



**Ingeniería en Desarrollo de Software**  
**3<sup>er</sup> semestre**

Programa de la asignatura:  
**Programación de sistemas operativos**

**Unidad 1. Administración de procesador, memoria y  
dispositivos de E/S**

**Clave:**

Ingeniería  
**15142317**

TSU  
**/ 16142317**

**Universidad Abierta y a Distancia de México**





### Índice

Unidad 1. Administración de procesador, memoria y dispositivos e/s .....	3
Presentación de la unidad .....	3
Propósito .....	4
Competencia específica.....	5
1.1. Procesador .....	5
1.1.1. Definición y conceptos .....	8
1.1.2. Procesos e hilos.....	10
1.1.3. Multiprocesamiento .....	15
1.1.4. Paralelismo .....	16
1.2. Memoria.....	17
1.2.1. Administración de la memoria .....	17
1.2.2. Jerarquía de la memoria .....	19
1.2.3. Multiprogramación con particiones fijas y variables.....	20
1.2.4. Memoria real .....	22
1.2.5. Memoria virtual .....	24
1.3. Entrada / Salida .....	27
1.3.1. Principios de hardware y software de E/S .....	28
1.3.2. Dispositivos y manejadores de dispositivos .....	29
1.3.3. Mecanismos y funciones de manejadores de dispositivos .....	32
1.3.4. Estructura de datos en el manejo de dispositivos.....	35
1.3.5. Operaciones de E/S .....	38
Cierre de la unidad .....	41
Para saber más .....	42
Fuentes de consulta .....	42



### Unidad 1. Administración de procesador, memoria y dispositivos E/S

#### Presentación de la unidad

Bienvenido(a) a la asignatura de Programación de sistemas operativos. En esta primera unidad se trabajará y entenderá cómo el sistema operativo hace uso del procesador administrándolo. El procesador es el cerebro de la computadora o donde se realizan las operaciones, también es conocido como CPU. En esta unidad también se trabajará con las partes que forman al CPU, cómo cambia de un proceso a otro, cómo manipula las actividades; así mismo, se entenderá cómo es que controla varios procesos a la vez o los hace de forma paralela. Cuando se entienda cómo el sistema operativo administra el procesador, se entenderá la forma en que controla los accesos a la memoria descargando y cargando información en ella, se entenderá la diferencia entre memoria real y virtual y para qué se usa cada una de ellas, y por último la gran ventaja de controlar las operaciones que se realizan en los dispositivos de entrada y salida.

El procesador está formado por planificadores en los cuales se decide qué se va a ejecutar primero y que se ejecuta después, además de qué va a dar respuesta de manera inmediata y qué puede esperar, todo esto con la finalidad de optimizar varios puntos, como: velocidad en respuesta, certeza, excelencia en respuesta, etc., de esta manera se entiende que los planificadores son las partes lógicas del procesador que analizan los datos recibidos y toman decisiones de respuesta respecto a la velocidad y prioridad.

Existen diferentes tipos de planificadores en un procesador los cuales se enlistan a continuación:

- **Planificador a largo plazo:** con éste el procesador decide el orden de ejecución con una adecuada organización de los recursos para que el trabajo se ejecute de forma ordenada y eficiente, según el modo de procesamiento.
- **Planificador a medio plazo:** este planificador es responsable de cargar y descargar trabajos desde el disco a la memoria y de la memoria al disco considerando el grado de sobrecarga del sistema.



- **Planificador a corto plazo:** Es quien decide cuándo, cómo y por cuánto tiempo recibe el procesador un proceso que está preparado para ejecutar.

Se debe considerar que el primer componente para que un sistema operativo funcione es el procesador.

El segundo componente es la memoria; la principal necesidad para cualquier sistema operativo es que la memoria sea rápida para que al ejecutar una instrucción la CPU o procesador no se vea afectado por la memoria de bajo rendimiento.

Dentro de un sistema operativo, la memoria no es el único recurso que debe administrarse, los dispositivos de Entrada/Salida también interactúan con el sistema operativo y por lo regular constan de dos partes: una tarjeta controladora y el dispositivo en sí, la tarjeta controladora es un chip o un conjunto de chips montados en una tarjeta insertable, que controla físicamente al dispositivo. Dicha tarjeta controladora acepta comandos del sistema operativo y los ejecuta.

Considerando entonces que el tercer componente son los dispositivos en sí. Los dispositivos tienen interfaces relativamente simples, debido a que no tienen mucha capacidad para estandarizarlos, esto último es necesario para que cualquier controladora de disco IDE pueda manejar cualquier disco IDE (electrónica integrada en la unidad, Integrated Drive Electronics. Por sus siglas en inglés).

### Propósito

Al término de esta unidad lograrás:

- Identificar cómo el sistema operativo interactúa, maneja el control del procesador y sus procesos, de la memoria y sus tipos de particiones y administración; sin dejar de tomar en cuenta el control de los dispositivos de E/S; sus principios, manejadores, mecanismos, estructuras y operaciones.



### Competencia específica

- Describir la administración del procesador, memoria y dispositivos de entrada y salida para reconocer las características que distinguen cómo interactúan con el sistema operativo mediante las transiciones entre ellos.

### 1.1. Procesador

El procesador o unidad central de procesamiento, comúnmente conocido como CPU por sus siglas en inglés Central Processing Unit, es la unidad que procesa todas las tareas que realiza la computadora, permitiendo el procesamiento de información numérica ingresada en formato binario, así como el proceso de instrucciones almacenadas en la memoria.

La CPU es un dispositivo electrónico el cual funciona enviando pulsos eléctricos a razón de Hertz, una CPU con 50 MHz enviará 50.000.000 pulsos eléctricos por segundo, mientras el procesador ejecuta una acción que corresponde a su vez a una instrucción o bien a una parte de ella.

La medida **CPI** (ciclos por instrucción) representa el número promedio de ciclos necesarios para que el microprocesador ejecute la instrucción.

La potencia del microprocesador se mide por el número de instrucciones por segundo que es capaz de procesar.

Cada CPU ejecuta un conjunto de instrucciones específicas a su arquitectura, por lo tanto no se pueden ejecutar programas de una arquitectura de un tipo de procesador a otro con estructura de manejo de proceso diferente. Tener acceso a la memoria para obtener una instrucción o una palabra de datos tarda mucho más que ejecutar una instrucción, todas las CPU contienen algunos registros para guardar variables importantes y resultados temporales. Así, el conjunto de instrucciones por lo general incluyen instrucciones para



cargar una palabra de la memoria en un registro, y para almacenar en la memoria una palabra que está en un registro.

“El sistema operativo debe tener conocimiento de todos los registros, al multiplicar en el tiempo la CPU, es común que el sistema operativo tenga que detener el programa en ejecución para iniciar o reiniciar otro. Cada vez que el sistema operativo detiene un programa en ejecución, debe guardar todos los registros para que puedan restablecer cuando el programa continúe su ejecución justo donde se quedó” (Tanenbaum, 2003, p. 22).

En la actualidad muchas CPU cuentan con recursos para ejecutar más de una instrucción a la vez, una CPU podría tener unidades individuales para tomar, decodificar y ejecutar de modo que mientras esté ejecutando la instrucción, también pueda estar decodificando la instrucción  $n+1$  y tomando la instrucción  $n+2$ .

El procesador realiza e interpreta una serie de instrucciones contenidas de un software y procesa los datos que este genera. Existen un gran número de modelos y marcas de procesadores que van dependiendo del tipo de socket, CoreClock, FSB, Cache, etc.

Instrucciones es una operación que el procesador debe de realizar; esas instrucciones se almacenan en la memoria principal del procesador y utilizan dos campos: código de operación que representa la acción que el procesador debe ejecutar; código operando, que es el que define los parámetros de la acción.

Las instrucciones se agrupan a su vez en categorías como:

- Acceso a memoria: es la transferencia de información entre registros.
- Operaciones aritméticas: es la parte que se encarga de realizar las operaciones, como suma, resta multiplicación o división.
- Operaciones lógicas: se encarga de realizar las operaciones Y, O, No, etc.
- Control: se encarga de realizar las conexiones condicionales, etc.





Debido a la exclusión de elementos de hardware complejos, la realización del trabajo por la capa de software, los procesadores poseen una estructura interna simple y muy eficiente, compuesta por:

- Unidades enteras (Integer ALU).
- Unidad de coma flotante (floating point unit).
- Unidad de memoria para operaciones de lectura y escritura (Load/Store Unit).
- Unidad de salto (Branch unit).

Al ejecutarse todas las instrucciones, la información que se genera se almacena de forma temporal en ubicaciones de memoria local, denominada registros.

Algunos de los registros comúnmente utilizados son:

- Acumulador (ACC), que almacena los resultados de las instrucciones de operación aritmética y lógica.
- Estado (PSW), que contiene los indicadores de estado del sistema.
- Instrucción (RI), que almacena las instrucciones que se va procesando.
- Contador ordinal (OC).

Casi todas las computadoras tienen varios registros especiales que puede ver el programador. Uno de ellos es el contador de programa, el cual contiene la dirección de memoria en la que se encuentra la siguiente instrucción que será tomada. Una vez obtenida esa instrucción, el contador de programa se actualizará de modo que apunte a su sucesora. Otro registro es el apuntador de pila, que apunta a la parte superior de la pila actual en la memoria. La pila contiene un marco por cada procedimiento en el que se ha entrado, pero del cual no se ha salido aún. El marco de pila de un procedimiento contiene los parámetros de entrada, variables locales y variables temporales que no se guardan en registros.



### 1.1.1. Definición y conceptos

El procesador es el encargado de procesar los datos que están almacenados en una computadora para producir cualquier tipo de información de interés para los usuarios.

Un microprocesador se divide en dos componentes básicos, los cuales son:

1. La unidad lógica/aritmética (ALU), que realiza operaciones aritméticas y lógicas, dadas a través de microprogramación.
2. La unidad de control (CU), que extrae instrucciones de la memoria, las descifra y ejecuta, llamando a la ALU cuando es necesario; sincroniza todas las operaciones de áreas de almacenamiento principal llamadas registros y la memoria caché.

El procesador se necesita para el funcionamiento de ciertas áreas de almacenamiento y que son de dimensiones mínimas; sin embargo, tienen la ventaja de su rapidez. Comparados con los accesos a RAM, los registros son como mínimo 10 veces más veloces.

Existen 4 registros llamados AX, BX, CX y DX, cada uno de ellos tienen asignada una actividad específica.

