realizar, nos puede dar una cantidad de tiempo bastante importante. De modo que la tecnología que se utilice puede dar resultados totalmente distintos incluso utilizando el mismo procesador. Por el momento, en un futuro cercano además de contar con la arquitectura de 0.25 micras, podremos disfrutar de una de 0.07 para el año 2011, lo que supondrá la introducción en el procesador de mil millones de transistores y alcanzando una velocidad de reloj cercana a los 10000 MHz, es decir, 10 GHz.

### **La tecnología MMX**

Aunque no podamos considerar la tecnología MMX como un procesador en sí mismo, sería injusto no hablar de ella en un informe como éste. Es uno de los mayores pasos que ha dado Intel en la presente década, y según ellos mismos, todos los procesadores que fabriquen a partir de mediados del próximo año llevarán incorporada esta arquitectura. Para su desarrollo se analizaron un amplio rango de programas para determinar el funcionamiento de diferentes tareas: algoritmos de descompresión de vídeo, audio o

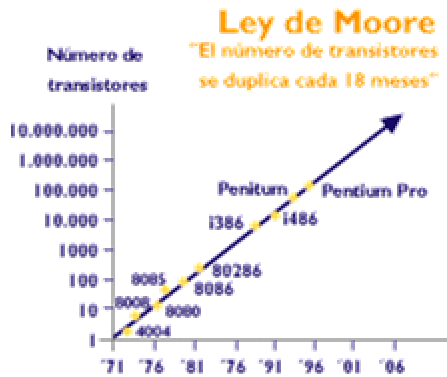


gráficos, formas de reconocimiento del habla o proceso de imágenes, etc. El análisis dio como resultado que numerosos algoritmos usaban ciclos repetitivos que ocupaban menos del 10% del código del programa, pero que en la práctica suponían el 90% del tiempo de ejecución. De modo que nació la tecnología MMX, compuesta por 57 instrucciones y 4 tipos de datos nuevos, que se encargan de realizar esos trabajos cíclicos consumiendo mucho menos tiempo de ejecución. Antes, para manipular 8 bytes de datos gráficos requería 8 repeticiones de la misma instrucción; ahora, con la nueva tecnología, se puede utilizar una única instrucción aplicada a los 8 bytes simultáneamente, obteniendo de este modo un incremento del rendimiento de 8x.

### Especificaciones técnicas de los microprocesadores Intel

|                 | Fecha de presentación | Velocidad de reloj           | Ancho de bus | Número de transistores | Memoria direccionable | Memoria virtual | Breve descripción                                    |
|-----------------|-----------------------|------------------------------|--------------|------------------------|-----------------------|-----------------|--|
| 4004            | 15/11/71              | 108 KHz.                     | 4 bits       | 2.300 (10 micras)      | 640 byte              |                 | Primer chip con manipulación aritmética              |
| 8008            | 1/4/72                | 108 KHz.                     | 8 bits       | 3.500                  | 16 KBytes             |                 | Manipulación Datos/texto                             |
| 8080            | 1/4/74                | 2 MHz.                       | 8 bits       | 6.000                  | 64 KBytes             |                 | 10 veces las (6 micras) prestaciones del 8008        |
| 8086            | 8/6/78                | 5 MHz.<br>8 MHz.<br>10 MHz.  | 16 bits      | 29.000 (3 micras)      | 1 MegaByte            |                 | 10 veces las prestaciones del 8080                   |
| 8088            | 1/6/79                | 5 MHz.<br>8 MHz.             | 8 bits       | 29.000                 |                       |                 | Idéntico al 8086 excepto en su bus externo de 8 bits |
| 80286           | 1/2/82                | 8 MHz.<br>10 MHz.<br>12 MHz. | 16 Bits      | 134.000 (1.5 micras)   | 16 Megabytes          | 1 Gigabyte      | De 3 a 6 veces las prestaciones del 8086             |
| Microprocesador | 17/10/85              | 16 MHz.                      | 32 Bits      | 275.000                | 4 Gigabytes           | 64 Terabytes    | Primer chip x86 capaz de manejar                     |

|                                  |         |  |         |  |             |                 |   |
|----------------------------------|---------|--|---------|--|-------------|-----------------|---|
| Intel 386 DX®                    |         | 20 MHz.<br>25 MHz.<br>33 MHz.  |         | (1 micra)                              |             |                 | juegos de<br>datos de 32<br>bits  |
| Microprocesador<br>Intel 386 SX® | 16/6/88 | 16 MHz.<br>20 MHz.   | 16 Bits | 275.000<br>(1 micra)                   | 4 gigabytes | 64<br>Terabytes | Bus capaz<br>de<br>direccionar<br>16 bits<br>procesando<br>32bits a bajo<br>coste             |
| Microprocesador<br>Intel 486 DX® | 10/4/89 | 25 MHz.<br>33 MHz.<br>50 MHz.  | 32 Bits | (1 micra, 0.8<br>micras en 50<br>MHz.) | 4 Gigabytes | 64<br>Terabytes | Caché de<br>nivel 1 en el<br>chip   |
| Microprocesador<br>Intel 486 SX® | 22/4/91 | 16 MHz.<br>20 MHz.<br>25 MHz.<br>33 MHz.   | 32 Bits | 1.185.000<br>(0.8 micras)              | 4 Gigabytes | 64<br>Terabytes | Idéntico en<br>diseño al<br>Intel 486DX,<br>pero sin<br>coprocesador<br>matemático            |
| Procesador<br>Pentium®           | 22/3/93 | 60 MHz.<br>66 MHz.<br>75 MHz.<br>90 MHz.<br>100 MHz.<br>120 MHz.<br>133 MHz.<br>150 MHz.<br>166 MHz.<br>200 MHz. | 32 Bits | 3,1 millones<br>(0.8 micras)           | 4 Gigabytes | 64<br>Terabytes | Arquitectura<br>escalable.<br>Hasta 5<br>veces las<br>prestaciones<br>del 486 DX a<br>33 MHz. |
| Procesador                       | 27/3/95 | 150 MHz.   | 64 Bits | 5,5 millones                           | 4 Gigabytes | 64              | Arquitectura<br>de ejecución  |



|                          |        |                                  |         |                               |             |                 |   |
|--------------------------|--------|----------------------------------|---------|-------------------------------|-------------|-----------------|---|
| PentiumPro®              |        | 180 MHz.<br>200 MHz.             |         | (0.32 micras)                 |             | Terabytes       | dinámica con procesador de altas prestaciones                 |
| Procesador<br>PentiumII® | 7/5/97 | 233 MHz.<br>266 MHz.<br>300 MHz. | 64 Bits | 7,5 millones<br>(0.32 micras) | 4 Gigabytes | 64<br>Terabytes | S.E.C., MMX,<br>Doble Bus<br>Indep.,<br>Ejecución<br>Dinámica |

## La ley de Moore

El Dr. Gordon Moore, uno de los fundadores de Intel Corporation, formuló en el año 1965 una ley que se ha venido a conocer como la "Ley de Moore". La citada ley que está reflejada en el gráfico adjunto, nos viene a decir que el número de transistores contenidos en un microprocesador se dobla más o menos cada dieciocho meses. Esta afirmación, que en principio estaba destinada a los dispositivos de memoria, pero también los microprocesadores han cumplido la ley. Una ley que significa para el usuario que cada dieciocho meses, de forma continua, pueda disfrutar de una tecnología mejor, algo que se ha venido cumpliendo durante los últimos 30 años, y de lo que se espera siga vigente en los próximos quince o veinte años. De modo que el usuario puede disponer de mejores equipos, aunque también significa la necesidad de cambiar de equipo cada poco tiempo, algo que no todo el mundo se puede permitir. Y eso que el precio aumenta de forma absoluta pero no relativa, puesto que la relación MIPS/dinero está decreciendo a velocidad vertiginosa. Algo que sin embargo no sucede con la industria del automóvil por ejemplo, ya que la potencia de los coches no se ha multiplicado de la misma forma que los precios. En cualquier caso, queda claro que en los próximos años nos espera una auténtica revolución en lo que a rendimiento de los procesadores se refiere, como ya predijera Moore hace más de treinta años.



## Coprocesador matemático 8087

Junto al microprocesador es común encontrar un zócalo de 40 pines para la instalación del circuito integrado 8087, que es la unidad de punto flotante (FPU) diseñada para el 8086/8088 (también llamada unidad de procesamiento numérico NPU). Este coprocesador agrega instrucciones numéricas y registros de punto flotante, lo que incrementa notablemente la velocidad de procesamiento, en especial cuando los programas requieren operaciones numéricas de alta precisión.

El agregado de ésta opción sólo requiere la instalación del coprocesador en el zócalo y la habilitación de un switch en la placa madre, que indica la existencia del coprocesador. Este switch permite que el microprocesador reciba la interrupción de excepción del coprocesador.

## Controlador de DMA 8237

También podemos observar el controlador 8237, que proporciona cuatro canales de DMA para dispositivos de entrada salida que requieran de este servicio. Estos canales se denominan con las siglas DMA0 a DMA3, y al igual que las interrupciones se encuentran preasignados para cumplir con las funciones básicas. El canal 0 se emplea para realizar el refresco de la memoria RAM dinámica, por lo que se conecta directamente al timer del sistema. Esto permite que se realice periódicamente cada intervalos de tiempo definidos para evitar que el contenido de la memoria se desvanezca por completo.

Los restantes canales se encuentran disponibles en el conector de expansión con sus señales DRQ y DACK para que puedan ser usados por los dispositivos de entrada salida que lo requieran. A continuación se proporciona una lista de la asignación de los canales en la PC.

| Canal de DMA | Aplicación          |
|--------------|---------------------|
| CANAL 0      | REFRESCO DE MEMORIA |
| CANAL 1      | SDLC                |
| CANAL 2      | DISCO FLEXIBLE      |
| CANAL 3      | NO ASIGNADO         |

El 8237 se encuentra ubicado en la posición 00h, pero requiere del agregado de un puerto para los 4 bits de direcciones superiores (A16-A19) que se encuentra en la posición 80h.

## Timer 8253

Este chip es un contador/timer que provee a la PC de 3 timers programables utilizados para todas las mediciones de tiempo. Algunas de las aplicaciones ya fueron mencionadas cuando se describieron los controladores 8259 y 8237, aunque no fueron completamente analizadas.

El 8253 provee tres canales independientes, cada uno de ellos programable de 6 modos diferentes. Las entradas de reloj pueden ser distintas para cada canal, aunque en la PC es única y de 1,19 MHz.

## Diagrama en bloques

En la PC los canales se encuentran asignados como se muestra en la figura siguiente.

Como se puede ver el canal 0 se encuentra siempre habilitado y se utiliza como contador del sistema para proveer una base de tiempo confiable. La salida del timer proporciona una interrupción, comúnmente llamada timer tic, que se conecta a la IRQ0. La misma es utilizada por el BIOS para incrementar un contador de 4 bytes en 0040:006C que se utiliza para el cómputo de la fecha y la hora. El canal 1 está también siempre habilitado y programado para una cuenta de 15uS, pero su salida se conecta al canal 0 de DMA. El final de cuenta de este timer dispara un ciclo de lectura de memoria ("dummy") que permite que no transcurra demasiado tiempo sin que ésta sea leída. Este ciclo conocido como refresco de memoria es imprescindible para que el contenido no se desvanezca. Por último el canal 2 se encuentra disponible, aunque como se pueda ver se encuentra conectado al altavoz. Si se quisiera usar con otros fines, se debe deshabilitar el parlante a través de la salida B1 del 8255, y se podrá habilitar el timer a través de B0. El final de cuenta puede detectarse por la misma 8255, a través de la entrada C5.

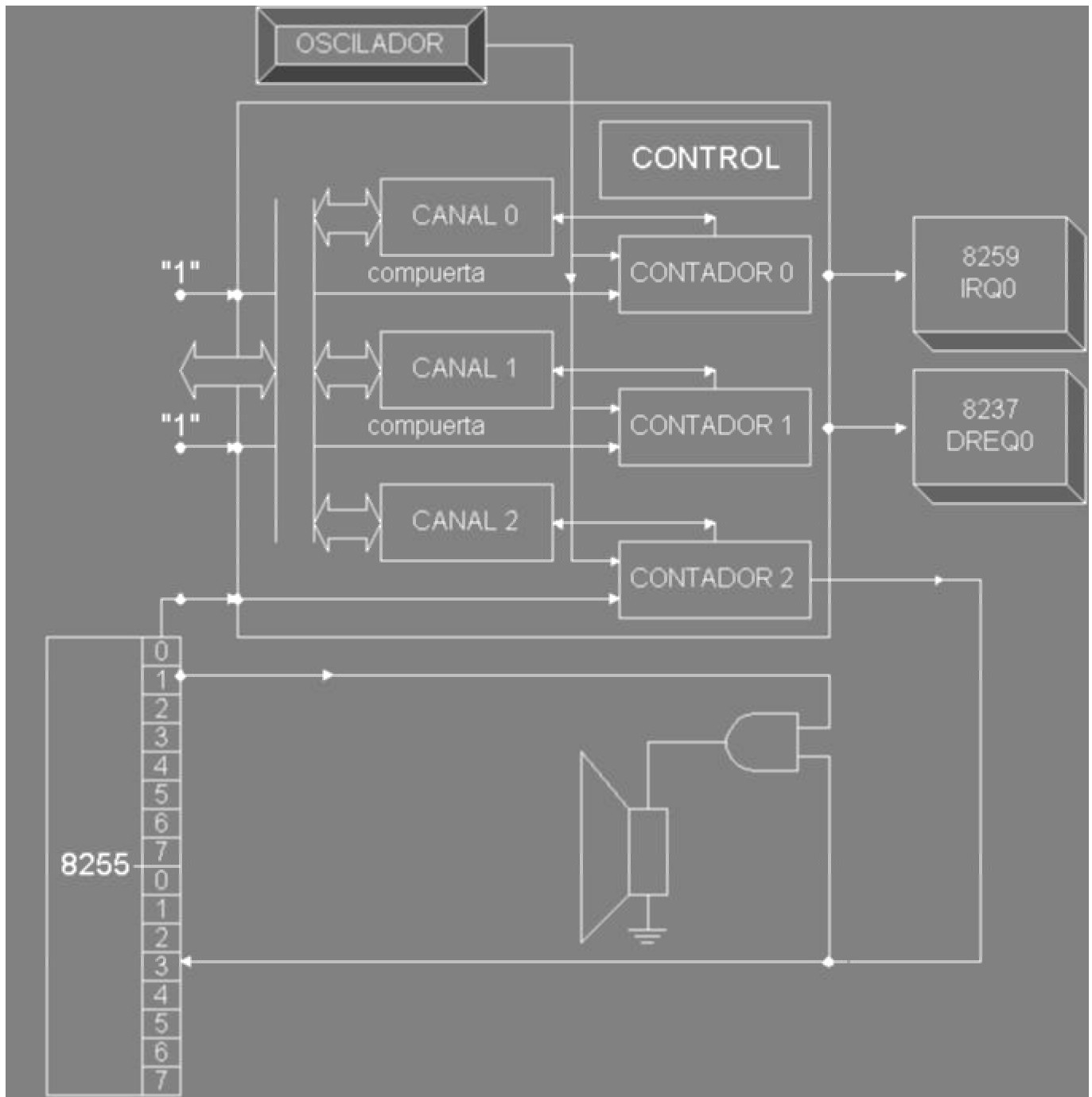


Diagrama en bloques

## Interfase 8255

La 8255 es una Peripheral Programmable Interface (PPI) que contiene tres puertos de E/S de 8 bits. En la PC IBM-compatible está configurado de modo de obtener dos puertos de entrada (60h y 62h) y un puerto de salida (61h).

El puerto A (60h) se usa principalmente para la lectura del teclado. También se lo puede utilizar para leer el estado de los switch de configuración de la placa principal (mother-board), aunque esto ocurre una sola vez durante la inicialización del sistema, en el proceso denominado POST (power on self test).

El puerto C (62h) utiliza los cuatro bits inferiores para la lectura de los switches que indican la capacidad de memoria instalada. El bit 4 estaba reservado para leer la entrada de cinta. El bit 5 se usa para el canal del temporizador y los bits 6 y 7 se usan para leer el código de error de la rutina NMI.

El puerto B (61h) se configura como salida y se utiliza para diversos fines. La lista siguiente muestra la utilización de los distintos bits:

|   |  |
|---|--|
| 0 | timer 2, habilitación - controlador del parlante |
| 1 | habilitación directa del parlante                |
| 2 | mux del puerto 62h                               |
| 3 | control motor cinta                              |
| 4 | habilitación chequeo de paridad                  |
| 5 | id   |
| 6 | control del teclado                              |
| 7 | id - mux del puerto 60h                          |

## Rom BIOS

Se mencionó que existe un hardware estándar que compone la base de la PC. Asimismo, con la primera PC, IBM proveyó también un programa estándar de control para muchos de los componentes de este hardware. Este programa reside en la ROM del sistema y se lo llama comúnmente ROM BIOS (Basic Input/Output System).

Esta interfaz permite que los diseñadores de software de aplicación puedan realizar programas que funcionen en sistemas de configuración variada, sin tener que realizar modificaciones para cada caso. De esta forma también los fabricantes pueden realizar modificaciones en el hardware, pero manteniendo esta interfaz tal cual es, no se necesitan modificar los programas. Igualmente versiones posteriores de BIOS incorporan nuevas funciones, sin perjuicio de las anteriores. Esto posibilita que los programas viejos puedan ejecutarse correctamente, aunque por supuesto no harán uso de estas nuevas funciones.

La ROM BIOS posee tres funciones específicas:

La primera de ellas se denomina POST (Power On Self Test). El POST consiste en una rutina que se ejecuta cuando se enciende la computadora o cada vez que se realice un Reset. La función principal de esta rutina es testear el hardware e inicializar los dispositivos programables, tales como el 8259 o el 8237.

Otra de las funciones es contener los drivers de los dispositivos de I/O. Tanto la versión de IBM como las posteriores proveen una gran variedad de rutinas para el hardware usado más comúnmente.

La tercer función del BIOS es contener un set de servicios del sistema, que si bien no controla ningún dispositivo, provee datos valiosos acerca de la operación del sistema.

## El bus del sistema.

### Bus ISA:

Es un bus de 62 contactos numerados desde A1 hasta A31 y B1 hasta B31. Contiene un bus de datos de 8 bits y uno de direcciones de 20 bits, seis señales de interrupción y tres canales de DMA. A la frecuencia de 8,33 MHz permitía una velocidad de transferencia de 8,33 Mb/s.

Este bus se transformó rápidamente en un cuello de botella, por lo cual con la PC-AT, basada en el procesador 80286, se expandió el mismo a 16 bits de datos, 24 de direcciones y se agregaron cinco nuevas interrupciones y cuatro canales de DMA. Para ello, se tuvo que adicionar un conector de 36 contactos. La velocidad de transferencia se duplicó para la misma frecuencia (8,33 MHz).

### Bus Local PCI

El bus de expansión estándar de la industria de 16-bits/8-MHz ISA, durante una década instalado en la mayoría de los ordenadores, no puede transportar adecuadamente las grandes cantidades de datos generados por las aplicaciones de hoy día. Procesadores de alto rendimiento, trabajando a velocidades de hasta 133 MHz, se ven forzados a esperar que los discos duros, tarjetas de vídeo y otros periféricos envíen y reciban datos a lo largo de un camino, que no es solamente estrecho y lento, sino también ineficiente. Como resultado, graves cuellos de botella se producen cuando grandes ficheros compiten por el limitado ancho de banda que ofrece el bus ISA. Se han introducido buses mejorados, como el bus EISA y Micro Channel\* para aliviar estos problemas, pero ninguno ha conseguido ofrecer una solución a largo plazo con un coste adecuado (ver gráfico "Requerimientos de ancho de banda para periféricos").

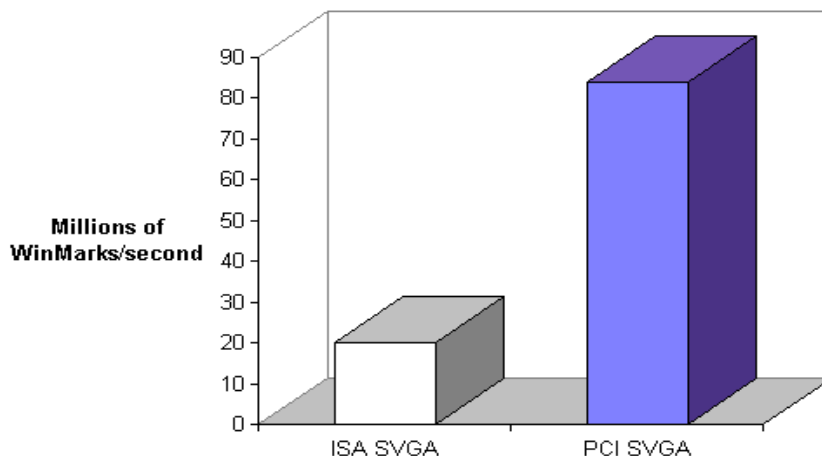


Figura 2 - performance de interfaces de video

Para resolver este problema, la industria del PC ha desarrollado una especificación de bus local basada en el bus del procesador Intel486(TM) y un conector de bus local. Conocido como VESA local bus, permite mejorar el cuello de botella de I/O de los sistemas de bus tradicionales, permitiendo periféricos de alto rendimiento y encontrando un atajo para el procesador del sistema. A pesar de que la especificación es una buena solución intermedia, y ha resuelto alguno de los problemas de I/O, tiene limitaciones. La más significativa es la incompatibilidad; dado que el diseño del bus VL está estrechamente relacionado a la velocidad del procesador, no todas las tarjetas VL bus son compatibles con cada sistema basado en el bus VL.

Para eliminar incompatibilidades y lograr una solución más robusta para los sistemas basados en procesadores Pentium, los laboratorios de Intel, junto con sus socios en la industria, han desarrollado la especificación para el bus local PCI (Peripheral Component Interconnect). PCI tiene un avanzado diseño de bus local de alto rendimiento que soporta múltiples periféricos. Está optimizado para aprovecharse de las ventajas la tecnología de los microprocesadores y ordenadores personales de hoy, y ofrecer una solución total al sistema. Suministra un incremento de rendimiento para adaptadores de Red, discos duros, vídeo, gráficos y la amplia gama de periféricos de alta velocidad de hoy día, al mismo tiempo que mantiene la compatibilidad con los ya existentes buses de expansión ISA/EISA/Micro Channel.

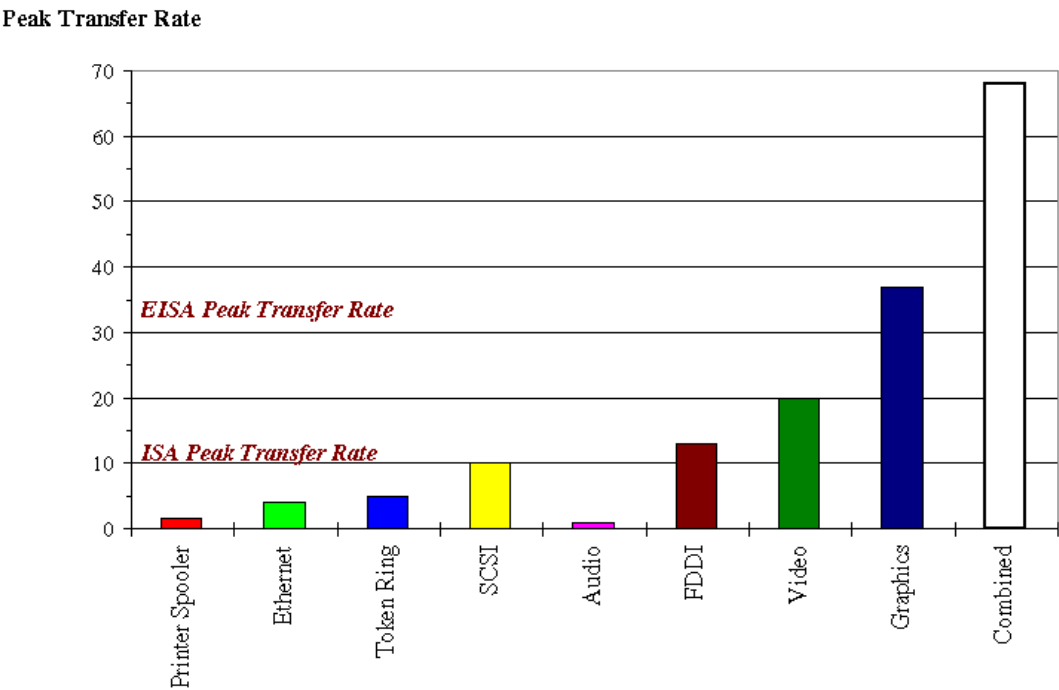


Figura 3 - performance de buses de PC

Corriendo a una velocidad de reloj de 33 MHz, el bus local PCI usa un bus de datos de 32-bits que soporta periféricos y tarjetas para un ancho de banda de hasta 132 Mbytes/segundo -- una substancial y considerable mejora con respecto a la transferencia de 5 Mbyte/segundo del bus ISA. Este incremento

sobre el ancho de banda, permite al bus PCI un rendimiento sobre gráficos de más de cuatro veces del bus ISA (ver gráfico "Pruebas WinBench").

El bus PCI, sin embargo, ofrece mucho más que un alto ancho de banda. Permite a los periféricos beneficiarse de la potencia de proceso disponible, sin ser dependientes de la arquitectura ni de la velocidad del procesador. Soporta también autoconfiguración Plug and Play, y tiene incorporada la posibilidad de actualización, a fin de adaptarse a los futuros avances técnicos.

Para tener una visión mas detallada sobre los buses de PC, se puede consultar el artículo

**Buses en la arquitectura IBM-PC compatible**

**El bus ISA**

El bus ISA (*Industry Standard Architecture*), fue el primer bus utilizado en los equipos PC y XT de IBM como sistema de arquitectura abierta. Esto permitió en gran parte la proliferación de clones y la gran variedad de tarjetas de expansión e interface de bajo costo que aún se encuentran en el mercado. De los buses ISA hay dos versiones: el de 8 bits y el de 16 bits.

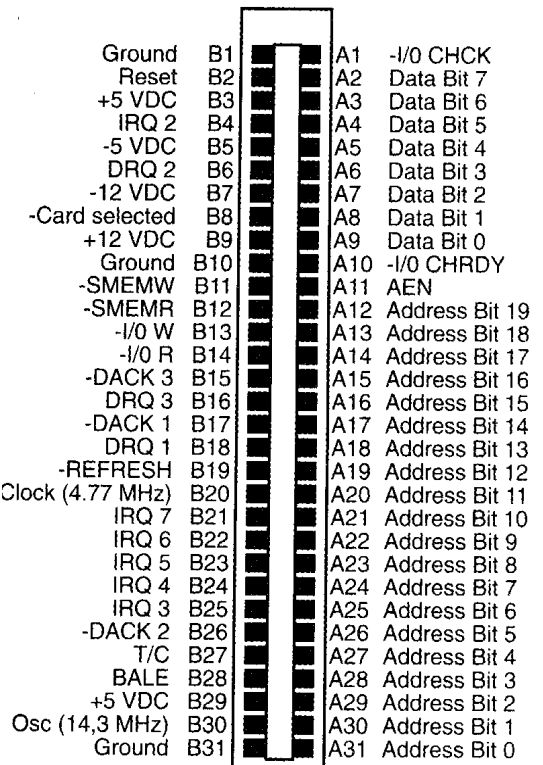
**El bus ISA de 8 bits**

El bus ISA de 8 bits salió al mercado en el año 1982 con el PC y luego se conservó en los XT's. Este bus se encuentra en forma de conector en línea de doble lado y tiene 62 contactos, 31 por cada lado. Sus pines o contactos están numerados por un lado como A1 hasta A31 y por el otro lado como B1 hasta B31. En la figura 1 se muestra la estructura física, la numeración y el nombre de cada una de las señales.

Como se puede observar en la figura, en una ranura o *slot* se encuentran el bus de datos, el bus de direcciones, las señales de control y de reloj y los voltajes de alimentación. Casi todas estas señales van conectadas al microprocesador a través de circuitos acopladores (*drivers o buffers*) con el fin de protegerlo de posibles cortocircuitos o conexiones equivocadas. Como podemos ver, en el bus ISA de 8 bits hay un bus de datos de 8 bits (de ahí su nombre) y un bus de direcciones de 20 líneas (lo que permite un direccionamiento hasta de 1 MByte).

El bus contiene seis señales de interrupción (IRQ2 a IRQ7), tres canales de DMA y una señal de reloj de 4.77 MHz. Aunque cada conector en el bus se supone que trabaja de la misma forma, los primeros PC's fabricados con ocho ranuras requerían tener por lo menos una tarjeta conectada en el *slot* 8 (el más cercano a la fuente de poder), con el fin de obtener una señal llamada "*card selected (tarjeta seleccionada)*" en el pin B8. Esta señal era una línea diseñada para recibir un adaptador especial de IBM llamado el 3270PC, pero la mayoría de los fabricantes de clones y compatibles no siguió esta configuración.

El pin del oscilador (B30), entrega una señal de 14.3 MHz. Cuando se debe hacer *reset* en el PC, el pin RESET DRV (B2) reinicia todo el sistema. Cuando se recibe una dirección



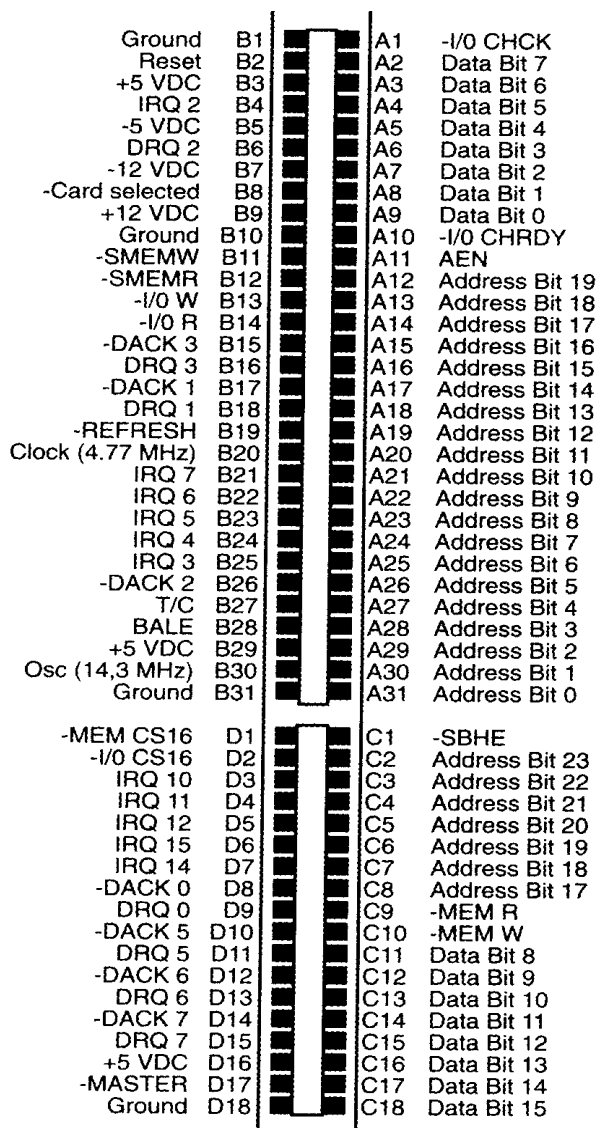
válida, el pin AEN (A11) le indica al sistema que se puede decodificar esta dirección. El pin -I/O CHCK o *I/O Channel Check* (A1), avisa a los circuitos de la tarjeta principal (*motherboard*) que ha ocurrido un error en la tarjeta de expansión; el signo negativo indica que es una señal activa baja. El pin -I/O CHRDY o *I/O Channel Ready* (A10) se activa cuando una tarjeta de expansión está lista. Si este pin está en nivel bajo (0), el microprocesador extiende el ciclo del bus generando estados de espera (*wait states*).

