

## 1. INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

Un Sistema operativo es un conjunto de **programas encargados** de administrar y explotar los recursos de un sistema de computador para ofrecer un conjunto de servicios a los usuarios. Los componentes de un sistema de computador son hardware, software y datos. Es el sistema operativo quien proporciona las facilidades para aprovechar adecuadamente estos recursos.

Podemos ver el sistema operativo como un encargado de asignar los recursos. Dado que puede haber muchas solicitudes de recursos que seguramente no podrán satisfacerse simultáneamente, el sistema operativo debe decidir a cuáles solicitudes se les va asignando recursos para poder operar el sistema de cómputo de una manera eficiente y justa.

El objetivo primordial de un sistema operativo es proporcionar comodidad al usuario. Es mucho más fácil realizar tareas en un centro de cómputo con la ayuda de un sistema operativo que sin él. Un segundo objetivo de un sistema operativo es la eficiencia y crece a medida que nos encontremos con grandes sistemas de cómputo multiusuarios.

### 1.1 ELEMENTOS BÁSICOS DE HARDWARE DE UN COMPUTADOR

Un sistema informático consta de 3 componentes básicos que de alguna manera deben estar interconectados para poder ejecutar los programas.

- **Procesador**: Controla toda la operación del computador y es quien ejecuta las instrucciones de los programas.
- **Memoria principal** (Real): Almacena los datos y programas en ejecución; normalmente es volátil, es decir, una vez termina la ejecución del programa se pierden los datos.

- **Módulos de entrada/salida:** Son los encargados de transportar los datos entre el mundo exterior y el computador, y viceversa.
- **Interconexión de sistemas:** Son los mecanismos que permiten la comunicación entre los 3 componentes anteriores.

Como ya decíamos anteriormente, el procesador es quien lleva el control de todo lo que pase en el sistema. Una de sus principales funciones es el intercambio permanente de datos con la memoria principal

## **1.2 REGISTROS DEL PROCESADOR**

Para el intercambio de datos entre el procesador, la memoria principal y dispositivos, el procesador utiliza dos registros:

- Registro de direcciones de memoria (**MAR**): Contiene la dirección de memoria donde el procesador efectuará la próxima lectura o escritura de datos.
- Registro de buffer de memoria (**MBR**): Contiene los datos que van a ser escritos a memoria o los que fueron leídos de ella.
- Registro de dirección de entrada/salida (**IOAR**): Especifica el dispositivo de entrada i/o salida.
- Registro de buffer de entrada/salida (**IOBR**): Área temporal utilizada para intercambiar los datos entre el procesador y el dispositivo de entrada y salida especificado en el IOAR.

Dentro del procesador existen otros registros que simulan una memoria que es mucho más rápida y pequeña que la memoria principal. Básicamente tienen 2 funciones:

- Registros visibles de usuario: Se utilizan para almacenar algunos datos y minimizar así las referencias a memoria principal. En algunos lenguajes se

permite al programador indicar qué variables se deben almacenar en estos registros.

- Registros de Control y de estado: Son utilizados por el procesador para el control de las operaciones y ejecución de programas. Los más importantes son:

- ✓ MAR, MBR, IOAR, IOBR.
- ✓ Contador de programa (PC): Contiene la dirección de la próxima instrucción a ser ejecutada.
- ✓ Registro de Instrucción (IR): Contiene la última instrucción leída.
- ✓ Palabra de estado del programa (PSW): Contiene códigos de condición. Los indicadores más comunes son:
  - Signo: Contiene bit del signo de la última operación aritmética efectuada
  - Cero: Se activa cuando el resultado de una operación aritmética es cero
  - Acarreo: Se activa cuando en una operación aritmética se produce un acarreo más allá del bit más significativo
  - Igualdad: Se activa si una comparación lógica da como resultado igualdad
  - Desbordamiento: Se activa para indicar un desbordamiento aritmético
  - Habilitar/inhabilitar interrupciones: Empleado para habilitar o deshabilitar las interrupciones
  - Supervisor: Indica si el procesador está ejecutando en modo supervisor o modo usuario.

### 1.3 EJECUCIÓN DE INSTRUCCIONES

La tarea básica de cualquier computador es ejecutar programas; éstos constan de un conjunto de instrucciones almacenadas en la memoria principal. Es el procesador el encargado de ejecutar estas instrucciones y lo realiza en dos pasos:

1. Trae las instrucciones desde memoria (una a la vez).
2. Ejecuta cada instrucción.

La ejecución de un programa consiste en la repetición de estos dos pasos y cada instrucción puede involucrar varias operaciones.