- AX: nombrado acumulador; contiene el operador que interviene en las operaciones aritméticas y lógicas. En general las instrucciones que trabajan con este registro tienen un microcódigo más simple que la misma instrucción ejecutada con otro registro.
- BX: conocido como registro base, comúnmente la tarea de este registro se basa en identificar la dirección o índice de cada registro dentro de una tabla de valores para su direccionamiento.
- CX: es el contador, las instrucciones que se realizan durante un proceso dentro del sistema operativo conocidas como bucle (LOOP) utilizan este registro como contador de n veces que un ciclo repite dentro de un proceso.





- DX: es el registro de datos multiuso, se utiliza en operaciones de multiplicación y división junto con AX, su aplicación suele estar dentro de operaciones de entrada/salida de puertos IN/OUT.

Los registros de segmento son utilizados para integrar las direcciones de otros segmentos para su aplicación en conjunto con otros registros que señalan las direcciones concretas dentro de estos segmentos, lo que permite la gestión total de la memoria direccional, para estos segmentos básicamente se contemplan cuatro tipos:

- Segmento de código CS: apunta la ubicación del segmento de código del programa, el cual se está ejecutando en ese instante.
- Segmento de datos DS: señala la ubicación del segmento de datos del programa en ejecución.
- Segmento de Pila SS: apunta la ubicación del segmento donde está la pila del programa en ejecución.
- Segmento extra ES: es un segmento auxiliar, es utilizado para indicar la existencia de espacio extra de algunos de los segmentos anteriores, es útil también para guardar de forma temporal las direcciones intermedias entre segmentos.

Los registros de puntero están destinados a contener direcciones que son desplazamientos dentro de los segmentos indicados por los registros de segmento.

- Puntero de instrucción IP y contador de programa PC, establece la dirección de la instrucción e indican el desplazamiento de la instrucción siguiente a ejecutar que ubica dentro del segmento CS.
- Puntero de Pila SP: señala la ubicación final de la pila dentro del segmento de pila SS.
- Puntero base BP: señala el desplazamiento donde se encuentra el origen de la zona ocupada por las variables dinámicas.

También, dentro de los registros se encuentra uno especial llamado registro de estado (FLAGS), que es un indicador del estado del procesador y el resultado de algunas operaciones, ya que su función se basa en el manejo de bits que actúan como semáforo.



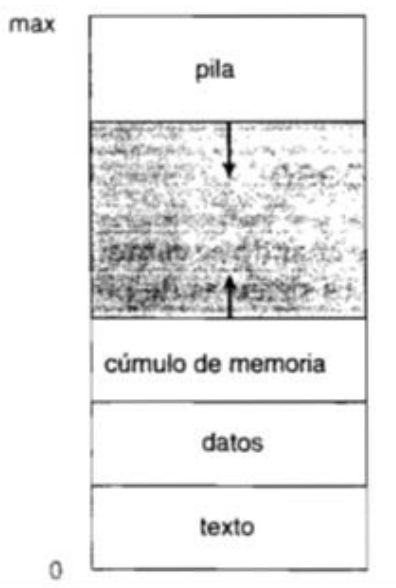
### 1.1.2. Procesos e hilos

Puede considerarse como definición de un proceso a la ejecución de un programa dentro de un sistema operativo, el cual puede necesitar de ciertos recursos determinantes para su ejecución tales como: tiempo de CPU, archivos de Entrada/salida, dispositivos de entrada/salida, y memoria principalmente, todos los recursos que utiliza se asignan al proceso en el momento de que es creado o bien al momentos de ser ejecutado. Cabe mencionar que cada uno los procesos puede tener su propia CPU virtual.

En la gran mayoría de los sistemas, los procesos suelen ser la determinante para la realización de los trabajos. Cada sistema contiene sus propios procesos, y se dividen en los procesos del sistema operativo que son los encargados de ejecutar el código del sistema y los procesos de usuarios encargados de ejecutar el código del usuario, por lo tantos estos procesos se pueden ejecutar de manera simultánea para la realización de tareas interactivas.

Durante la ejecución del código de un proceso, también utiliza la actividad actual que está representada por el valor del contador de programa y por los contenidos de los registros del procesador. Por lo general de la misma forma, un proceso también incluye la pila del proceso, lo cual puede contener los datos temporales: parámetros de funciones, direcciones de retorno y las variables locales además de una sección de datos que contienen las variables globales.

Un proceso puede incluir una acumulación de memoria, que es asignada de forma dinámica al proceso en tiempo de ejecución, en la siguiente figura, se presenta la estructura de un proceso ubicado dentro de la memoria, indicando el estado que va tomando a medida que el proceso es ejecutado.



Ejecución de un proceso ubicado en la memoria  
Tomada de Silberschatz (2006, p. 74).

A continuación se expondrá la diferencia entre un programa y un proceso. Un programa por su estructura propia no es un proceso, ya que es una entidad pasiva, por ejemplo un archivo lleva incluido una cantidad de instrucciones que se almacenan en disco. Un proceso es una entidad activa, la cual contiene las instrucciones que se deben ejecutar y un conjunto de recursos asociados a los registros de la CPU para su funcionamiento.

Un programa se convierte en proceso al cargarse en memoria, lo cual lo convierte en archivo ejecutable. Un ejemplo que ayude a distinguir la diferencia es el de una persona que se dedica a la construcción, y está a punto construir una casa, para esto cuenta con planos de la construcción y todo el material necesario para realizar la obra: cemento, cal, arena, tabiques, varilla, alambre, clavo, madera, etc., en este ejemplo los planos son el programa, la persona de la construcción es la CPU y los elementos o materiales para la construcción son los datos de entrada. El proceso es la actividad de la persona que consiste en leer los planos, obtener los materiales y construir la casa.

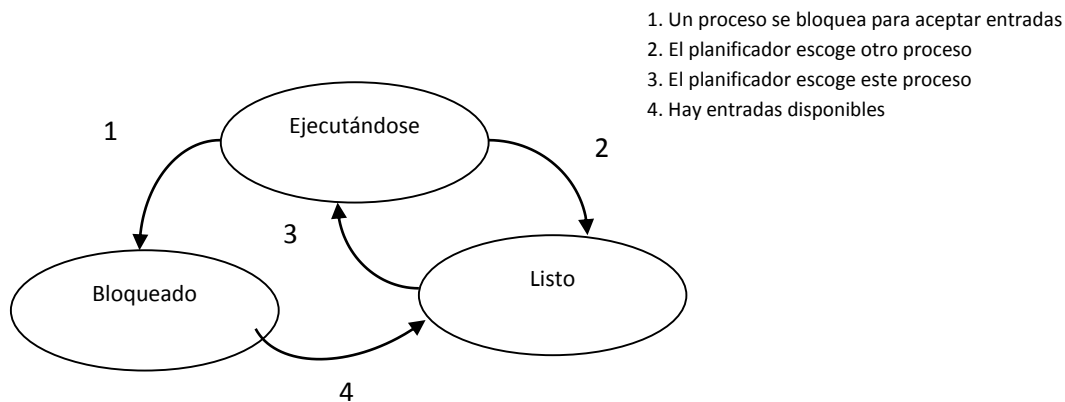
La idea del proceso consiste en que es una actividad de algún tipo: tiene programa, entrada, salida y un estado. Se puede compartir un procesador entre varios procesos, usando algún



algoritmo de planificación para determinar cuándo debe dejarse de trabajar en un proceso para atender a uno distinto.

La jerarquía de los procesos que manejan el concepto de proceso debe contar con algún mecanismo para crear todos los procesos necesarios, en los sistemas muy sencillos o en los diseñados para ejecutar sólo una aplicación es posible que, cuando el sistema se inicia, todos los procesos que puedan necesitarse estén presentes. Sin embargo, en la mayor parte de los sistemas se necesita algún mecanismo para crear y destruir procesos según sea necesario durante la operación.

En el estado de procesos, aunque cada proceso es una entidad independiente, con su propio contador de programa y estado interno, a menudo necesitan interactuar con otros procesos. Un proceso podría generar ciertas salidas que otro proceso utiliza como entradas. Mientras un proceso es ejecutado va modificando el estado, dependiendo de la actividad que esté en memoria, en la siguiente figura se muestra el estado de un proceso



Un proceso puede estar en el estado de ejecutándose, bloqueado o listo. Las transiciones entre estos tres estados son las que se muestran en el esquema. Tomada de Tanenbaum (2003, p. 51).

Habitualmente, la mayoría de los procesos están creados por el sistema operativo de tal forma que los usuarios o el programa de aplicación no conozcan su procedencia. Sin embargo, es bastante útil para el rendimiento de memoria y administración de recursos dentro de los procesos de la CPU, que de un proceso se pueda derivar en otro proceso; el



nuevo proceso se podría ejecutar de forma paralela con la aplicación y podría ser activado cada que se mande a llamar.

El bloqueo de control de procesos en un sistema operativo es representado mediante un bloque de control de procesos. Un bloque puede integrar algunos de los elementos de información que especifiquen el estado de un proceso, algunos de estos elementos son:

- Estado del proceso: en este estado el proceso puede tomar la forma básica de inicio, nuevo, listo, ejecutando, en espera, detenido, finalizado, etc.
- Contador de programa: el contador de programa identifica la ubicación de la siguiente instrucción que se ejecutara por el proceso.
- Registros en CPU: los diferentes registros que se pueden utilizar para el proceso dependen del tipo y número, así como de la arquitectura de la CPU, estos registros se incluyen los tratados en el tema anterior.
- Estado y planificación de CPU: la planificación de CPU establece qué proceso tiene mayor prioridad para el buen funcionamiento y manejo de recursos.
- Administración de memoria: muestra el valor de los registros, tablas de páginas y de segmentos.
- Información contable: este elemento contiene la cantidad de CPU de tiempo real empleado, límites de tiempo asignados y números de cuenta.
- Estado de E/S: este elemento muestra la lista de los dispositivos de E/S asignados al proceso.