Las seis señales de interrupción por *hardware* (IRQ2 a IRQ7), se utilizan por las tarjetas de expansión para demandar atención por parte del microprocesador. Las interrupciones 0 y 1 no están disponibles en el bus ya que ellas tienen las prioridades más altas del temporizador principal y el teclado. Las señales *I/O Read* y *I/O Write* indican que el microprocesador o el controlador de DMA quieren transferir datos hacia o desde el bus de datos. Las señales de lectura y escritura de memoria (-MEMR y -MEMW) le indican a la tarjeta de expansión que la CPU o el DMA van a leer o escribir datos a la memoria principal. El bus XT tiene tres señales de requisición de DMA (DRQ1 a DRQ3), que le permiten a la tarjeta de expansión transferir datos hacia o desde la memoria.

## El bus ISA de 16 bits

Con el avance de la tecnología, las limitaciones del bus ISA de *8 bits* se evidenciaron rápidamente. Las seis interrupciones disponibles se coparon con la unidad de disco flexible, el disco duro, los puertos seriales y el puerto paralelo, dejando pocas posibilidades de expansión. Así mismo, de los tres canales de DMA, la unidad de disco y el disco duro ocupaban dos, quedando uno solo libre. La capacidad de direccionamiento de solo 1 MB y el bus de datos de *8 bits* también se constituyeron en un cuello de botella, que hacía los equipos muy lentos para las nuevas aplicaciones generadas en el desarrollo de nuevo *software*.

Con la aparición del microprocesador 80286 de 16 *bits* y los computadores AT en 1984, se diseñó en forma muy inteligente un nuevo bus, el ISA de 16 *bits*, que utilizaba en gran parte el bus anterior ISA de *8 bits*





agregándole un segundo conector de 36 pines alineado con el primero con nuevas señales, figura 2. De esta forma, se podían utilizar la gran cantidad de tarjetas periféricas y de expansión que existían en el mercado. En esencia se agregaron otros 8 *bits* de datos, más direcciones, cinco interrupciones y cuatro canales de DMA y algunas señales de control. Así mismo, se incrementó la velocidad a 8.33 MHz.

Vale la pena mencionar que el bus ISA de 16 *bits* o bus AT, fue ampliamente utilizado por los fabricantes de clones y compatibles, revolucionando desde el punto de vista de los precios, la industria de los PC's. Este bus todavía se encuentra en la mayoría de los computadores personales fabricados actualmente, ya que para él se fabrican una gran cantidad de tarjetas para periféricos y de expansión.

### **El bus MCA de 32 bits**

Con la introducción de los microprocesadores de 32 *bits* como el 80386 y el 80486, el bus ISA de 16 *bits* se quedó nuevamente atrás y se creó la necesidad de un nuevo bus. En 1987, IBM consideró que se debía archivar el bus tipo ISA y diseñó el bus MCA (*Micro Channel Architecture*) o Micro Canal en su línea de computadores PS/2.

Su diseño buscaba, además del objetivo de mejorar el rendimiento de los equipos, frenar el avance de la industria de los clones y los compatibles, ya que IBM no entregó sus características tratando de fabricar sus propias tarjetas de expansión y periféricos. Finalmente, este intento no tuvo éxito y se convirtió en un *boomerang* contra el gigante azul, que más tarde tuvo que retroceder y volver a los buses tradicionales, debido a que el bus MCA no llegó a ser muy popular entre los fabricantes de tarjetas de expansión y a que el público se negó a abandonar sus inversiones en *hardware* y *software* para cambiar por un nuevo sistema.

La principal diferencia entre los buses ISA y MCA consiste en la configuración física, ya que el último tiene una separación entre los pines de 0.050" con el fin de acomodar más conexiones en el mismo espacio. Esto genera una incompatibilidad total entre los dos sistemas, lo que fue un error de IBM al no considerar las inversiones en tecnología por parte de los usuarios.

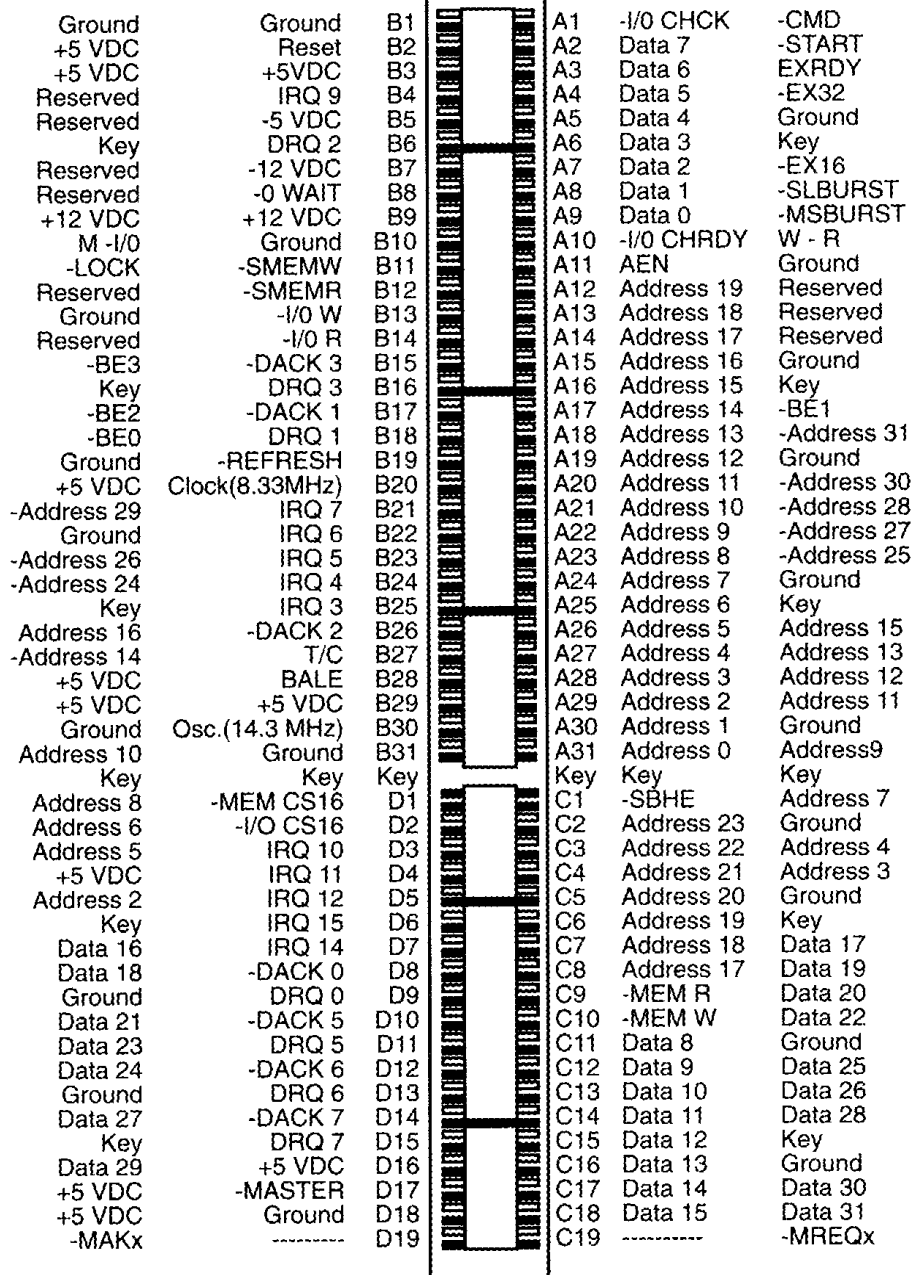
### **El bus EISA de 32 bits**

En 1988 y 1989 fue desarrollado el bus EISA (*Extended Industry Standard Architecture*) de 32 *bits* para satisfacer las necesidades de mayor velocidad y desempeño que proporcionaban los microprocesadores .. 80386 y 80486 y para no dejar solo a IBM, y su bus MCA, con este mercado.

Este bus, cuyo éxito es la compatibilidad con los buses anteriores, el ISA de 8 y 16 *bits*, es un producto de un consorcio (mas conocido como la banda de los nueve), formado por Compaq, Hewlett Packard, NEC, Zenith, AST, Epson, Wyse, Olivetti y Tandy.

Para lograr la compatibilidad entre los dos sistemas ISA y EISA, se diseñó una solución muy ingeniosa que tardó alrededor de dos años para ser perfeccionada. El conector que recibe las tarjetas tiene dos filas de pines, con diferente altura o nivel y la transferencia entre los dos buses se realiza automáticamente dependiendo de la forma del conector de la tarjeta.

Los pines superiores corresponden a las tarjetas tipo ISA y en el conector inferior hay cinco topes de plástico que no permiten que éstas hagan contacto con los pines inferiores, que corresponden al bus EISA. Las tarjetas EISA tienen una serie de ranuras que coinciden con los topes y éstas se pueden deslizar hasta el fondo obteniendo el contacto con sus respectivos pines.



El bus EISA tiene 99 pines por cada lado en los cuales se ha reducido su espaciamiento de 0.100" a 0.050". En la figura 3 tenemos la distribución de las señales en este tipo de bus. Como se puede observar en la figura, el bus EISA tiene 30 líneas de direccionamiento, 32 *bits* de datos, 15 niveles de interrupción y 7 canales de DMA.

Una de las principales ventajas del bus EISA es su capacidad de manejo autónomo del bus (*bus-mastering*), que en pocas palabras podría explicarse como la capacidad para permitir el intercambio de información entre dos periféricos sin la intervención de la CPU o P Aunque la velocidad del bus EISA se conservó en 8.33 MHz, para permitir la compatibilidad con ISA, su velocidad para transferencia de los datos es mucho mayor debido al aumento en la anchura del bus; esta llega a ser de unos 33 Mb/s (*Megabits* por segundo).

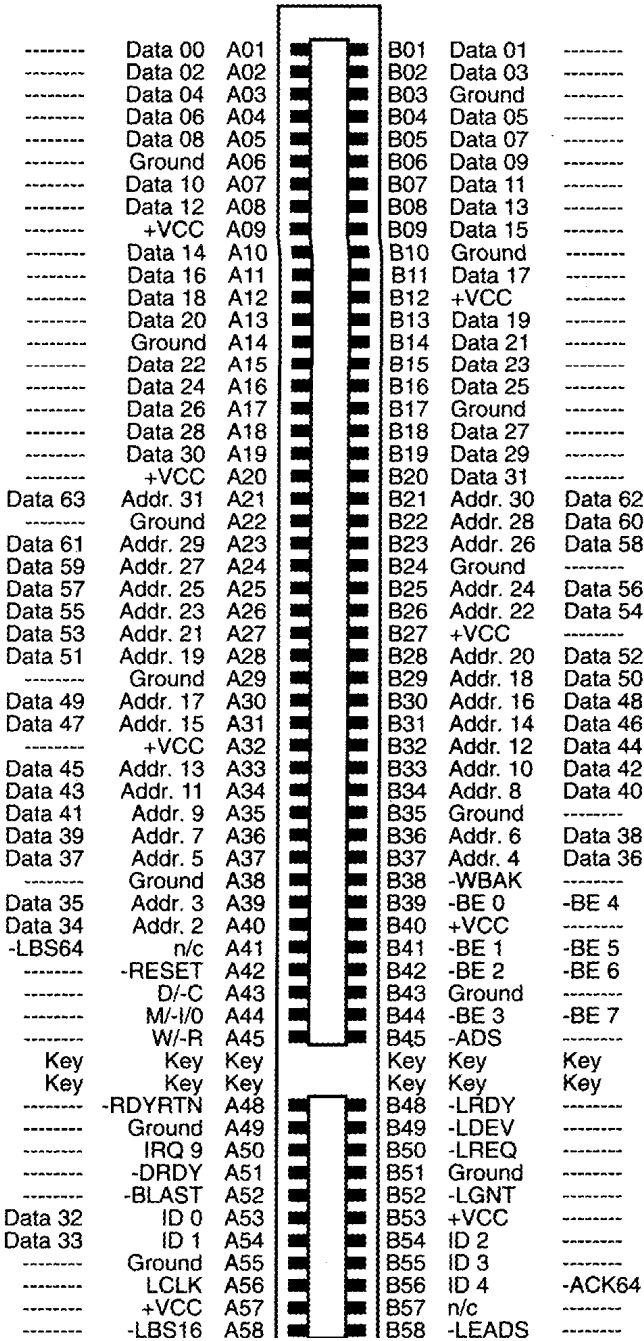
Desafortunadamente, muchas tarjetas de interface tipo EISA no son más rápidas que sus similares tipo ISA, debido a que sus fabricantes no utilizan todas las propiedades que el bus ofrece. Así mismo, la baja velocidad del disco duro (comparada con otros periféricos); es un problema que no depende del bus en sí.

LOS BUSES LOCALES

Los buses locales surgieron por la constante demanda de una mayor velocidad en las operaciones que involucran gráficas, video y sonido presentes en la gran cantidad de aplicacipnes con multimedia y otras similares, tan populares actualmente. La baja velocidad de los buses ISA, MCA y EISA sólo se podía superar conectándose directamente a las líneas de datos y direccionamiento del microprocesador y de la memoria. De esta forma, se pueden lograr velocidades hasta de 66 MHz en comparación a los 8 0 10 MHz de los otros buses. Los dos tipos principales de buses locales son el Video Local y el PCI.

El bus local de video o Video Local (VL) bus

Este tipo de bus fue propuesto en 1992 por la *Video Electronics Standards Association* y por eso se le llama *VESA Video Local Bus*. Irónicamente, este bus utiliza el mismo conector de 62 pines del bus MCA. Sus pines se muestran en la figura 4. Las tarjetas que utilizan este bus tienen dos conectores, uno en cada extremo y para cada bus. Los conectores o ranuras del *motherboard*, se colocan alineados con los conectores o ranuras ISA.El bus VL es expandible de 32 a 64 *bits* permitiendo su utilización en sistemas con el microprocesador Pentium, por medio de otro conector tipo MCA alineado con el primero.



## El bus local PCI

El bus local PCI (*Peripheral Component Interconnect*) fue propuesto por Intel en 1992 y apoyado por la mayoría de los fabricantes de computadores personales, incluyendo a Apple Computer, quien lo ha adoptado para su línea de equipos mas reciente que utilizan el microprocesador PowerPC 604.

La tendencia actual es la utilización del bus VESA VL para equipos de bajo perfil y del bus PCI para equipos de alta velocidad y configuraciones de buen tamaño (RAM, disco duro, alta resolución, etc.). Con el bus PCI se espera mejorar considerablemente el desempeño de los equipos en operaciones de alta intensidad en intercambio de datos como son el manejo de sonido con calidad de CD, así como el manejo de redes y discos duros de alta velocidad.

El bus PCI, de 188 pines, también utiliza conectores de 62 pines tipo MCA que se instalan algunas veces alineados con las ranuras ISA o EISA. En cambio del bus VL, el bus PCI utiliza una técnica de multiplexación de direcciones y datos que permite una considerable reducción en el número de pines. Esto crea la necesidad de circuitos de acoplamiento (*buffers*) que reducen un poco la velocidad de transferencia.

El bus PCI, de 32 *bits*, es expandible a 64 *bits* y soporta la lógica de 3.3 voltios junto a la normal de 5 voltios. De esta forma hay cuatro versiones, la de 5 voltios y 32 *bits*, la de 3.3 voltios y 32 *bits*, la de 5 voltios y 64 *bits* y la de 3.3 voltios y 64 *bits*. En la figura 5 tenemos la disposición de los pines y el nombre de las diferentes señales.

Las ranuras PCI estarán dedicadas para los periféricos más críticos que pueden estar conectados directamente a la tarjeta principal o por medio de tarjetas de interface como son las

|           |           |     |     |              |           |
|-----------|-----------|-----|-----|--------------|-----------|
| -12VDC    | TCK       | B2  | A2  | +12VDC       | +12VDC    |
| Ground    | Ground    | B3  | A3  | TMS          | TMS       |
| TDO       | TDO       | B4  | A4  | TDI          | TDI       |
| +5VDC     | +5VDC     | B5  | A5  | +5VDC        | +5VDC     |
| +5VDC     | +5VDC     | B6  | A6  | -INTA        | -INTA     |
| -INTB     | -INTB     | B7  | A7  | -INTC        | -INTC     |
| -INTD     | -INTD     | B8  | A8  | +5VDC        | +5VDC     |
| -PRSNT1   | -PRSNT1   | B9  | A9  | Reserved     | Reserved  |
| Reserved  | Reserved  | B10 | A10 | +3.3VDC(I/O) | +5VDC     |
| -PRSNT2   | -PRSTN2   | B11 | A11 | Reserved     | Reserved  |
| Ground    | Key       | B12 | A12 | Key          | Ground    |
| Ground    | Key       | B13 | A13 | Key          | Ground    |
| Reserved  | Reserved  | B14 | A14 | Reserved     | Reserved  |
| Ground    | Ground    | B15 | A15 | -RST         | -RST      |
| Clock     | Clock     | B16 | A16 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B17 | A17 | -GNT         | -GNT      |
| -REQ      | -REQ      | B18 | A18 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B19 | A19 | Reserved     | Reserved  |
| Adr/Dat31 | Adr/Dat31 | B20 | A20 | Adr/Dat30    | Adr/Dat30 |
| Adr/Dat29 | Adr/Dat29 | B21 | A21 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B22 | A22 | Adr/Dat28    | Adr/Dat28 |
| Adr/Dat27 | Adr/Dat27 | B23 | A23 | Adr/Dat26    | Adr/Dat26 |
| Adr/Dat25 | Adr/Dat25 | B24 | A24 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B25 | A25 | Adr/Dat24    | Adr/Dat24 |
| C/-BE3    | C/-BE3    | B26 | A26 | IDSEL        | IDSEL     |
| Adr/Dat23 | Adr/Dat23 | B27 | A27 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B28 | A28 | Adr/Dat22    | Adr/Dat22 |
| Adr/Dat21 | Adr/Dat21 | B29 | A29 | Adr/Dat20    | Adr/Dat20 |
| Adr/Dat19 | Adr/Dat19 | B30 | A30 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B31 | A31 | Adr/Dat18    | Adr/Dat18 |
| Adr/Dat17 | Adr/Dat17 | B32 | A32 | Adr/Dat16    | Adr/Dat16 |
| C/-BE2    | C/-BE2    | B33 | A33 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B34 | A34 | -FRAME       | -FRAME    |
| -IRDY     | -IRDY     | B35 | A35 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B36 | A36 | -TRDY        | -TRDY     |
| -DEVSEL   | -DEVSEL   | B37 | A37 | Ground       | Ground    |
| Ground    | Ground    | B38 | A38 | -STOP        | -STOP     |
| -LOCK     | -LOCK     | B39 | A39 | +3.3VDC      | +5VDC     |
| -PERR     | -PERR     | B40 | A40 | SDONE        | SDONE     |
| +5VDC     | +3.3VDC   | B41 | A41 | -SBO         | -SBO      |
| -SERR     | -SERR     | B42 | A42 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B43 | A43 | PAR          | PAR       |
| C/-BE1    | C/-BE1    | B44 | A44 | Adr/Dat15    | Adr/Dat15 |
| Adr/Dat14 | Adr/Dat14 | B45 | A45 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B46 | A46 | Adr/Dat13    | Adr/Dat13 |
| Adr/Dat12 | Adr/Dat12 | B47 | A47 | Adr/Dat11    | Adr/Dat11 |
| Adr/Dat10 | Adr/Dat10 | B48 | A48 | Ground       | Ground    |
| Ground    | Ground    | B49 | A49 | Adr/Dat9     | Adr/Dat9  |
| Key       | Ground    | B50 | A50 | Ground       | Key       |
| Key       | Ground    | B51 | A51 | Ground       | Key       |
| Adr/Dat8  | Adr/Dat8  | B52 | A52 | C/-BE0       | C/-BE0    |
| Adr/Dat7  | Adr/Dat7  | B53 | A53 | +3.3VDC      | +5VDC     |
| +5VDC     | +3.3VDC   | B54 | A54 | Adr/Dat6     | Adr/Dat6  |
| Adr/Dat5  | Adr/Dat5  | B55 | A55 | Adr/Dat4     | Adr/Dat4  |
| Adr/Dat3  | Adr/Dat3  | B56 | A56 | Ground       | Ground    |
| Ground    | Ground    | B57 | A57 | Adr/Dat2     | Adr/Dat2  |
| Adr/Dat1  | Adr/Dat1  | B58 | A58 | Adr/Dat0     | Adr/Dat0  |
| +5VDC     | +3.3VDC   | B59 | A59 | +3.3VDC      | +5VDC     |
| -ACK64    | -ACK64    | B60 | A60 | -REQ64       | -REQ64    |
| +5VDC     | +5VDC     | B61 | A61 | +5VDC        | +5VDC     |
| +5VDC     | +5VDC     | B62 | A62 | +5VDC        | +5VDC     |
| Key       | Key       | Key | Key | Key          | Key       |
| Key       | Key       | Key | Key | Key          | Key       |
| Reserved  | Reserved  | B63 | A63 | Ground       | Ground    |
| Ground    | Ground    | B64 | A64 | C/-BE7       | C/-BE7    |
| C/-BE6    | C/-BE6    | B65 | A65 | C/-BE5       | C/-BE5    |
| C/-BE4    | C/-BE4    | B66 | A66 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B67 | A67 | PAR64        | PAR64     |
| Adr/Dat63 | Adr/Dat63 | B68 | A68 | Adr/Dat62    | Adr/Dat62 |
| Adr/Dat61 | Adr/Dat61 | B69 | A69 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B70 | A70 | Adr/Dat60    | Adr/Dat60 |
| Adr/Dat59 | Adr/Dat59 | B71 | A71 | Adr/Dat58    | Adr/Dat58 |
| Adr/Dat57 | Adr/Dat57 | B72 | A72 | Ground       | Ground    |
| Ground    | Ground    | B73 | A73 | Adr/Dat56    | Adr/Dat56 |
| Adr/Dat55 | Adr/Dat55 | B74 | A74 | Adr/Dat54    | Adr/Dat54 |
| Adr/Dat53 | Adr/Dat53 | B75 | A75 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B76 | A76 | Adr/Dat52    | Adr/Dat52 |
| Adr/Dat51 | Adr/Dat51 | B77 | A77 | Adr/Dat50    | Adr/Dat50 |
| Adr/Dat49 | Adr/Dat49 | B78 | A78 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B79 | A79 | Adr/Dat48    | Adr/Dat48 |
| Adr/Dat47 | Adr/Dat47 | B80 | A80 | Adr/Dat46    | Adr/Dat46 |
| Adr/Dat45 | Adr/Dat45 | B81 | A81 | Ground       | Ground    |
| Ground    | Ground    | B82 | A82 | Adr/Dat44    | Adr/Dat44 |
| Adr/Dat43 | Adr/Dat43 | B83 | A83 | Adr/Dat42    | Adr/Dat42 |
| Adr/Dat41 | Adr/Dat41 | B84 | A84 | +3.3VDC      | +5VDC     |
| Ground    | Ground    | B85 | A85 | Adr/Dat40    | Adr/Dat40 |
| Adr/Dat39 | Adr/Dat39 | B86 | A86 | Adr/Dat38    | Adr/Dat38 |
| Adr/Dat37 | Adr/Dat37 | B87 | A87 | Ground       | Ground    |
| +5VDC     | +3.3VDC   | B88 | A88 | Adr/Dat36    | Adr/Dat36 |
| Adr/Dat35 | Adr/Dat35 | B89 | A89 | Adr/Dat34    | Adr/Dat34 |
| Adr/Dat33 | Adr/Dat33 | B90 | A90 | Ground       | Ground    |
| Ground    | Ground    | B91 | A91 | Adr/Dat32    | Adr/Dat32 |
| Reserved  | Reserved  | B92 | A92 | Reserved     | Reserved  |
| Reserved  | Reserved  | B93 | A93 | Ground       | Ground    |
| Ground    | Ground    | B94 | A94 | Reserved     | Reserved  |

unidades para multimedia (CD-ROM), tarjetas para gráficas y video, unidades de disco duro y tarjetas para conexión en red. De todas maneras, los sistemas seguirán incorporando ranuras ISA.

Cada paso que se avanza en la industria de los computadores personales trae cambios tanto técnicos como de adaptación del usuario y requiere un consenso de las empresas que dominan el mercado. Si no se tiene en cuenta al usuario o no se logra una norma aprobada por un buen número de fabricantes, como en el caso del bus MCA, los resultados pueden ser muy negativos. El bus PCI, aceptado por las dos principales plataformas, PC y Mac, y en versiones de 32 y 64 bits, se vislumbra como uno de los que permanecerán por más tiempo en el mercado.

## **El Puerto de Gráficos Acelerado (AGP)**

La necesidad de aumentar el ancho de banda (ancho de banda) entre el procesador principal y el subsistema video lleva originalmente al desarrollo del bus de I/O local en el PCs, empezando con el

VESA local bus y en el futuro llevando al popular bus PCI. Esta tendencia continúa, con la necesidad de ancho de banda de video que ahora empieza a empujar contra los límites de incluso el bus PCI.

Tal como anteriormente fue el caso con el bus ISA, el tráfico en el bus PCI está empezando a ponerse pesado en las PCs de extremo alto, con video, disco duro y datos de periféricos todos que compiten para el mismo ancho de banda de I/O. Para combatir la saturación eventual del bus PCI con información de video, una nueva interface ha sido introducida por Intel, diseñado específicamente para el subsistema de video. Se llama el Puerto de Gráficos Acelerado o AGP.

AGP se desarrolló en respuesta a la tendencia hacia los requisitos de mayores y mayores desempeño para el video. Cuando el software evoluciona y el uso de la computadora continúa en las áreas previamente inexploradas como aceleración 3D y reproducción de video en pleno movimiento, el procesador y los *chipset* videos necesitan procesar cada vez más información. El bus de PCI está alcanzando su límite de desempeño en estas aplicaciones, sobre todo con disco duro y también otros periféricos en su lucha para el mismo ancho de banda.

Otro problema ha sido las demandas crecientes para la memoria de video. Cuando la informática en 3D se vuelve más corriente principal, se requieren cantidades muy grandes de memoria, no sólo para la imagen de la pantalla sino también para hacer los cálculos en 3D. Esto tradicionalmente ha significado poner más memoria en la tarjeta de video por hacer este trabajo. Hay dos problemas con esto:

**Costo:** La memoria de la tarjeta de video es muy cara comparada a la RAM normal del sistema.

**Tamaño limitado:** La cantidad de memoria en la tarjeta de video está limitada: si usted decide poner 6 MB en la tarjeta y usted necesita 4 MB para el buffer del marco, usted tiene 2 MB por encima de para trabajo de procesamiento y nada más (a menos que usted hace una actualización del hardware). No es fácil de extender esta memoria, y usted no puede usarla para nada más que no sea para el proceso de video.

AGP va en solución de estos problemas permitiendo el procesador de video acceder la memoria del sistema principal por hacer sus cálculos. Esto es más eficaz porque esta memoria puede compartirse

dinámicamente entre el procesador del sistema y el procesador de video, dependiendo de las necesidades del sistema.

La idea detrás de AGP es simple: crear una interface más rápida y especializada, entre el *chipset* de video y el procesador del sistema. La interface es sólo entre estos dos dispositivos; esto tiene tres ventajas mayores: Es más fácil de implementar el puerto, hace más fácil de aumentar la velocidad del AGP, y hace posible implementar mejoras en el diseño que son específicas del video.

AGP es considerado un puerto, y no un bus, porque sólo involucra dos dispositivos (el procesador y tarjeta del video) y no es expansible. Una de las grandes ventajas de AGP es que aísla el subsistema de video del resto de la PC de manera que no hay casi ninguna disputa sobre el ancho de banda de I/O como sucedía con el PCI. Con la tarjeta de video quitada del bus de PCI, otros dispositivos de PCI también se beneficiarán del ancho de banda mejorado.

AGP es una nueva tecnología y se presentó al mercado en el tercer trimestre de 1997. El primer soporte para esta nueva tecnología será de *chipset* 440LX Pentium II de Intel. Más información sobre AGP será venidera cuando se vuelve más corriente y se vea más en el mercado de la informática general. Interesantemente, se supuso uno de las metas de Intel con AGP para ser hacer el video del extremo alto más económico sin requerir sofisticadas tarjetas de video 3D. Si éste es el caso, realmente me hace preguntarse por qué ellos están haciendo sólo AGP disponible para su línea del procesador Pentium II de alto extremo. : ^) Originalmente, AGP fue rumoreado para ser un rasgo en el *chipset* 430TX de Pentium socket 7, pero no materializó. *Via* y otras compañías están llevando la bandera ahora para el desarrollo futuro del *chipset* socket 7, ese que Intel ha dejado caer, y varios *chipsets* no-Intel con soporte AGP estarán entrando en el mercado en 1998.

Las secciones debajo discuten varios aspectos del AGP, incluyendo sus características y problemas de desempeño. Para obtener más información adicional, revise la página de AGP de Intel.