El proceso requerido para ejecutar cada instrucción (traída y ejecución) se llama ciclo de instrucción.

## **1.4 INTERRUPCIONES**

Una vez que un computador empieza a funcionar luego de su arranque (incluye carga del sistema operativo) espera a que ocurra algún suceso. La ocurrencia de un suceso generalmente se indica con una interrupción sea del hardware o del software. El hardware puede generar una interrupción en cualquier momento enviando una señal al procesador a través del bus del sistema. El software puede generar una interrupción ejecutando una operación especial denominada llamada al sistema.

Una interrupción no es más que un mecanismo mediante el cual otros módulos(entrada/salida, memoria) pueden interrumpir la ejecución normal del procesador. Son utilizadas como una herramienta para mejorar la eficiencia del procesamiento. Hay muchos sucesos que pueden generar una interrupción, por ejemplo, la finalización de una operación de entrada/salida, una división por cero, un acceso no válido a la memoria, una solicitud a algún servicio del sistema operativo, el intento de ejecutar una instrucción ilegal de la máquina, el desbordamiento aritmético. Se cuenta con una rutina de servicio para cada una de estas interrupciones, la cual se encarga de atender la interrupción.

Cuando el procesador se interrumpe, suspende lo que está haciendo y de inmediato transfiere la ejecución a la rutina de servicio para esa interrupción. Dicha rutina se ejecuta y, cuando termina, reanuda la operación que fue interrumpida. El acontecimiento de una interrupción involucra la siguiente serie de sucesos:

- El dispositivo emite una señal de interrupción al procesador.
- El procesador termina la ejecución de la instrucción actual.
- El procesador comprueba la interrupción y envía una señal de reconocimiento al dispositivo que la envió.
- Antes de pasar el control a la rutina de servicio de interrupción, el procesador salva el contenido del PSW y del PC en la pila de control.
- El procesador carga en el PC la dirección de la rutina de servicio de interrupción.
- Se salva el contenido de los otros registros del procesador.
- Se ejecuta la rutina de servicio de interrupción.
- Se restaura el contenido de los otros registros del procesador.
- Se restauran de la pila de control los registros PSW y PC.

Con las interrupciones, el procesador se puede dedicar a la ejecución de otras tareas mientras una operación de entrada/salida está en proceso. Para dar cabida a las interrupciones, se añade un ciclo de interrupción al ciclo de instrucción

Ejemplo:

En el siguiente ejercicio, el programa principal hace una serie de llamadas a imprimir, intercaladas con el procesamiento. Los segmentos de código 1, 2 y 3 se refieren a secuencias de instrucciones que no implican I/O. Las llamadas a imprimir son llamadas a un programa I/O e implican:

- Una operación 4, que incluye la copia de los datos hacia un buffer especial.
- La orden concreta de I/O. Como se puede ver en la figura 1.1, mientras que la impresora lleva a cabo su función, el programa debe esperar.
- Una operación 5 que puede incluir la verificación que indique el éxito o fracaso de la operación.

Debido a que generalmente la operación de I/O puede tardar un tiempo relativamente grande, el programa principal se puede quedar colgado esperando a que se complete la operación. Suponiendo que cada operación tarde un segundo, el tiempo de retorno sería 9 segundos, de los cuales, 2 segundos el programa estaría “parado” haciendo nada.