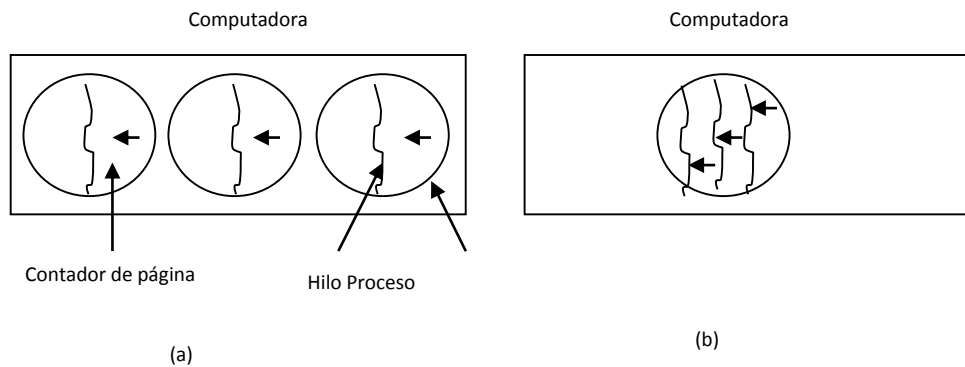
Por otro lado, en la finalización de un proceso debe ser determinante que un proceso indique la finalización de su tarea. Para esto la aplicación deberá detectar una llamada o petición explícita sobre el servicio del sistema operativo que determine la finalización del proceso demandante.

Durante un proceso tradicional, que representa a una actividad en un proceso tradicional, puede existir un solo hilo de control y un solo contador de programa en cada proceso. Sin embargo, algunos sistemas operativos modernos manejan múltiples hilos de control dentro de un proceso.





La siguiente figura presenta un modelo de solicitud, donde la existencia de un proceso puede utilizar hilos simples y múltiples hilos, que reciben solicitudes para procesar el archivo y la devolución de datos o actualización.



(a) Estructura de tres procesos con un hilo simple y espacio de direcciones distinto. (b) Estructura de un proceso con tres hilos que comparte el espacio de direcciones Tomada de Tanenbaum (2003, p. 54).

Estos hilos de control normalmente se llaman sólo hilos, en la siguiente figura, se podrá observar tres procesos tradicionales, cada proceso tiene su propio espacio de direcciones y un solo hilo de control. A diferencia, en el segundo esquema se observa un solo proceso con tres hilos de control. Aunque en ambos casos se tienen tres hilos, cada uno de ellos opera en un espacio de direcciones distinto, mientras que en otro caso los tres hilos comparten el mismo espacio de direcciones.

Una determinante de rendimientos es vista en la figura anterior (b), cuando una solicitud procesada es entregada a un hilo, si este hilo se bloquea, los demás hilos pueden continuar ejecutándose de manera que se podrán realizar nuevas solicitudes, ya sea inclusive de E/S de disco. A diferencia del modelo (a), que no es muy conveniente, porque no todos los hilos comparten el mismo espacio de direcciones y caché de memoria. Algunos de los sistemas operativos anteriormente no utilizaban estos procesos y los hilos se manejaban de forma simple de pilas. La principal desventaja que se tenía con este método era que si el hilo se bloqueaba el proceso quedaría inconcluso.



Para otros sistemas, el sistema operativo utiliza los hilos múltiples por proceso, esta forma de hilos múltiples tiene la ventaja del anterior que cuando un hilo se bloquea, el sistema operativo selecciona el hilo que se ejecutará en seguida, ya sea para el mismo proceso o un proceso distinto. Para realizar la planificación, el kernel deberá saber la existencia de la tabla de hilos que relacione los hilos del sistema.

La diferencia entre un hilo simple y uno múltiple, consiste en el rendimiento que varía de forma significativa entre uno y otro. La modificación de hilos es más ágil cuando la administración de hilos se realiza en el espacio de usuario que cuando se realiza una llamada al kernel; ya sea que los hilos se puedan administrar por el kernel o que en el espacio de usuario se introduzcan varios problemas que se deberán resolver y modificar considerablemente en el modelo de programación para un mejor rendimiento del sistema operativo.

### 1.1.3. Multiprocesamiento

Un sistema operativo que tiene múltiples CPU puede compartir la carga de procesamiento; sin embargo, el problema de la planificación se hace más difícil. La idea para enfocarse sobre el multiprocesamiento es con base en los sistemas, en lo que los procesadores son idénticos, homogéneos en relación con su funcionalidad, pero existen diferentes limitaciones que afectan a la planificación.

Un método para planificar las CPU para un sistema multiprocesador consiste en que todas las decisiones sobre la planificación, el procesamiento de E/S y varias actividades del sistema sean administrados por un mismo procesador. Este multiprocesamiento asimétrico parece simple, porque sólo existe un procesador que accede a las estructuras de datos del sistema, minimizando la necesidad de compartir datos.

Otro método es el multiprocesamiento simétrico, que tiene como cualidad que cada uno de los procesadores se autoplanifica, y los procesos pueden estar en una cola común de procesos preparados, o bien cada procesador puede ser propietario de su cola de procesos



preparados de forma independiente. En este método se puede establecer que el planificador de cada procesador analice la cola de procesos preparados y seleccione un proceso para ejecutarlo. Se considerará una programación eficiente del sistema a aquella en la que el procesador pueda seleccionar un proceso que esté dentro de una estructura de datos y que dos procesadores no seleccionen el mismo proceso y también que se evite la pérdida de procesos de la cola.

En la actualidad, la mayoría o prácticamente todos los sistemas operativos modernos tienen la facultad de soportar el multiprocesamiento simétrico, por mencionar algunos de ellos: Windows XP y superior, Linux en todas sus distribuciones, Mac OSx, etc.

### 1.1.4. Paralelismo

Para la programación de sistemas operativos, el término paralelismo implica la existencia de varios procesadores en el sistema, la técnica de programación paralela determina el poder dividir la ejecución de un proceso en distintos módulos del sistema.

Mientras la CPU está ejecutando un programa, también puede estar leyendo un disco, esto se realiza en el transcurso de muy poco tiempo, ya que ésta puede trabajar con varios programas simulando que el sistema realiza varios procesos al mismo tiempo. El término pseudoparalelismo, se utiliza para referirse a esta rápida conmutación de la CPU entre programas, para diferenciar del verdadero paralelismo de hardware de los sistemas multiprocesador, en el proceso de evolución de desarrollo de sistemas operativos se ha ido mejorando el modelo de procesos secuenciales que facilita el manejo del paralelismo.

El paralelismo para los programas puede ser explícito, una de sus principales características es que el programador especifica mediante una construcción de concurrencia de exclusión mutua y sincronización, para determinar la prioridad de los procesos. En este método puede ser viable para utilizar los procesadores por separado y ejecutar cada uno los procesos, por su dificultad de implementación se pueden generar varios errores de programación.



### 1.2. Memoria

La memoria en un sistema operativo puede llegar a ser un recurso muy importante que se debe administrarse con cuidado. La memoria principal puede ser considerada como un arreglo lineal de localidades de almacenamiento de un byte de tamaño, cada localidad de almacenamiento tiene asignada una dirección que la identifica.

La memoria principal es el lugar donde la CPU registra las instrucciones que se van a ejecutar, así como algunos datos a emplear, una de las funciones básicas que debe implementar un sistema operativo es la administración de la memoria para tener un control sobre los lugares donde están almacenados los procesos y datos que actualmente se están utilizando.

#### 1.2.1. Administración de la memoria

Se debe tener cuidado en la gestión de la administración de la memoria para un sistema multiproceso, porque se puede determinar el estado del procesador dependiendo de la cantidad de procesos que se estén ejecutando. La memoria principal es la parte donde se ejecutan los programas y procesos, y está definida por el espacio disponible para que se puedan ejecutar los procesos.

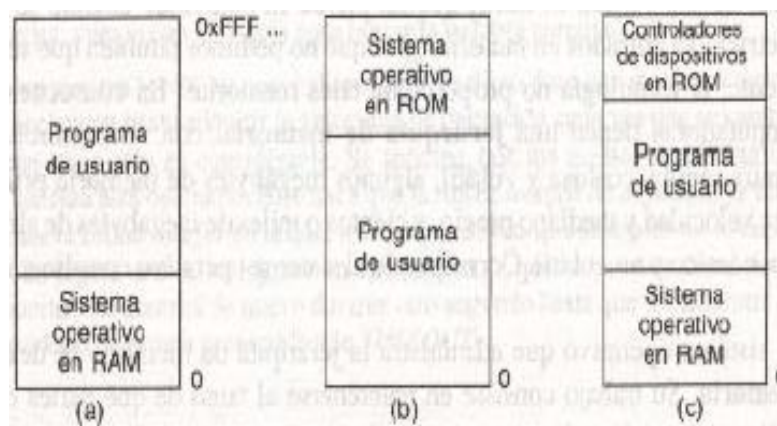
La administración de la memoria es la encargada de gestionar la jerarquía de los procesos que se ejecutan dentro de la misma, su principal función es determinar qué partes de la memoria real están en uso y qué partes están libres, también asigna memoria a los procesos cuando la necesitan recuperar y cuando termina; además controla el intercambio entre la memoria principal y el disco cuando la primera es demasiado pequeña.

Para un sistema operativo la administración básica de la memoria se puede dividir en dos clases, la que intercambian los procesos entre la memoria y algún dispositivo, esto mediante el proceso de ejecución. La otra clase es lo contrario a la anterior, porque aquí están catalogados lo que no realizan el intercambio y paginación.



El intercambio y la paginación son utilizados como una técnica para las situaciones causadas cuando la memoria principal es escasa, para mantener todos los programas al mismo tiempo. Esto reduce el costo de la memoria principal y aumenta el rendimiento del procesador, comúnmente es conveniente el replantear la administración de la memoria a medida que el sistema tenga mayor crecimiento, pues esto implica que la memoria u otro pueden hacerse obsoletos, ya que los sistemas crecen con mayor rapidez que las memorias.

La monoprogramación que no tiene intercambio ni paginación es un esquema más simple y tiene la posibilidad de ejecutar sólo un programa a la vez, compartiendo la memoria entre el programa y el sistema operativo, en la siguiente figura plantea tres variantes sobre el estado de memoria.



Tres variantes de organización de la memoria entre un programa y el sistema operativo  
Tomada de Tanenbaum (2003, p. 310).

Las variantes de la figura anterior muestran (a) que el sistema operativo puede estar alojado dentro de la memoria de acceso aleatorio RAM; la (b) indica que el sistema operativo puede estar alojado en la parte superior de la memoria ROM, memoria de sólo lectura; en la variante (c) los controladores de los dispositivos se encuentran en la parte superior de la memoria ROM seguido de los programas de usuario y al final del esquema se encuentra el sistema operativo por encima de la memoria RAM.