## **Interfaz AGP**

La interfaz AGP todavía es de muchas maneras bastante similar a PCI. El propio *slot* es físicamente similar en forma y tamaño, pero se desplaza más allá del borde del motherboard en donde se ubican los *slots* PCI. La especificación de AGP es de hecho basado en la especificación PCI 2.1 que incluye un alto- ancho de banda velocidad de 66 MHz que nunca fue implementado en el PC. Los *motherboards* de AGP tienen un solo *slot* de tarjeta de expansión para la tarjeta de video AGP, y normalmente un *slot* PCI menos, y es por otra parte bastante similar a las *motherboards* PCI.

## **Ancho, Velocidad y Ancho de banda del Bus AGP**

El bus AGP es de 32 bits ancho, sólo igual que el PCI, pero en lugar de correr a la mitad de la velocidad del bus sistema (memoria) de la manera que lo hace PCI, corre a velocidad plena del bus. Esto significa que en un motherboard Pentium II normales que AGP ejecuta a 66 MHz en lugar del 33 MHz del bus PCI. Esto duplica el ancho de banda del puerto por supuesto inmediatamente; en lugar del límite de 133 MB/s como con PCI, AGP en su modo de velocidad más bajo tiene un ancho de banda de 266 MB/s. Más por supuesto los beneficios de no tener que compartir ancho de banda con otros dispositivos de PCI.

Además de duplicar la velocidad del bus, AGP ha definido un modo 2X que usa señalización especial para permitir enviar dos veces tanto datos encima del puerto a la misma velocidad del reloj. Lo que el hardware hace es enviar información sobre los flancos de subida y bajada de la señal de reloj -- normalmente sólo se envía información una vez por cada ciclo del reloj, pero con este modo 2X puede enviarse dos veces información. El resultado es que el desempeño se duplica de nuevo, a 533 MB/s el ancho de banda teórico. Hay también un plan para llevar a cabo un modo 4X que realizará cuatro transferencias por ciclo del reloj: ¡un enorme ancho de banda de 1.066 GB/s!

Esto es ciertamente muy excitante, pero nosotros debemos temprar esta excitación un poco (y no sólo porque AGP es nuevo y nosotros no tenemos mucho que es práctico evaluar todavía). Es gran diversión para hablar aproximadamente ancho de banda de 1 GB/s para la tarjeta de video, pero hay sólo un problema: ¡esto está por encima del ancho de banda del bus del sistema entero de un PC moderno! Si usted revoca, el bus de datos de una PC clase Pentium o posterior es de 64 bits ancho y corre a 66 MHz. Esto da un ancho de banda total de 533 de MB/s, así que el máximo de 1 GB/s no va a hacer muy bueno hasta que nosotros consigamos el bus de datos que corre más rápido que 66 MHz. Los chipsets de futuras motherboard tomarán el bus del sistema a 100 MHz que aumentarán ancho de banda de memoria total a 800 MB/s, un paso definido en la dirección correcta, pero todavía no bastante para hacer factible transferencias a 4X.

También vale recordar que el CPU también necesita tener acceso a la memoria del sistema, no sólo el subsistema video. ¿Si todo el 533 MB/s de ancho de banda del sistema es tomado por el video sobre AGP, qué va a hacer el procesador? De nuevo aquí, yendo el sistema a la velocidad de 100 MHz ayudará inmensamente.

## 2.2 Estructura de E/S

### INTRODUCCION:

Una de las funciones principales de un sistema operativo es controlar todos los dispositivos de entrada salida de la computadora, este debe emitir comandos a los dispositivos, capturar interrupciones y manejar errores. También debe de proporcionar una interfaz entre los dispositivos y el resto del sistema que sea simple y fácil de utilizar.

Hasta donde sea posible, la interfaz debe ser la misma para todos los dispositivos. El código de entrada salida representa una fracción significativa del sistema operativo en su totalidad.

Los dispositivos de E/S se dividen en dos categorías: dispositivos de bloque (paralelos) y dispositivos de caracteres (seriales).

Dispositivos de bloque: son aquellos que almacenan información en bloques de tamaño fijo, cada uno con su dirección. Los tamaños van de los 128 bytes a los 1024 bytes. Su característica es que pueden ser leídos o escritos en forma independiente de los demás por parte del sistema.

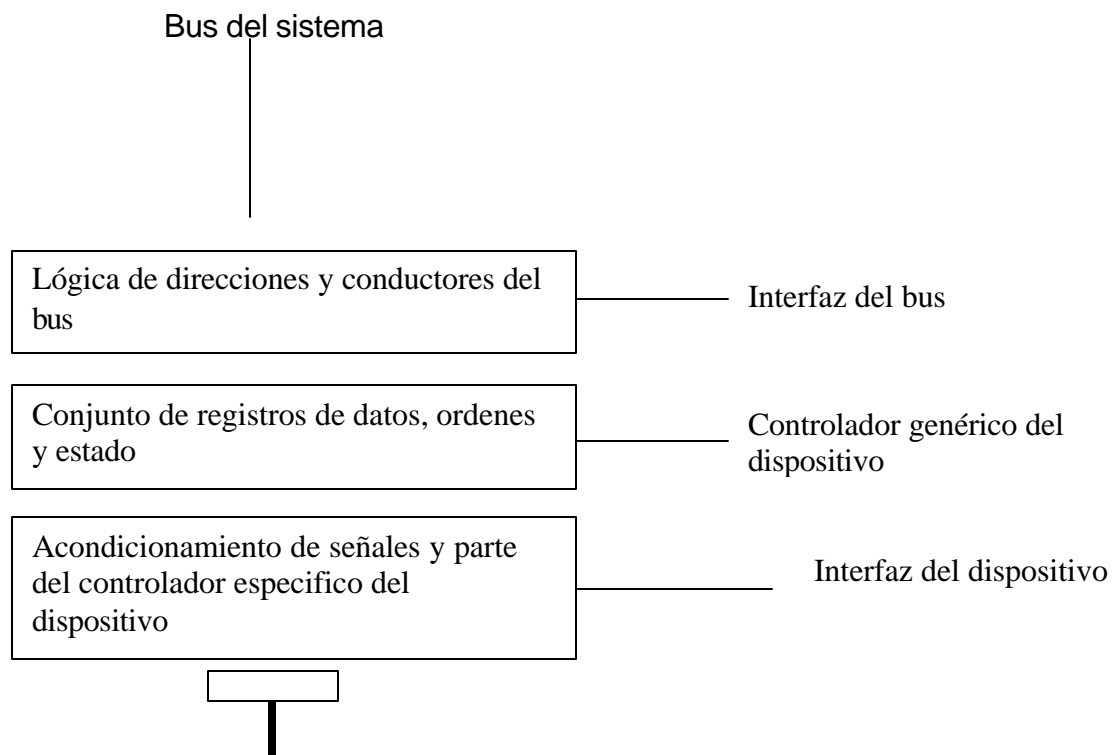
En cambio un dispositivo de E/S de caracteres entrega o acepta un flujo de caracteres, sin importar de cual estructura de bloque se trate. No es direccionable y no tiene ninguna operación de

localización. Las terminales, las impresoras de líneas, el mouse, las interfaces de red entre otros, pertenecen a esta categoría.

#### Interfaces de E/S.

Las unidades de E/S constan de una parte mecánica y otra electrónica, la parte electrónica se denomina controlador del dispositivo o adaptador el cual generalmente se identifica como una tarjeta que se inserta en uno de los slots de expansión de la computadora los cuales son parte del bus del sistema, o bien se conecta a uno de los puertos de salida de la misma. El componente mecánico es el dispositivo de E/S.

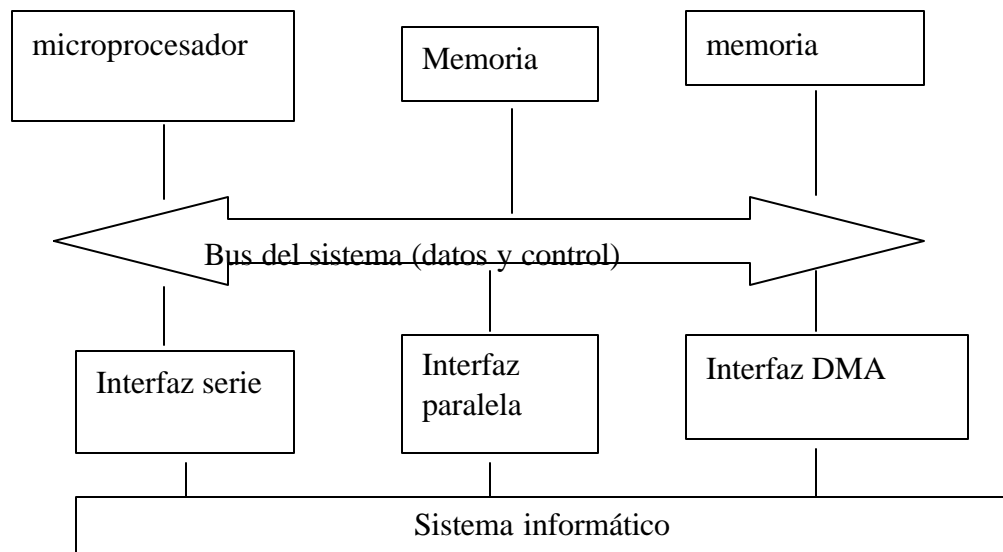
Por otro lado es necesario tomar en cuenta que el complemento del controlador, son los drivers o programas de configuración del dispositivo.





(Dispositivo físico)

***Diagrama esquemático de un controlador de E/S***



Buffers.

de entrada. La función de un buffer de entrada es guardar un dato hasta que el procesador este preparado para aceptarlo. Esto es mas notorio en sistema de multiprogramación en donde un procesador esta multiplexado entre varias actividades concurrentes.

De salida. Sirven principalmente al propósito de almacenar los datos de salida escritos por el procesador de modo que los dispositivos externos puedan disponer de señales continuas adecuadas a sus exigencias de temporización.

Registros de ordenes.

Están encargados de transferir ordenes de E/S entre un microprocesador y un dispositivo de E/S.

Registros de estado.

Se utilizan para proporcionar información al microprocesador con respecto al estado del dispositivo de E/S. Incluyen información tal como si el dispositivo esta preparado u ocupado, el bufer esta vacio o lleno e indicaciones de error.

## 2.3 Estructura de almacenamiento

### Memoria

La memoria sirve para guardar datos cuando no son trasladados de un lugar a otro. Hay cuatro tipos de memoria en las computadoras:

1. De acceso aleatorio
2. Virtual
3. De sólo lectura
4. CMOS

- **Memoria de Acceso Aleatorio o RAM**

Este tipo de memoria sirve para guardar datos temporalmente y sólo guarda los datos mientras la computadora esté encendida, una vez que se apaga desaparecen los datos almacenados en RAM.

En los chips de RAM hay pequeños capacitores que al estar encendidos representan un 1 y al estar apagados un 0. Por lo que es lógico pensar que los capacitores están en grupos de ocho.

El RAM se mide actualmente en Mb, por lo que los datos se guarda por millones.

La arquitectura física de la RAM es denominada como **SIMM** (Single In-Line Memory Module) y es una pequeña tarjeta de circuito que tiene conectados de tres a nueve **DIPs**. Un **DIP** es un módulo que almacena 256 kilobits.

- **Memoria Virtual**

La memoria virtual es usada por la computadora para guardar datos simulando la memoria RAM en el almacenamiento del disco. Esta es una gran ayuda, ya que una máquina con poca memoria RAM puede ejecutar programas grandes gracias a la memoria virtual.

El problema de la memoria virtual es que es más lenta que la RAM, y las computadoras que la usan tardan mucho en recuperar datos guardados en memoria virtual ya que el disco es un dispositivo mecánico.

- **Memoria de Sólo Lectura o ROM**

Esta se encuentra en los chips de ROM y son instrucciones permanentes que usa la computadora para preparar tareas de procesamiento. En ella están las instrucciones que le dicen a la computadora cómo llegar a las unidades de disco y buscar el sistema operativo. Al encender la computadora, ésta sigue las instrucciones del ROM para inicializarse.

- **Memoria CMOS**

Esta permite que algunas instrucciones de inicialización no sean permanentes al emplear corriente de una batería para guardar información del sistema vital, como las especificaciones del disco duro.

Al cambiar la configuración del sistema, se debe actualizar la información en el CMOS, por ejemplo: al cambiar un disco duro.

## 2.4 Protección del hardware

El sistema operativo interactúa en armonía con el hardware de la computadora y en consecuencia debe contemplar esquemas de tipo software tendientes a protegerlo a través del manejo de cada uno de los elementos del hardware como son la memoria o los buses e incluso mediante un adecuado manejo de las interrupciones.

## 2.5 Sistema de interrupciones

Controlador de interrupciones 8259. Otro de los chips de la familia Intel que integran la PC es el controlador programable de interrupciones (PIC) 8259. Este chip permite ampliar a ocho las entradas de interrupción, generándose las señales llamadas IRQ0 a IRQ7. El diseño de la placa madre de la PC prevee la asignación de algunas de éstas para funciones específicas, tal es el caso de las IRQ0 e IRQ1. La primera de ellas se conecta a la salida del contador del timer y sirve para proveer una base de tiempo confiable. En la programación inicial del 8259 se le asigna a esta entrada el vector de interrupción 8, por lo que las restantes poseen los vectores siguientes (9 a 15). La IRQ1 (vector 9) se emplea para la lectura del teclado.

Las restantes interrupciones se encuentran en el slot de expansión del BUS, lo que permite que sean usadas por cualquier dispositivo conectado en dicho lugar. Sin embargo la estandarización del hardware y software mencionada con anterioridad hizo que los controladores más comunes conserven una interrupción como predeterminada. Esto implica que las mismas no deben ser utilizadas por otros dispositivos cuando se encuentre instalado alguno de esos controladores. A continuación se proporciona una lista de las interrupciones y se mencionan los controladores más comunes que hacen uso de ellas.

| Interrupción | Aplicación     |
|--------------|----------------|
| IRQ2         | RESERVADA      |
| IRQ3         | COM/SDLC       |
| IRQ4         | COM/SDLC       |
| IRQ5         | DISCO DURO     |
| IRQ6         | DISCO FLEXIBLE |
| IRQ7         | LPT            |

La IRQ2 que figura como reservada permite que se la utilice para agregar un segundo chip 8259 en cascada, lo que implica un aumento de 7 interrupciones. Esto se incorporó a la placa madre de la PC-AT que sucedió a la PC, e incorporaba el microprocesador 80286 como unidad central de proceso.

## 2.6 El mapa de direcciones

### El Bus

Los datos van de una parte de la computadora a otra por medio de un trayecto o circuito llamado **“bus”**. Funciona de la misma forma que un autobús: Recoge pasajeros (datos) los lleva de un lugar a otro y los deja. Después recoge otros pasajeros y así sucesivamente.

**bus se divide en dos tipos de líneas:**

1. Las líneas de datos
2. Las líneas de direccionamiento

Las líneas de datos son las que llevan los datos de un lugar a otro y las líneas de direccionamiento llevan la ubicación de los datos para que la computadora sepa donde están.

El bus tiene su velocidad, y se mide en Mhz. Hay varios tipos de bus, y se listan a continuación junto con sus velocidades correspondientes:

| Nombre         | Bits que transporta | Velocidad (Mhz) |
|----------------|---------------------|-----------------|
| Bus XT         | 8                   | 4.77            |
| Bus ISA        | 16                  | 8               |
| Bus EISA       | 32                  | 6 - 8.33        |
| Bus Microcanal | 16 ó 32             | 10              |
| Bus Local VESA | 32                  | 33 - 66         |
| Bus Local PCI  | 32 ó 64             | 33 - 66         |

## 2.7 Estructura de red

### 3. Procesos

#### 3.1 Introducción a los procesos

Todas las computadoras modernas pueden realizar varias tareas al mismo tiempo. En estos sistemas de multiprogramación la CPU debe cambiar de un programa a otro, ejecutando cada uno en decenas o cientos de milisegundos. Pero en cualquier instante de tiempo la CPU puede estar ejecutando solo un programa.

Todo el software ejecutable en la computadora, que con frecuencia incluye al sistema operativo, se organiza en varios procesos secuenciales. Un proceso es básicamente un programa ejecutante donde intervienen los valores corrientes del contador de programa, registros y variables.

#### Concepto de proceso.

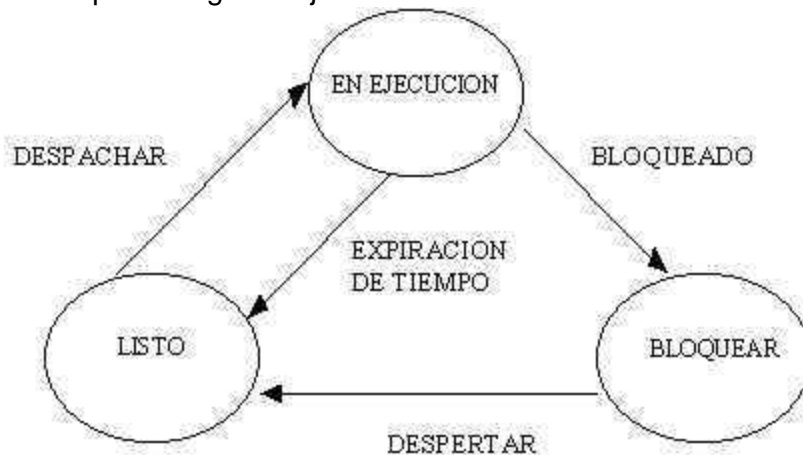
El término "**PROCESO**", fue utilizado por primera vez por los diseñadores del sistema Multics en los años 60's. Desde entonces, el término proceso, utilizado a veces como sinónimo de tarea, ha tenido muchas definiciones . A continuación se presentan algunas:

- *Un programa en ejecución*
- *Una actividad asíncrona*
- *El "espíritu animado" de un procedimiento*
- *El "centro de control" de un procedimiento en ejecución*
- *Lo que se manifiesta por la existencia de un "bloque de control del proceso" en el sistema operativo*
- *La entidad a la que se asignan los procesadores*
- *La unidad "despachable"*

Aunque se han dado muchas otras definiciones, no hay una definición universalmente aceptada, pero el concepto de "Programa en ejecución" parece ser el que se utiliza con

mas frecuencia. Un programa es una entidad inanimada; sólo cuando un procesador le "infunde vida" se convierte en la entidad "activa" que se denomina proceso.

Un proceso pasa por una serie de datos discretos. Se dice que un proceso se está ejecutando (estado de ejecución), si tiene asignada la UCP. Se dice que un proceso está listo (estado listo) si pudiera utilizar una UCP en caso de haber una disponible. Un proceso está bloqueado (estado bloqueado) si está esperando que suceda algún evento antes de poder seguir la ejecución.



### ***Estados de un proceso.***

Un proceso puede variar entre 5 distintos estados:

**New:** cuando el proceso esta siendo creado.

**Running:** cuando el proceso se esta ejecutando.

**Waiting:** cuando el proceso esta esperando que se cumpla algún otro evento.

**Ready:** cuando el proceso esta pronto para ejecutar, esperando por la CPU.

**Terminated:** cuando el proceso esta terminado.

Bloque de control del proceso PCB (Process Control Block).-

Internamente los procesos se almacenan en una lista encadenada en donde cada nodo almacena la información anterior.

Cada nodo tiene información de:

- Nombre del proceso
- Estado del proceso
- Program Counter

- Los registros de la CPU
- La información de Scheduling de la CPU
- La información de uso (tiempo que ha usado la CPU, hora de despachado, etc.)
- Información de E/S(Lista de dispositivos "alocados", archivos abiertos, etc.)

### *Scheduling.*

*Colas de Scheduling:* Los procesos que están en estado de espera se quedan en una lista llamada lista o cola de ready. Los procesos que hacen uso de E/S se guardan en una cola de E/S. Hay una cola de E/S por cada dispositivo.

*Schedullers:* Componente del sistema operativo responsable de decidir quien hará uso de la CPU.

### Round Robin

Este es un algoritmo basado en FCFS. Trata la cola de ready como una lista circular. Introduce el concepto de "Quantum" o "Time slice" : mayor tiempo de cpu que podrá hacer uso un proceso en cada vuelta. Si el valor del Quantum fuese muy grande, el algoritmo funcionara como un FCFS. Si el Quantum fuera muy chico se produce un overhead por context switch (significa que el Quantum se setea en un tiempo menor al que demora el context switch). Este algoritmo es fuertemente dependiente del Quantum o Time Slice. El Quantum debe ser mayor que el 80% de los tiempos de CPU que hagan uso los procesos, pero no el proceso entero sino por cada burst.

MQS (Multilevel Queue Scheduling)

Este algoritmo parte la cola de ready en un numero de colas n. Luego existe un criterio para clasificar en que cola será colocado un proceso cuando que queda en estado de ready. Cada cola puede manejar su propio algoritmo de scheduling

MFQS (Multilevel Feed Back Queue Scheduling)

Define los siguientes parámetros:

- Numero de colas
- Algoritmo de scheduling usado en cada cola
- Método para decidir a que cola entrara un proceso cuando entre a estado de ready
- Método para decidir cuando un proceso será enviado a una cola de menor prioridad.

### Múltiple CPU

Para que mas de una CPU no tomen el mismo proceso de la cola de ready se utilizan mecanismos de sincronización. Otro mecanismo seria partir la cola en n colas (n CPUs), pero si se hiciera esto podría llegar a suceder que los procesos de una CPU fueran todos cortos y los de otra fueran largos por lo cual habría una CPU que quedaría libre y otra ejecutando. Otra forma sería que una CPU decidiera cual CPU va a ejecutar cual proceso. Esto puede llevar a que la CPU que esta seleccionando quede un poco mas cargada porque en algún momento estará ejecutando el algoritmo de selección y un proceso asignado a ella.

## **3.2 Problemas clásicos de la Comunicación entre procesos.**

### 3.2.1 Comunicación entre procesos

Hay dos tipos de procesos: los *independientes* que no afectan ni pueden ser afectados por ningún otro proceso y los *cooperativos* que afectan y pueden ser afectados por algún otro proceso del sistema operativo.

Los procesos con frecuencia precisan comunicarse entre si, Para esto existen ciertos contratiempos que debieron ser superados. Para prevenir este problema en las situaciones en la que interviene cualquier recurso compartido se debe impedir que mas de un proceso haga uso del recurso compartido al mismo tiempo. Lo que se necesita es exclusión mutua, La parte del programa que accesa la memoria compartida se le llama *sección crítica*. Para tener una solución adecuada los procesos deben cumplir estos cuatro puntos:

1. Nunca dos procesos pueden encontrarse simultáneamente en sus secciones críticas.
2. No se hacen suposiciones acerca de las velocidades relativas de los procesos o del numero de CPU.
3. Ningún proceso suspendido fuera de la sección crítica debe bloquear a otros procesos.
4. Nunca un proceso debe querer entrar en forma arbitraria en su sección crítica.

### Algoritmos de Solución al Problema de Sección Crítica

#### Técnicas con Espera Ocupada

##### Deshabilitación de Interrupciones

La solución mas simple es que cada proceso desactive todas las interrupciones justo después de entrar en su sección crítica y las vuelva a activar antes de salir de ella. Con las interrupciones deshabilitadas no pueden ocurrir interrupciones al reloj por lo cual no se cambiara a otro proceso. Este método no es muy atractivo ya que no es muy prudente dar a los procesos del usuario el poder de desactivar las interrupciones.. Si un usuario lo hiciera y nunca las volviera a activar sería el fin del sistema. Además si fuera un sistema con mas de una CPU la desactivación de interrupciones solo afectaría una de ellas y las otras podrían accesar la memoria compartida.

#### Algoritmo de Variables de Cierre

Este algoritmo utiliza un flag. Al entrar a la sección crítica se fija si es uno o cero si es cero lo pone en uno y entra a la sección crítica; si es uno espera hasta que valga cero. Antes de salir de la sección crítica iguala el flag a cero.

```
Repeat
If (flag = 0) then {
Flag:=1
Seccion crítica
Flag:=0
Seccion no crítica}
Until 0 = 1
```

Este algoritmo si embargo no resuelve el problema de la sección crítica porque si hubiera una interrupción justo después de comprobar el estado del flag y se accediera a la memoria compartida antes de cambiar el flag, otro proceso podría acceder a la memoria compartida, cambiar el flag a uno y habrían dos procesos accesando la memoria compartida.

#### Algoritmo de alternación Estricta



En este algoritmo dos procesos comparten una variable flag, pero en vez de utilizar el mismo código se usa distinto código para cada proceso.

El problema aquí radica en que si un proceso tiene una sección no crítica muy larga el otro tendrá que esperar para entrar en su sección crítica a pesar de que no se esté ejecutando nada que la afecte.

Algoritmo de Peterson (o Algoritmo de Dekker)

Este algoritmo combina las dos ideas anteriores. Se comparten una variable flag y una variable turno.

Este algoritmo soluciona el problema de la sección crítica y evita que un proceso quede en espera mientras que otro proceso esté en su sección no crítica. Para  $n$  procesos la idea es un poco más compleja:

Flag::array[0.. $n-1$ ]

Turn::0.. $n-1$

El while se tornará en una evaluación de  $n$  condiciones

### 3.2.2 Semáforos

Son una herramienta de sincronización. Es una variable protegida que solo puede ser modificada por la rutina de inicialización y por otras dos operaciones atómicas:

$P() \leftarrow \text{wait}$

$V() \leftarrow \text{signal}$

Las operaciones a las cuales se puede acceder son:

Inicialización: Crea un nuevo semáforo asignándole un valor inicial

$P(s)$ : while ( $s=0$ ) do no\_op ATÓMICA

$s:=s-1$

$V(s)$ :  $s:=s+1$  ATÓMICA

Existen básicamente dos tipos de semáforos:

- Semáforos contadores: Toman valores positivos mayores o iguales a 0. Se utilizan para sincronización de procesos.
- Semáforos binarios: Toman los valores 0 ó 1 y se utilizan para exclusión mutua.

A continuación se presenta una solución mediante semáforos del problema productor/consumidor.

```
#define N 100
semaphore mutex = 1;
semaphore empty = N;
semaphore full = 0;
void productor(void)
{int item;
while(1){
produce_item(item);
down(empty);
down(mutex);
enter_item(item);
up(mutex);
```

```

up(full);}
}
void consumidor(void)
{ int item;
while(1){
down(full);
down(mutex);
remove_item(&item);
up(mutex);
up(empty);
consume_item(item);}
}

```

En este caso tenemos la utilización de 3 semáforos, uno destinado al control de la exclusión mutua y los otros dos destinados a la sincronización de los procesos para el control de buffer lleno y vacío.

Podríamos utilizar semáforos para un algoritmo de espera ocupada con n procesos, pero los n procesos que están ejecutando el while de la función P(s) van a la cola de ready en un instante de tiempo reduciendo la performance general del equipo.

Para evitar la espera ocupada se crea un semáforo que sea un registro de un nuevo tipo:

Semáforo = Record

Valor: Integer

L: Lista\_de\_Procesos

End

P(s) {

s.Valor:= s.valor - 1

if s.Valor < 0 then agregar este proceso a s.L

bloquear;

end}

V(s){

s.Valor:=s.Valor + 1

if s.Valor = 0 then quitar un proceso y a s.L

despertar(y) }

### 3.2.3 Monitores

Es un tipo de procedimientos, variables y estructuras de datos que se agrupan en un tipo de modulo especial. Tienen una propiedad importante: solo un proceso puede estar activo en un monitor en un instante de tiempo.

Los monitores proveen un nuevo tipo de variables de condición con dos operaciones que operan sobre el (solo se usan dentro del procedimiento de el monitor).

*Wait* -> *wait(a)* : produce que el proceso que ejecuta la instrucción sea interrumpido y removido de la cola de ready hasta que otro proceso lo habilite ejecutando la instrucción *signal()* con la misma variable de condición.

*Signal* -> *signal(a)* : Habilita la ejecución de algún proceso en espera por la ejecución de la instrucción *wait* con la misma variable de condición.

### Deadlocks

Si un conjunto de procesos esta en estado de espera por recursos y nunca cambia de estado porque los recursos por los que espera están siendo utilizados por otros procesos en estado de espera tenemos un DEADLOCK. Los recursos son la memoria, dispositivos de E/S, semáforos, archivos, etc. La forma en que un proceso hace uso de un recurso es:

*Solicitud:* si el recurso esta disponible se le otorga, si no el proceso pasa a estado de espera.

*Uso:* El proceso utiliza el recurso

*Liberación:* el proceso debe liberar el recurso.

*Condiciones Necesarias para que se produzca DEADLOCK*

Si se presentan simultáneamente las cuatro siguientes condiciones el sistema esta en DEADLOCK.

1. EXCLUSION MUTUA: Por lo menos un proceso que tenga otorgado un recurso en forma exclusiva.
2. USO Y ESPERA: Debe existir al menos un proceso que este haciendo uso de un recurso y que este esperando por otros recursos asignados a otros procesos.
3. NO INTERRUPCION: Los recursos no pueden ser retirados al proceso. Si un proceso hace uso de un recurso no le podrá ser retirado hasta que voluntariamente el proceso lo libere.
4. ESPERA CIRCULAR: Debe existir un conjunto de procesos  $\{P1.....Pn\}$  tal que P1 espera por un recurso utilizado por P2,.....,Pn espera por un recurso utilizado por P1.

*Resource Allocation Graph(Grafo de utilizacion de recursos)*

Conjunto de Vértices:  $U = P \cup R$

$P=\{P1,P2,....,Pn\}$

$R=\{R1,R2,....,Rn\}$

Conjunto de Aristas:

Una arista de un proceso  $P_i$  a un Recurso  $R_j$  significa que el proceso  $i$  esta haciendo una solicitud por el recurso  $R_j$ .

Una arista del recurso  $R_j$  al proceso  $P_i$ , significa que el recurso esta asignado al proceso.

Si un RAG no tiene ciclos el sistema no esta en DEADLOCK, sis si los tiene no se puede afirmar nada.

*Mecanismos para tratar con Deadlocks*

Prevención de Deadlocks

La estrategia consiste en anular alguna de las cuatro condiciones necesarias para que se produzca un Deadlock.

1. No puede ser anulada porque existen recursos que deben ser usados en modalidad exclusiva.
2. La alternativa sería hacer que todos los procesos solicitaran todos los recursos que habrán de utilizar antes de utilizarlos al momento de su ejecución lo cual sería muy ineficiente.
3. Para anular esta condición cuando un proceso solicita un recurso y este es negado el proceso deberá liberar sus recursos y solicitarlos nuevamente con los recursos adicionales. El problema es que hay recursos que no pueden ser interrumpidos.
4. Espera Circular: esta estrategia consiste en que el sistema operativo numere en forma exclusiva los recursos y obligue a los procesos a solicitar recursos en forma ascendente. El problema de esto es que quita posibilidades a la aplicación.