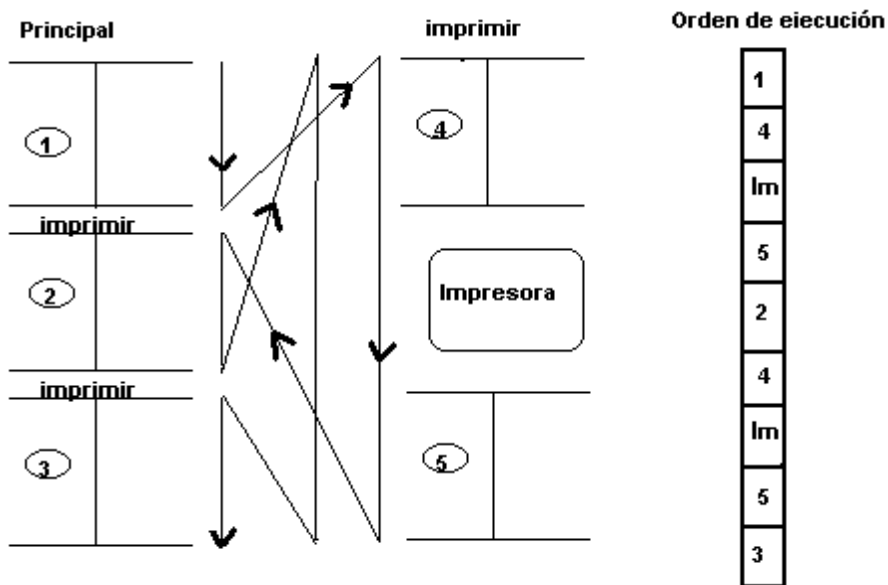


Figura 1.1 (Sin interrupciones)

Si el sistema manejara interrupciones, en el momento en que el dispositivo I/O empezara a trabajar, el programa no tendría que quedarse colgado esperando a que la impresora terminara. Como se puede ver en la figura 1.2, mientras que la impresora hace su trabajo, el procesador va realizando la operación dos; en el momento en que la impresora termina genera una interrupción al procesador para que “acabe” de ejecutar el procedimiento imprimir, más específicamente la operación 5. Cuando se termina la operación dos, se presenta un nuevo llamado a imprimir; cuando empieza nuevamente la impresora, pasa lo mismo y el procesador atiende el programa principal con la operación tres. En este caso, el tiempo de retorno sólo sería de 7 segundos, de los cuales el programa nunca tuvo que esperar.

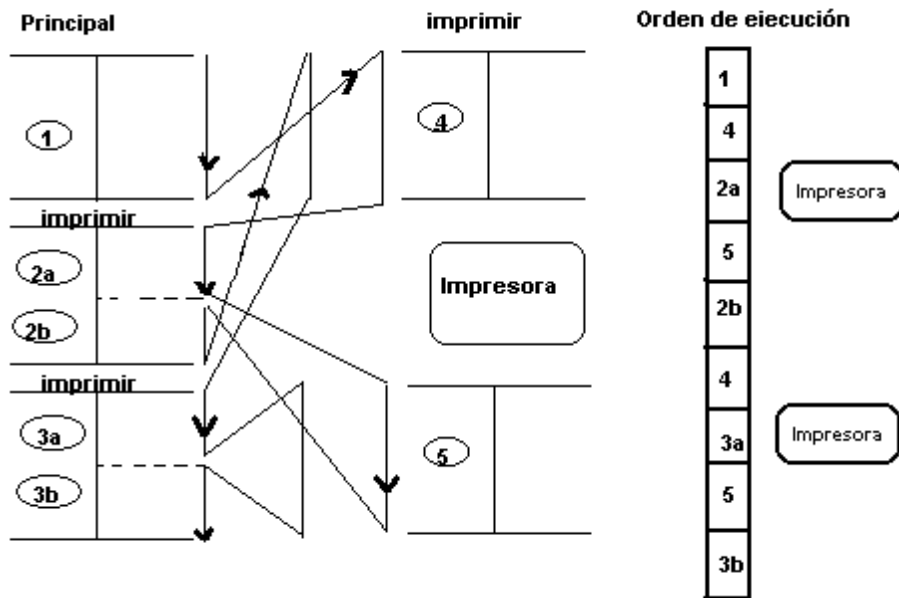


Figura 1.2 (Con interrupciones)

#### 1.4.1 INTERRUPCIONES MÚLTIPLES

Existen dos tratamientos para el caso en el cual se presenten interrupciones múltiples. El primero es inhabilitar las interrupciones, ignora las prioridades, es decir, se hace un tratamiento secuencial de ésta, una a la vez. El segundo caso es definir prioridades para las interrupciones y permitir que una interrupción de prioridad más alta pueda interrumpir a la rutina de servicio de interrupción de prioridad más baja; se hace un tratamiento anidado de interrupciones. Ver figura 1.3