En los esquemas anteriores, se puede ver que se permite ejecutar sólo un proceso a la vez, para que sea copiado a la memoria y se ejecute. Al finalizar el proceso en el sistema





operativo se deberá mostrar un indicativo que muestre que está a la espera de un nuevo comando, así entonces el sistema operativo recibe este nuevo comando para cargar el nuevo programa en memoria sobre escribiendo el anterior. Existen diferentes técnicas para la administración de la memoria, las cuales se clasifican como:

- Multiprogramación con particiones fijas
- Multiprogramación con particiones variables
- Segmentación
- Paginación
- Administración de memoria con mapas de bits
- Administración de memoria con listas enlazadas

Estas técnicas de administración de memoria, por la capacidad de procesamiento obligan muchas veces a modificar la arquitectura del computador para obtener el rendimiento deseado a partir del nuevo hardware.

### 1.2.2. Jerarquía de la memoria

La jerarquía de memoria sirve para organizar de forma incremental de pirámide o niveles. La principal función de este método es conseguir el rendimiento con aumentar la velocidad al costo de una memoria simple.

Los niveles que comúnmente conforman la jerarquía de la memoria son:

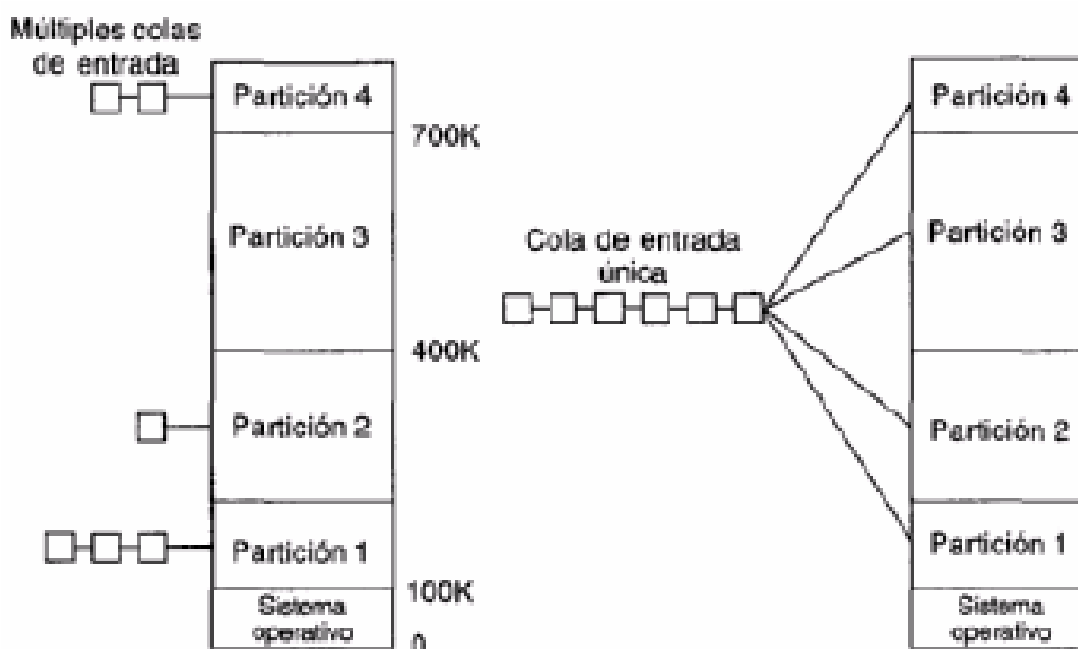
- Nivel 0 de registros
- Nivel 1 de memoria cache
- Nivel 2 memoria principal
- Nivel 3 de disco duro
- Nivel 4 virtual

De esta manera, lo que se busca con la jerarquía de la memoria es establecer una determinante para la administración de la memoria.



### 1.2.3. Multiprogramación con particiones fijas y variables

Existe otro esquema para la administración de la memoria que permite la ejecución de múltiples procesos en un mismo tiempo, este esquema es conocido como multiprogramación con particiones fijas. En los sistemas con tiempo compartido para los procesos resulta que éste si se encuentra bloqueado en espera de que finalice una E/S, otro proceso podrá utilizar la CPU. Así se incrementa el rendimiento tanto de la memoria como de la CPU, una posibilidad de lograr la multiprogramación consiste en dividir la memoria en n particiones, ya sea homogénea o desigual. A la llegada de algún proceso, se podrá colocar en la cola de entrada de la partición pequeña que puede contener, ya que las particiones están fijas, pero si por algún motivo el proceso no utilizó alguna de las particiones ésta será desperdiciada. En la siguiente figura, se ilustra un esquema con diferentes tipos de entradas a particiones fijas.



Esquema de particiones fijas con múltiples colas de entrada (a), particiones fijas con cola única de entrada (b)

Tomada de Tanenbaum (2003, p. 311).

La figura anterior muestra un esquema de dos tipos distintos de asignación de procesos mediante las colas de entrada, se puede observar que al utilizar particiones fijas con colas distintas (a) se ve claro que el rendimiento del sistema baja, ya que al encontrar una

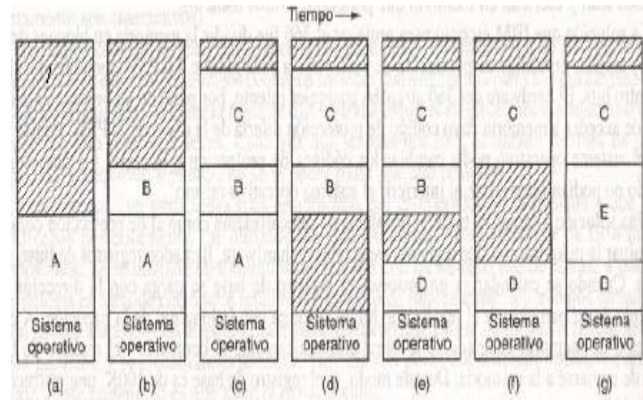


partición demasiado grande y vacía y una partición pequeña y llena esto es una gran desventaja. Una alternativa para una posible solución a esta desventaja puede ser lo que se ilustra en la figura anterior (b), en donde se puede ver que se mantiene una sola cola de asignación para que cada que se libera una partición, se seleccione el proceso ubicado en la partición siguiente inmediata para ser ejecutada.

Dentro del desarrollo para los sistemas de tiempo compartido, la situación se torna de forma distinta para la multiprogramación de particiones variables. La desventaja de este método consiste en que en muchas ocasiones no existirá la cantidad suficiente de memoria principal para almacenar todos los procesos que se encuentren activos. Se podrá utilizar la administración de memoria, pero dependiendo del hardware disponible.

Existen dos estrategias para este método, una se utiliza para buscar y cargar en la memoria cada proceso en su totalidad con la finalidad de que se pueda ejecutar durante el tiempo que sea necesario, para que después se pueda colocar de nueva cuenta sobre el disco, a esta estrategia comúnmente se le conoce como llamadas de intercambio. La segunda estrategia, conocida como memoria virtual, permite ejecutar a los programas a pesar de que estos se encuentren parcialmente en memoria principal.

En el funcionamiento de un sistema con particiones variables los primeros o el primer proceso se cargan en la memoria creando la partición, para dar seguimiento a los demás procesos de acuerdo con la necesidad del proceso y liberando la memoria al término de cada proceso cargado. En la siguiente figura se puede observar un claro ejemplo del uso de particiones variables, donde se carga un solo proceso y crea la partición que se utilizará (a), posteriormente se pueden cargar demás procesos asignando el espacio dentro de la memoria (b), seguido se muestra el acceso de un tercer proceso utilizando el espacio asignado para este (c), Luego sale el proceso A, dejando el espacio libre para dejar el paso al proceso D (d,e), Posteriormente termina B y entra E (f,g).



Asignación de memoria mediante la multiprogramación de particiones variables

Tomada de Tanenbaum (2003, p. 314).

En la asignación de memoria que se muestra en la figura anterior, se puede observar que mientras los procesos entran en memoria y salen el espacio se libera para dejarlo listo para el siguiente proceso que se mande llamar. El tamaño de estas particiones puede variar, tiene la flexibilidad de no estar limitado a la cantidad de particiones fijas establecidas para la asignación de procesos.

### 1.2.4. Memoria real

La memoria real o principal es donde se guardan los datos y se ejecutan los programas, por lo general esta memoria es volátil. La CPU extrae instrucciones de la memoria de acuerdo con el valor del contador de programa, durante el ciclo de ejecución de algún proceso debería extraer la instrucción de la memoria, la cual se decodifica y puede hacer que se procesen una serie de operaciones. El resultado de haber ejecutado esas operaciones se almacena de nuevo en la memoria.

Comúnmente el costo de esta memoria es mayor que el de la memoria secundaria, pero la gran diferencia está en que la memoria real es de rápido acceso a la información contenida; la memoria cache es la única que podría llegar a ser mucho más rápida que la principal pero su costo también es aún mayor que el de las memorias real y secundaria.



La organización y administración de la memoria real de un sistema ha sido y es uno de los factores más importantes en el diseño de los sistemas operativos. Muy a menudo los términos memoria y almacenamiento se consideran equivalentes, los programas y datos deben estar en el almacenamiento principal para poderlos ejecutar y referenciarlos directamente.

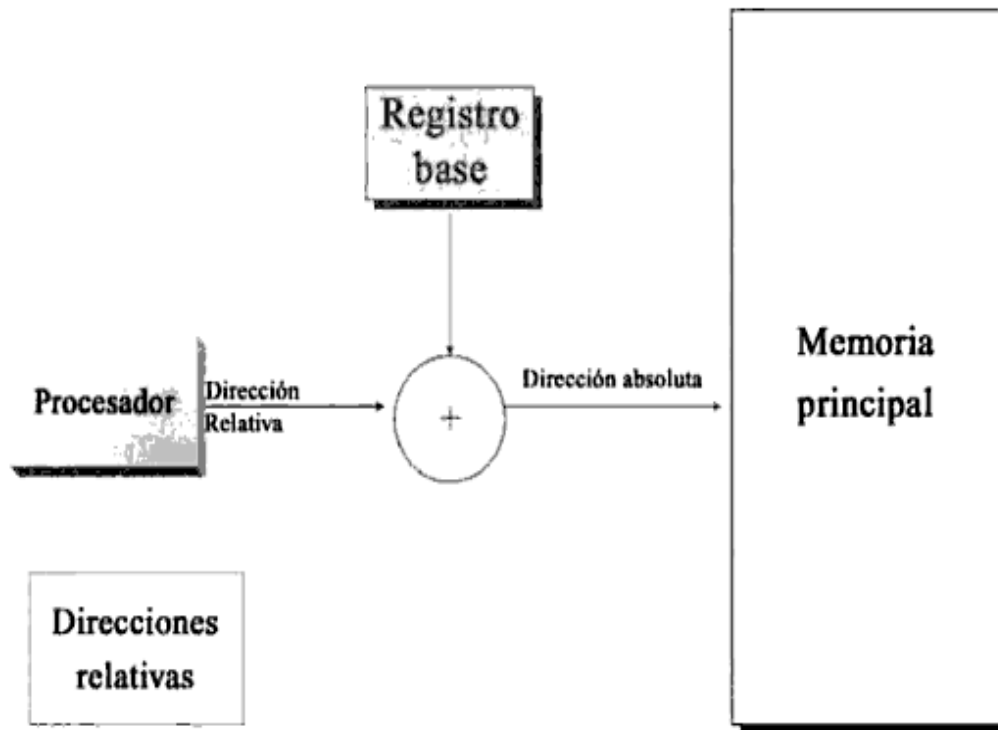
Se considera almacenamiento secundario o auxiliar al que es soportado por lo general en discos, la parte del sistema operativo que administra la memoria se llama administrador de la memoria. Lleva un registro de las partes de memoria que se están utilizando y de aquellas que no, asigna espacio en memoria a los procesos cuando éstos la necesitan y libera espacio de memoria asignada a procesos que han terminado.