Evasión de Deadlocks

Si se tiene cuidado al en la forma de asignar los recursos se pueden evitar situaciones de Deadlock. Supongamos un ambiente en el que todos los procesos declaren a priori la cantidad máxima de recursos que habrá de usar.

Estado Seguro: un estado es seguro si se pueden asignar recursos a cada proceso (hasta su máximo) en algún orden sin que se genere Deadlock. El estado es seguro si existe un ordenamiento de un conjunto de procesos  $\{P1...Pn\}$  tal que para cada  $P_i$  los recursos que  $P_i$  podrá utilizar pueden ser otorgados por los recursos disponibles mas los recursos utilizados por los procesos  $P_j, j < i$ . Si los recursos solicitados por  $P_i$  no pueden ser otorgados,  $P_i$  espera a que todos los procesos  $P_j$  hayan terminado.

## Algoritmo del banquero de Dijkstra

Asigna peticiones de recursos siempre que las mismas den como resultado estados seguros. Solicitudes que den como resultado estados inseguros serán negadas hasta que puedan ser satisfechas. Este algoritmo evita situaciones de Deadlock asignando los recursos en forma correcta.

Las debilidades del algoritmo son: que requiere que la cantidad de recursos del sistema sea constante, requiere una cantidad de procesos constante y requiere que los procesos garanticen que los recursos van a ser devueltos en un intervalo finito de tiempo.

### Detección y Recuperación de Deadlocks

#### Algoritmos de Detección de Deadlock

1. Cuando hay una única ocurrencia de cada recurso. (variante del grafo de "wait for graph"). Consiste en reducir el grafo, retirando las aristas que van de recursos a procesos y de procesos a recursos, colocando nuevas aristas que reflejan la situación de espera entre procesos. Si el grafo reducido tiene ciclos el sistema está en Deadlock. Orden de operaciones =  $n^2$ ,  $n$  = cantidad de aristas.
2. Cuando hay múltiples ocurrencias de cada recurso

#### Procedure detecta\_deadlock

var Disponible:array[1..n] of integer *## de recursos*

Usados:array[1..n] of integer

Solicitados:array[1..n] of integer

Finalizado:array[1..n] of boolean

Begin

Work:=Disponibles;

For i:=1..n do if(usados[i] > 0) then finish[i]:=false

Else finish[i]:=true;

While(encontrar\_indice\_i = true) do {

Work:=work + usados[i];

Finish[i]:=true; }

If ((finish[i] = false) para algun i),  $1 \leq i \leq n$  then El sistema está en Deadlock.

End

Function encontrar\_indice\_i : Boolean

Begin

If (existe i tal que (Finish[i]=false && solicitados[i] <= work)

Then -> true

Else -> false

End

#### Recuperación ante Deadlocks

1. Cancelación de procesos
  - a. Cancelación de todos los procesos involucrados. Esto resuelve la situación de Deadlock pero tiene un costo muy alto de reprocesamiento.
  - b. Cancelación de un proceso por vez hasta resolver la situación de Deadlock. La ventaja de esto es que el costo de reprocesamiento de la información es menor pero cada vez que se cancela un proceso debe ejecutarse el algoritmo de detección de deadlock. Los criterios para elegir el candidato a ser cancelado son: por prioridad, por tiempo de uso de CPU, por cantidad de recursos utilizados y por cuantos recursos adicionales habrá de utilizar.

2. Obtención de recursos (resource Preemption). El sistema operativo selecciona un proceso y le quita los recursos otorgados. Los criterios para seleccionar el candidato son los mismos que para la cancelación. El proceso seleccionado se le quitan los recursos y se le lleva a un estado consistente (Rollback).

### **3.2.4 Problemas clásicos de comunicación entre procesos.**

- a) El problema de la cena de los filósofos.
- b) El problema de los lectores y los escritores
- c) El problema del barbero dormilón.

### **3.3 Gestión de procesos.**

Los servicios del sistema operativo para la gestión de procesos se refieren a las llamadas al sistema operativo y que son realizadas en tiempo de ejecución. Dichas llamadas pueden ser invocadas por procesos de usuario o bien directamente, mediante llamadas al supervisor incorporadas al código del usuario, o bien indirectamente, mediante ordenes tecleadas en el terminal y traducidas a llamadas al sistema operativo por la rutina monitor de consola.

Las llamadas típicas al sistema operativo para gestión de procesos son:

**Crear, eliminar, dividir/unir, abortar, suspender, reanudar, retardar, leer\_atributos y modificar\_prioridad.**

### **3.4 Planificación de procesos**

En épocas pasadas de los sistemas de procesamiento por lotes, con una entrada en forma de imágenes de tarjetas en una cinta magnética, el algoritmo de planificación era sencillo: sólo había que ejecutar el siguiente trabajo en la cinta. En los sistemas de multiusuario de tiempo compartido, que se combinaban con un fondo de trabajos procesados en lote, el algoritmo era más complejo. En forma invariable, existían varios usuarios en espera de servicio y podían existir también otros trabajos para ser procesados en lote. Incluso en los sistemas puros de tiempo compartido existen con frecuencia los trabajos colaterales, como el sistema de correo electrónico, que a menudo se ejecuta todo el tiempo para enviar o recibir correo o noticias.

Cuando más de un proceso es ejecutable, el Sistema Operativo debe decidir cual de ellos deberá ejecutarse primero. Hay que tener una planificación de los procesos que quieren ejecutarse en el sistema. La planificación es una función primordial del Sistema Operativo. La mayoría de los recursos, si no es que todos, se planifican antes de que se utilicen.

La asignación de procesadores físicos a los procesos hace posible que éstos realicen su trabajo, y tal asignación es un problema complejo manejado por el Sistema Operativo.

En este trabajo se describe el concepto de planificación, así como sus principales objetivos.

### **CONCEPTO DE PLANIFICACIÓN**

La planificación es una función fundamental del Sistema Operativo. La mayor parte de los recursos se planifican antes de que se usen. Desde luego, la UCP es uno de los principales recursos de la computadora, por lo que su planificación es importante para el diseño de los Sistemas Operativos.

Se conoce como planificación al estudio de los problemas: ¿cuándo asignar procesadores? y ¿a cuáles procesos asignarlos?.

La planificación se refiere a un conjunto de políticas y mecanismos que poseen los actuales Sistemas Operativos y por los que se rige el orden en que se completa el trabajo que hay que realizar por los procesos.

En Windows NT los subprocesos tienen asignadas prioridades. El rango de prioridades va de 1 a 31, siendo 31 la prioridad mayor. La UCP trata de manera idéntica a todos los subprocesos de la misma prioridad. Esto es simplemente asigna el primer subproceso de prioridad 31 a una UCP, y una vez que su intervalo de tiempo ha finalizado asigna el siguiente subproceso de prioridad 31 a la UCP. Cuando todos los subprocesos de prioridad 31 han tenido un intervalo de tiempo de la UCP, el kernel de Windows NT asigna de nuevo el primer subproceso de prioridad 31 a la UCP. Como se ve, si siempre existe un subproceso de prioridad inferior de 31 nunca se ejecutará.

## OBJETIVOS DE LA PLANIFICACIÓN

El objetivo principal de la planificación es optimizar el rendimiento del sistema y proporcionar un buen servicio a todos los procesos que se encuentren en él.

En general, la planificación trata de cubrir los siguientes objetivos:

- Justicia. La planificación debe ser lo más justa posible con todos los procesos, sin favorecer a unos y perjudicar a otros.
- Máxima capacidad de ejecución. Debe dar un servicio eficiente para que todos los trabajos se realicen lo más rápidamente posible. Esto se puede lograr disminuyendo el número de cambios de proceso.
- Máximo número de usuarios interactivos. En los sistemas de tiempo compartido se tratará de que puedan estar trabajando el mayor número de usuarios al mismo tiempo.
- Predecibilidad. La planificación debe realizarse de tal forma que en todo momento pueda saberse como será su ejecución.
- Minimización de la sobrecarga. La computadora debe tener poca sobrecarga ya que ésta afecta directamente al rendimiento final del sistema: a menor sobrecarga mayor velocidad de proceso.
- Equilibrio en el uso de recursos. Para obtener un mayor rendimiento en el uso de los recursos y que estos estén ocupados equitativamente el mayor tiempo posible.
- Seguridad de las prioridades. Si un proceso tiene mayor prioridad que otro, éste debe ejecutarse más rápidamente.

- Evitar la postergación indefinida. Esto se logra aumentando la prioridad de un proceso mientras espere por un recurso. La prioridad llegará a ser tan alta que al proceso le será asignado el recurso que pidió.

### 3.4.1 Algoritmos de planificación

#### Planificadores

El planificador o scheduler es un componente del Sistema Operativo que se encarga de elegir la tarea siguiente que hay que admitir en el sistema y el proceso siguiente que hay que ejecutar.

La finalidad del planificador es asignar procesos para que sean ejecutados por el procesador o procesadores con el fin de obtener mejores tiempos de respuesta, mayor productividad o rendimiento y eficiencia del procesador. En la mayoría de los sistemas esta actividad de planificación se realiza en tres tipos de planificadores que se explican a continuación.

En UNÍX, las tareas de planificación las resuelve el scheduler (planificador) mediante un mecanismo de prioridades. Cada proceso tiene asignada una prioridad. Las prioridades de los procesos de los usuarios son siempre menores que la prioridad más pequeña de un proceso del sistema.

El componente que se encarga de administrar los procesos de Windows 95 trabaja con dos modelos de Multitarea:

- Multitarea Cooperativa
- La Multitarea con Derecho Preferente

Mediante el método de Multitarea Cooperativa, el Sistema Operativo deja que sea el proceso quien compruebe la cola de procesos que quieren ejecutarse y que cada proceso, deje tiempo de UCP al resto de aplicaciones que se están ejecutando. Puede ocurrir que el programa que esta en ejecución, no compruebe periódicamente la cola y se apodere completamente de la UCP, con lo cual ningún proceso responderá hasta que termine el que está en ejecución.

En la Multitarea con Derecho Preferente, es el Planificador de Procesos quien quita o da el control a un proceso y además determina el tiempo que dicho proceso va a ocupar la UCP. El cambio de un proceso a otro, generalmente está dado a un suceso o evento (un clic del ratón, minimizar una ventana). En la Multitarea con Derecho, el Planificador asigna unas prioridades a los procesos que determinan la forma de ejecución, sin embargo estas prioridades varían según avanza la ejecución, ya que, en caso contrario, cuando un proceso de máxima prioridad tomase la UCP nunca la soltaría.

Windows 95 solo utiliza la Multitarea Cooperativa, con las aplicaciones de 16 bits. Todas las aplicaciones de 32 bits se ejecutan en Multitarea con Derecho Preferente.

#### Planificador a largo plazo (Planificador de trabajos)

Cada vez que un usuario inicia la ejecución de un programa, el planificador a largo plazo recibe la orden de ejecución, crea el proceso correspondiente y lo pasa al planificador a corto plazo, colocándose en la cola de procesos listos. Este planificador es, por tanto, el encargado de crear los procesos. Cada vez que termina un trabajo, el planificador tomara la decisión de añadir uno o mas trabajos nuevos.

### **Planificador a mediano plazo (planificador de intercambio)**

Este planificador decide si un proceso que esta en estado bloqueado o suspendido debe ser retirado de la memoria temporalmente. Posteriormente, cuando el sistema se encuentra mas descargado, devolverá dicho proceso a la memoria y al estado de ejecución. Esta técnica se conoce con el nombre de intercambio (swapping). Este nivel, por tanto, gestiona los procesos suspendidos en espera de algún recurso no disponible en el momento de la suspensión.

### **Planificador a corto plazo (Planificador del procesador)**

El planificador acorto plazo también conocido como despachador, es el encargado de decidir como y cuando tendrá acceso al procesador un proceso que esta listo para utilizarlo. Solo considera los procesos que están en memoria. Ha de ser rápido y eficiente ya que se ejecuta muy a menudo. Este nivel, es donde se debe de dar un buen servicio a los procesos para que el usuario no perciba, o lo haga un pequeño grado, que esta compitiendo por el procesador junto con otros usuarios.

### **Planificación de Procesos**

Cuando más de un proceso es ejecutable, el Sistema Operativo debe decidir cual de ellos debe ejecutarse en primer termino. Esa parte del sistema operativo que debe de llevar a cabo esa decisión se llama planificador y el algoritmo que utiliza se llama algoritmo de planificación.

El planificador intenta conseguir con su administración de procesos lo siguiente:

**Equidad:** Garantizar que cada proceso obtenga su proporción justa de la CPU. Es decir, que ningún proceso llegue a apoderarse por completo de la CPU.

**Eficiencia:** Mantener ocupada la CPU al 100 %, esto con el fin de evitar los tiempos ociosos que pueda tener el CPU.

**Rendimiento:** Maximizar el número de tareas procesadas por hora, es decir, que el CPU pueda atender todos y cada una de las peticiones que le fueron hechas.

Para garantizar que ningún proceso se ejecute un tiempo excesivo, casi todas las computadoras tienen un cronómetro electrónico o un reloj incluido, que provoca una interrupción en forma periódica. En cada interrupción del reloj, el sistema operativo logra ejecutarse y decidir si el proceso que se ejecuta en ese momento tiene permiso de continuar o si tiene el tiempo suficiente en la CPU por el momento, para después suspenderlo para que otro proceso utilice la CPU.

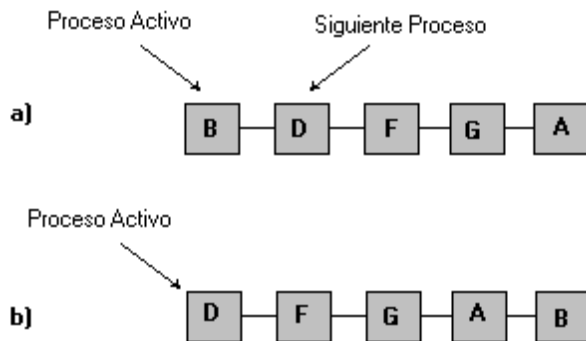
Algunos algoritmos específicos de planificación .

### **Planificación Round Robin**

Uno de los más antiguos, sencillo, justo y de uso más amplio es el round robin. Cada proceso tiene asignado un intervalo de tiempo de ejecución, llamado su quantum. Si el proceso en ejecución al final



de su quantum, otro proceso se apropia de la CPU. Si el proceso esta bloqueado o a terminado antes de consumir su quantum, se altera el uso de la CPU. El Round Robin es muy fácil de implantar. Todo lo que necesita el planificador es mantener una lista de procesos ejecutables. Cuando el quantum de un proceso se consume, se le coloca al final de la lista, como se muestra a continuación.



- Planificación Round Robin.

(a) La lista de procesos ejecutables.

(b) La lista de procesos ejecutables después de agotarse el quantum B.

El único aspecto interesante del round robin es la longitud del quantum. La alternancia entre un proceso y otro necesita cierta cantidad de tiempo para administración.

- Planificación por prioridades

La idea fundamental es directa: cada proceso tiene asociada una prioridad y el proceso ejecutable con máxima prioridad es el que tiene el permiso de ejecución.

Para evitar que los procesos de alta prioridad se ejecuten en forma indefinida, el planificador puede disminuir la prioridad del proceso en ejecución en cada interrupción de reloj. Si la prioridad del siguiente proceso con alta prioridad es mayor, se alternan los procesos.

Las prioridades se pueden asignar en forma estática o dinámica.

## ALGORITMO DE PLANIFICACIÓN FIFO (First Input – First Output)

Este algoritmo a su vez es llamado First Come - First Served (Primero en llegar- Primero en servir).

Tal vez la disciplina más simple de planificación sea la de primeras entradas–primeras salidas (PEPS). Los procesos se despachan de acuerdo con su tiempo de llegada a la cola de procesos listos. Cuando un proceso tiene la UCP, se ejecuta hasta terminar. Es justo en el sentido formal, pero algo injusta en

cuanto a que los trabajos largos hacen esperar a los cortos y los trabajos sin importancia hacen esperar a los importantes. FIFO ofrece variaciones relativamente pequeñas en los tiempos de respuesta y por lo tanto es más predecible que los otros esquemas. No es útil en la planificación para los usuarios interactivos porque no puede garantizar buenos tiempos de respuesta.

El esquema FIFO rara vez se usa como esquema principal en los sistemas actuales, pero a menudo está incorporado en otros sistemas. Por ejemplo, muchos esquemas de planificación despachan los procesos de acuerdo con la prioridad, pero los procesos con la misma prioridad se despachan de acuerdo con el esquema FIFO.

Este es un algoritmo que no usa apropiación, y que consiste en atender a los procesos por estricto orden de llegada a la lista de procesos listos. Cada procesos se ejecuta hasta que termina, o hasta que hace una llamada bloqueante (de E/S). Se trata de una política muy simple y sencilla de llevar a la práctica, pero muy pobre en cuanto a su comportamiento.

Las características principales de este algoritmo son las siguientes:

- No es apropiativa
- Es justa, aunque los procesos largos hacen esperar mucho a los cortos.
- Es una política predecible
- El tiempo promedio de servicio es muy variable ya que está en función del número de procesos y la duración promedio que tenga

| Proceso | Tiempo de UCP |
|---------|---------------|
| P1      | 24            |
| P2      | 3             |
| P3      | 3             |

Media del tiempo de espera:

Caso 1)  $(0 + 24 + 27) / 3 = 17$

Caso 2)  $(6 + 0 + 3) / 3 = 3$

En este esquema se tienen tres procesos (P1, P2, P3) listos para ejecutarse, con un tiempo de ejecución de 24, 3 y 3 unidades de tiempo (para facilidad tomaremos milisegundos como unidad de tiempo) respectivamente. Los procesos se ejecutan en ese mismo orden. El primer proceso se ejecuta de inmediato y no espera nada.

El segundo proceso espera todo lo que dura en ejecutarse el primer proceso que son 24 milisegundos. Por último el tercer proceso esperará la suma de lo que duran en ejecutarse los dos procesos anteriores, o sea, 27 segundos. Todo esto da como resultado un tiempo promedio de espera de 17 milisegundos. Si en cambio, el orden en que se ejecuten los procesos es como el caso 2), entonces el proceso P2 se ejecutaría de inmediato, P3 esperaría lo que tarde en ejecutarse P2 (3 milisegundos) y P1 esperaría lo que duran los dos procesos anteriores, obteniendo un tiempo promedio de espera de 3 milisegundos.

Como puede verse el tiempo promedio de espera que tengan los procesos depende por completo del orden en que llegan los procesos a ejecutarse.

## EL TRABAJO MAS CORTO PRIMERO

Llamado también SJF (Shortest Job First). Este algoritmo de planificación reduce la preferencia en favor de los procesos más largos dando más importancia a los procesos cortos, como se muestra en la figura.

Esta política funciona otorgándole el procesador al proceso más corto. En general, esta política mejora el funcionamiento del sistema en términos del tiempo de respuesta, pero gran cambia demasiado cuando se trata de procesos largos.

|  | PROCESO | DURACIÓN |  |
|--|---------|----------|--|
|  | P1      | 7        |  |
|  | P2      | 4        |  |
|  | P3      | 1        |  |
|  | P4      | 4        |  |

|       |    |
|-------|----|
| P3    | 0  |
| P2    | 1  |
| P4    | 5  |
| P1    | 9  |
| Total | 16 |

Media del tiempo de espera:  $(9+1+0+5)/4=3.75$

En la figura anterior tenemos cuatro procesos (P1, P2, P3, y P4) que quieren ejecutarse. Los procesos tienen una duración de 7, 4, 1 y 4 milisegundos respectivamente. Como P3 es el proceso más corto, será el primero ejecutarse. Como P2 y P4 tienen la misma duración, el que se ejecutará será el que llegó primero, en este caso será P2 el que se ejecute primero y después P4. Por último se ejecutará el proceso P1. Cada uno de los procesos esperará lo que duren en ejecutarse anteriores a él, con excepción del primer proceso. El tiempo promedio de espera es de 3.75 milisegundos, lo que representa un promedio de espera aceptable.

Una de las dificultades con este método es que debe saber el tiempo que cada proceso va a utilizar el procesador para realizar su trabajo, lo cual es una tarea difícil, pero es posible a través de diversos métodos como pueden ser la información dada por el propio usuario, por el propio programa.

Las características de este algoritmo son las siguientes:

- El tiempo de espera aumenta de acuerdo con el tamaño de los procesos, pero el tiempo promedio de espera con respecto a otros algoritmos es óptimo.
- Es poco predecible.
- No es justo con los procesos largos.
- Tiene buen tiempo de servicio.
- Resulta difícil de llevarse a la práctica por los datos que necesita para realizarse la planificación.

Su característica principal es que cuando se activa el planificador, éste elige el proceso de menor duración. Es decir, introduce una noción de prioridad entre procesos.

La ventaja que presenta este algoritmo sobre el algoritmo FIFO es que minimiza el tiempo de finalización promedio, como puede verse en el siguiente ejemplo:

Ejemplo: Supongamos que en un momento dado existen tres procesos listos R1, R2 y R3, sus tiempos de ejecución respectivos son 24, 3 y 3 ms. El proceso R1 es la que lleva más tiempo ejecutable, seguido del proceso al que pertenece R2 y del de R3. Veamos el tiempo medio de finalización (F) de los procesos aplicando FIFO y SJF:

$$\text{SJF } F = (3 + 6 + 30) / 3 = 13 \text{ ms.}$$

$$\text{FIFO } F = (24 + 27 + 30) / 3 = 27 \text{ ms.}$$

Se puede demostrar que este algoritmo es el óptimo. Para ello, consideremos el caso de cuatro procesos, con tiempos de ejecución de a, b, c y d. El primer proceso termina en el tiempo a, la segunda termina en el tiempo a+b, etc. El tiempo promedio de finalización es  $(4a+3b+2c+d)/4$ . Es evidente que a contribuye más al promedio que los demás tiempos, por lo que debe ser el proceso más corto, b la siguiente, y así sucesivamente. El mismo razonamiento se aplica a un número arbitrario de procesos.

No obstante, este algoritmo sólo es óptimo cuando se tienen simultáneamente todos los procesos. Como contraejemplo, considérense cinco procesos desde A hasta E, con tiempo de ejecución de 2, 4, 1, 1 y 1 respectivamente. Sus tiempos de llegada son 0, 0, 3, 3 y 3. Primero se dispone de A y B, puesto que los demás procesos no han llegado aún. Con el algoritmo SJF las ejecutaríamos en orden A, B, C, D, y E con un tiempo de finalización promedio de 4.6. Sin embargo, al ejecutarlas en orden B, C, D, E y A se tiene un promedio de finalización de 4.4.

## PLANIFICACION POR PRIORIDAD

A cada proceso se le asigna una prioridad, de manera que el procesador se le asignará al proceso que tenga la mayor prioridad.

Las prioridades pueden ser definidas interna o externamente. En el primer caso, el Sistema Operativo se basa en una serie de informaciones medibles para el cálculo y asignación de dichas prioridades (tiempo necesitado del procesador, necesidad de memoria, etc.). En las externas la asignación de prioridades se basa en la categoría que tenga el usuario.

| PROCESO | PRIORIDAD | TIEMPO UCP |
|---------|-----------|------------|
| P1      | 2         | 7          |
| P2      | 4         | 4          |
| P3      | 1         | 1          |
| P4      | 3         | 4          |

Media del tiempo de espera  $(1 + 12 + 0 + 8)/4 = 5.25$

El ejemplo de la figura cuenta con los mismos datos que el de la figura, solo que ahora se le agrego la prioridad de cada proceso que será de 2, 4, 1, 3, respectivamente (el número menor indica la mayor prioridad). Como este algoritmo se basa en las prioridades de los procesos, el primero en ejecutarse será el proceso p3 por ser el de mayor prioridad.

Después vendrán a ejecutarse los procesos P1, P4 y p2 en ese orden. Este orden de ejecución de los procesos nos da un tiempo promedio de espera de 5.25 milisegundos. Como podrá observarse el tiempo de espera es mayor que el del algoritmo anterior, ocasionado por darle mas importancia a los procesos que tiene mayor prioridad.

El principal problema de este algoritmo es el bloqueo o postergación indefinida, ya que un proceso de baja prioridad puede estar esperando su turno al grado de no ejecutarse nunca. Para evitar esta postergación se aumenta poco a poco las prioridades de los procesos que están, a la espera de utilizar el procesador.

Cualquier algoritmo que se base en esta política puede ser apropiativo o no apropiativo. En el primer caso, un proceso puede ser retirado del procesador si aparece otro de mayor prioridad.

El algoritmo SJF es un caso especial de planificación por prioridad, donde la prioridad es el tamaño del proceso.

## ROUND-ROBIN (RR)

Volviendo a FCFS, una forma de mejorarlo es agregando apropiación de tal forma que cada proceso no retenga el procesador por mas de un quantum o periodo de tiempo predefinido. Consiste en darle a cada proceso en ejecución un cierto periodo de tiempo, y una vez finalizado el tiempo, si el proceso no ha terminado, es colocado al final de la cola de procesos listo, otorgándose el procesador al siguiente proceso. Trata de ser mas justo en cuanto a tiempos de respuesta tanto de los procesos cortos como de los procesos largos.

El punto interesante es encontrar el quantum adecuado. Si es muy grande, puede caer en un algoritmo FCFS, pero tampoco puede ser demasiado pequeño, porque entonces el costo de cambiar de un proceso a otro es muy alto. Los valores del quantum varían entre 10 y 100 milisegundos, siendo recomendable que el 80% de los tiempos de respuesta de los procesos sean menores que el quantum.

| PROCESO | DURACION |
|---------|----------|
| P1      | 7        |
| P2      | 4        |
| P3      | 1        |
| P4      | 4        |

Media del tiempo de espera:  $(9 + 7 + 4 + 9)/4 = 7.25$

Ahora explicaremos el ejemplo de la figura 64. Como primer punto establecemos el quantum que tendrán los procesos para ejecutarse que será de 2 milisegundos. Primero se ejecuta el proceso P1 durante los 2 milisegundos que están permitidos. Como este tiempo no es suficiente para que se ejecute por completo, pasa al final de la lista de procesos para su posterior ejecución. Luego viene el proceso P2, al cual le sucede lo mismo que al proceso anterior, por lo que se va al final de la lista.

Ahora le toca el turno al proceso P3, pero a éste si le alcanza el tiempo para ejecutarse totalmente. Posteriormente se ejecuta P4 obteniendo los mismos resultados que los dos primeros procesos. Ahora el turno le corresponde de nuevo al primer proceso que se ejecutó. De nuevo le sucede lo mismo que la primera vez que se ejecutó. Le corresponde ahora el turno al segundo proceso, el cual, en esta ocasión ya se ejecuta por completo. Lo mismo le pasa al siguiente proceso que se ejecuta (P2).

Otra vez le toca ejecutarse al proceso P1 y de nuevo no se puede ejecutar por completo. Como ahora es el único proceso que queda en el sistema, se vuelve a ejecutar de nuevo, terminando esta vez su ejecución.

Las características de este algoritmo de planificación son:

- Baja sobrecarga si el cambio entre un proceso y otro es eficiente y los procesos siempre están en la memoria principal
- El tamaño óptimo del quantum depende de:
  - El tipo de sistema.
  - Las cargas que vaya a soportar el sistema.
  - El número de procesos en el sistema y su tipo.
- Es la política mas usada para tiempo compartido.
- Ofrece un servicio igual para todos los procesos.
- Es una política apropiativa.
- Mantiene más equilibradas las colas de procesos listos y bloqueados.

### 3.4.2 Evaluación del rendimiento

El rendimiento en los SO es dependiente de su efectividad en la asignación de recursos del sistema. En general un proceso activo requiere para ejecutarse memoria principal, dispositivos de E/S, almacenamiento secundario, y el CPU simultáneamente. La misión de los planificadores es asignar los recursos de manera que sea justa, que optimice el criterio de rendimiento elegido y que este libre de complicaciones en tiempo de ejecución tales como interbloqueos. Lo anterior es complicado porque la carga presentada a los sistemas reales es generalmente impredecible. En la práctica, los planificadores son implementados utilizando heurísticas que operan sobre propiedades generalmente conocidas de disciplinas de servicio particulares.

La importancia de la asignación de recursos y la naturaleza inexacta de los planificadores prácticos hace de la evaluación del rendimiento una herramienta importante para determinar la efectividad de los sistemas existentes y para estimar el comportamiento de nuevos sistemas mientras estén siendo diseñados.

Las herramientas de evaluación de los SO son las simulaciones y los modelos analíticos.

#### Algoritmos de Scheduling

##### FCFS (First Come First Served)

Cuando un proceso llega a la cola de ready su PCB es agregado al final de la lista. El uso de la CPU es otorgado al primero de la lista y una vez que un proceso comienza a ejecutar no deja de hacerlo hasta que se termina. El tiempo medio de espera para este algoritmo suele ser bastante alto.

##### SJF (Shortest Job First)

Una vez que un proceso ejecuta no deja de hacerlo hasta que voluntariamente cambia de estado (no hay interrupción por tiempo). Asocia a cada proceso el tiempo de CPU que habrá de usar en su próxima vuelta y va a decidir por el más pequeño. Si hubiera mas de uno utiliza FCFS para desempatar. El mayor problema de este algoritmo radica en el cálculo de tiempo de uso de CPU. Este se puede aproximar a:

$$T_{n+1} = t_n + (1 - \alpha) T_n$$

$0 < \alpha < 1$  Tiempo calculado en la vuelta  $n$

Próximo uso de CPU Tiempo usado en la vuelta  $n$

El problema de este algoritmo es que el tiempo de espera para los procesos largos puede ser demasiado largo. Constantemente se están entregando los procesos mas cortos y el más grande nunca será ejecutado.

El FJS se puede subdividir en 2 tipos de algoritmos: PREEMPTIVO o NO PREEMPTIVO

Preemptivo significa que si mientras un proceso se esta ejecutando, entra a la cola de ready un proceso mas corto, el proceso en la cola de ready se apropia de la CPU y comienza su ejecución.

#### Priority Scheduling

Asocia a cada proceso una prioridad. Luego selecciona el proceso con mas prioridad para desempatar. En caso de que hubieran dos o mas procesos con la misma prioridad, se usa FCFS para desempatar. Hay dos tipo de prioridad en los procesos: la *prioridad externa* definidas a través del sistema operativo y la *prioridad interna* definida por el tiempo de uso de la CPU, el control de E/S, etc. Este algoritmo también se puede ejecutar de dos maneras: preemptivo y no preemptivo. El algoritmo soluciona el problema del looping pero el problema es que los procesos con prioridad muy baja tienen chance de no ejecutarse nunca. Para solucionar este problema de espera infinita el envejecimiento de un proceso eleva su prioridad.