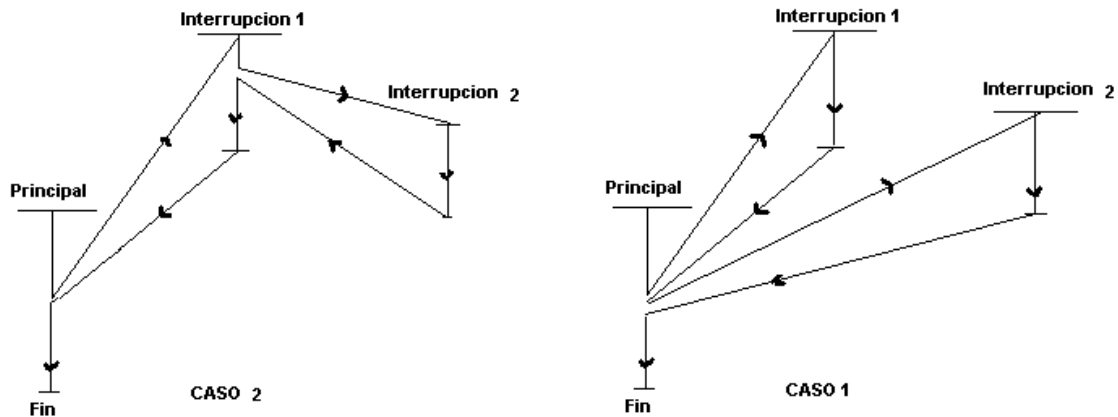


Figura 1.3

## 1.5 JERARQUÍA DE MEMORIA

Las limitaciones de diseño de la memoria de un computador se resumen en tres variables:

- ¿Qué cantidad?
- ¿Qué velocidad?
- ¿Qué costo?

Estas tres características compiten entre sí y generalmente se cumplen las siguientes relaciones:

- A menor tiempo de acceso, mayor costo por bit
- **A mayor capacidad, menor costo por bit**
- A mayor capacidad, mayor tiempo de acceso

A medida que se desciende por la jerarquía (figura 1.4) se tienen las siguientes condiciones:

- Disminución del costo por bit
- Aumento de la capacidad



- Aumento del tiempo de acceso (menos velocidad)
- Disminución de la frecuencia de acceso a la memoria por el procesador

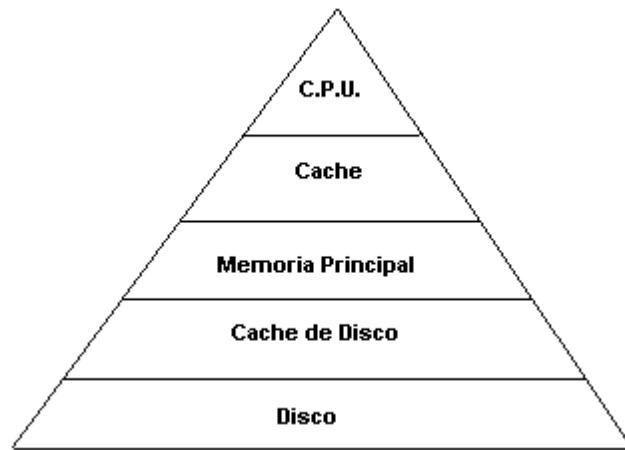


Figura 1.4

Este último punto es la clave del éxito y está permitiendo que memorias caras, rápidas y pequeñas estén siendo reemplazadas por memorias baratas, lentas y grandes. La base del cumplimiento de la última condición es un principio conocido como cercanía de referencias. Durante el curso de ejecución de un programa, las referencias a memoria por parte del procesador, tanto para instrucciones como datos, tienden a estar agrupadas. Durante un largo período, las agrupaciones en uso cambian, pero, en un período corto, el procesador trabaja principalmente con grupos fijos de referencias a memoria.

## 1.6 MEMORIA CACHE

En todos los ciclos de instrucción, el procesador accede a la memoria para leer la instrucción y los operandos y/o almacenar los resultados. La frecuencia con que el procesador puede ejecutar las instrucciones está claramente limitada por el tiempo de ciclo de memoria, que de por sí es mucho más lento que el del procesador.

La memoria caché intenta obtener una velocidad cercana a las memorias más rápidas y, aporta una memoria mucho más grande a unos precios menos costosos. Hay una memoria principal más lenta y relativamente grande, junto a una memoria cache más pequeña y rápida. Cuando el procesador intenta leer una palabra de la memoria, se comprueba si la palabra está en la memoria cache. Si es así, la palabra se envía al procesador. Si no, se rellena la cache con un bloque de memoria principal, formado por un número fijo de palabras y, después la palabra es enviada al procesador. Debido al principio de cercanía de referencias, cuando se carga en la caché un bloque de datos para satisfacer una sola referencia a memoria, es probable que ya se hayan hecho antes otras referencias a palabras del mismo bloque. Ver figura 1.5.