En lo que respecta a la reasignación de direcciones, el contenido del registro frontera indicará el punto a partir del cual puede cargarse el programa del usuario. Para ello será necesario reasignar las direcciones del programa en función de la frontera, existen dos formas de hacerlo, una estática y otra dinámica.

La asignación estática se realiza durante la compilación o bien durante la carga del programa en memoria. Esto implica que cualquier variación del tamaño del sistema operativo exigirá una nueva compilación o carga del programa. Es una técnica fácil de llevar a cabo pero demasiado rígida. La alternativa es asignar las direcciones reales dinámicamente durante la ejecución. Un dispositivo hardware especial interceptará cada dirección lógica generada por el programa y le sumará el contenido del registro frontera.

El resultado será la dirección real correspondiente, como se puede observar en la siguiente figura. Con esta técnica el usuario no podrá manejar direcciones reales. Su programa utilizará direcciones relativas que podrán variar de 0 al máximo permitido por el sistema operativo. Éste será su espacio lógico de direcciones.





Reasignación dinámica Tomada de Morera y Pérez (2002, p. 262).

Posteriormente, el sistema con la ayuda del hardware establecerá la correspondencia con el espacio físico de direcciones. El cual estará limitado por la dirección frontera y el tamaño de la memoria real. Esta separación entre la visión del usuario (espacio lógico) y la memoria real (espacio físico) permite gestionar a ésta con mayor eficacia.

### 1.2.5. Memoria virtual

La memoria virtual tiende a simplificar y facilitar la tarea de programación, debido a que con esta alternativa de memoria ya no se limita a la cantidad de memoria física disponible. La memoria virtual incluye la separación de memoria lógica y física, esta división permite proporcionar a los programadores una memoria virtual extremadamente grande, cuando sólo se tiene disponible una memoria física de menor tamaño.



La memoria virtual contiene espacios de direcciones virtuales de un proceso que hace referencia a la forma lógica de almacenar un proceso en la memoria, las ventajas de la memoria virtual son importantes para la asignación de procesos.

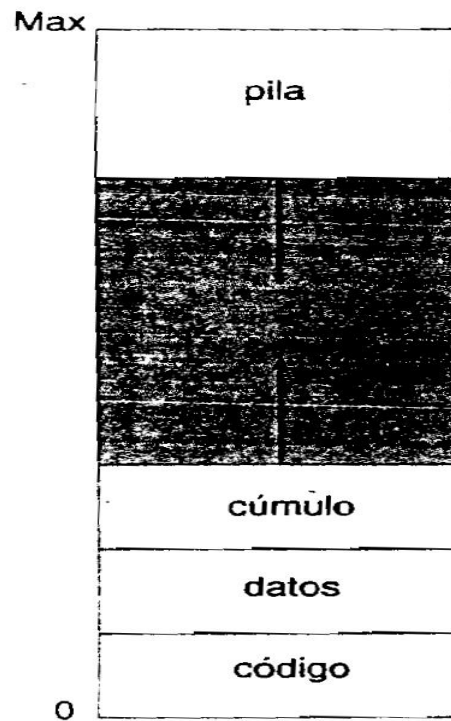
La memoria virtual tiene la posibilidad de poder ser utilizada en un sistema de multiprogramación, manteniendo segmentos de muchos programas dentro. En lo que un programa está esperando que se extraiga de la memoria una de sus partes, también está a la espera de E/S, mientras no puede ejecutarse, por lo que podrá otorgarse la CPU a otro proceso, lo mismo que en cualquier otro sistema de multiprogramación.

Mientras el sistema realiza el trabajo real de intercambiar los recubrimientos, el programador tiene que encargarse de dividir en trozos apropiados el programa. La tarea de dividir programas grandes en pequeños trozos modulares era laboriosa y tediosa.

La idea básica de la memoria virtual es que el tamaño combinado del programa, sus datos y su pila pueden exceder la cantidad de memoria física disponible. El sistema operativo mantiene en la memoria principal aquellas partes del programa que se están usando en cada momento, manteniendo el resto de las partes del programa en el disco. Por ejemplo, un programa de 16 MB puede ejecutarse sobre una máquina de 4MB eligiendo cuidadosamente que 4 MB se tendrán en la memoria en cada instante, e intercambiando partes del programa entre el disco y la memoria, según sea necesario.

La memoria virtual puede funcionar también en un sistema multiprogramado, con diversos fragmentos de muchos programas en memoria a la vez. Mientras un programa espera a que se traiga del disco una parte de sí mismo está esperando por una E/S y no puede ejecutarse, por lo que debe asignarse la CPU a otro proceso de la misma forma que en cualquier otro sistema multiprogramado.

La siguiente figura muestra el cúmulo de memoria, que es utilizada para la asignación dinámica de memoria. Este método permite que la pila crezca hacia abajo en la memoria con las sucesivas llamadas a función.



Espacios de las direcciones en memoria virtual

Tomada de Silberschatz (2006, p. 281).

El espacio vacío entre el cúmulo y la pila forma parte de la memoria virtual, los espacios de estas direcciones incluyen espacios de direcciones dispersos, este tipo de direcciones tienen ventaja, porque se puede utilizar un espacio disperso, ya que estos huecos pueden llenarse a medida que crecen los segmentos de pila. Aparte de que se separan la memoria lógica y física, la memoria virtual también, permite que dos o más procesos compartan los archivos con la memoria, mediante mecanismos de compartición de páginas.

Este tipo de dispersión de espacio tiene una serie de ventajas, entre las cuales se puede clasificar: las bibliotecas del sistema se pueden compartir con otros procesos, la memoria virtual facilita a los procesos compartir la memoria física y permite que un proceso cree una región de memoria que los pueda compartir con otros procesos, además de que se comparten páginas durante la creación de procesos mediante la llamada al sistema incrementando la creación de tareas y ejecución de procesos.



### 1.3. Entrada / Salida

La administración de dispositivos de Entrada /Salida resulta ser un tema muy amplio en el diseño de los sistemas operativos. La entrada / salida (E/S) es una amplia variedad de dispositivos y sus variadas dispositivos y sus diversidad de aplicaciones.

Una de las principales funciones de la administración de dispositivos es el controlar todos los dispositivos de E/S; el sistema operativo debe enviar comandos a los dispositivos, atender las interrupciones y gestionar los errores que se generen.

Los dispositivos externos de E/S que interactúan con las computadoras pueden clasificarse en las siguientes categorías:

- Dispositivos físicos para humanos: este tipo de dispositivos son propiamente para establecer la comunicación computadora-usuario. Unos de los principales ejemplos se tiene son: teclado, mouse, pantalla, terminal de lector laser, etc.
- Dispositivos para máquina: los dispositivos para máquina son los que se utilizan para comunicar las partes electrónicas entre computadora-sistema, como discos, tarjetas de video, controladores.
- Dispositivos de caracteres: estos dispositivos proporcionan o acepta un flujo de caracteres, sin tener en cuenta ninguna estructura de bloque. Comúnmente este tipo de dispositivos son los que se conocen como manejadores de los dispositivos como son los driver o controladores.

Cada dispositivo tiene sus propias características de acuerdo con su clasificación de pertenencia, cada una de estas diferencias básicamente se enfoca en la velocidad de los datos, por lo que es muy posible una gran diferencia de varios datos de transmisión. Las aplicaciones son también una de las diferencias de los dispositivos, porque por lo general cada tipo de dispositivo necesita su propio software que será de utilidad para que el sistema operativo gestione el dispositivo; la complejidad del control es necesario poder contar con la interfaz de control al dispositivo.



### 1.3.1. Principios de hardware y software de E/S

Es muy común tener diferentes perspectivas respecto al hardware y software de E/S, dependiendo del tipo de manejo que se realice de forma directa con los dispositivos será el punto de vista que se tenga referente hacia este término. Un programador no tendrá el mismo punto de vista que un ingeniero en electrónica de dispositivos de E/S, que como la parte física tangible que puede interactuar con la máquina, y un programador es quien diseña esa parte intangible, la relación tarjeta-máquina.

Los dispositivos de E/S se pueden dividir en dos grandes grupos:

- Dispositivos por bloques: se encarga de almacenar la información en bloques de tamaño fijo, cada uno de ellos deberá contar con su propia dirección.
- Dispositivos por caracteres: este tipo de dispositivo es el que se encarga de proveer y aceptar una serie de caracteres que son convertidos en corriente eléctrica, este tipo de dispositivos no considera la estructura de bloque que contempla el anterior tipo de dispositivos, tampoco tiene la posibilidad de ser direccionable ni puede realizar la búsqueda.

Existen varios dispositivos que no se contemplan este tipo de clasificación, porque por lo general no se pueden ajustar a esta clasificación. Un ejemplo claro de ello son los relojes, este tipo de dispositivos no están considerados como dispositivos por bloques ni tampoco como dispositivos por caracteres, su función principal de los relojes consiste en suministrar interrupciones a intervalos que ya estén definidos.

En lo que respecta al software, el principal objetivo de este dispositivo de E/S, es prácticamente sencillo de plantear, debido a que se debe organizar el software mediante capas que puedan ser utilizadas para establecer la prioridad del hardware, para que las capas superiores sean las encargadas de establecer la interfaz con el sistema lo cual ayuda a comunicar de forma más ágil el dispositivo-sistema.