### 3.4.3 Comunicación y sincronización de procesos

La sincronización entre procesos se realiza mediante un conjunto de protocolos y mecanismos utilizados para preservar la integridad y consistencia del sistema cuando varios procesos concurrentes comparten recursos que son reutilizables en serie

#### **4. Administración de la memoria**

Un recurso importante a administrar por un sistema operativo, es la memoria y la parte que la administra se denomina administrador de la memoria. Su labor consiste en llevar un registro de las partes de memoria que se estén utilizando y aquellas que no, con el fin de asignar espacio en memoria a los procesos cuando estos la necesiten y liberar espacio cuando terminen, así como administrar el intercambio entre la memoria principal y el disco, en los casos en que la memoria no pueda albergar a todos los procesos.

##### *Utilización de la memoria.*

Los programas residen en disco como ejecutables binarios y deben ser traídos a memoria para su ejecución. El conjunto de programas residentes en disco esperando para ser cargados en memoria forman la cola de entrada.

Los programas pueden direccionar memoria en varias formas:

1. *Direccionamiento al momento de la compilación:* se genera código absoluto (es decir que siempre ejecuta en una dirección de memoria de inicio fija, si esta ocupada no ejecuta)
2. *Direccionamiento al momento de la carga:* maneja direcciones relativas al momento de carga del programa (no genera código absoluto). Se generan direcciones relativas a la posición inicial de memoria (desplazamiento). Genera código reubicable.
3. *Dinamic Linking:* se difiere la link edición hasta el momento de ejecución se usa normalmente con bibliotecas del sistema. Al momento de la link edición (generación del ejecutable) se entrega un trozo de código indicando la referencia a la rutina que se debe cargar.
4. *Overlays:* Esta técnica permite ejecutar un programa en un espacio de memoria menor que su tamaño. Mantiene en memoria toda aquella parte del programa que es utilizada en un instante determinado. Las rutinas independientes son cargadas a medida que se necesitan ejecutar (son partes del programa que se dividen en forma independiente).

#### **4.1 Administración de la memoria sin intercambio o paginación.**

Los sistemas de administración de memoria se pueden clasificar en dos tipos. *Los que desplazan los procesos de la memoria principal al disco y viceversa durante la ejecución (intercambio y paginación) y aquellos que no.*

1. Monopogramación sin intercambio o paginación.

Es en forma secuencial pues solo se tiene un objeto en memoria en cada instante, el usuario carga toda la memoria con un programa, esto implica que cada proceso debe contener controladores de dispositivo para cada uno de los dispositivos E/S que utilice.



## 2. Multiprogramación y uso de la memoria.

La multiprogramación facilita la programación de una aplicación al dividirla en dos o mas procesos. La mayoría de los procesos tardan cierto tiempo en la espera de datos de dispositivos E/S.

Un modelo para el uso y aprovechamiento de la CPU es el modelo probabilístico dado por la fórmula :

$$\text{Uso de la CPU} = 1 - p^n$$

## 3. Multiprogramación con particiones fijas

El objetivo en todo esto es tener mas de un proceso en memoria a la vez, solución posible sería dividir la memoria en n partes al inicio de una sesión de uso de la máquina, pero aún así se obtiene el desperdicio de particiones grandes con una tarea pequeña, la respuesta puede ser tener particiones pequeñas también.

Las tareas que van llegando se forman hasta que una partición adecuada está disponible, en cuyo momento la tarea se carga en esa partición y se ejecuta hasta terminar.

### 4.2 Particiones e intercambio.

#### 4.2.1 Intercambio

En un sistema por lotes la organización de la memoria en particiones fijas es adecuado pero en un ambiente multiusuario la situación es distinta con el tiempo compartido, ya que existen mas usuarios de los que puede albergar la memoria, por lo que es conveniente albergar el exceso de los procesos en disco., por supuesto para ser ejecutados estos procesos deben ser trasladados a la memoria principal. Al traslado de procesos de disco a memoria y viceversa se le llama **intercambio**.

#### 4.2.2 Multiprogramación con particiones variables.

Mediante un algoritmo de administración de memoria las particiones variables varían de forma dinámica durante el uso de la máquina, evitando desperdicio de memoria

Otros métodos de administración de memoria que tenemos son:

la administración de memoria con mapa de bits

la memoria se divide en unidades de asignación, a cada asignación le corresponden un bit en el mapa de bits, un mapa de bits es una forma sencilla para llevar un registro de las palabras de la memoria en una cantidad fija de memoria.

la administración de memoria con listas ligadas

otra forma de mantener un registro en memoria es mediante una lista ligada donde cada entrada de la lista especifica un hueco o un proceso.

la administración de memoria con el sistema de los asociados

basado en el sistema binario o utiliza para las direcciones.

### ***Administración de Memoria Real.***

Asignación contigua de almacenamiento de un solo usuario.

- Un solo usuario a la vez
- Todos los recursos a disposición de un solo usuario
- El tamaño de los programas esta limitado por la cantidad de memoria, aunque se pueden usar overlays para sacar mayor provecho a esta.

Protección de Memoria: Se incorpora un registro de limite (limit register) a la cpu. Cada petición de acceso a la memoria realizada por el usuario se compara con el registro limite. Si la dirección a la que se quiere acceder es menor o igual al registro limite se cancela la solicitud.

Multiprogramación de partición fija.

La memoria se divide en particiones de tamaño fijo (puede ser distinto el tamaño de cada partición). Originalmente los programas se compilaban y link editaban para ejecutar en una partición en particular (direcciones absolutas). Posteriormente los compiladores y link editores generan código reubicable para que un programa pudiera ejecutar en cualquier partición de memoria suficientemente grande. Con esta estructura de administración de memoria se desperdicia memoria y tiempo de CPU (si hay un programa corriendo los demás quedan encolados aunque haya otra partición libre).

Multiprogramación de partición variable.

Cada programa o usuario utiliza tanta memoria como sea necesaria siempre que quepa en el almacenamiento real. Cuando los programas van terminando su ejecución se van generando agujeros en memoria.

Métodos de solución al problema de agujeros en memoria:

Combinación.

Cuando un programa termina su ejecución el sistema operativo verifica si hay algún bloque de memoria contigua libre (antes o después) en cuyo caso los combina para generar un único bloque de memoria libre de tamaño igual a la suma de ambos.

Compactación o Compresión.

Agrupar toda la memoria disponible en un único bloque de memoria al final.

Desventajas:

- Alto consumo de CPU (casi dedicada exclusivamente a esto)
- Mientras se ejecuta el algoritmo la actividad del sistema debe ser detenida.
- La movida de programas implica reubicación

En definitiva no es conveniente utilizar este método para solucionar el problema de los agujeros en memoria debido al alto costo.

### ***Estrategias de colocación en el almacenamiento.***

En cualquiera de estas estrategias cuando a una petición le es asignada una partición y sobra memoria el sistema operativo se da cuenta y deja a este espacio sobrante como una nueva partición.

Best Fit.-

Los bloques de memoria son asignados según las necesidades del programa que se esta ejecutando

First fit.-

Asignan la primera partición disponible en la que pueda entrar el programa. Esta estrategia es mas rápida pero se desperdicia mucho espacio de memoria.

Worst Fit.-

Consiste en asignar la partición en la que sobra mas espacio.

### ***4.3 Memoria Virtual.***

Las direcciones de memoria no están necesariamente contenidas en el almacenamiento real. El S.O. se apoya en el almacenamiento secundario para extender el uso de la memoria.

Las direcciones referidas por los procesos son direcciones virtuales. Los procesos no tienen por que saber en que posición de almacenamiento real o secundario están almacenados. Las direcciones de memoria que direcciones de memoria que referencian al almacenamiento primario (memoria real) se denominan "direcciones reales". Por mas que los procesos hagan referencia a direcciones virtuales, estos solo pueden ejecutar en el almacenamiento primario; por lo tanto las direcciones virtuales deben ser transformadas en direcciones reales mientras el proceso esta en ejecución.

Como el almacenamiento real es compartible por varios procesos, solo se mantiene al mismo tiempo una pequeña parte de cada uno de ellos en memoria real.

El mecanismo de traducción de direcciones deberá mantener mapas que ilustren que direcciones del almacenamiento virtual se encuentran en memoria real, y donde se encuentran.

La información se agrupa en bloques y el sistema operativo debe estar informado del lugar en que se encuentra almacenada cada página).

Cuando los bloques son del mismo tamaño se denominan *páginas* y el mecanismo de administración del almacenamiento virtual asociado se denomina *paginación*.

Si los bloques son de tamaño variable se denominan *segmentos* y el mecanismo de administración del almacenamiento virtual *segmentación*.

Las direcciones en un esquema de almacenamiento virtual son un par (b,d) donde b indica el bloque y d el desplazamiento desde el inicio del bloque.

#### Reemplazo de Página.

Si al momento de generarse un page fault no hubiera ningún frame disponible, se debe proceder a reemplazar una página de memoria. La rutina de page fault debe ser modificada para incluir reemplazo de páginas:

1. Encontrar un frame disponible.
2. Si hay un frame disponible usarlo
3. Si no hay frames disponibles utilizar algoritmo de reemplazo de páginas para seleccionar la víctima
4. Escribir la página víctima a disco y actualizar tabla de páginas .

#### 4.4 Algoritmos de reemplazo de página.

FIFO (First In First Out).

Asocia a cada página el tiempo en que fue cargada en memoria. Cuando debe reemplazar una página, se selecciona la que hace mas tiempo que esta en memoria. También se puede implementar mediante la utilización de una lista. Se reemplazan las páginas de la cabeza y se agregan al final.

LRU (Least Recently Used).-

Se selecciona la página que hace mas tiempo que no se utiliza. Asocia cada página el tiempo de la última vez que fue utilizada. En LRU se necesita llevar tiempo de última utilización de cada página:

*Contadores:* Se asocia a cada entrada en la tabla de páginas, el tiempo de la última vez que se uso. Cada vez que se accede a una página se actualiza el contador.

*Stack:* Se lleva un stack con los números de página cada vez que una página es accedida, se quita del stack y se quita al tope. En el tope del stack estará la página que fue utilizada por última vez, y en el fondo la página que hace mas tiempo que no se accede.

Algoritmo de Segunda Chance.

Es una variante del algoritmo FIFO. Cada vez que se selecciona una página, se inspecciona un bit de referencia. Si el valor es cero se procede a reemplazar la página. Si el valor es uno, se da una segunda oportunidad a la página y se selecciona la siguiente víctima por criterio FIFO. Cada vez que una página recibe una segunda oportunidad, se carga en el bit de referencia. Si una página vuelve a ser accedida, se incrementa el bit de referencia a uno.

LFU (Least Frequently Used).

Establece que la página con menos cantidad de referencias debe ser reemplazada. Una contra del algoritmo es que una página puede haber sido muy utilizada en la fase de carga de un proceso, pero no ser vuelta a utilizar, por lo cual se estaría manteniendo en memoria páginas que no son necesarias para la ejecución.

MFU (Most Frequently Used).

La página con mayor cantidad de referencias es la víctima a ser reemplazada. El argumento es que si una página fue utilizada pocas veces, se debe a que recién fue cargada en memoria y aún deba ser utilizada.

### Page Buffering.

Se lleva un pool de frames disponibles. Cuando ocurre un page fault se elige una víctima utilizando alguno de los mecanismos ya vistos (anteriores). En lugar de hacer el cambio de frame en una sola operación se carga la pagina en memoria en uno de los frames disponibles (pool) permitiendo que el proceso retome su ejecución. El S. O. Se encargará luego de descargar la página y agregarla al pool de frames disponibles (es decir que agrego a la lista de paginas del pool la página liberada en memoria)

### Trashing.

Cuando un proceso consume mas tiempo paginando que ejecutando se dice que el proceso esta en situación de trash. Una forma de evitar esta situación es utilizar un mecanismo de reemplazo de paginas "local". Cuando un proceso comienza a paginar, no puede robar frames de otros procesos solamente podrá reemplazar páginas de su propio proceso.

### Working-Set.

Es el conjunto de la páginas activas de un proceso que son almacenadas en memoria. Se define un número  $n$  por proceso. Las  $n$  páginas más activas del proceso son mantenidas en memoria.

PFF (Page Fault Frequency).-

Se controla el page fault rate de cada proceso. Si un proceso tiene page fault rate muy bajo probablemente se deba a que este utilizando demasiados frames en memoria. Si el page fault rate es muy alto, probablemente no tenga suficientes frames otorgados. La idea de esta técnica es definir limite superior e inferior para page fault.

## 4.4 Paginación.

Las direcciones en un sistema de paginación están dadas por:

Desplazamiento

Página

Las páginas se transfieren del almacenamiento secundario al primario en bloque llamados page frames (los cuales tienen el mismo tamaño de las páginas).

El sistema lleva una tabla de páginas para cada proceso. De esta forma se asegura la protección.

Cuando un proceso debe ser cargado en memoria par ejecutar se siguen determinados pasos:

1. Se calcula su tamaño en páginas
2. Se carga el programa en memoria
3. se construye la tabla de paginas del proceso.

### Soporte de Hardware.

Como la tabla de páginas de cada proceso puede llegar a tornarse demasiado grande, se buscan mecanismos que permiten acelerar el tiempo de búsqueda en esa información. La solución mas simple es la utilización de un conjunto de registros dedicados (no usar tabla de paginación de cada proceso). El CPU dispatcher será el encargado de cargar y actualizar la información de estos registros con cada context switch. Esta solución sería razonable si se hablara de pocos registros pero en la realidad no se utiliza.

La solución mas común es apoyarse en un cache de acceso muy rápido llamado TLB (Translation Location Buffer). La TLB solo puede contener algunas entradas de la tabla de páginas. Con cada context switch la TLB debe ser variada para evitar que un proceso haga referencia a direcciones de

memoria utilizadas por otro proceso (protección). Si se hace referencia a una dirección que no estaba cargada en TLB, se debe cargar en esta, eliminando alguna otra dirección si fuera necesario.

Protección.

Por la forma en que se construyen las tablas de página los procesos solo van a hacer referencia a direcciones de su espacio virtual. La protección se implementa por un conjunto de bits que son asociados a cada frame. Un bit puede definir si una página es read-only o read-write y cada vez que se hace una operación sobre un page frame se verifica sobre estos bits.

Multilevel Paging.

En algunos sistemas se busca minimizar el tiempo de búsqueda en la tabla de páginas utilizando un sistema de paginación de 2 niveles.

Una dirección es de la forma: (P1,P2,d) donde P1 y P2 corresponde al número de página y d al desplazamiento.

Inverted Page Table.

Se lleva una tabla de páginas para todos los procesos. Cada entrada de página contiene la siguiente información: **<process-id, nro. Pagina, desplazamiento>**

Paginas Compartidas.

Es posible que los procesos compartan código común (reentrante) El código común no podrá ser modificado durante su ejecución, por lo cual se deberán proteger mediante los bits de protección.

#### 4.6 Segmentación.

En un esquema de segmentación un espacio de direcciones lógicas es un conjunto de segmentos. Cada segmento tendrá un nombre y un largo. Las direcciones van a hacer referencia tanto al nombre como al desplazamiento dentro del segmento.

**<nro.segmento, desplazamiento>**

Mecanismo de traducción de direcciones.

Se lleva una tabla de segmento por cada proceso, cada entrada a la tabla de segmento lleva la siguiente información: segment base (dirección base del segmento)  
segment limit (largo del segmento)

Protección.

La protección se asegura verificando cada acceso a la memoria con la tabla de segmentos para asegurar que se esta direccionando dentro del espacio de direcciones lógicas del proceso.

Además el mecanismo de traducción de direcciones asegura que no se direcciona fuera de un segmento en particular. Existen también bits de protección para cada entrada de la tabla de segmentos que indicaran si el segmento es read only o read-write.

Segmentos Compartidos.-

Similar a lo visto para paginación. Se comparte la totalidad de un segmento

## Fragmentación.

El sistema operativo deberá asignar memoria utilizando algunos algoritmos ya vistos (first fit, worst fit, best fit) Cuando el sistema operativo intenta cargar un segmento y no hay ningún segmento disponible de tamaño suficiente para almacenarlo se procederá a compactar la memoria.

### Paginación por demanda.-

Es similar a lo visto para la paginación introduciendo el concepto de swapping. Los procesos residen en el disco y al ser ejecutados deben ser cargados en memoria. Cuando un proceso va a ser ejecutado, el mismo es swapeado a memoria, utilizando lazy swapping. El lazy swapping nunca trae paginas a memoria si no van a ser ejecutadas. Se necesita determinar si un pagina esta en memoria o en disco, por lo cual se utiliza el bit de válido / inválido de la tabla de páginas. Si el bit = 1 la página es valida y esta cargada en memoria si es 0 la página es inválida y no esta cargada en memoria (esta en disco).

Cuando un proceso intenta acceder a una página que no esta cargada en memoria ocurre un *page fault* (tomo de página). El procedimiento para manejar un page fault es el siguiente:

1. Verificar si la referencia a la pagina es valida (se utiliza una tabla interna generalmente llevada en PCB donde se indica las paginas validas).
2. Si la referencia no es valida, se cancela la ejecución del proceso.
3. Encontrar un frame disponible para cargarla (la página esta en disco)(por ejemplo de la free frame list)
4. Solicitar operación de I/O para leer la página de disco cargarla en el frame obtenido.
5. Modificar la tabla interna y la tabla de paginas para que ahora esta pagina figure como que esta en memoria.
6. Continuar con la ejecución del proceso en la instrucción en la que fue interrumpido.

## Reemplazo de Página.

Si al momento de generarse un page fault no hubiera ningún frame disponible, se debe proceder a reemplazar una pagina de memoria. La rutina de page fault debe ser modificada para incluir reemplazo de páginas:

5. Encontrar un frame disponible.
6. Si hay un frame disponible usarlo
7. Si no hay frames disponibles utilizar algoritmo de reemplazo de paginas para seleccionar la víctima
8. Escribir la pagina víctima a disco y actualizar tabla de paginas .

## Prepaginado.

Es una estrategia para evitar el alto nivel de paginación inicial, consiste en cargar en memoria de una vez todas las páginas que el proceso necesite.

### Segmentación por demanda.-

Un proceso no necesita tener todos los segmentos cargados en memoria para ejecutar. Solamente se deben cargar en memoria aquellos que estén siendo activamente utilizados. La tabla de segmentos mantiene un bit de validez (idem paginación por demanda) para indicar si un segmento esta en memoria o en disco. Cada vez que se direcciona un segmento se chequea el bit de validez. Si el segmento no esta cargado en memoria se produce un segmento fault. Una vez detectada la necesidad de un segment fault, se utiliza la siguiente técnica:

1. Verificar si hay suficiente memoria libre para acomodar el segmento. Si no hay suficiente memoria libre, se puede llegar a utilizar compactación de memoria
2. Si luego de la compactación sigue sin haber espacio para acomodar el segmento se debe producir un reemplazo de segmento.
3. El segmento al final de la lista es seleccionado y es swapeado a disco.
4. Si el nuevo espacio libre es suficientemente grande para acomodar el segmento de carga. Si no se repite el paso 3. hasta obtener un espacio suficientemente grande.

## 5. Sistemas de archivos

Un sistema de archivos ( file system ) es una estructura de directorios con algún tipo de organización el cual nos permite almacenar, crear y borrar archivos en diferentes formatos. En esta sección se revisarán conceptos importantes relacionados a los sistemas de archivos.

### 5.1 Archivos

#### 5.1.2 Generalidades.

En un sistema de cómputo es evidente que existe la necesidad por parte de los usuarios y aplicaciones de almacenar datos en algún medio, a veces por periodos largos y a veces por instantes. cada aplicación y cada usuario debe tener ciertos derechos con sus datos, como son el poder crearlos y borrarlos, o cambiarlos de lugar; así como tener privacidad contra otros usuarios o aplicaciones. El subsistema de archivos del sistema operativo se debe encargar de estos detalles, además de establecer el formato físico en el cual almacenará los datos en discos duros, cintas o discos flexibles. Debe ser conocido por todos, que tradicionalmente la información en los sistemas modernos se almacena en discos duros, flexibles y unidades de disco óptico, y en todos ellos se comparten algunos esquemas básicos para darles formato físico: las superficies de almacenamiento son divididas en círculos concéntricos llamados "pistas" y cada pista se divide en "sectores". A la unión lógica de varias pistas a través de varias superficies "paralelas" de almacenamiento se les llama "cilindros", los cuales son inspeccionados al momento de lectura o escritura de datos por las respectivas unidades físicas llamadas "cabezas". Las superficies de almacenamiento reciben el nombre de "platos" y generalmente están en movimiento rotatorio para que las cabezas accedan a las pistas que los componen. Los datos se escriben a través de los sectores en las pistas y cilindros modificando las superficies por medio de las cabezas.

El tiempo que una cabeza se tarda en ir de una pista a otra se le llama "tiempo de búsqueda" y dependerá de la distancia entre la posición actual y la distancia a la pista buscada. El tiempo que tarda una cabeza en ir del sector actual al sector deseado se le llama tiempo de latencia y depende de la distancia entre sectores y la velocidad de rotación del disco. El impacto que tiene las lecturas y escrituras sobre el sistema está determinado por la tecnología usada en los platos y cabezas y por la forma de resolver las peticiones de lectura y escritura, es decir, los algoritmos de planificación.

#### 5.1.2 Atributos.

Cada archivo tiene su nombre y datos además de los siguientes atributos: tipo, tamaño, protección (información de control de acceso asociado), fecha y hora de creación, fecha y hora de modificación, usuario propietario, según el sistema operativo varían los atributos de archivos.



### 5.1.3. Algoritmos de planificación de peticiones

Los algoritmos de planificación de peticiones de lectura y escritura a discos se encargan de registrar dichas peticiones y de responderlas en un tiempo razonable. Los algoritmos más comunes para esta tarea son:

Primero en llegar, primero en ser servido ( FIFO ): Las peticiones son encoladas de acuerdo al orden en que llegaron y de esa misma forma se van leyendo o escribiendo las mismas. La ventaja de este algoritmo es su simplicidad y no causa sobrecarga, su desventaja principal es que no aprovecha para nada ninguna característica de las peticiones, de manera que es muy factible que el brazo del disco se mueva muy ineficientemente, ya que las peticiones pueden tener direcciones en el disco unas muy alejadas de otras. Por ejemplo, si se están haciendo peticiones a los sectores 6,10,8,21 y 4, las mismas serán resueltas en el mismo orden. \_ Primero el más cercano a la posición actual: En este algoritmo las peticiones se ordenan de acuerdo a la posición actual de la cabeza lectora, sirviendo primero a aquellas peticiones más cercanas y reduciendo, así, el movimiento del brazo, lo cual constituye la ventaja principal de este algoritmo. Su desventaja consiste en que puede haber solicitudes que se queden esperando para siempre, en el infortunado caso de que existan peticiones muy alejadas y en todo momento estén entrando peticiones que estén más cercanas. Para las peticiones 6,10,8,21 y 4, las mismas serán resueltas en el orden 4,6,8,10 y 21.

Por exploración ( algoritmo del elevador ): En este algoritmo el brazo se estará moviendo en todo momento desde el perímetro del disco hacia su centro y viceversa, resolviendo las peticiones que existan en la dirección que tenga en turno.

En este caso las peticiones 6,10,8,21 y 4 serán resueltas en el orden 6,10,21,8 y 4; es decir, la posición actual es 6 y como va hacia los sectores de mayor numeración (hacia el centro, por ejemplo), en el camino sigue el sector 10, luego el 21 y ese fue el más central, así que ahora el brazo resolverá las peticiones en su camino hacia afuera y la primera que se encuentra es la del sector 8 y luego la 4. La ventaja de este algoritmo es que el brazo se moverá mucho menos que en FIFO y evita la espera indefinida; su desventaja es que no es justo, ya que no sirve las peticiones en el orden en que llegaron, además de que las peticiones en los extremos interior y exterior tendrán un tiempo de respuesta un poco mayor.

Por exploración circular: Es una variación del algoritmo anterior, con la única diferencia que al llegar a la parte central, el brazo regresa al exterior sin resolver ninguna petición, lo cual proveerá un tiempo de respuesta más cercano al promedio para todas las peticiones, sin importar si están cercas del centro o del exterior.

### 5.1.4. Asignación del espacio de almacenamiento

El subsistema de archivos se debe encargar de localizar espacio libre en los medios de almacenamiento para guardar archivos y para después borrarlos, renombrarlos o agrandarlos. Para ello se vale de localidades especiales que contienen la lista de archivos creados y por cada archivo una serie de direcciones que contienen los datos de los mismos. Esas localidades especiales se llaman directorios. Para asignarle espacio a los archivos existen tres criterios generales que se describen enseguida.

Asignación contigua: Cada directorio contiene los nombres de archivos y la dirección del bloque inicial de cada archivo, así como el tamaño total de los mismos. Por ejemplo, si un archivo comienza en el sector 17 y mide 10 bloques, cuando el archivo sea accesado, el brazo se moverá inicialmente al bloque 17 y de ahí hasta el 27. Si el archivo es borrado y luego creado otro más pequeño, quedarán huecos inútiles entre archivos útiles, lo cual se llama fragmentación externa.

Asignación encadenada: Con este criterio los directorios contienen los nombres de archivos y por cada uno de ellos la dirección del bloque inicial que compone al archivo. Cuando un archivo es leído, el brazo va a esa dirección inicial y encuentra los datos iniciales junto con la dirección del siguiente bloque y así sucesivamente. Con este criterio no es necesario que los bloques estén contiguos y no existe la fragmentación externa, pero en cada "eslabón" de la cadena se desperdicia espacio con las direcciones mismas. En otras palabras, lo que se crea en el disco es una lista ligada.

Asignación con índices ( indexada ): En este esquema se guarda en el directorio un bloque de índices para cada archivo, con apuntadores hacia todos sus bloques constituyentes, de manera que el acceso directo se agiliza notablemente, a cambio de sacrificar varios bloques para almacenar dichos apuntadores. Cuando se quiere leer un archivo o cualquiera de sus partes, se hacen dos accesos: uno al bloque de índices y otro a la dirección deseada. Este es un esquema excelente para archivos grandes pero no para pequeños, porque la relación entre bloques destinados para índices respecto a los asignados para datos es incoesteable.

#### 5.1.5. Métodos de acceso en los sistemas de archivos

Los métodos de acceso se refiere a las capacidades que el subsistema de archivos provee para acceder datos dentro de los directorios y medios de almacenamiento en general. Se ubican tres formas generales: acceso secuencial, acceso directo y acceso directo indexado.

Acceso secuencial: Es el método más lento y consiste en recorrer los componentes de un archivo uno en uno hasta llegar al registro deseado. Se necesita que el orden lógico de los registros sea igual al orden físico en el medio de almacenamiento. Este tipo de acceso se usa comunmente en cintas y cartuchos.

Acceso directo: Permite acceder cualquier sector o registro inmediatamente, por medio de llamadas al sistema como la de seek. Este tipo de acceso es rápido y se usa comúnmente en discos duros y discos o archivos manejados en memoria de acceso aleatorio. \_ Acceso directo indexado: Este tipo de acceso es útil para grandes volúmenes de información o datos. Consiste en que cada archivo tiene una tabla de apuntadores, donde cada apuntador va a la dirección de un bloque de índices, lo cual permite que el archivo se expanda a través de un espacio enorme. Consume una cantidad importante de recursos en las tablas de índices pero es muy rápido.

#### 5.1.6. Operaciones soportadas por el subsistema de archivos

Independientemente de los algoritmos de asignación de espacio, de los métodos de acceso y de la forma de resolver las peticiones de lectura y escritura, el subsistema de archivos debe proveer un conjunto de llamadas al sistema para operar con los datos y de proveer mecanismos de protección y seguridad. Las operaciones básicas que la mayoría de los sistemas de archivos soportan son:

Crear ( create ) : Permite crear un archivo sin datos, con el propósito de indicar que ese nombre ya está usado y se deben crear las estructuras básicas para soportarlo.

Borrar ( delete ) : Eliminar el archivo y liberar los bloques para su uso posterior.

Abrir ( open ) : Antes de usar un archivo se debe abrir para que el sistema conozca sus atributos, tales como el dueño, la fecha de modificación, etc. \_ Cerrar ( close ) : Después de realizar todas las operaciones deseadas, el archivo debe cerrarse para asegurar su integridad y para liberar recursos de su control en la memoria.

Leer o Escribir ( read, write ) : Añadir información al archivo o leer el caracter o una cadena de caracteres a partir de la posición actual. \_ Concatenar ( append ) : Es una forma restringida de la llamada 'write', en la cual sólo se permite añadir información al final del archivo. \_ Localizar ( seek ) : Para los archivos de acceso directo se permite posicionar el apuntador de lectura o escritura en un registro aleatorio, a veces a partir del inicio o final del archivo.

Leer atributos: Permite obtener una estructura con todos los atributos del archivo especificado, tales como permisos de escritura, de borrado, ejecución, etc.

Poner atributos: Permite cambiar los atributos de un archivo, por ejemplo en UNIX, donde todos los dispositivos se manejan como si fueran archivos, es posible cambiar el comportamiento de una terminal con una de estas llamadas.

Renombrar ( rename ) : Permite cambiarle el nombre e incluso a veces la posición en la organización de directorios del archivo especificado. Los subsistemas de archivos también proveen un conjunto de llamadas para operar sobre directorios, las más comunes son crear, borrar, abrir, cerrar, renombrar y leer. Sus funcionalidades son obvias, pero existen también otras dos operaciones no tan comunes que son la de 'crear una liga' y la de 'destruir la liga'. La operación de crear una liga sirve para que desde diferentes puntos de la organización de directorios se pueda acceder un mismo directorio sin necesidad de copiarlo o duplicarlo. La llamada a 'destruir una liga' lo que hace es eliminar esas referencias, siendo su efecto la de eliminar las ligas y no el directorio real. El directorio real es eliminado hasta que la llamada a 'destruir liga' se realiza sobre él.

## 5.2 Directorios.

Para llevar un registro de los archivos, los sistemas de archivos tienen por lo general directorios, los cuales, en muchos sistemas, son a su vez archivos.  
Estructuras de directorios.

### Single Level Directory.

Todos los archivos son contenidos en un único directorio, lo cual hace que la búsqueda sea muy fácil pero no se pueden repetir los nombres de archivos.

### Two Level Directory.

Cada usuario va a tener un single level directory lo cual soluciona el problema de los nombres. Estos directorios de cada usuario se llaman UFD (User File Directory). Se mantiene un directorio MFD (Master File Directory) el cual indica dado un usuario que directorios posee. El problema de esta estructura es que no permite los archivos compartidos.