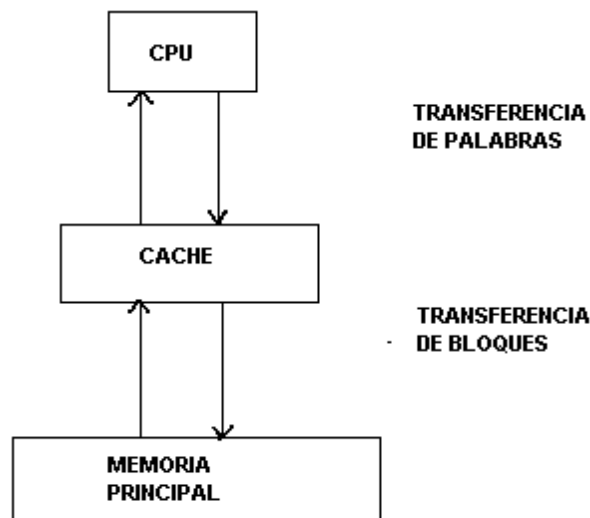


Figura 1.5

## 1.7 TÉCNICAS DE COMUNICACIÓN DE ENTRADA/SALIDA

Para cualquier tipo de operación de entrada/salida, son posibles las tres técnicas siguientes:

- Entrada/salida programada: Cuando el procesador encuentra una operación de entrada/salida, ejecuta dicha instrucción enviando una orden al módulo de

entrada/salida correspondiente, pero es responsabilidad del procesador comprobar periódicamente el estado del módulo hasta saber que se ha completado la operación. El procesador es el único responsable de extraer o poner los datos en memoria.

- Entrada/salida dirigida por interrupciones: El problema de la entrada/salida programada es que el procesador tiene que esperar largos períodos de tiempo mientras que el módulo de entrada/salida esté listo para recibir o transmitir más datos. Una alternativa es que el procesador envíe una orden al módulo y se dedique a realizar alguna otra tarea útil. Cuando el módulo esté listo para intercambiar más datos interrumpirá al procesador para que ejecute la transferencia de datos.
- Acceso directo a memoria (DMA): Cuando se tienen que mover grandes volúmenes de datos, se necesitan técnicas más eficientes. El DMA es una de ellas, es un módulo de Hardware independiente. Cuando el procesador requiere efectuar una operación de entrada/salida emite una orden al DMA, enviándole la siguiente información:
  - Si la operación es de lectura o escritura
  - La dirección del dispositivo de entrada/salida involucrado
  - Dirección inicial de memoria donde se va a leer o escribir
  - Número de palabras a leer o escribir

El procesador continúa con otro trabajo y el módulo DMA es el único responsable de la transferencia de datos. Una vez ésta ha terminado, interrumpe al procesador. El procesador solo interviene al inicio y al final de la transferencia.

## **1.8 CONTROL DE PROCEDIMIENTOS**

Una técnica común para controlar las llamadas a los procedimientos y los retornos es usar una pila. Se necesitan tres direcciones para las operaciones, las cuales se almacenan a menudo en los registros del procesador.

- Un puntero de pila: Contiene la dirección de la cima de la pila. Se incrementa cada que entra un nuevo elemento o se decrementa si se saca uno.
- Base de la pila: Contiene la dirección del fondo de la pila. Puede producir errores de Underflow.
- Límite de la pila: Contiene La dirección del otro extremo de la pila. Puede producir errores de Overflow.

Cuando el procesador ejecuta una llamada, pone la dirección del PC en la pila.  
 Cuando ejecuta un retorno, extrae una dirección de la pila y la carga en el PC.

Ejemplo:

Principal		procedimiento A		procedimiento B	
Dirección	Instrucción	Dirección	Instrucción	Dirección	Instrucción
1000	.....	2000	Call B	3000	.....
1001	Call A	2001	.....	3001	.....
1002	.....	2002	Call B	3002	.....
1003	.....	2003	Return	3003	Return

#### ESTADO DE LA PILA DE CONTROL

.	.	.	.	.	.	.
	1002	1002	1002	1002	1002	
		2001		2003		
a) Contenido de la Pila Inicial	b)Luego de llamar a A	c)Luego de llamar a B	d) Retorno de B	e)Luego de llamar a B	e) Retorno de B	F) retorno de A