En el ámbito del software, existe un concepto clave para poder diseñar el software de E/S, llamado independencia del dispositivo. El cual tiene la utilidad de que los programas puedan





interactuar entre dispositivo hardware y dispositivo software, para que esto pueda funcionar el sistema operativo deberá tener la facilidad de resolver los problemas causados por las diferencias de dispositivos.

Los errores que se manejan dentro del software de E/S, se deben manejar lo más cercano posible del hardware. Cuando el controlador descubre un error de lectura deberá tratar de corregirlo, de lo contrario el sistema operativo es quien podría tratar de corregirlo. Muchas ocasiones los errores son transitorios al momento de la lectura de E/S, y estos por lo general se solucionan reiniciando el proceso de lectura de cada dispositivo.

Uno de los aspectos claves es cuando ocurre la transferencia, ya puede ser por bloqueo o controladas por interrupciones. El dispositivo de E/S físico es por lo general asíncrona, tal como la CPU que inicia la transferencia y se libera mientras está a la espera de una nueva interrupción. Los programas por lo general resultan ser síncronas, después del comando de lectura, el programa se detiene hasta la disponibilidad de datos.

Los dispositivos de uso exclusivo y no exclusivo, son utilizados por lo general al mismo tiempo, ya que no existen conflictos de mantener abiertos varios archivos en el disco al mismo tiempo. Las unidades de cinta son un ejemplo de dispositivos exclusivo, porque sólo un proceso de lectura puede estar activo a la vez.

### 1.3.2. Dispositivos y manejadores de dispositivos

Los dispositivos juegan un papel muy importante en el desarrollo de sistemas operativos pues sirven para introducir datos, estos datos se leerán por los dispositivos de entrada y se almacenan en la memoria principal. Los dispositivos de E/S se pueden agrupar en tres categorías:

- **Dispositivos legibles para los usuarios:** son aquellos dispositivos que permiten la comunicación entre los usuarios y la computadora. Dentro de este grupo se incluyen todos los dispositivos que sirven para proporcionar interfaz con el usuario,



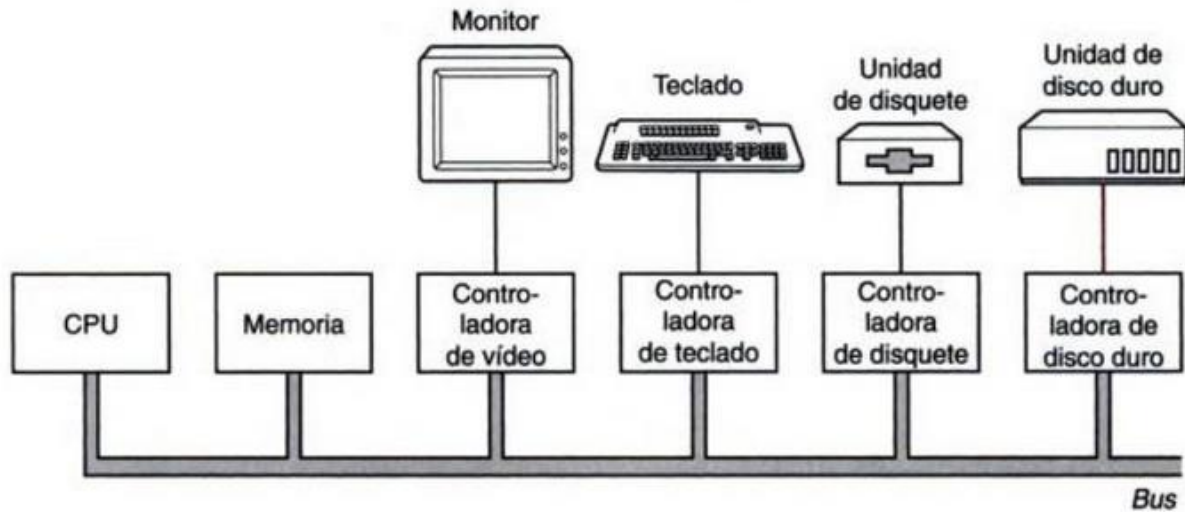
tanto para entrada (ratón, teclado, monitor TouchScreen, scanner, lectores de huella, etc.) como para salida (impresoras, pantalla, etc.).

- **Dispositivos de almacenamiento:** son utilizados para abastecer el almacenamiento no volátil de datos y memoria. Su principal función consiste en proveer los datos y almacenar en los programas que se ejecutan en la CPU.
- **Dispositivos de comunicaciones:** permiten conectar a la computadora con otras computadoras a través de una red. Los dos tipos de dispositivos más importantes de esta clase son los módem, para comunicación vía red telefónica, y las tarjetas de interfaz a la red, para conectar la computadora a una red de área local.

Los dispositivos de acuerdo con su función se pueden clasificar de la siguiente manera:

- **Dispositivos de E/S programada:** mientras el procesador se encuentra en ejecución de E/S, emite la señal para que el proceso esté a la espera del término de la operación para poder continuar con su ejecución.
- **Dispositivos de E/S por interrupciones:** el procesador es el encargado de emitir la orden de E/S para los procesos, estos ejecutan las instrucciones y son interrumpidos al finalizar su trabajo.
- **Acceso directo a memoria (DMA).** Este módulo puede controlar el intercambio de datos entre la memoria principal y un módulo de E/S.

La mayoría de las unidades de E/S constan normalmente de un componente mecánico y un componente electrónico. En la mayoría de los casos es posible separar las dos partes y tener un diseño más modular y general. El componente electrónico se denomina controlador del dispositivo o manejador. En una computadora el componente electrónico llega a ser una tarjeta de circuito impreso el cual puede ajustarse en alguna ranura de expansión, donde el componente mecánico es el dispositivo en sí, la organización de los dispositivos se puede ejemplificar en la figura siguiente:



Algunos de los componentes de un ordenador personal sencillo.

Tomada de Tanenbaum (2003, p. 272).

El manejador del dispositivo se provee usualmente de un conector, en el cual puede conectarse un cable que va al dispositivo. Muchas controladoras pueden manejar dos, cuatro o incluso ocho dispositivos idénticos. Si la interfaz entre la controladora y el dispositivo es un interfaz estándar, ya sea un estándar ANSI, IEEE o ISO oficial, o un estándar de facto, eso permite que cualquier fabricante de hardware pueda manufacturar controladores o dispositivos que se ajusten a esa interfaz. Por ejemplo, muchas compañías de hardware fabrican unidades de disco compatibles con la interfaz IDE o SCSI.

La interfaz entre el controlador y el dispositivo es a menudo una de muy bajo nivel. Por ejemplo, un disco puede formatearse con 256 sectores de 512 bytes por pista. Sin embargo, lo que en realidad sale de la unidad es un flujo de bits en serie que comienza por un preámbulo, seguido de los 4096 bits de un sector y terminando con una suma de verificación (checksum), también llamada un código de corrección de errores (ECC, Error Correcting Code). El preámbulo se escribe cuando se formatea el disco, y contiene el número de cilindro y de sector, el tamaño del sector y datos similares, así como información de sincronización. La tarea del controlador consiste en convertir ese flujo de bits en serie en un bloque de bytes y realizar cualquier corrección de errores que sea necesaria. Normalmente primero se ensambla el bloque de bytes, bit a bit, en un búfer que está dentro



del controlador. Una vez comprobado su checksum y declarado el bloque libre de errores, puede procederse a copiarlo en la memoria principal.

A un nivel igual de bajo, el controlador de un monitor (CRT, tubo de rayos catódicos), también opera como un dispositivo de bits en serie: lee de la memoria bytes que contienen los caracteres a visualizar y genera las señales que sirven para modular el haz de electrones del CRT para producir la escritura en la pantalla. El controlador genera también las señales que hacen que el haz del CRT efectúe un retrasado horizontal al terminar cada barrido de una línea, así como las señales que realizan el retrasado vertical una vez que se ha barrido toda la pantalla. Si no fuera por el controlador del CRT, el programador del sistema operativo tendría que programar de forma explícita el barrido analógico del tubo de figura del monitor. Con el controlador, el sistema operativo inicializa el controlador con unos pocos parámetros, tales como el número de caracteres o píxeles por línea y el número de líneas de la pantalla, y deja que el controlador sea realmente quien se encargue de dirigir el haz del CRT.

Existe una gran variedad de controladores y tipos de periféricos, por lo que es frecuente encontrar más de un controlador posible para el mismo dispositivo, cada uno ofreciendo un nivel distinto de funcionalidades.

Los manejadores de dispositivos son muy variados, casi tanto como los dispositivos de E/S. Muchos de ellos, como los de disco, pueden controlar múltiples dispositivos

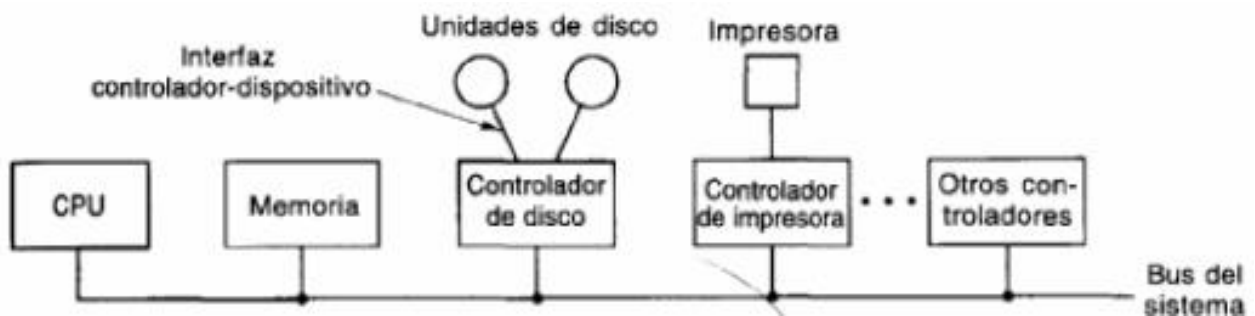
### 1.3.3. Mecanismos y funciones de manejadores de dispositivos

Las funciones de manejadores de los dispositivos de entrada y salida consisten en un componente mecánico y otro eléctrico, en la mayoría de las ocasiones las dos partes se pueden separar con el objetivo de contar con el diseño modular y general. La parte eléctrica se nombra adaptador de dispositivo, por lo general al hablar de la parte eléctrica se puede apreciar una tarjeta de circuitos impresos que se puede insertar en una ranura de la tarjeta



de la computadora, este componente también se puede considerar como componente mecánico.