### Estructura de Árbol.

Esta es la estructura más común. Existe un directorio raíz que contiene un conjunto de archivos y subdirectorios. Hay un bit en la estructura de directorios que indica si una entrada es un archivo o un subdirectorio.

Grafo Acíclico.

Si un archivo puede estar referenciado por dos o mas entradas de un directorio el sistema operativo deberá proveer mecanismos que eviten procesar mas de una vez la información en operación que recorren la estructura de directorios. El sistema operativo debe llevar una lista de referencias de los archivos (file reference list) para poder determinar en que momento se debe liberar el espacio ocupado al borrar un archivo.

Grafos.

Los algoritmos de búsqueda en la estructura de directorios deben contemplar la posibilidad de entrar en loop infinito.

### 5.3 Implementación de sistemas de archivos(File Systems).

Método de Asignación Contigua.

Cada vez que se crea un archivo el mismo es almacenado en disco en forma contigua (conjuntos de bloques contiguos en disco). Simplifica los mecanismos de acceso a la información del archivo. La estructura de directorios sería:

| archivo | Bloque de inicio | Largo |
|---------|------------------|-------|
| Datos 1 | 2                | 5     |
| Datos 2 | 15               | 23    |
| mail    | 52               | 3     |

Los archivos no se pueden extender, se les asigna el disco cuando se crea con un tamaño que se cree conveniente, por mas que los cálculos sean buenos, se puede llegar a desperdiciar espacio.

La dificultad es que no siempre se tiene la información del espacio que halla e ocupar un archivo al momento de su creación.

Asignación Encadenada.-

Cada archivo en un conjunto de bloques encadenados que pueden estar distribuidos en cualquier lugar del disco. Los archivos pueden crecer libremente siempre que haya espacio en sico. Para encontrar el bloque i-ésimo es necesario i bloques en memoria para poder seguir la cadena.

Variante: Tabla FAT (Ms-DOS)

Asignación Indexada.

Cada archivo mantiene un bloque que indica en donde se almacena los puentes a todos los bloques ocupados por el archivo. Un archivo puede crecer libremente y en la búsqueda es más rápido. Soluciona los problemas de la asignación continua (los archivos pueden crecer libremente mientras haya espacio en disco y espacio en el archivo de índices). El problema de este método es que se puede producir pérdida de espacio ya que cada archivo por chico que sea tiene un bloque índice. Por otro lado existe un límite para el tamaño de un archivo dado por la cantidad de índices. La ventaja del método es la velocidad en el acceso a la información en el archivo.

*Manejo de espacio libre.*

La lista de espacio libre (Free Sapce List), que lleva información de todos los bloques de disco que no estén siendo ocupados por ningún archivo o directorio. Cada vez que un archivo es creado, el sistema operativo recorre esta lista asignando los bloques necesarios.

Implementación de la lista de espacio libre.

Bit Vector (Vector de Bits).

La lista de espacio libre es llevada como un vector de bits, donde 1 indica que el bloque esta ocupado y 0 indica un bloque libre. Es simple encontrar cual es el primer bloque disponible, o n bloque consecutivo. El problema es que la técnica deja de ser eficiente a menos que el vector sea cargado en memoria principal, lo que puede ser un problema.

### **5.3.1 Implementación de Directorios.**

Lista Encadenada.

Consiste en llevar una lista de nombres de archivos con punteros a los bloques de datos. La búsqueda de esta implementación consume considerable tiempo cada vez que se referencia a un archivo, debe recorrerse toda la estructura de la lista.

Hash.

Consiste en llevar una estructura de hash para representar la información de directorio. El principal problema de esta implementación es fijar el largo de la tabla de hash.

Si la tabla es muy grande (como es un vector) la búsqueda es rápida pero se desperdicia mucho espacio. Si la tabla es chica los tiempos de búsqueda casi como los de la lista encadenada.

*Estructura del Almacenamiento Secundario.-*

Estructura de disco.

Las direcciones en disco son referidas como una dirección compuesta, donde se incluye el numero de dispositivo, la cara, la pista y el sector. El conjunto de sectores que pueden ser leídos sin que se muevan las cabezas del disco se denominan cilindro. El sector es la menor unidad que puede ser leída o escrita de un disco.

Los tiempos involucrados en una operación de E/S son:

- **Tiempo de búsqueda:** es el tiempo que le lleva al disco mover las cabezas hasta llegar a la pista adecuada.
- **Tiempo de latencia:** es el tiempo que demora el disco en girar hasta ubicar el sector dentro de la pista
- **Tiempo de transferencia:** es el tiempo que lleva transferir la información entre la memoria y el disco.

Mientras el disco esta resolviendo una solicitud, las solicitudes adicionales serán encoladas en cola de E/S. Existen algoritmos para resolver mas eficientemente el problema de cómo ordenar la cola de E/S. Algoritmos de Selección de Cola de I/O.-

FCFS (First Come First Served).

Se accede a las direcciones en el disco de la misma forma en que fueron entrando.

SSJF (Shortest Seek Time First).

La idea es servir juntos todos los requerimientos que estén cerca de la posición actual de la cabeza del disco. Los requerimientos que están muy apartados de la posición de la cabeza no van a ser tomados en cuenta nunca (la cola es dinámica, siempre están llegando requerimientos). Es decir que pueden haber requerimientos que no sean satisfechos nunca o que sean postergados indefinidamente.

Scan.

Propone resolver todos los requerimientos en el sentido en que se esta moviendo la cabeza del disco hasta llegar a su extremo o a que no hay requerimientos para resolver en ese sentido, en cuyo caso comienza a resolver en sentido contrario.

C-Scan.

Similar al anterior pero los requerimientos se resuelven siempre en el mismo sentido. Si no hay mas requerimientos para satisfacer en ese sentido o se llegó al extremos del disco se salta al comienzo del mismo y se continúa resolviendo.

Formateo.

Formateo Físico.

Consiste en escribir un conjunto de sectores en el disco en cada pista que las cabezas puedan leer. Cada uno de estos sectores contiene el numero de sector, información de ECC (error Correctin Code) el cual permite identificar errores en la lectura de un sector del disco. Cada vez que se escribe un sector se calcula un valor aplicando una formula a todos los bytes del sector y se almacena en este. Al momento de leer el sector, se recalcula el valor y se compara con el valor guardado. Si son distinto el sector puede estar dañado y el sistema operativo puede decidir marcarlo como malo. (bad blocks).

Formateo Lógico.

Inicializa la información del file systems: directorio vacío, información de i-nodos, fat, etc, free space list, etc.

Manejo del espacio de Swap.

El espacio de swap puede ser utilizado de diferentes maneras dependiendo del sistema operativo en cuestión. El espacio de swap puede residir en el file system o en una partición separada. Si es un archivo mas las primitivas normales del file system se utilizan para crearlo, asignarle espacio, escribirlo, etc. Compite con los demás requerimientos de E/S del file system.

Lo mas común es almacenar el espacio de swap en una partición separada donde no se coloque ningún file sstem ni estructura de directorios (raw devices). Se utilizan algoritmos de gerenciamiento del espacio de swap para asignar y desasignar bloques. La ventaja es la velocidad ya que se evitan las primitivas de E/S del file system y no se compite con la cola de E/S del dispositivo.

## 5.4 Seguridad

Los sistemas de archivos contienen información muy valiosa para los usuarios. La protección de esa información en contra del uso no autorizado es una preocupación esencial de todos los sistemas de archivos.

### 5.4.1 la seguridad tiene muchas facetas:

las mas importantes son: a) La perdida de datos y b) los intrusos.

Las principales causas de la perdida de datos son:

siniestros

Errores del hardware.

Errores humanos.

En cuanto a los intrusos, estos tienen dos vertientes:

Los intrusos pasivos que solo quieren leer archivos que no están autorizados a leer.

Los intrusos pasivos que desean hacer cambios no autorizados a los datos.

Es importante conocer estos tipos situaciones pues ayudaran a diseñar los esquemas de protección adecuados con el fin de lograr mejores niveles de seguridad.

Los niveles de seguridad se diseñaran para:

Evitar curiosidad casual.

Conocidos husmeando

Vulneración del sistema para obtener dinero.

Espionaje comercial o militar.

Terrorismo informático

Virus

## 5.5 Mecanismos de protección

Refiere a los mecanismo para controlar el acceso de programas, usuarios o proceso a los recursos definido en el sistema. Un sistema es un conjunto de recursos (CPU, memoria, impresoras, etc.) en donde cada objeto o recurso tiene un nombre único. Las operaciones que se pueden hacer sobre cada objeto dependen del objeto.

Dominios de protección.

Cada proceso opera dentro de su dominio de protección, que indica las operaciones que el proceso puede hacer sobre un determinado conjunto de objetos. La habilidad de un proceso de ejecutar una operación sobre un objeto se denomina derecho (access right). Un dominio es un conjunto de derechos cada uno de los cuales es un par ordenado **<objeto, lista de derechos>**

Un dominio puede estar definido de varias formas:

Un usuario es un dominio. El conjunto de objetos que pueden ser accedidos dependen de la identidad del usuario.

Un proceso es un dominio. El conjunto de objetos que pueden ser accedidos dependen de la identidad del proceso.

Un procedimiento es un dominio.

Matriz de Acceso.

El modelo de protección puede ser visto mediante una matriz de acceso. Las filas representan dominios y las columnas representan objetos. Las entradas de la matriz representan el conjunto de derechos para el objeto en cuestión, para determinado dominio.

### **5.5.1 Algunas facilidades extras de los sistemas de archivos**

Algunos sistemas de archivos proveen herramientas al administrador del sistema para facilitarle la vida. Las más notables es la facilidad de compartir archivos y los sistemas de 'cotas'.

La facilidad de compartir archivos se refiere a la posibilidad de que los permisos de los archivos o directorios dejen que un grupo de usuarios puedan accederlos para diferentes operaciones" leer, escribir, borrar, crear, etc. El dueño verdadero es quien decide qué permisos se aplicarán al grupo e, incluso, a otros usuarios que no formen parte de su grupo. La facilidad de 'cotas' se refiere a que el sistema de archivos es capaz de llevar un control para que cada usuario pueda usar un máximo de espacio en disco duro. Cuando el usuario excede ese límite, el sistema le envía un mensaje y le niega el permiso de seguir escribiendo, obligándolo a borrar algunos archivos si es que quiere almacenar otros o que crezcan. La versión de UNIX SunOS contiene esa facilidad.

### **5.5.2. Sistemas de Archivos Aislados**

Los sistemas de archivos aislados son aquellos que residen en una sola computadora y no existe la posibilidad de que, aún estando en una red, otros sistemas puedan usar sus directorios y archivos. Por ejemplo, los archivos en discos duros en el sistema MS-DOS clásico se puede ver en esta categoría.

### **5.5.3. Sistemas de Archivos Compartidos o de Red**

Estos sistemas de archivos es factible accederlos y usarlos desde otros nodos en una red. Generalmente existe un 'servidor' que es la computadora en donde reside el sistema de archivos físicamente, y por otro lado están los 'clientes', que se valen del servidor para ver sus archivos y directorios de manera como si



estuvieran localmente en el cliente. Algunos autores les llaman a estos sistemas de archivos 'sistemas de archivos distribuidos' lo cual no se va a discutir en este trabajo.

Los sistemas de archivos compartidos en red más populares son los provistos por Netware, el Remote File Sharing ( RFS en UNIX ), Network File System ( NFS de Sun Microsystems ) y el Andrew File System ( AFS ). En general, lo que proveen los servidores es un medio de que los clientes, localmente, realicen peticiones de operaciones sobre archivos los cuales con 'atrapadas' por un 'driver' o un 'módulo' en el núcleo del sistema operativo, el cual se comunica con el servidor a través de la red y la operación se ejecuta en el servidor. Existen servidores de tipo "stateless y no-stateless". Un servidor "stateless" no registra el estado de las operaciones sobre los archivos, de manera que el cliente se encarga de todo ese trabajo. La ventaja de este esquema es que si el servidor falla, el cliente no perderá información ya que ésta se guarda en memoria localmente, de manera que cuando el servidor reanude su servicio el cliente proseguirá como si nada hubiese sucedido. Con un servidor "no-stateless", esto no es posible.

La protección sobre las operaciones se lleva a cabo tanto en los clientes como en el servidor: si el usuario quiere ejecutar una operación indebida sobre un archivo, recibirá un mensaje de error y posiblemente se envíe un registro al subsistema de 'seguridad' para informar al administrador del sistema de dicho intento de violación.

En la práctica, el conjunto de permisos que cada usuario tiene sobre el total de archivos se almacena en estructuras llamadas 'listas de acceso' ( access lists ).

#### **5.5.4. Tendencias actuales**

Con el gran auge de las redes de comunicaciones y su incremento en el ancho de banda, la proliferación de paquetes que ofrecen la compartición de archivos es común. Los esquemas más solicitados en la industria es el poder acceder los grandes volúmenes de información que residen en grandes servidores desde las computadoras personales y desde otros servidores también. Es una realidad que la solución más socorrida en las empresas pequeñas es usar Novell Netware en un servidor 486 o superior y acceder los archivos desde máquinas similares.

A veces se requieren soluciones más complejas con ambientes heterogéneos:

diferentes sistemas operativos y diferentes arquitecturas. Uno de los sistemas de archivos más expandidos en estaciones de trabajo es el NFS, y prácticamente todas las versiones de UNIX traen instalado un cliente y hasta un servidor de este servicio.

Es posible así que una gran cantidad de computadoras personales (de 10 a 80 ) accedan grandes volúmenes de información o paquetería (desde 1 a 8 Gygabytes ) desde una sola estación de trabajo, e incluso tener la flexibilidad de usar al mismo tiempo servidores de Novell y NFS. Soluciones similares se dan con algunos otros paquetes comerciales, pero basta ya de 'goles'. Lo importante aquí es observar que el mundo se va moviendo poco a poco hacia soluciones distribuidas, y hacia la estandarización que, muchas veces, es 'de facto'.

## 6. Entrada/Salida

### 6.1 Introducción.

Las computadoras realizan cuatro funciones básicas:

1. Reciben entradas. La entrada son los datos que se capturan en un sistema de computación para su procesamiento.
2. Producen salidas. La salida es la presentación de los resultados del procesamiento.
3. Procesan información
4. Almacenan información

Todo sistema de cómputo tiene componentes de hardware dedicados a estas funciones:

1. Dispositivos de entrada
2. Dispositivos de salida
3. Unidad central de procesamiento. Es la computadora real, la "inteligencia" de un sistema de computación.
4. Memoria y dispositivos de almacenamiento.

Cada dispositivo de entrada es sólo otra fuente de señales eléctricas; cada dispositivo de salida no es más que otro lugar al cual enviar señales; cada dispositivo de almacenamiento es lo uno o lo otro, dependiendo de lo que requiera el programa; no importa cuáles sean los dispositivos de entrada y salida si son compatibles.

Los elementos fundamentales que justifican el uso de las computadoras, radican en que las computadoras son:

- **Útiles.**
- **Baratas:** tanto con respecto a sí mismas como con respecto al costo de la mano de obra.
- **Fáciles** de utilizar.

### Descripción del procesador

Los procesadores se describen en términos de su tamaño de palabra, su velocidad y la capacidad de su RAM asociada (v.g.: 32 bits, 333MHz, 64 MB)

- **Tamaño de la palabra:** Es el número de bits que se maneja como una unidad en un sistema de computación en particular. Normalmente, el tamaño de palabra de las microcomputadoras modernas es de 32 bits; es decir, el bus del sistema puede transmitir 32 bits (4 bytes de 8 bits) a la vez entre el procesador, la RAM y los periféricos.
- **Velocidad del procesador:** Se mide en diferentes unidades según el tipo de computador:

**MHz** (Megahertz): para microcomputadoras. Un oscilador de cristal controla la ejecución de instrucciones dentro del procesador. La velocidad del procesador de una micro se mide por su frecuencia de oscilación o por el número de ciclos de reloj por segundo. El tiempo transcurrido para un ciclo de reloj es  $1/\text{frecuencia}$ . Por ejemplo un procesador de 50MHz (o 50 millones de

ciclos de reloj) necesita 20 nanosegundos para concluir un ciclo. Cuanto más breve es el ciclo de reloj, más veloz es el procesador.

**MIPS** (Millones de instrucciones por segundo): Para estaciones de trabajo, minis y macrocomputadoras. Por ejemplo una computadora de 100 MIPS puede ejecutar 100 millones de instrucciones por segundo.

**FLOPS** (*floating point operations per second*, operaciones de punto flotante por segundo): Para las supercomputadoras. Las operaciones de punto flotante incluyen cifras muy pequeñas o muy altas. Hay supercomputadoras para las cuales se puede hablar de GFLOPS (Gigaflops, es decir 1.000 millones de FLOPS).

**Capacidad de la RAM:** Se mide en términos del número de bytes que puede almacenar. Habitualmente se mide en KB y MB, aunque ya hay computadoras en las que se debe hablar de GB.

**Tipos de computadoras:** La característica distintiva de cualquier sistema de computación es su tamaño, no su tamaño físico, sino su *capacidad de cómputo*. El tamaño o capacidad de cómputo es la cantidad de procesamiento que un sistema de computación puede realizar por unidad de tiempo.

- **Macrocomputador:** Máquina de enormes dimensiones, que usan las grandes organizaciones y que tienden a ser invisibles para el público en general, ya que están escondidas en salas con clima controlado. Son capaces de comunicarse simultáneamente con varios usuarios por la técnica de tiempo compartido; éste también permite que los usuarios con diversas necesidades computacionales compartan costosos equipos de computación.
- **Minicomputador:** También es una máquina multiusuario (es decir que usa la técnica de tiempo compartido). Es más pequeño y económico que un macrocomputador, pero mayor y más potente que una computadora personal.
- **Estación de trabajo:** Computador de escritorio que tiene el poder de un minicomputador, pero a una fracción del costo. Es de uso muy común entre personas cuyas tareas requieren gran cantidad de cálculos (científicos, analistas bursátiles, ingenieros). Aunque muchas estaciones de trabajo son capaces de dar servicio a varios usuarios al mismo tiempo, en la práctica a menudo son usadas por una sola persona a la vez.
- **Microcomputadora o Computador personal:** PC (*Personal computer*). Computador habitualmente monousuario (aunque puede configurarse para usuarios múltiples) de propósito general. En una micro se monta el microprocesador, los circuitos electrónicos para manejar los dispositivos periféricos y los chips de memoria en un solo tablero de circuitos, el tablero de sistema o *tablero madre* (*mother board*). El microprocesador y los otros chips se montan en una portadora antes de fijarlos al tablero madre. Las portadoras tienen conectores de agujas de tamaño estándar que permiten que se conecten los chips en el tablero de sistema. La mother board es lo que distingue a una computadora de otra. La PC puede ser de escritorio o portátil. Dentro de los computadores portátiles encontramos:
  - **Laptop:** alimentado por baterías, con pantalla plana y que pueden cargarse como un portafolios.
  - **Notebook:** Más livianas que las anteriores y que pueden transportarse dentro de un portafolios.
  - **Palmtop:** o computador manual, o PC de bolsillo. Tan pequeñas que caben en un bolsillo. Atiende las necesidades de usuarios para los cuales la movilidad es más valiosa que un teclado o una pantalla de tamaño usual.

## Canales, puertos y ranuras de expansión

- **Canales:** Grupos de cables o pistas de impreso a través de los cuales viaja la información entre los componentes del sistema. Tienen 8, 16 o 32 cables y este número indica la cantidad de bits de información que puede transmitir al mismo tiempo. Los canales más anchos pueden transmitir información con más rapidez que los canales angostos.
- **Ranuras de expansión:** Se conectan al bus eléctrico común. Algunos canales están conectados a ellas en la caja del computador. Los usuarios pueden personalizar sus máquinas insertando tarjetas de circuitos (o *tarjetas*) de propósito especial en estas ranuras. Existen tarjetas de expansión de RAM, adaptadores de color y de gráficos, fax módem, puertos, coprocesadores (procesadores adicionales que incrementan la capacidad o velocidad de procesamiento del sistema), etc.
- **Puertos:** Son puntos de conexión en la parte exterior del chasis de la computadora a los que se conectan algunos canales. El puerto permite una conexión directa con el bus eléctrico común de la PC. los puertos pueden ser:
  - Puertos series: Permiten la transmisión en serie de datos, un bit a la vez. Este tipo de puertos permiten una interfaz con impresoras y módems de baja velocidad.
  - Puertos paralelos: Permiten la transmisión paralela de datos, es decir que se transmiten varios bits simultáneamente. Permiten la interfaz con dispositivos tales como impresoras de alta velocidad, unidades de cinta magnética de respaldo y otras computadoras.

Las ranuras de expansión y los puertos simplifican la adición de dispositivos externos o periféricos

## Tareas que realizan las computadoras

Son pocas, sólo 4, pero con rapidez y precisión:

1. **Capturar datos:** Llevar mensajes del entorno al sistema.
2. **Calcular:** en rigor sólo suman, pero así logran realizar las 4 operaciones básicas.
3. **Comparar:** En sí misma, la comparación no sirve de nada; sólo si ayuda a la toma de decisiones. Sólo realizan comparaciones elementales (con dos posibilidades). La combinación secuencial de comparaciones permite la comparación compleja, y por ende la toma de decisiones complejas.
4. **Registrar:** Tanto en el sentido de mostrar (pantalla, impresora), es decir llevar a un lenguaje humano algo que está guardado en el computador, como en el sentido de guardar algo en el computador.

## 6.2 HARDWARE DE ENTRADA/SALIDA

### 6.2.1 Unidades y soportes de entrada

Los dispositivos de entrada traducen los datos a una forma que la computadora pueda interpretar, para luego procesarlos y almacenarlos.

#### Dispositivos manuales

- **Teclado alfanumérico:** El estándar es actualmente el teclado de 101 letras con la distribución QWERTY, 12 teclas de funciones, un teclado o pad numérico, teclas de función y teclas para el control del cursor. Algunos teclados están diseñados para aplicaciones específicas, permitiendo una interacción rápida con los sistemas de

computación (v.g.: caja registradora). El teclado es un circuito en forma de matriz; cada circuito está conectado al dispositivo controlador, que reconoce la letra o código que envía el usuario cuando se cierra o abre un circuito. La configuración del teclado puede ser modificado por software.

- **Teclado para perfoverificación:** cada bit se representa como perforado o no perforado. Cada columna de la tarjeta es barrida por un cepillo metálico, cuando hay una perforación al pasar el cepillo se cierra un circuito.

## Dispositivos apuntadores

- **Ratón:** La efectividad de las GUI depende de la capacidad del usuario para hacer una selección rápida de una pantalla con íconos o menús. En estos casos el mouse puede colocar el apuntador (o cursor gráfico) sobre un ícono con rapidez y eficiencia. Los más comunes tienen una esfera en su parte inferior que puede rodar en un escritorio.
- **Bola rastreadora (trackball) o bola palmar:** Es una bola insertada en una pequeña caja que se hace girar con los dedos para mover el cursor gráfico.
- **Palanca de mando (joystick):** también llamada palanca de control de juegos. Es una palanca vertical que mueve el cursor gráfico en la dirección en que se mueve la palanca.
- **Pantalla sensible al tacto:** Sirven cuando hay muchos usuarios no familiarizados con las computadoras. Puede ser sensible al tacto por la presión o por el calor. Son de muy baja velocidad.

## Dispositivos ópticos

- **Lector de marcas o rastreador de marca óptica:** Usa la luz reflejada para determinar la ubicación de marcas de lápiz en hojas de respuestas estándar y formularios similares.
- **Lector de código de barras** Usa la luz para leer **UPC** (*Universal Product Codes*, Códigos universales de productos), códigos de inventario y otros códigos creados con patrones de barras de anchura variable. Los códigos de barra representan datos alfanuméricos variando el ancho y la combinación de las líneas verticales adyacentes. La ventaja de los códigos de barras sobre los caracteres es que la posición u orientación del código que se lee no es tan importante para el lector.
- **Lector de vara (lápiz óptico):** Usa luz para leer caracteres alfabéticos y numéricos escritos con un tipo de letra especial, siendo también legible para las personas este tipo de letra; muchas veces estos lectores están conectados a terminales **POS** (*point-of-sale*, punto de venta). Cuando se usan de esta forma el computador lleva a cabo un reconocimiento óptico de caracteres (**OCR**, *optical character recognition*).
- **Rastreador de páginas** Rastrea e interpreta los caracteres alfanuméricos de las paginas impresas normales. Se usa para convertir una copia dura a un formato que la máquina puede leer. Este tipo de rastreador puede reducir al mínimo o eliminar la captura de datos mediante el teclado.

## Dispositivos magnéticos

- **MICR (*magnetic ink character recognition*, reconocimiento de caracteres en tinta magnética) o Lectora de caracteres magnéticos:** lee los caracteres impresos con tinta magnética en los cheques. En ellos el número de cuenta y el número de cheque se encuentran codificados; la fecha de la transacción se registra automáticamente para todos los cheques procesados ese día; por tanto, sólo se debe teclear el importe en un inscriptor MICR. Un lector-ordenador MICR lee los datos de los cheques y los ordena para el procesamiento que corresponda. Estos dispositivos de reconocimiento son más rápidos y precisos que los OCR.
- **Lectora de bandas magnéticas:** Las bandas magnéticas del reverso de las tarjetas de crédito, por ejemplo, ofrece otro medio de captura de datos directamente de la fuente (como los dispositivos ópticos).

Se codifican las bandas con datos apropiados para la aplicación. Las bandas magnéticas contienen muchos más datos por unidad de espacio que los caracteres impresos o los códigos de barras. Además, dado que no se pueden leer visualmente, son perfectos para almacenar datos confidenciales.

## Digitalizadores

Para que un computador pueda reconocer texto manuscritos, primero tiene que digitalizar la información, convertirla en alguna forma digital para poder almacenarla en la memoria del computador. Hay diferentes dispositivos de entrada para capturar y digitalizar información:

- **Digitalizador de imágenes (scanner):** Puede obtener una representación digital de cualquier imagen impresa. Convierte fotografías, dibujos, diagramas y otra información impresa en patrones de bits que pueden almacenarse y manipularse con el soft adecuado
- **Cámara digital:** Es un digitalizador de imágenes que permite tomar fotografías del mundo real y obtener imágenes digitales; es decir que no se limita a capturar imágenes impresas planas, puede registrar las mismas cosas que una cámara normal, sólo que en lugar de registrar las imágenes en película, las cámaras digitales almacenan patrones de bits en discos u otros medios de almacenamiento digital.
- **Digitalizador de audio:** Permite digitalizar sonidos de micrófonos y otros dispositivos de sonido. Para que el computador interprete correctamente la entrada de voz digitalizada como si fueran palabras se requiere software de inteligencia artificial. Una unidad de respuesta auditiva o un sintetizador de voz hace que la conversación sea un diálogo. El reconocimiento del habla funciona de la siguiente manera:
  - *Se dice la palabra.* Cuando se habla en un micrófono, cada sonido se divide en sus diversas frecuencias.
  - *Se digitaliza la palabra.* Se digitalizan los sonidos de cada palabra de modo que la computadora los pueda manejar.
  - *Se compara la palabra.* Se compara la versión digitalizada contra modelos similares del diccionario electrónico de la computadora. El modelo digitalizado es una forma que las computadoras pueden almacenar e interpretar.
  - *Se presenta la palabra o se realiza el comando.* Cuando se encuentra una igualdad, se presenta en una VDT o se realiza el comando adecuado.

En el reconocimiento del habla, la creación de los datos se conoce como *capacitación*. La mayor parte de los sistemas de reconocimiento del habla son dependientes del locutor, es decir que responde a la voz de un individuo particular.

La tecnología más reciente permite sistemas independientes del locutor, pero necesitan una base de datos muy grande para aceptar el patrón de voz de cualquier persona.

- **Digitalizador de vídeo:** Es una colección de circuitos que puede capturar entradas de una fuente de vídeo y convertirla en una señal digital que puede almacenarse en la memoria y exhibirse en pantallas de computador. Cuando se pone en operación el sistema, éste compara la imagen digitalizada que se debe interpretar con las imágenes digitalizadas registradas previamente en la base de datos. Estos sistemas de entrada de visión son apropiados para tareas especializadas, en que sólo se encuentran unas cuantas imágenes.
- **Dispositivos sensores** diseñados para hacer seguimientos de la temperatura, la humedad, la presión y otras cantidades físicas, proporcionan datos útiles en robótica, control ambiental, pronósticos meteorológicos, supervisión médica, biorretroalimentación, investigación científica y cientos de aplicaciones más.

## Otras entradas

- **Tarjetas inteligentes:** Son una versión mejorada de las tarjetas con banda magnética. Contienen un microprocesador que almacena algunos datos de seguridad y personales en su memoria en todo momento. Dado que las tarjetas inteligentes pueden tener más información, que tienen cierta capacidad de procesamiento y que es casi imposible duplicarlas, seguramente sustituirán a las tarjetas con bandas magnéticas.
- **Analógicas:** Sensores que miden magnitudes físicas escalares o vectoriales.

## Conceptos vinculados

- **Documentos retornables:** Un documento retornable es una salida generada por computadora que finalmente regresa como una entrada que la máquina puede leer.
- **Sistemas OCR (optical character recognition):** Es un proceso de naturaleza topológica (analiza la forma por medio de funciones matemáticas) y neuronal (actúa como las neuronas de las personas; el problema es que a veces falla la conexión entre ellas). El primer paso en el reconocimiento óptico de caracteres consiste en digitalizar la imagen de la hoja en la memoria del computador mediante un digitalizador (scanner), una cámara digital o un fax módem. La imagen digitalizada no es más que un patrón de bits en la memoria. Antes de que el computador pueda procesar el texto de la página, debe reconocer los caracteres individuales y convertirlos en códigos de texto. El software de OCR localiza e identifica los caracteres impresos que aparecen en la imagen, "lee" el texto. Los programas de OCR se valen de varias técnicas:
  - la segmentación de la página en imágenes, bloques de texto y (finalmente) caracteres individuales;
  - tecnología de sistemas expertos, a una escala menor, para reconocer las reglas básicas de distinción de letras;
  - "expertos" en contextos para ayudar a identificar letras ambiguas de acuerdo con su contexto;
  - aprendizaje a partir de ejemplos reales y retroalimentación de un entrenador humano.

## Unidades y soportes de salida

Estos dispositivos traducen los bits y bytes a una forma comprensible para el usuario.

## Monitores

Una **VDT** (*video display terminal, terminal de despliegue visual*) sirve como dispositivo de salida para recibir mensajes del computador. Las imágenes de un monitor se componen de pequeños puntos llamados **píxeles** (picture elements) o elementos de imagen. La cantidad de ellos que hay por cada pulgada cuadrada determina la **definición** del monitor que se expresa en puntos por pulgada o **dpi** (*dots per inch*). Cuanto más alta es la definición, más cercanos están los puntos.

La salida de un monitor es temporal y se la designa como copia blanda o efímera.

Pueden ser monocromáticos o a colores; la mayoría de estos últimos combinan el rojo, el verde y el azul para lograr un espectro y por ello se llaman monitores **RGB** (*red, green, blue*).

Los monitores pueden ser de dos clases:

- **CRT (*cathode ray tube*), tubo de rayos catódicos:** como en un televisor. Son los preferidos para los computadores de escritorio por su claridad y velocidad de respuesta.
- **De pantalla plana::** Más compactos y ligeros, dominan el mercado de las computadoras portátiles. Utilizan 3 tipos de tecnología:
  - **LCD (*liquid crystal display*), pantalla de cristal líquido.** Consumen relativamente poca energía.
  - **Plasma de gas.**
  - **EL (*electroluminiscencia*).** Ofrecen mayor ángulo de visión.

## Impresoras

Una impresora permite obtener una **copia dura** o física de cualquier información que pueda aparecer en pantalla. Hay dos grupos básicos de impresoras:

- de Impacto: Dependen de la tecnología de **matriz de puntos**. Forman las imágenes golpeando un martillo contra una cinta y el papel; al hacer contacto con el papel pueden producir copias al carbón junto con el original. entre ellas encontramos:
- **de línea:** Son rápidas y ruidosas. Tienen la desventaja de estar limitadas a la impresión de caracteres, por lo que no son apropiadas para aplicaciones donde los gráficos son un ingrediente esencial del producto acabado. imprimen una línea de puntos a la vez. Se alinean martillos similares a agujas sobre el ancho del papel.
- **en serie:** Imprimen texto y gráficos. Usa martillos del tamaño de un alfiler para transferir la tinta a la página. Una página impresa es una matriz de pequeños puntos, algunos blancos y otros negros (o color). Este tipo de impresora tiene una baja definición, inferior a las 100 dpi. Forma las imágenes, un carácter a la vez, a medida que la cabeza de impresión se mueve sobre el papel. Las impresoras en serie son bidireccionales, es decir que imprimen sin importar hacia que lado se este moviendo la cabeza de impresión. La cabeza de impresión contiene una o varias columnas de *agujas*, que se activan independientemente para crear la imagen del carácter. El número de puntos de la matriz puede variar, y la calidad de la impresión se relaciona con la densidad de estos puntos. Las más densas son impresoras de modo dual, porque pueden imprimir en calidad de borrador o NLQ (*near-letter-quality*, calidad casi tipo carta).
- De no impacto o de página: Han ido reemplazando a las anteriores, salvo cuando hay que imprimir formularios con varias copias 9imprimen una sola copia a la vez); usan sustancias químicas, rayos láser y calor para crear imágenes en el papel; tienen una definición mucho mayor (300 dpi o más) y pueden ser:
- **de chorro de tinta:** rocían tinta directamente sobre el papel. Utilizan varias cámaras de inyección controladas de manera independiente para inyectar pequeñas gotas de tinta sobre el papel.
- **láser:** un rayo láser crea patrones de cargas eléctricas en un tambor giratorio; estos patrones atraen tonificador (toner) y lo transfieren al papel conforme gira el tambor.

## Trazadores



Un trazador o graficador es un instrumento automatizado para dibujar que puede producir dibujos a escala de elevada finura moviendo una pluma o el papel como respuesta a mandatos del computador.

## Respuesta audible

Hay dos tipos de unidades de respuesta de voz: uno utiliza la reproducción de una voz humana y la el otro utiliza un sintetizador de voz. Las salidas de respuesta audible ofrecen una salida de copia blanda o temporal.

En el caso de unidades de respuesta de **voz grabada**, las grabaciones análogas reales de sonidos se convierten en datos digitales que luego se almacenan permanentemente en discos o en un chip de memoria. Cuando los sonidos se almacenan en un disco el usuario tiene la flexibilidad de actualizarlos.

Los **sintetizadores** sirven para generar música, ruido o cualquier sonido intermedio. Muchas PC tienen sintetizadores incorporados que producen sonidos que van mas allá del bip básico. Casi todos los computadores se pueden conectar a sintetizadores independientes para controlar el instrumento. para producir la voz, estos dispositivos combinan sonidos similares a los fonemas (unidades de sonido básicas) que conforman la voz.

## Salidas analógicas

Muchos dispositivos de salida funcionan tomando patrones y convirtiéndolos en movimientos o mediciones no digitales. Por ejemplo los brazos robóticos, los conmutadores telefónicos, el equipo automatizado de las fábricas reciben sus órdenes de una computadora.

## Otras salidas

- **Terminales no inteligentes:** La mayoría de las terminales se clasifican como no inteligentes. Estas sólo presentan texto y se deben conectar a un procesador para usuarios múltiples. Únicamente permiten la entrada/salida de una sola aplicación.
- **Terminales X:** Tienen capacidades de procesamiento y RAM comparables a las de algunas micros y estaciones de trabajo; no están diseñadas para operar en forma independiente; permiten la interacción con el usuario por medio de una GUI. Permiten el trabajo con varias aplicaciones a la vez, desplegándose cada aplicación en su propia ventana.
- **Terminales telefónicas:** Se pueden capturar datos alfanuméricos en el teclado numérico de un teléfono (teclado) o hablando en el micrófono (entrada de voz), recibándose una salida de voz generada por computadora.
- **Terminales para funciones especiales:** Están diseñadas para una aplicación específica (v.g.: cajero automático, etc.)

### 6.2.2 Controladores de dispositivos.

Los controladores son el enlace entre el elemento mecánico (dispositivo) y la computadora y son la parte electrónica de un dispositivo de entrada o salida funcionan como un adaptador y así como pueden instalarse en uno de los slots de la computadora o como en los modelos actuales de

Mother Boards, pueden encontrarse ya instalados en ella principalmente la tarjeta de red, de sonido de video o el faxmodem.

El enlace con el dispositivo de entrada – salida es a través de un conector normalizado tipo USB, DB25, DB9 o RJ45. es necesario tener presente la diferencia entre un controlador y un dispositivo porque el S.O. siempre va a trabajar con el controlador directamente y el flujo de datos dependera del controlador, pueden existir flujos en serie o en paralelo y no es de esperarse una velocidad de trabajo como se da al interior de la computadora.

La labor del controlador es convertir el flujo de bits en serie o en paralelo en un bloque de bytes y llevar a cabo cualquier corrección de errores necesaria. Lo común es que el bloque de bytes se ensamble, bit a bit en un buffer dentro del controlador. Después de verificar la suma y declarar al bloque libre de errores, se le puede copiar en la memoria principal.

Cada controlador tiene unos cuantos registros que utiliza para la comunicación con el CPU. En algunas computadoras, estos registros son parte del espacio normal de direcciones de la memoria. Este esquema se llama E/S mapeada a memoria. La asignación de direcciones asignados a algunos dispositivos se lleva a cabo mediante una lógica de descodificación del bus asociada al controlador. La asignación de direcciones puede variar según el fabricante de computadoras y ello incide en los programas que trabajan directamente con el hardware incluyendo al sistema operativo.

Un ejemplo de lo anterior es el que se refiere al IBM PC que se muestra continuación:

| Controlador de E/S | dirección de E/S | Vector de interrupcion |
|--------------------|------------------|------------------------|
| Reloj              | 040-043          | 8                      |
| Teclado            | 060-063          | 9                      |
| Rs232 secundario   | 2F8-2FF          | 11                     |
| Disco duro         | 320-32F          | 13                     |
| Impresora          | 378-37F          | 15                     |

|                       |         |    |
|-----------------------|---------|----|
| Monitor monocromatico | 380-3BF | -  |
| Monitor color         | 3D0-3DF | -  |
| Floppy                | 3F0-3F7 | 14 |
| RS232 primario        | 3F8-3FF | 12 |

El sistema operativo realiza la E/S al escribir comandos en los registros de los controladores los cuales varían en cantidad según en controlador al que se haga referencia. Muchos de los comandos tienen parámetros, los cuales también se cargan en los registros del controlador. Al aceptar un comando, el CPU puede dejar al controlador y dedicarse a otro trabajo. Al terminar el comando, provoca una interrupción para permitir que el SO obtenga el control del CPU y verifique los resultados de la operación. El microprocesador obtiene los resultados y el estado del dispositivo al leer uno o mas bytes de información de los registros del controlador.

### 6.3 Software de entrada/salida

#### Software (I)

##### **Concepto general**

El software permite comunicar al computador los problemas y hace posible que nos comunique las soluciones. Los programas son el software del computador. Es una estructura de instrucciones (o programas) que la máquina es capaz de leer. Son programas que dirigen las actividades del sistema de computación

**Programas:** conjuntos de instrucciones de computador diseñados para resolver problemas. Confieren a la computadora capacidad para llevar a cabo las funciones deseadas. Secuencia de instrucciones (enunciados) que se ejecutan una después de otras. Estas instrucciones pueden ser de:

- Entrada/salida: dirigen a la computadora para interactuar con un periférico.
- Cómputo: permiten realizar las operaciones aritméticas.
- Control (decisión y/o ramificación): pueden alterar la secuencia de la ejecución del programa o terminar la ejecución. Hay dos tipos de instrucciones de control:
  - *de bifurcación incondicional:* interrumpen la secuencia normal de la ejecución, originando una subrutina.
  - *de rama condicional:* o enunciados SI (if); si se cumplen ciertas condiciones se crea una ramificación en cierta parte del programa.
- Transferencia de datos y asignación: permiten que se asigne a un sitio determinado de la RAM una constante de cadena o valor literal.

- Formato: se usan junto con las instrucciones de entrada o salida y describen la manera en que se deben realizar la entrada y salida de datos de la RAM.

El software alimenta a la memoria de la máquina a través de dispositivos de entrada; como el software se almacena en la memoria, la computadora puede pasar de una tarea a otra y luego regresar a la primera sin que sea necesario modificar el hardware.

## **Software (II)**

El software es el enlace de comunicación entre el ser humano y el computador. El software se almacena en memoria, y no permanentemente en los circuitos, por lo cual puede modificarse para satisfacer las necesidades del usuario.

Casi todo el software corresponde a una de 3 categorías generales:

- **programas de traducción:** permiten escribir programas en lenguajes parecidos al inglés y traducirlos al lenguaje de máquina.
- **aplicaciones de software:** facilitan a los usuarios comunicar sus necesidades al computador, sin tener que aprender a programar. Las aplicaciones simulan y extienden las propiedades de las herramientas comunes de la vida real, como las máquinas de escribir, los pinceles y los archiveros.
- **software de sistema:** el sistema operativo funciona tras bambalinas, traduciendo las instrucciones del software a mensajes que el hardware puede comprender. Un sistema operativo sirve como administrador del computador, encargándose de los múltiples detalles necesarios para que la máquina funcione.

## **Lenguajes**

Los **lenguajes naturales** son aquellos con los cuales hablamos y escribimos en nuestras actividades cotidianas. Los **lenguajes de programación** ocupan una posición intermedia entre los lenguajes naturales humanos y los precisos **lenguajes de máquina**.

Gracias a la progresiva complejidad de los lenguajes traductores que permiten convertir las instrucciones de un lenguaje de programación al lenguaje de máquina, la programación puede usar lenguajes de computación que se parecen cada vez más a los lenguajes naturales.

También se habla de lenguajes naturales para referirse al software que permite que los sistemas de computación acepten, interpreten y ejecuten instrucciones en el lenguaje materno o "natural" que habla el usuario final, por lo general el inglés. La mayor parte de los lenguajes naturales comerciales están diseñados para ofrecer a los usuarios un medio de comunicarse con una base de datos corporativa o con un sistema experto.

## **Lenguajes de programación**

Un lenguaje de programación consiste en un conjunto de órdenes o comandos que describen el proceso deseado. Cada lenguaje tiene sus instrucciones y enunciados verbales propios, que se combinan para formar los programas de cómputo.

Los lenguajes de programación no son aplicaciones, sino herramientas que permiten construir y adecuar aplicaciones.

## Características de la programación

Un programa útil tiene 5 atributos:

1. **Exactitud y precisión:** Un programa debe aceptar datos, procesarlos y generar resultados sin errores de sintaxis o de lógica; de ahí la necesidad de exactitud y precisión.
2. **Integridad o completitud:** Un programa es completo sólo si ejecuta todas las operaciones que el usuario espera al procesar un conjunto de datos.
3. **Generalidad:** La generalidad se entiende en dos sentidos:
  - Un programa bien diseñado se puede generalizar; es decir puede procesar conjuntos de datos que varían en el número de transacciones y en la naturaleza de los datos mismos. Las instrucciones se deben elaborar para que no se limiten a un número específico de transacciones en el conjunto de datos. No debe suponerse que una parte específica de datos siempre sea de igual tamaño.
  - El programa contiene diferentes opciones que pueden o no ser aplicadas durante una corrida. Los usuarios pueden seleccionar la opción adecuada cada vez.
4. **Eficiencia:** Debe asegurarse que las instrucciones para procesar datos se seleccionen tan rápida y fácilmente como sea posible. La rapidez y la eficiencia del procesamiento de datos puede variar debido a:
  - modo como se introducen los datos,
  - orden en que se procesan los datos,
  - instrucciones particulares que se empleen para controlar el proceso,
  - etc.
5. **Documentación:** Consiste en la inclusión de auxiliares que explican con claridad como ocurre el procesamiento de los datos en un programa. La documentación incluida en un programa es de dos tipos:
  - **externa:** No está incluida en el programa mismo sino que va por separado. Comprende:

- diagrama de flujo que muestre los pasos y los procedimientos y el orden en el cual tienen lugar;
- explicación que describa en el lenguaje común cómo es el procesamiento de los datos;
- descripción de los datos que van a introducirse y procesarse, incluyendo su tipo y tamaño.
- **interna:** Consiste en comentarios y descripciones que se insertan entre los enunciados ejecutables de un programa. Explica la sucesión de los pasos del procesamiento y los objetivos de los diferentes grupo de enunciados del programa

## Sistema operativo

El sistema operativo supervisa y controla todas las actividades de entrada/salida y procesamiento de un sistema de computación. Todo el hardware y el software se controla por medio del sistema operativo.

El sistema operativo distribuye y controla (es decir administra) recursos limitados del equipo de cómputo. Los programas que constituyen el sistema operativo son desarrollados generalmente por los fabricantes de computadoras y proporcionados a los compradores. Están diseñados para hacer el mejor uso de los componentes de cada sistema de computación.

Pueden existir diferentes sistemas operativos incluso en los sistemas de un solo fabricante.

El conjunto de programas de control del sistema operativo trabaja en forma conjunta y está diseñado para maximizar la eficacia total del sistema de cómputo.

Este sistema supervisa todas las actividades, recabando programas especiales de sistema cuando sea necesario, asignando recursos del sistema y programando los trabajo de tareas para la operación continua del sistema.

Los componentes del sistema operativos son de 2 tipos:

- **residentes:** también llamados supervisores. Son componentes que residen permanentemente en la memoria central durante todo el procesamiento.
- **transientes:** residen sólo cuando se los necesita, estando almacenados en periféricos cuando no están en la memoria central.

### *Funciones básicas del sistema operativo*

**1. Proveer servicios para la ejecución y desarrollo de los programas de aplicación:** administrar los recursos en proceso, obtener automáticamente la rutina apropiada y mantener la operación del computador, sin necesidad de intervención manual.

**2. Actuar como entorno de la aplicación en la cual el programa es ejecutado:** Supervisar mediante un conjunto de rutinas las operaciones del computador, controlar el desarrollo de las operaciones internas de la CPU, dirigir el desenvolvimiento de las operaciones de entrada/salida.

### *Carga inicial del sistema operativo*

Casi todas las máquinas tienen una parte de su S.O. en la ROM y lo demás se carga en memoria durante el proceso de arranque. Una de las pocas ocasiones en que el usuario debe comunicarse directamente con el S.O. es al arrancar una PC, momento en el que el S.O. aparece en primer plano y espera instrucciones.

No todo el S.O. está en memoria permanentemente. La parte que reside siempre en memoria durante la ejecución de los distintos programas se llama residente o supervisor.

Una vez encendido el computador habrá que cargar el residente en memoria para poder comenzar a ejecutar los distintos programas. Hay un primer programa especial (stand-alone), llamado IPL o BOOT, que forma parte del S.O., cuya única misión es traer a memoria al segundo y principal programa, el residente. Este programa puede ser llamado pulsando una tecla; una vez que se encuentra en memoria, se comienza a ejecutar cumpliendo una serie de tareas que difieren de acuerdo al computador, entre ellas:

- pedir la fecha del día al operador (la primera y más común);
- permitir la ejecución de programas stand-alone (todavía no se ha cargado el supervisor);
- recibir la configuración del equipo y la asignación de memoria a las particiones; puede obtener estos datos:
  - por parte del operador; o
  - directamente de un archivo creado al efecto por un programa stand-alone (utilitario configurador).

Finalmente, solicitará al operador el dispositivo donde deberá buscar al residente para traerlo a memoria. Una vez cargado el residente en memoria, este programa desaparece. Por tanto el IPL o BOOT es un programa transiente del sistema operativo.

## 6.4 Discos

### ***Almacenamiento secundario: Unidades y soportes de entrada-salida***

#### **Concepto y organización del almacenamiento secundario**

A diferencia de la RAM, que olvida todo en cuanto se apaga la máquina, y la ROM, que no puede aprender nada nuevo, los dispositivos de almacenamiento secundario permiten que la computadora registre información en forma semipermanente, para que pueda ser leída después por el mismo u otro computador. El almacenamiento secundario es más barato y de mayor capacidad que el almacenamiento primario.

- **Procesamiento secuencial:** Es el que se da en medios de almacenamiento en el cual el usuario debe pasar secuencialmente por la información, en el mismo orden en que fue grabada, hasta llegar a la que le interesa. Un archivo secuencial se procesa de principio a fin. Todo el archivo se debe procesar, aun cuando se actualice sólo un registro. Este tipo de procesamiento requiere de:
  - un *archivo maestro*, fuente permanente de todos los datos;

- un *archivo de transacción*, refleja la actividad diaria.

Antes del procesamiento, los registros en ambos archivos se clasifican y ordenan en secuencia ascendente por clave. Ambos archivos constituirán entradas y el *nuevo archivo maestro* será la salida, reflejando las actualizaciones. En este procesamiento siempre se crea un nuevo archivo maestro para las actualizaciones realizadas.

- **Procesamiento aleatorio:** Se tiene acceso a los programas y datos deseados directamente del medio de almacenamiento. En este tipo de procesamiento sólo se necesita el valor del campo clave del registro para recuperar o actualizar un registro.

## Cintas magnéticas

La cinta pasa debajo de una cabeza de escritura/lectura y se realiza la operación ordenada. Una unidad de cinta se clasifica por la densidad con que los datos se pueden almacenar, así como por la velocidad de la cinta cuando pasa por debajo de la cabeza de escritura/lectura. Combinadas, éstas determinan la **velocidad de transferencia** o el número de caracteres por segundo que se pueden transmitir a la RAM. La **densidad de cinta** se mide en bytes por pulgada (**bpi**, *bytes per inch*) o el número de caracteres (bytes) que se pueden almacenar por pulgada lineal de cinta.

Una cinta magnética puede almacenar enormes cantidades de información en un espacio pequeño y a un costo relativamente bajo. La preferida es la **DAT** (digital audio tape, cinta de audio digital). Su desventaja es que se trata de un medio de **acceso secuencial**; por ello el uso principal es para el respaldo de datos y algunas otras operaciones en las cuales el tiempo no es un factor decisivo. En cualquier sesión, una sola cinta es para entrada o salida, no para ambas.

## Discos magnéticos

Gracias a su capacidad de acceso aleatorio, son el medio más popular para el almacenamiento de datos. Los hay de dos tipos:

- **Discos flexibles o diskettes o discos magnéticos intercambiables:** Es una pequeña oblea de plástico flexible, con sensibilidad magnética encerrada en un paquete de plástico que puede ser rígido o flexible. Es económico, práctico y confiable, pero no tiene la capacidad de almacenamiento ni la velocidad necesaria para trabajos de gran magnitud. Estos discos se pueden almacenar fuera de línea y cargarlos según sea necesario.
- **Discos duros o discos magnéticos fijos:** es un disco rígido, con sensibilidad magnética, que gira continuamente a gran velocidad dentro del chasis del computador o en una caja aparte conectada a éste. Se instalan en forma permanente, aunque existen unidades portátiles. El disco duro se la microcomputadora se llama disco **Winchester**. Contiene varios platos de disco rígidos apilados en un solo eje giratorio. El movimiento de rotación pasa todos los lados debajo o sobre una cabeza de escritura/lectura, permitiendo tener acceso a todos los datos del disco en cada giro; un disco fijo tiene por lo menos una cabeza de escritura/lectura para cada superficie de grabación. Las cabezas se montan en brazos de acceso que se mueven juntos y flotan encima o bajo las superficies de grabación giratorias. Los datos se almacenan en pistas concéntricas magnetizando la superficie para representar configuraciones



de bits. El espacio de las pistas, es decir la **densidad de pista**, se mide en pistas por pulgada (**TPI**, *tracks per inch*). La **densidad de grabación** se mide en bits por pulgada (de pista). Los discos usan la **organización de sector** para almacenar y recuperar datos; la cantidad de sectores depende de la densidad del disco. Cada sector tiene un número único, por lo tanto para una dirección de disco de una superficie de la cara del disco en particular, todo lo que se necesita es el número de sector y el número de pista; la **dirección de disco** representa la ubicación física de un conjunto de datos o un programa determinados. Un **cilindro** en particular se refiere a cada pista con el mismo número en todas las superficies de grabación. Cuando se lee o se escribe en un disco Winchester todos los brazos de acceso se mueven hacia el cilindro correcto. El **tiempo de acceso** del disco es el intervalo entre el momento en que la computadora pide la transferencia de datos de un dispositivo de almacenamiento en disco a la RAM y el momento en que la operación se completa; este tiempo de acceso se compone del *tiempo de búsqueda* (la mayor parte del tiempo, consiste en el tiempo que el brazo de acceso mecánico necesita para mover la cabeza de escritura/lectura hacia el lugar deseado), el *retardo rotacional* (tiempo que ocupan los datos para colocarse debajo de la cabeza de escritura/lectura) y el *tiempo de transmisión* (tiempo necesario para transmitir los datos al almacenamiento primario; es insignificante).

## Discos ópticos

Una unidad de disco óptico usa rayos láser en lugar de imanes para leer y escribir la información en la superficie del disco. Aunque no son tan rápidos como los discos duros, los discos ópticos tienen mucho más espacio para almacenar datos.

Las unidades de **CD-ROM** (compact disc-read only memory, disco compacto-memoria sólo de lectura) son unidades ópticas capaces de leer CD-ROM, discos de datos físicamente idénticos a un disco compacto musical.

Los discos ópticos son menos sensibles a las fluctuaciones ambientales y proporcionan mayor almacenamiento a un costo menor.

## 6.4 Relojes

El reloj de la computadora realiza las siguientes funciones:

1. Mantener la hora al día
2. Evitar que los procesos se ejecuten durante mas tiempo que el permitido.
3. Mantener el registro del uso del CPU.
4. Controlar la llamada al sistema ALARM por parte de los procesos del usuario.
5. proporcionar cronómetros guardianes de partes del propio sistema.
6. realizar resúmenes , monitoreo y recolección de estadísticas.

## 6.5 Terminales

cada computadora tiene una o mas terminales que se utilizan para comunicarse con ella, las terminales tienen distintas formas. El manejador de la terminal se encarga de ocultar todas estas diferencias, de forma que la parte independiente del dispositivo en el SO y los programas del usuario no tienen que volverse a escribir para cada tipo de terminal.

Hardware para terminal. Este se clasifica según el tipo de conector, así habrá para terminales RS323 que permitirá una comunicación serial; constan de un teclado y un monitor, de impresión, Las otras terminales con las mapeadas a memoria las cuales no se comunican con la CPU mediante una línea serial sino que son parte integral de la computadora y esta se comunica con ellas como al resto de la memoria.

## 7. Bloqueos

### 7.1 Recursos

### 7.2 El algoritmo del avestruz

### 7.3 Evasión de bloqueos

### 7.4 Prevención de bloqueos

## 9. Sistemas operativos distribuidos

### 9.1 Conceptos básicos, Historia y objetivos de diseño.

### 9.2 Comunicación en los sistemas distribuidos

### 9.3 Sincronización

### 9.4 Procesos y procesadores en sistemas distribuidos

#### 9.4.1 Hilos

#### 9.4.2 Modelos de sistemas

#### 9.4.3 Asignación de procesadores

#### 9.4.4 Planificación en sistemas distribuidos

### 9.5 Sistemas distribuidos de archivos

#### 9.5.1 Diseño

#### 9.5.2 Implantación

#### 9.5.3 Tendencias en los sistemas distribuidos de archivos

### 9.6 Memoria compartida distribuida.

### 9.7 Sistemas distribuidos: CORBA

#### 98.7.1Sistemas basados y orientados a objetos.

#### 9.7.2 Componentes de CORBA.

#### 9.7.3 CORBA y la distribución

## 8. Sistemas operativos actuales, esquema de trabajo y operación.

### 1. CASO DE ESTUDIO: OS/2

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

#### Manejo de archivos en OS/2

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

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

En la tabla 1.1 se muestran algunas llamadas para la manipulación de archivos.

| Llamada        | Descripción                            |
|----------------|--|
| DosBufReset    | Graba al disco los buffers del archivo |
| DosClose       | Cierra el archivo                      |
| DosDelete      | Borra el archivo                       |
| DosDevIOCtl    | Establece parámetros de trabajo        |
| DosMkDir       | Crea un directorio                     |
| DosNewSize     | Cambia el tamaño de archivo            |
| DosFileInfo    | Obtiene información sobre el archivo   |
| DosSetFileInfo | Establece información del archivo      |
| DosOpen        | Abre un archivo                        |
| DosSetFileMode | Establece el modo de operación         |
| DosRmdir       | Borra un directorio vacío              |
| DosSelectDisk  | Selecciona un disco para trabajar      |

Tabla 1.1 Algunas llamadas de OS/2 para archivos

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

### Manejo de procesos en OS/2

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

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

La calendarización de procesos o 'threads' se hace por prioridad y dándoles un intervalo de ejecución a cada proceso o 'thread'. Se manejan tres niveles de prioridades: procesos preferentes, procesos preferentes interactivos y procesos normales.

OS/2 eleva a la categoría de preferentes a aquellos procesos que hacen mucha E/S.

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

### Manejo de memoria en OS/2

La versión inicial de OS/2 usaba segmentación pura debido sobre todo a las restricciones de los procesadores. Pero ya que el 80386 soportaba segmentación y paginación, IBM prometió un manejo de memoria virtual más sofisticado. El algoritmo de sustitución de segmentos era el 'Menos Recientemente

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

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

### Manejo de entrada/salida en OS/2

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

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

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

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

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

## 2. CASO DE ESTUDIO: WindowsNT

Windows NT es el nuevo sistema operativo de Microsoft. Fue diseñado para tomar ventaja de todo el poder que ofrecen los procesadores más avanzados de Intel, así como algunos de los procesadores RISC. Windows NT es la respuesta de Microsoft a UNIX. NT ofrece los mismos servicios que UNIX, interopera con redes UNIX pero reemplaza los comandos críticos de UNIX, su estructura de archivos

ARCANE y la mezcla de GUIs con una simple y estandarizada interfaz para el usuario como lo es Windows. Además, NT tiene las características que originalmente iba a tener el OS/2: un avanzado sistema operativo de 32 bits y compatibilidad con Windows GUI, además de soportar las aplicaciones hechas en DOS pero liberándose de las limitaciones de éste. Las características de diseño que hacen de Windows NT un sistema operativo avanzado son [MJS Jul-Ago92]:

**Extensibilidad:** El código podrá ser alterado (crecer o cambiar) de manera sencilla según cambien las necesidades del mercado.

**Portabilidad:** El código podrá utilizar cualquier procesador sin que esto afecte su desempeño de manera negativa.

**Confiabilidad y robustez:** El sistema deberá auto-protegerse tanto de los malos funcionamientos internos como de los externos. Así mismo se deberá comportar de manera predecible en cualquier momento y las aplicaciones no deberán afectar su funcionamiento en forma negativa.

**Compatibilidad:** El sistema se extenderá hacia la tecnología existente pero al mismo tiempo sus API y sus UI serán compatibles con los sistemas ya existentes de Microsoft.

**Multiprocesamiento y escalabilidad:** Las aplicaciones podrán tomar ventaja de cualquier computadora y los usuarios podrán correr las mismas aplicaciones tanto en una computadora de un procesador como en una multiprocesador.

**Cómputo distribuido:** NT será capaz de repartir sus tareas computacionales a otras computadoras en la red para dar a los usuarios más poder que el que tenga cualquier computadora por sí misma en la red. Podrá usar computadoras tanto local como remotamente de manera transparente al usuario (efecto de sinergia en red).

**Desempeño:** El sistema debe responder y ser lo más rápido posible en cada plataforma HW.

**Compatibilidad con POSIX:** POSIX (Portable Operating System based on UNIX) es un estándar especificado por el gobierno de los EU, el cual deberán de cumplir todos los contratos en el área computacional que sean vendidos a ese gobierno. NT puede proporcionar un ambiente opcional para la ejecución de aplicaciones POSIX.

**Seguridad certificable por el gobierno de EU:** El gobierno de EU estableció niveles de seguridad computacional como guías a cumplir para todas las aplicaciones gubernamentales. El rango de estos niveles va desde la D (menor) hasta la A (mayor), en donde la C y B tienen varios subniveles. NT puede soportar el C2 (el dueño del sistema tiene el derecho de decidir quién tiene permiso de acceso y el sistema operativo puede detectar cuándo los datos son accesados y por quién) pero en futuras versiones puede ser mejorada para alcanzar niveles de seguridad más altos.

## Características de WindowsNT

Un sistema operativo es un programa complejo que necesita un modelo unificado para asegurarse que el sistema puede acomodar sus características propias sin que éstas alteren el diseño. El diseño de Windows

NT fue guiado por una combinación de diversos modelos que fueron unidos en Windows NT. Los rasgos característicos de NT son [LenF93]:

- Direccionamiento de 32-bits.
- Soporte de memoria virtual.
- Preemptive multitasking.
- Soporte para multiprocesador.
- Arquitectura cliente/servidor.
- Seguridad e integridad del sistema.
- Compatibilidad con otros Sistemas Operativos.
- Independencia de plataformas.
- Networking (Interoperatividad).