La mayoría de los controladores pueden manejar uno o varios dispositivos idénticos, Si la interfaz es de mismo tipo estándar con la controladora se podrán ajustar para su funcionamiento. Muchas de las ocasiones en los procesos de E/S de dispositivos el sistema operativo siempre interactúa con el controlador y no con el dispositivo. La siguiente figura muestra la forma de cómo se interconectan los dispositivos.



Modelo básico, para la implementación de dispositivos, entre la CPU y los controladores de dispositivos de E/S. Tomada de Tanenbaum (2003, p. 155).

La comunicación entre la CPU y los controladores utiliza un bus de comunicación simple, pero en equipos de macro computadoras por lo general utilizan un modelo muy distinto como posibles bus múltiples de datos. La actividad del controlador es convertir un flujo de bits a un bloque de bytes y realiza las acciones de corrección de errores necesarias, por lo general, se va armando por bloques de bytes en un buffer dentro del controlador.

El controlador tiene algunos registros que se pueden utilizar para comunicar con la CPU, en algunas computadoras los registros forman parte del espacio de direcciones de la memoria básica, lo que se conoce como mapeo de memoria de E/S.

El código dependiente se aloja dentro de los controladores de dispositivos. Cada controlador administra un tipo de dispositivo o clase de dispositivos similares. En términos generales, el controlador de dispositivos de software acepta peticiones del software independiente al dispositivo para que puedan ser atendidas, las peticiones más comunes es el leer un bloque n. Si algún controlador está disponible o libre al llegar una petición,



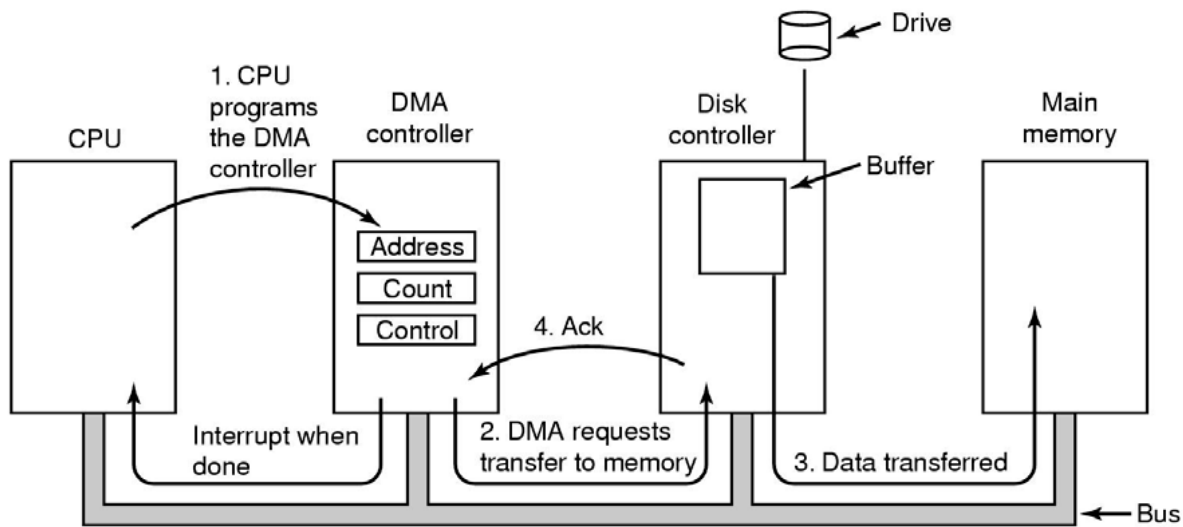


inicia atendiéndola, pero si se encontrara ocupado en otra petición a la llegada de una nueva, éste la coloca en la cola de peticiones y la deja como pendiente hasta que sea atendida.

Lo primero que se realiza para atender la petición de E/S es que se debe traducir de términos abstractos a concretos. Esto deberá calcular en que parte del módulo el proceso se encuentra ubicado para decidir la prioridad de operaciones del hardware. Al decidir la prioridad se comienza a generar la escritura en los registros del dispositivo de éste.

La CPU necesitará direccionar los controladores de dispositivos para intercambiar datos con ellos, puede solicitar datos del controlador de E/S byte a byte, pero haciéndolo así estaría desperdiciándose mucho tiempo de CPU. Por ese motivo normalmente se utiliza un esquema diferente, denominado acceso directo a memoria (DMA; Direct Memory Access). El sistema operativo sólo puede utilizar DMA si el hardware dispone de un controlador de DMA, por lo que la mayoría de los sistemas cuentan con él. A veces ese controlador está integrado en los controladores de disco o en otros controladores, pero tal diseño significa tener un controlador de DMA por cada dispositivo. Comúnmente, se tiene un único controlador de DMA (por ejemplo en la placa madre) para regular las transferencias con múltiples dispositivos, a menudo de forma concurrente.

“En cualquier ubicación física el controlador de DMA tiene acceso al bus del sistema independientemente de la CPU, como se muestra en la siguiente figura. El controlador contiene varios registros en los que la CPU puede leer y escribir. Éstos incluyen un registro de dirección de memoria, un registro contador de bytes y uno o más registros de control. Los registros de control especifican el puerto de E/S que se utilizará, la dirección de la transferencia (leyendo del dispositivo de E/S o escribiendo en el dispositivo de E/S), la unidad de transferencia (un byte a la vez o una palabra a la vez) y el número de bytes que se transferirán en cada ráfaga.” (Tanenbaum, 2003, p. 276).



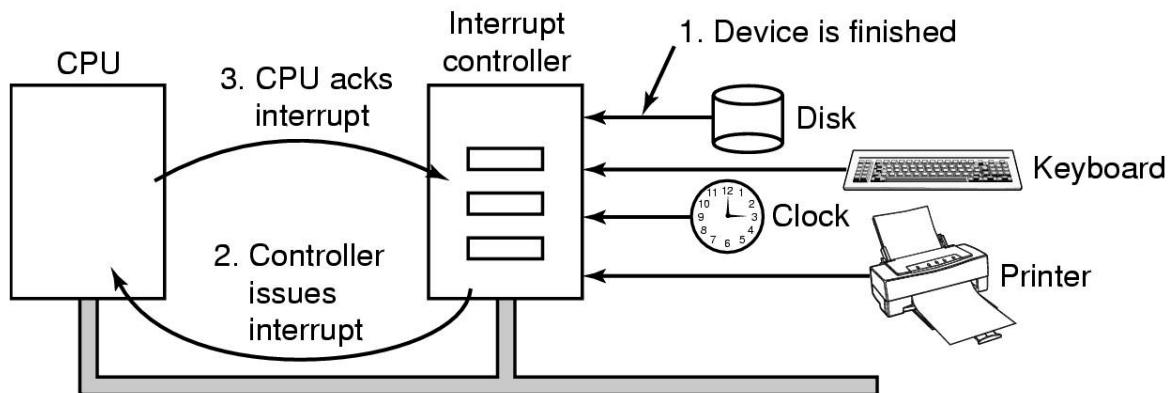
Funcionamiento de una transferencia con DMA

Tomada de Tanenbaum (2003, p. 276).

El controlador es el encargado de leer el bloque de cada sector de la unidad, hasta que esté todo el bloque en el búfer interno del controlador. Posteriormente se calcula el checksum para validar que no se producirán errores de lectura que pudieran provocar interrupciones. Al ejecutar el sistema operativo se podrá leer el bloque de disco del búfer de la controladora.

### 1.3.4. Estructura de datos en el manejo de dispositivos

Dentro de una computadora típica, la estructura de las interrupciones es como se muestra en la figura siguiente, a nivel de hardware, las interrupciones funcionan como se describe a continuación: una vez que el dispositivo termina el proceso de información que se le ordenó, provoca una interrupción (considerando que el sistema operativo es quien habilita las interrupciones). Esto lo hace aplicando una señal a una línea del bus que se le asignó. El chip controlador de interrupciones situado en la placa madre detecta esa señal y decide lo que se va hacer a continuación.



Estructura básica para la representación de la estructura de manejo de dispositivos

Tomado de Tanenbaum (2003, p. 286).

La figura anterior muestra la forma en la que se produce una interrupción. En la realidad las conexiones entre los dispositivos y el controlador de interrupciones utilizan líneas del bus en vez de cables dedicados.

Si no hay otras interrupciones pendientes, el controlador de interrupciones procesa la interrupción inmediatamente. Si está atendiéndose alguna otra interrupción en ese momento, o si otro dispositivo ha realizado una petición simultánea sobre una línea de petición de interrupción de mayor prioridad, el primer dispositivo será ignorado momentáneamente. En este caso, el dispositivo seguirá aplicando la señal de interrupción al bus hasta que reciba de la CPU el servicio deseado. Para gestionar la interrupción, el controlador vuelca un número en las líneas de dirección del bus especificando qué dispositivo requiere atención y aplica una señal que interrumpe a la CPU.

La estructura de datos, en el manejo de dispositivos, tiene la siguiente clasificación:

- **Estructura simple:** es la que se basa en el acceso a operaciones básicas de entrada/salida para escribir directamente en pantalla o disco, lo que se le nombra como libre acceso, otra estructura simple es la separación del kernel y los programas de sistemas.
- **Estructura por capas:** esta estructura se basa en nuevas versiones tales como unix donde se diseñaron para hardware más avanzado, para dar mayor soporte al mismo hardware. La ventaja de la estructura por capas es que cada capa cumple



con una serie de funciones y servicios que brinda a las otras capas, esto permite una mejor organización del sistema operativo y una depuración más fácil de éste.

Existen estructuras que se generan en memoria durante la ejecución de los procesos relacionadas con el uso que estos hacen de los archivos. El sistema posee una tabla general de archivos abiertos, en donde se procesa la información sobre los archivos activos e índices de referencia al archivo. La llamada al sistema toma el nombre del archivo busca en el directorio, copia la entrada del directorio en la tabla de archivos abiertos. Todo esto si estas acciones son permitidas de acuerdo con los modos de protección del archivo.

En los sistemas multiusuario, un archivo puede ser abierto por más de dos usuarios simultáneamente, cada uno con su puntero. Existen dos tipos de tablas, la de proceso, que contiene información sobre los archivos abiertos por el proceso y contiene un apuntador a la segunda tabla, la de los archivos abiertos en todo el sistema, esta segunda tabla tiene información independiente del proceso que lo ejecute como ubicación del archivo en el disco.

El sistema operativo partiendo de su estructura por capas, está organizado en cuatro capas, las cuales tienen su función propiamente establecida y una interfaz con la capa adyacente.

La organización se estructura en las siguientes capas:

- Manejador de interrupción: activa al manejador al terminar la E/S.
- Manejador de dispositivos o drivers: inicia los registros del dispositivo, comprobando el estado.
- Software de E/S independiente de los dispositivos: este software está formado por la parte de alto nivel de los manejadores, el gestor de cache, el gestor de bloques y el servidor de archivos.
- Interfaz del sistema operativo: llamadas al sistema que usan las aplicaciones de usuario.

Los manejadores de interrupción, administran las interrupciones que generan los controladores de dispositivos una vez que éstos están listos para la transferencia de datos, o bien han leído o escrito los datos de memoria principal en caso de acceso directo a



memoria. Para administrar la interrupción se ejecuta el manejador de interrupción que tendrá como efecto es el de salvar los registros, comunicar el evento al manejador del dispositivo y restaurar la ejecución de un proceso.

Cuando una interrupción ocurre muy frecuentemente, caso del reloj, o cuando la cantidad de información a transferir es muy pequeña, caso del teclado, sería muy costoso comunicar siempre el evento al manejador de dispositivo asociado. En estos casos, el propio manejador de interrupción registra la ocurrencia del evento, mediante el incremento de una variable global para el reloj o mediante la acumulación de caracteres en un buffer del teclado.

Los manejadores de dispositivos, tienen un manejador asociado en el sistema operativo, dicho manejador incluye un código independiente del dispositivo que otorga al nivel superior del sistema operativo una interfaz de alto nivel y el código dependiente del dispositivo necesario para programar el controlador del dispositivo a través de sus registros y datos. La principal actividad de un manejador de dispositivo es aceptar las peticiones en formato abstracto de la parte del código de E/S independiente del dispositivo, traducir dichas peticiones a términos que entienda el controlador, enviar al mismo las órdenes adecuadas en la secuencia correcta y esperar a que se cumplan.

### 1.3.5. Operaciones de E/S

En la mayoría de las aplicaciones, el archivo es el elemento central. Cualquiera que sea la finalidad de la aplicación, implicará la generación y uso de información.

Normalmente cada sistema dispone de programas de utilidad que se ejecutan como aplicaciones privilegiadas. Sin embargo, un sistema de gestión de archivos necesita como mínimo algunos servicios especiales del sistema operativo.

Para la E/S programada como la basada en interrupciones, la CPU debe encargarse de realizar las operaciones de lectura y escritura sobre ficheros.





- La lectura o bien entrada de datos, permite recibir valores desde los dispositivos o archivos.
- La escritura, realiza la operación de escritura de resultados en dispositivos de salida o archivos.

En un proceso de usuario que desea leer bloques de datos de una cinta, uno cada vez, siendo cada bloque de 100 bytes, los datos van a ser leídos en una zona de datos del proceso de usuario situada en las direcciones virtuales 1000 a 1009. La forma simple de realizar sería emitir una orden de E/S a la unidad de cinta y esperar a que los datos estén disponibles. La espera podría ser activa o de manera más práctica, suspender al proceso en espera de una interrupción.

Existen dos problemas con este enfoque:

- El programa se queda colgado esperando a que la operación de E/S termine.
- Dificulta las decisiones de intercambio del sistema operativo.

Las ubicaciones virtuales 1000 a 1009 deben permanecer en memoria principal durante el curso de la transferencia del bloque, de lo contrario parte de los datos se perderán.

Si un proceso emite una orden de E/S queda suspendido a la espera del resultado, se le expulsa antes de comenzar la operación y se bloquea esperando a que la operación termine. Mientras tanto, la operación de E/S queda bloqueada esperando a que el proceso vuelva a memoria, para evitar este interbloqueo, la memoria de usuario implica en la operación de E/S debe quedar fija en la memoria principal, inmediatamente después de emitir la petición de E/S, incluso aunque la operación de E/S se encole y pueda no ejecutarse por algún tiempo.

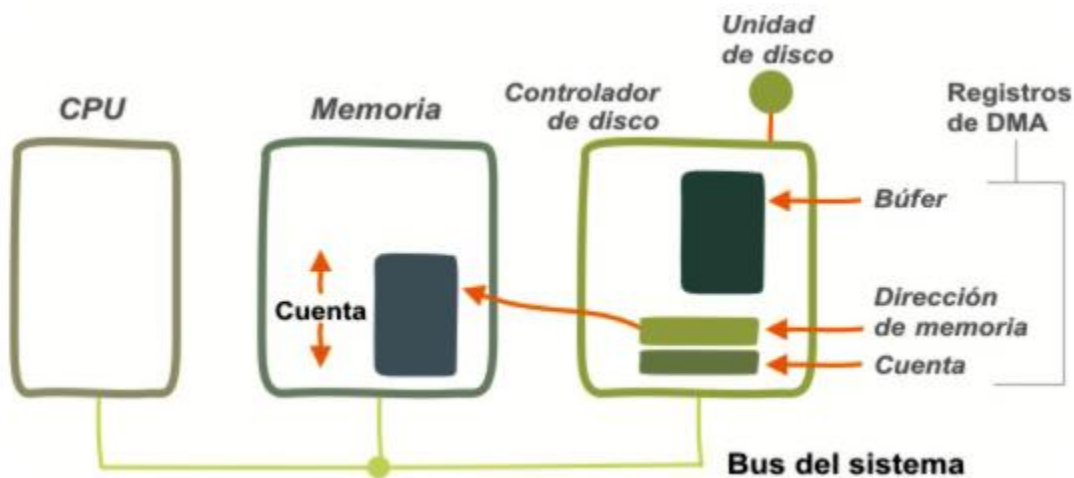
Acceso directo a memoria (DMA). Varios controladores.

Los dispositivos por bloques manejan el acceso directo a memoria o DMS. Primero el controlador lee el bloque de la unidad en serie, bit por bit, hasta que todo el bloque está en el buffer interno del controlador. A continuación, el controlador calcula la suma de verificación para comprobar que no ocurrieron errores de lectura y luego causa una interrupción. Cuando el sistema operativo comienza a ejecutarse, puede leer el bloque del



disco del buffer del controlador byte por byte o palabra por palabra, ejecutando un ciclo, leyéndose en cada iteración un byte o una palabra de un registro del controlador y almacenándose en la memoria.

La finalidad del DMA es liberar a la CPU de un ciclo programado donde se leen los bytes del controlador uno por uno, porque eso tiene pérdida de tiempo en la CPU, cuando se utiliza DMA la CPU proporciona al controlador dos elementos de información además de la dirección en disco del bloque: la dirección de memoria donde debe colocarse el bloque y el número de bytes que deben transferirse, como se muestra en la figura siguiente.



Una transferencia DMA es realizada totalmente por el controlador

Tomada de Tanenbaum (2003, p. 158).

El procedimiento a seguir en una operación de E/S con DMA es:

- Programación de la operación de E/S. Se indica al controlador la operación, los datos a transferir y la dirección de memoria sobre la que se efectuará la operación.
- El controlador contesta aceptando la petición de E/S.
- El controlador le ordena al dispositivo que lea (para operación de lectura) una cierta cantidad de datos desde una posición determinada del dispositivo a su memoria interna.
- Cuando los datos están listos, el controlador los copia a la posición de memoria que tiene en sus registros, incrementa dicha posición de memoria y disminuye el contador de datos pendientes de transferir.



- Los pasos 3 y 4 se repiten hasta que no quedan más datos por leer.
- Cuando el registro de contador está a cero, el controlador interrumpe a la CPU para indicar que la operación de DMA ha terminado.

### **Inicio y control de los programas de canal.**

Estos programas residen en la memoria principal de la CPU y se ejecutan en el canal. La CPU puede utilizar para estos fines, son las siguientes:

- START I/O: inicia una operación de E/S. El campo de dirección de la instrucción se emplea para especificar el canal y el dispositivo de E/S que participa en la operación.
- HALT I/O: finaliza la operación del canal.
- TEST CHANNEL: prueba el estado del canal.
- TEST I/O: Prueba el estado del canal, el subcanal y el dispositivo de E/S.

Una operación de E/S se inicia con la instrucción START I/O. La ubicación del programa de canal en la memoria principal viene definida en la palabra de dirección de canal (CAW: Channel Address Word).

### **Cierre de la unidad**

Has concluido la primera unidad del curso. A lo largo de ésta se recordaron conceptos básicos sobre hilos y que es el multiprocesamiento y el paralelismo, también conociste cómo el sistema operativo maneja la memoria, cuál es su jerarquía, sus particiones así como el manejo que hace el sistema operativo de la memoria real y virtual, identificaste cómo los dispositivos de entrada y salida son manejados por el sistema operativo sus mecanismos y estructura; así como las operaciones que realizan, siendo este el último tema realizado por una investigación de los principales dispositivos de entrada y salida E/S.

Es aconsejable que revises nuevamente la unidad en caso de que los temas que se acaban de mencionar no te sean familiares, o no los recuerdes, de no ser este tu caso, ya estás preparado(a) para seguir con la unidad 2, en donde se abordará el sistema de archivos, sus conceptos básicos y su jerarquía. Así como los tipos de archivos: reales virtuales y



componentes; cómo el sistema operativo funciona física y lógicamente sus operaciones y su implementación. Todo ello con el fin de obtener el prototipo final al terminar la tercera y cuarta unidad del curso de Programación de sistemas operativos.

### Para saber más

Si deseas saber más acerca de cómo el procesador AMD maneja la energía, virtualización, arquitectura, administración y compatibilidad con diferentes sistemas operativos, visita la siguiente dirección electrónica:

- <http://www.amd.com/LA/PRODUCTS/TECHNOLOGIES/Pages/technologies.aspx>

### Fuentes de consulta

#### Bibliografía básica

- Martínez, P., Cabello M. y Díaz, M. (1997). *Sistemas operativos*. Madrid: Díaz de Santos.
- Ortiz, H. (2005). *Sistemas operativos modernos*. Colombia: Universidad de Medellín.
- Santiago, C., García, R., Quezada, A., Santana, F. y Santos, J. (2007). *Fundamentos de sistemas operativos teorías y ejercicios resueltos*. España: Parainfo.
- Silberschatz, A. (2006). *Fundamentos de sistemas operativos*. España: McGraw-Hill.
- Stallings, W. (2007). *Sistemas operativos*. México: Pearson Prentice Hall.
- Tanenbaum, A. (2003). *Sistemas operativos modernos*. México: Pearson Prentice Hall.

#### Bibliografía complementaria

- Morera, J. y Pérez, A. (2002). *Conceptos de sistemas operativos*. Madrid: Comillas.