### El núcleo de WindowNT

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

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

### Entradas y salidas de tareas al sistema

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

### Proceso de interrupciones y excepciones

En Windows NT se manejan las interrupciones como en cualquier sistema operativo. La llegada de señales por el bus debido a fallas de los programas o por peticiones de entrada/salida de los periféricos son atrapadas por el núcleo. En la Figura 2.1 se pueden observar las partes del núcleo de WindowsNT. El paso de los subsistemas de OS/2, POSIX y Win32 hacia los servicios del sistema se hace a través de mensajes y de atrapado de interrupciones.

### Sincronización de los multiprocesadores.

Esta característica asegura que sólo una tarea puede acceder un mismo recurso a la vez. En un sistema basado en multiprocesadores con memoria compartida, dos o más procesadores pueden estar ejecutando

tareas que necesitan acceder la misma página de memoria o realizar operaciones sobre un mismo objeto. El núcleo y el ejecutivo de NT proveen mecanismos para asegurar la integridad del sistema a través de la sincronización; en el caso del kernel la sincronización es manejada a través de candados colocados en puntos críticos de las instrucciones del nivel despachador, de esta manera, ningún otro procesador puede ejecutar código o acceder datos protegidos por uno de los candados de tipo spin hasta que éste es liberado. El ejecutivo del NT realiza la sincronización a través de la familia de los objetos de sincronización.

## Recuperación del sistema

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

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

## Arquitectura cliente/servidor

Windows NT tiene dos modos de operar, modo usuario y modo privilegiado (kernel). Programas de aplicaciones como una base de datos, una hoja de cálculo, o un sistema de reservaciones de un hotel, siempre son ejecutados en modo usuario. El ejecutivo de NT es el corazón del sistema. El ejecutivo de NT realiza tareas como el manejo de entradas y salidas, la memoria virtual, y todos los procesos, además de controlar las ligas entre NT y el hardware de la computadora. El ejecutivo de NT es ejecutado en modo kernel, el cual es un modo de alta seguridad libre de interferencias de los procesos de los usuarios. Consúltese la figura 2.1.

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

Adicionalmente el API tiene reglas de seguridad que protegen a los otros subsistemas de interferencias entre ellos.

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

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



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

El ejecutivo NT (Ver figura 2.1) es un sistema operativo completo que no cuenta con interfaz y está compuesto de cuatro capas, siendo éstas las siguientes:

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

**Componentes del ejecutivo:** el ejecutivo de NT tiene seis componentes primarios cada uno de los cuales realiza el siguiente conjunto de operaciones críticas del sistema: manejador de objetos, monitoreo de la seguridad del sistema, manejador de procesos, facilidad para la llamada de procesos locales, manejador de la memoria virtual y manejador de las entradas y salidas.

### Manejador de Objetos

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

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

- Asignar memoria.

- Asigna un descriptor de seguridad del objeto el cual permite o prohíbe el acceso a dicho objeto.

- Coloca el nombre del objeto dentro de la posición adecuada en el directorio de objetos.

- Crea y regresa un "manejador" o apuntador al objeto el cual elimina la necesidad de llamar al objeto por su ubicación.

### Monitor de la seguridad del sistema

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

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

### Manejo de archivos en WindowsNT

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

FAT (DOS)

HPFS (OS/2)

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

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

### Manejo de procesos en WindowsNT

En la arquitectura de NT los procesos son segmentados en componentes más pequeños llamados 'threads'. Windows NT soporta varias tareas al mismo tiempo. Existen dos tipos de multitarea, el apropiativo (preemptive) y el no apropiativo (no preemptive). Con la multitarea apropiativa la ejecución de un 'thread' puede ser suspendida después de un tiempo determinado (time slice) por el sistema operativo para permitir que otro thread sea ejecutado. Mientras que con la multitarea no apropiativa, es el thread el que determina cuándo le regresará el control al sistema operativo para permitir que otro thread sea ejecutado. NT así como OS/2 y UNIX usan preemptive multitasking para soportar la ejecución "simultánea" de varios procesos.

### Manejador de Procesos.

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

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

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

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

## Soporte para multiprocesador

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

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

## Seguridad e integridad del sistema

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

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

La protección individual de los procesos y del sistema operativo, para que en caso de un bug o de un programa destructivo no pueda hacer que el sistema se caiga o afecte a otros programas o aplicaciones. (Integridad del sistema) en el primer punto, el control sobre el acceso al sistema se refiere al manejo de user names y passwords para poder acceder al sistema operativo, de esta manera se mantienen a los usuarios sin autorización fuera del sistema. El siguiente nivel de seguridad en cuanto a este punto se refiere, son los privilegios que tiene un usuario, todos los usuarios o grupos de usuarios a los directorios y archivos del sistema, p.e. el acceso a los archivos del sistema de NT está estrictamente limitado al administrador del sistema, mientras que las aplicaciones comunes como lo son hojas de cálculo o procesadores de palabras pueden ser accesados por todos los usuarios.

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

## Control de Acceso y Seguridad del sistema.

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

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

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

### Integridad del sistema

Entendemos por integridad del sistema a la habilidad del mismo de permanecer activo cuando una de sus aplicaciones falla.

Windows NT está diseñado para prevenir la caída catastrófica del sistema en caso de que algunas de sus aplicaciones falle y para esto establece los siguientes cuatro mecanismos de protección de memoria:

Espacio de direcciones separado: cada proceso maneja sus propias direcciones virtuales y el sistema prohíbe el acceso a espacios de memoria de otros procesos.

Modos de Kernel y usuarios separados: todas las aplicaciones corren en modo de usuario por lo tanto está prohibido el acceso o modificación del código o datos del sistema que residan en el kernel.

Banderas de páginas: cada página de la memoria virtual tiene una bandera la cual determina cómo puede ser accesada en modo usuario y en modo kernel.

Seguridad de los Objetos: el manejador virtual de la memoria crea un tipo especial de objeto llamado objeto-sección el cual funciona como una ventana hacia la memoria virtual, por lo tanto cada vez que un proceso accesa un objeto-sección el sistema determina si el proceso tiene los permisos de lectura y/o escritura sobre éste.

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

### Manejo de memoria en WindowNT

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

## Direccionamiento de 32 bits

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

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

## Soporte de memoria virtual

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

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

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

## Manejador de memoria virtual

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

- Reservar y liberar la memoria virtual

- La lectura y escritura de páginas de memoria virtual

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

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

## Memoria protegida

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

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

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

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

## Manejo de entrada/salida en WindowsNT

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

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

requisición ha sido pasada a un manejador de dispositivos éste es responsable del control de las mismas a través de sistemas de colas.

### Compatibilidad con otros Sistemas Operativos

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

### Independencia de plataformas

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

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

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

### Interoperatividad (Networking)

Windows NT ofrece cuatro tipos diferentes de soporte de redes:

Punto a punto: En las conexiones punto a punto con otros sistemas Windows NT y Windows para grupos.

Interoperabilidad: con otros sistemas operativos orientados a red como lo son : DEC Pathworks, Novell Network, BanyanVINES a través de la arquitectura de sistemas abiertos de Windows ( WOSA ) , al igual que sistemas UNÍS basados en TCP/IP.

SNA: Conexiones a host basados en redes SNA a través de una propia versión de los servidores de comunicaciones de Microsoft DCA.

Soporte para redes Microsoft basadas en sistemas operativo de red LAN Manager.

## 3. CASO DE ESTUDIO: UNIX

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

## Estandarización de UNIX

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

| Estándar | Descripción                     |
|----------|---------------------------------|
| 1003.0   | Introducción y repaso.          |
| 1003.1   | Llamadas al sistema.            |
| 1003.2   | Intérprete y comandos.          |
| 1003.3   | Métodos de prueba.              |
| 1003.4   | Extensiones para tiempo real.   |
| 1003.5   | Lenguaje Ada.                   |
| 1003.6   | Extensiones para la seguridad   |
| 1003.7   | Administración del Sistema.     |
| 1003.8   | Acceso transparente a archivos. |
| 1003.9   | Lenguaje Fortran.               |
| 1003.10  | Supercómputo.                   |

Tabla 3.1 Los Estándares de POSIX

Al momento del auge de los estándares de POSIX desgraciadamente se formó un grupo de fabricantes de computadoras (IBM, DEC y Hewlett-Packard) que lanzaron su propia versión de UNIX llamada OSF/1 (de Open Software Foundation). Lo bueno fue que su versión tenía como objetivo cumplir con todas los estándares del IEEE, además de un sistema de ventanas (el X11), una interfaz amigable para los usuarios (MOTIF) y las definiciones para cómputo distribuido (DCE) y administración distribuida (DME). La idea de ofrecer una interfaz amigable en UNIX no fue original de OSF, ya en la versión 3.5 de SunOS de Sun Microsystems se ofrecía una interfaz amigable y un conjunto de librerías para crear aplicaciones con interfaz gráfica técnicamente eficiente y poderosa llamada SunWindows o SunVIEW. Esta interfaz junto con sus librerías estaban evolucionando desde la versión para máquinas aisladas hacia una versión en red, donde las aplicaciones podían estarse ejecutando en un nodo de la red y los resultados gráficos verlos en



otro nodo de la red, pero Sun tardó tanto en liberarlo que le dio tiempo al MIT de lanzar el X11 y ganarle en popularidad.

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

## Filosofía de UNIX

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

Todo se maneja como cadena de bytes: Los dispositivos periféricos, los archivos y los comandos pueden verse como secuencias de bytes o como entes que las producen. Por ejemplo, para usar una terminal en UNIX se hace a través de un archivo (generalmente en el directorio /dev y con nombre ttyX).

Manejo de tres descriptores estándares: Todo comando posee tres descriptores por omisión llamados 'stdin', 'stdout' y 'stderr', los cuales son los lugares de donde se leen los datos de trabajo, donde se envían los resultados y en donde se envían los errores, respectivamente. El 'stdin' es el teclado, el 'stdout' y el 'stderr' son la pantalla por omisión (default).

Capacidades de 'entubar' y 'redireccionar': El 'stdin', 'stdout' y el 'stderr' pueden usarse para cambiar el lugar de donde se leen los datos, donde se envían los resultados y donde se envían los errores, respectivamente. A nivel comandos, el símbolo de 'mayor que' (>) sirve para enviar los resultados de un comando a un archivo. Por ejemplo, en UNIX el comando 'ls' lista los archivos del directorio actual (es lo mismo que 'dir' en DOS). Si en vez de ver los nombres de archivos en la pantalla se quieren guardar en el archivo 'listado', el redireccionamiento es útil y el comando para hacer la tarea anterior es 'ls > listado'. Si lo que se desea es enviar a imprimir esos nombres, el 'entubamiento' es útil y el comando sería 'ls | lpr', donde el símbolo "|" (pipe) es el entubamiento y 'lpr' es el comando para imprimir en UNÍ BSD.

Crear sistemas grandes a partir de módulos: Cada instrucción en UNIX está diseñada para poderse usar con 'pipes' o 'redireccionamiento', de manera que se pueden crear sistemas complejos a través del uso de comandos simples y elegantes. Un ejemplo sencillo de esto es el siguiente. Suponga que se tienen cuatro comandos separados A,B,C y D cuyas funcionalidades son:

A: lee matrices checando tipos de datos y formato.

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

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

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

Como se ve, cada módulo hace una actividad específica, si lo que se quiere es un pequeño sistema que lea un sistema de ecuaciones y como resultado se tenga un listado 'bonito', simplemente se usa el entubamiento para leer con el módulo A la matriz, que su resultado lo reciba el B para obtener la solución, luego esa solución la reciba el módulo C para que le ponga los encabezados 'bonitos' y finalmente eso lo tome el módulo D y lo imprima, el comando completo sería 'A | B | C | D'. ¿Fácil no ?.

## Sistema de Archivos en UNIX

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

| Comando en UNIX | Comando en VMS   | Utilidad                  |
|-----------------|------------------|---------------------------|
| rm              | delete           | borra archivos            |
| cpb             | copy             | copia archivos            |
| mv              | rename           | renombra archivos         |
| ls              | dir              | lista directorio          |
| mkdir           | create/directory | crea un directorio        |
| rmdir           | delete           | borra directorio          |
| ln              | -                | crea una 'liga simbolica' |
| chmod           | set protection   | maneja los permisos       |
| chown           | set uic          | cambia de dueño           |

Tabla 3.2 Manejo de Archivos en UNIX y VMS

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

|     |     |     |
|-----|-----|-----|
| RWX | RWX | RWX |
| 1   | 2   | 3   |

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

## El núcleo de UNIX

El núcleo de UNIX (kernel) se clasifica como de tipo monolítico, pero en él se pueden encontrar dos partes principales [Tan92]: el núcleo dependiente de la máquina y el núcleo independiente. El núcleo dependiente se encarga de las interrupciones, los manejadores de dispositivos de bajo nivel (lower half) y

parte del manejo de la memoria. El núcleo independiente es igual en todas las plataformas e incluye el manejo de llamadas del sistema, la planificación de procesos, el entubamiento, el manejo de sentildes;ales, la paginación e intercambio, el manejo de discos y del sistema de archivos.

## Los procesos en UNIX

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

## El manejo de memoria en UNIX

Los primeros sistema con UNIX nacieron en máquinas cuyo espacio de direcciones era muy pequeño (por ejemplo 64 kilobytes) y tenían un manejo de memoria real algo complejo. Actualmente todos los sistemas UNIX utilizan el manejo de memoria virtual siendo el esquema más usado la paginación por demanda y combinación de segmentos paginados, en ambos casos con páginas de tamaño fijo. En todos los sistemas UNIX se usa una partición de disco duro para el área de intercambio.

Esa área se reserva al tiempo de instalación del sistema operativo. Una regla muy difundida entre administradores de sistemas es asignar una partición de disco duro que sea al menos el doble de la cantidad de memoria real de la computadora. Con esta regla se permite que se puedan intercambiar flexiblemente todos los procesos que estén en memoria RAM en un momento dado por otros que estén en el disco. Todos los procesos que forman parte del kernel no pueden ser intercambiados a disco.

Algunos sistemas operativos (como SunOS) permiten incrementar el espacio de intercambio incluso mientras el sistema está en uso (en el caso de SunOS con el comando 'swapon'). También es muy importante que al momento de decidirse por un sistema operativo se pregunte por esa facilidad de incrementar el espacio de intercambio, así como la facilidad de añadir módulos de memoria RAM a la computadora sin necesidad de reconfigurar el núcleo.

## El manejo de entrada/salida en UNIX

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

| Llamada | Función |
|---------|---------|
|---------|---------|

|            |   |
|------------|---|
| open       | Obtener un descriptor entero.                 |
| close      | Terminar las operaciones sobre el archivo     |
| lseek      | Posicionar la entrada/salida.                 |
| read,write | Leer o escribir al archivo (dispositivo)      |
| ioctl      | Establecer el modo de trabajo del dispositivo |

Tabla 3.3 Llamadas al sistema de entrada/salida

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

#### 4.CASO DE ESTUDIO: **Sistema Operativo Solaris**

##### CARACTERÍSTICAS:

Entre las características de Solaris tenemos:

**PORTABILIDAD:** El software conformado por una ABI aplicación de interfaces binaria (Application Binary Interface) ejecuta con un Shrink-wrapped (Contracción envuelta) el software en todos los sistemas vendidos con la misma arquitectura del microprocesador. Esto obliga a los desarrolladores de aplicaciones a reducir el costo del desarrollo del software y traer productos al mercado rápidamente, y obliga a los usuarios a actualizar el hardware mientras retienen sus aplicaciones de software y minimizan sus costos de conversión.

**ESCALABILIDAD:** Las aplicaciones se usan con más frecuencia en el sobre tiempo, y requiere sistemas más poderosos para soportarlos. Para operar en un ambiente creciente, el software debe ser capaz de ejecutar en un rango de ancho poderosos y debe ser capaz de tomar ventajas del poder adicional que se está procesando.

**INTEROPERATIVIDAD:** La computación del ambiente heterogéneo es una realidad hoy. Los usuarios compran de muchos vendedores para implementar la solución que necesitan. La

estandarización y una clara interface son criterios para un ambiente heterogéneo, permitiendo a los usuarios desarrollar estrategias para comunicarse por medio de su red. El sistema operativo de Solaris puede interoperar con unos sistemas muy populares hoy en el mercado, y aplicaciones que se ejecutan en UNIX se pueden comunicar fácilmente.

**COMPATIBILIDAD:** La tecnología de la computación continua avanzando rápidamente, pero necesita permanecer en el ámbito competitivo para minimizar sus costos y maximizar sus ingresos.

### *CARACTERISTICAS PARA LOS USUARIOS*

Dentro de las características de los usuarios tenemos:

#### **ESPACIO DE TRABAJO PARA EL ADMINISTRADOR**

(A workspace manager): cuenta con una ventana de manejo de servicios rápidos (open, close, more, etc.), así como herramientas el cual le permite al usuario entallar su espacio de trabajo a sus necesidades personales.

**INTEGRACION DE SERVICIOS DESKTOP (Desktop Integration Services):** incluyen ToolTalk, Drag and Drop (arrastrar y soltar), y cut and paste (cortar y pegar), proporcionando la base para que a las aplicaciones puedan integrarse unos con otros.

**BIBLIOTECAS GRAFICAS (Graphics Libraries):** incluye XGL, Xlib, PEX, y XIL, proporcionando soporte para aplicaciones de 2D y 3D.

**ADMINISTRADOR DE CALENDARIO (Calendar Manager):** posee una aplicación de administrador de tiempo que despliega citas y todos los compromisos del día, semana, o un mes en una ojeada. También contiene un Multibrowse que hace un programa de reuniones entre un grupo de usuarios más fácil. Varios calendarios pueden ser cubiertos simultáneamente para determinar la conveniencia de la hora de una reunión en una ojeada.

**HERRAMIENTA DE IMAGEN (Image Tool):** permite cargar, ver y salvar imágenes en 40 diferentes formatos incluyendo PICT, PostScript (TM), TIFF, GIF, JFIF, y muchas más.

*Otras herramientas incluyen una herramienta de impresión, audio, shell, reloj, y editor de texto.*

## **CARACTERISTICAS PARA EL ADMINISTRADOR DEL SISTEMA**

*El Sistema Solaris ofrece una variedad de herramientas nuevas para el administrador como lo son:*

**Dispositivo de Información:** los administradores pueden usar estos accesorios opcionales para obtener información sobre dispositivos instalados incluyendo nombres, atributos, y accesibilidad.

**Sistema de Administración de Archivo:** estos accesorios permiten a los administradores crear, copiar, amontonar, depurar, reparar y desmontar sistemas de archivos, crear y remover cadenas de archivos y nombrar tuberías o pipes, y manejar volúmenes.

**Manejo del Proceso:** este controla la agenda de control del sistema. Usando estos accesorios, administradores pueden generar reportes sobre el desempeño, entrada de identificación, ubicación del acceso a discos, y buscar la manera de afinar el desempeño del sistema.

**Usuarios y el manejo del grupo:** con estos accesorios, un administrador puede crear y eliminar entradas en grupos y entradas de identificación del sistema, y asignar grupos y IDs de usuario.

**Seguridad:** El ASET (Automated Security Enhancement Tool) es un accesorio que incrementa la seguridad porque permite a los administradores de sistemas revisar archivos del sistema incluyendo permisos, pertenencia, y contenido del archivo. El ASET alerta a los usuarios acerca de problemas de seguridad potencial y donde es apropiado colocar el sistema de archivos automáticamente de acuerdo a los niveles de seguridad especificados.

## **PAQUETES DE SOFTWARE Y CLUSTERS**

El software del sistema de Solaris es entregado en unidades conocidos como paquetes. Un paquete es una colección de archivos y directorios requeridos para el producto de un software. Un cluster (racimo) es una colección de paquetes.

Hay 4 tipos de clusters:

**Núcleo del Soporte del Sistema (Core System Support):** es el software de configuración mínima; contiene solo el software necesario para iniciar el funcionamiento del computador y ejecutar el ambiente operativo de Solaris.

**Sistema de Soporte para Usuarios Finales (End User System Support):** contiene el Núcleo del Soporte del Sistema más el Sistema de soporte para usuarios finales, como lo es el Open

Windows sistema de ventanas y aplicaciones de archivos DeskSet relacionados; este cluster incluye el software recomendado para un usuario final.

Soporte de Sistemas Desarrollados (Developer System Support): contiene soporte de usuario final del sistema más librerías, incluye archivos y herramientas que se necesitan para desarrollar el software en el sistema de Solaris. Compiladores y depuradores no están incluidos en el sistema de Solaris 2.5.

Distribución Entera (Entire Distribution): contiene todo el ambiente de Solaris.

### *ADMINISTRACION DEL PAQUETE*

El manejo de paquetes de software simplifica la instalación y actualización del software. La administración es simplificada porque el método de manejo del software del sistema y aplicaciones de terceros son ahora consistentes. Las herramientas para crear paquetes de software están en un paquete de aplicaciones de herramientas de biblioteca.

Hay 2 herramientas que se pueden utilizar para instalar y remover paquetes:

Programa de Interface Gráfica (A graphical user interface program): se puede instalar un software en un sistema local o en un sistema remoto con Admintol (comenzando con el comando Admintol). Se utiliza Admintol para:

Ver el software instalado en un sistema local.

Instalar o remover un software en un sistema local.

El comando de línea de accesorios (The command-line utilities): se utiliza para instalar, remover, y revisar la instalación del paquete de software.

### SERVICE ACCESS FACILITY (SAF)

**El SAF es una herramienta usada para administrar terminales, modems, y otros dispositivos de red. En particular, el SAF permite:**

Añadir y administrar ttymon and listen monitores en puertos (usando el comando sacadm)

Añadir y administrar ttymon servicios de monitores en puertos (usando los comandos pmadm y ttyadm)

Añadir y administrar listen servicios de monitores en puerto (usando los comandos pmadm y nlsadmin)

## ~~Administra y troubleshoot de dispositivos TTY.~~

Administrar y troubleshoot entradas de requisitos de red para servicios de impresión.

Administrar y troubleshoot el controlador de acceso al servicio (Service Access Controller) usando el comando sacadm.

*El SAF no es un programa. Es una jerarquía de últimos procesos y comandos de administración. El nivel tope del programa SAF es el SAC. El SAC (service access controller) controla monitores de puerto que se pueden administrar por el comando sacdm. Cada puerto de monitor puede manejar uno ó más puertos.*

### **Funciones del SAF y Programas Asociados**

| <b>Función</b>         | <b>Programa</b> | <b>Descripción</b>                                      |
|------------------------|-----------------|---|
| Overall Administration | sacadm          | Comando para añadir y<br>Remover monitores de<br>puerto |

|  |            |                               |
|--|------------|-------------------------------|
| <b>Controlador de servicio de acceso</b> | <b>sac</b> | <b>Programa maestro SAF's</b> |
|--|------------|-------------------------------|

|                            |               |  |
|----------------------------|---------------|--|
| <b>Monitores de Puerto</b> | <b>ttymon</b> | <b>Monitores de puerto serial</b>                    |
|                            |               | Con requerimientos de login                          |
|                            | listen        | Requerimientos de monitores<br>para servicios de red |

|                                      |       |   |
|--------------------------------------|-------|---|
| Administrador de Monitores de puerto | pmadm | Servicios de control de monitores de puerto |
|--------------------------------------|-------|---|

|                  |                |                                     |
|------------------|----------------|-------------------------------------|
| <b>Servicios</b> | <b>logins;</b> | <b>Servicios al cual SAF provee</b> |
|                  | Llamada de     | acceso                              |



procedimiento  
remoto; etc.

## ***CONTROLADORES DE INTERFACES DE DISPOSITIVOS***

La intención de Solaris 2.5 SPARC DDI/DKI es de proporcionar una compatibilidad de los dispositivos que soporten las plataformas y para todas las futuras innovaciones del ambiente de Solaris 2.5 en esas plataformas.

En el ambiente operativo de Solaris 2.5 hay un nuevo conjunto de dispositivos de interfaces. Los dispositivos de interface en el ambiente operativo de Solaris2.5 están formalizados y son referidos como Solaris 2.5 SPARC DDI/DKI.

El término DDI/DKI es derivado de la especificación original que se utiliza como suministro del SVR4 (System V Release 4). DDI/DKI significa device driver interface/driver kernel interface. Las interfaces se dividen en 3 grupos:

DDI/DKI

DKI only

DDI only

DDI/DKI: se estandarizó en el SVR4, y son genéricos a lo largo de todas las implementaciones del SVR4, independientemente de la plataforma en la que se ejecuta.

DDI only: son genéricos como las interfaces de DDI/DKI y son soportados en todas las implementaciones del SVR4. Por otro lado, no son garantizados para ser soportados en el Solaris V.

DKI only: están destinados a ser de una arquitectura específica; por ejemplo, métodos para acceder y controlar dispositivos y sistemas de hardware específico (archivos de E/S, servicios de DMA, interrupciones, y memoria de mapeo). Estas interfaces no están garantizados para trabajar

en otras implementaciones de SVR4.

Estos dispositivos, combinados con un gran número de plataformas SPARC, son una ayuda a nuevos desarrolladores de hardware.

En el Solaris 2.5 DDI/DKI solo el DDI only son genéricos a todos los sistemas Solaris basados en SPARC que soportan Solaris 2.5 DDI/DKI.

## ***KERNEL***

El kernel del Solaris tiene multithread. En vez de una llave maestra. Hay muchas llaves pequeñas que protegen pequeñas regiones de código. Por ejemplo, puede haber una llave de kernel que protege el acceso a un nodo particular, y uno que protege un nodo. Solo un procesador puede estar ejecutando códigos relacionados con ese nodo a la vez, pero otro podría estar accediendo a un nodo. Esto permite mayor concurrencia.

El kernel de multithread tendrá mayor impacto en como está diseñado el controlador.

## ***INSTALACION***

Hay 3 formas para instalar un sistema Solaris las cuales son:

Que hacer antes de instalar un software de Solaris

Salvar información de configuración

Salvar la información del sistema de archivo

No importa como se planea manejar la transferencia de datos, debe hacer un respaldo de todos sus discos. Debido a que los nombres de conversión son diferentes en el ambiente operativo de Solaris, es posible confundirse e inadvertidamente seleccionar el disco de trabajo equivocado cuando se instala el software de Solaris. Hacer respaldo de los archivos del sistema antes de comenzar la instalación ofrece protección si esta clase de accidente llega a pasar.

Antes de comenzar el proceso de instalación, se debería tener una copia dura (es decir, en papel) de las particiones de discos existentes en el sistema. Si esta información se salva en línea, puede ser sobre escrita durante la instalación. Salvar esta información sirve como referencia de muchas decisiones que se hacen acerca de la configuración del sistema de Solaris.

## ***COMANDOS***

Entre los comandos de Solaris tenemos los siguientes:

/etc/system: Es en donde se encuentra la información de la configuración del sistema.

ufsboot: iniciar el comportamiento del computador de disco a disco

inetboot: iniciar el comportamiento del computador a través de la red

bootblk: descarga ufs al iniciar el funcionamiento del computador en el disco

unix: **bootable imagen de kernel**

/etc/rcs: **amontona y copia unix de la red**

/etc/rc2,/etc/rc3,/etc/rc2.d,/etc/rc3.: **scripts de la configuración del sistema**

mod load, /etc/system,add\_drv,rem\_drv: **customizes el sistema de kernel, carga, adiciona, y remueve modelos como sea necesario**

run states 0-6, and s: **niveles de ejecución del sistema**

add\_drv: **informa al sistema que hay un nuevo dispositivo controlador instalado**

rem\_drv: **informa al sistema que el controlador especificado ya no es valido**

modload: **carga el modulo especificado en el sistema en ejecución**

modunload: **descarga el modulo especificado del sistema en ejecución**

pkgadd: **para instalar un paquete**

pkgrm: **para remover un paquete**

pkgchk: **para revisar la instalación del paquete**

pkginfo: **para hacer un listado de paquetes instalados en el sistema**

## **VERSIONES**

Entre las versiones de Solaris tenemos:

Solaris 1.x al Solaris 2.x

Solaris DDI/DKI

Solaris V

## **BIBLIOGRAFIA**

**[WWW.SUN.COM](http://WWW.SUN.COM)**

## CONCLUSIÓN

*El ambiente del Sistema Operativo Solaris nos brinda algunas características, las cuales ayudarán a dar una buena portabilidad, escalabilidad, compatibilidad y seguridad en las aplicaciones para así operar en un ambiente creciente. También presenta características para los usuarios la cual cuenta con ventanas para manejo de servicios rápidos, al igual que integra servicios desktop, bibliotecas gráficas, administración de calendario y herramientas de imagen. Este sistema tiene características para el administrador de sistemas en donde se puede obtener información sobre dispositivos.*

*El sistema operativo Solaris brinda paquetes de software (colecciones de archivos y directorios) y clusters (colección de paquetes). El kernel de solaris tiene multithread en vez de llave maestra la cual brinda mayor impacto en como está diseñado el controlador. Su instalación se basa en 3 formas como los son el salvar la información de la configuración, del sistema de archivo y que hacer antes de instalar un software de Solaris.*

*Este sistema cuenta con comandos los cuales nos ayudarán a manejar el sistema; con algunos de ellos podemos revisar, instalar y remover paquetes.*

## 7. Bloqueos

### 7.1 Recursos

### 7.2 El algoritmo del avestruz

### 7.3 Evasión de bloqueos

### 7.4 Prevención de bloqueos

## 9. Sistemas operativos distribuidos

### 9.1 Conceptos básicos, Historia y objetivos de diseño.

### 9.2 Comunicación en los sistemas distribuidos

### 9.3 Sincronización

### 9.4 Procesos y procesadores en sistemas distribuidos

9.4.1 Hilos

9.4.2 Modelos de sistemas

9.4.3 Asignación de procesadores

9.4.4 Planificación en sistemas distribuidos

9.5 Sistemas distribuidos de archivos

9.5.1 Diseño

9.5.2 Implantación

9.5.3 Tendencias en los sistemas distribuidos de archivos

9.6 Memoria compartida distribuida.

9.7 Sistemas distribuidos: CORBA

9.7.1 Sistemas basados y orientados a objetos.

9.7.2 Componentes de CORBA.

9.7.3 CORBA y la distribución

## **Relación de practicas**

1. Estudio e instalación de Windows 9X: 98, NT, 2000 Server.
2. Estudio e instalación de Linux
3. Estudio e instalación de Solaris
4. Elaborar un programa para trabajar con las principales interrupciones BIOS
5. Crear un cargador de arranque en un lenguaje de alto nivel
6. Desarrollar un prototipo de SO por equipos de 3 alumnos en lenguaje C