SISTEMA OPERATIVO II

SISTEMAS OPERATIVOS MODERNOS

TERCERA EDICIÓN

ANDREW S. TANENBAUM

¿QUÉ ES UN SISTEMA OPERATIVO?

- REVISIÓN DEL HARDWARE DE COMPUTADORA
- LOS TIPOS DE SISTEMAS OPERATIVOS
- CONCEPTOS DE LOS SISTEMAS OPERATIVOS
- ESTRUCTURA DE UN SISTEMA OPERATIVO

PROCESOS E HILOS

COMUNICACIÓN ENTRE PROCESOS

PLANIFICACIÓN

ADMINISTRACION DE LA MEMORIA

MEMORIA VIRTUAL

Una computadora moderna consta de uno o mas

Procesadores.
Memoria principal.
Discos.
Impresoras.
Teclado.
Ratón.
Pantalla.
Monitor.
Interfaces de red.

Y otros dispositivos de entrada/salida. En general es un sistema complejo. Por esa razón para administrar de manera efectiva se debe tener un software que se encargue de administrar todos los recursos antes mencionados, este es el Sistema Operativo.

¿QUÉ ES UN SISTEMA OPERATIVO?

proporcionar a los programadores de aplicaciones un conjunto abstracto de recursos simples, en vez de los complejos conjuntos de hardware; y administrar estos recursos de hardware.

Una de las principales tareas del sistema operativo es ocultar el hardware y presentar a los programas (y a sus programadores) abstracciones gradables, elegantes, simples y consistentes con las que puedan trabajar.

El trabajo del sistema operativo es proporcionar una asignación ordenada y controlada de los procesadores, memorias y dispositivos de E/S, entre los diversos programas que compiten por estos recursos.

Los sistemas operativos modernos permiten la ejecución simultánea de varios programas.

El programa con el que los usuarios generalmente interactúan se denomina

SHELL/CLI - cuando está basado en texto.

GUI - cuando utiliza elementos gráficos o iconos.

En realidad no forma parte del sistema operativo, aunque lo utiliza para llevar a cabo su trabajo.

Por encima del hardware se encuentra el software.

La mayoría de las computadoras tienen dos modos de operación:

Modo Kernel/Supervisor: En este modo, el sistema operativo tiene acceso completo a todo el hardware y puede ejecutar cualquier instrucción que la máquina sea capaz de ejecutar.

Modo Usuario: En el cual sólo un subconjunto de las instrucciones de máquina es permitido.

Las instrucciones que afectan el control de la máquina o que se encargan de la E/S (entrada/salida) están prohibidas para los programas en modo usuario.

El programa de interfaz de usuario, shell o GUI, es el nivel más bajo del software en modo usuario y permite la ejecución de otros programas.

Navegador Web,
Lector de correo electrónico
Reproductor de música.

Estos programas también utilizan en forma intensiva el sistema operativo.

LOS TIPOS DE SISTEMAS OPERATIVOS

Sistemas operativos de mainframe.
Sistemas operativos de servidores.
Sistemas operativos de multiprocesadores.
Sistemas operativos de computadoras personales.
Sistemas operativos de computadoras de bolsillo.
Sistemas operativos integrados.
Sistemas operativos de nodos sensores.
Sistemas operativos en tiempo real.
Sistemas operativos de tarjetas inteligentes.

Los sistemas operativos modernos permiten la ejecución simultánea de varios Programas.

La administración de recursos incluye el **multiplexaje (compartir) de recursos en dos formas distintas:**

- **Tiempo y el Espacio (CPU/RAM).**

Cuando un recurso se multiplexa en el tiempo, los distintos programas o usuarios toman turnos para utilizarlo, uno de ellos obtiene acceso al recurso, después otro, y así en lo sucesivo. El sistema operativo debe estar al tanto de todos los registros. Cuando la CPU se multiplexa en el tiempo, el sistema operativo detiene con frecuencia el programa en ejecución para (re)iniciar otro. Cada vez que detiene un programa en ejecución, el sistema operativo debe guardar todos los registros para poder restaurarlos cuando el programa continúe su ejecución.

El otro tipo de multiplexaje es en el espacio. En vez de que los clientes tomen turnos, cada uno obtiene una parte del recurso.

ADMINISTRACION DE RECURSOS

REVISIÓN DEL HARDWARE DE COMPUTADORA

Un sistema operativo está íntimamente relacionado con el hardware de la computadora sobre la que se ejecuta. Extiende el conjunto de instrucciones de la computadora y administra sus recursos. Para trabajar debe conocer muy bien el hardware.

PROCESADOR

Obtiene las instrucciones de la memoria y las ejecuta. El ciclo básico de toda CPU es obtener la primera instrucción de memoria, decodificarla para determinar su tipo y operandos, ejecutarla y después obtener, decodificar y ejecutar las instrucciones subsiguientes.

Cada CPU tiene un conjunto específico de instrucciones que puede ejecutar.

Muchas CPUs modernas cuentan con medios para ejecutar más de una instrucción al mismo tiempo.

Ejemplo: una CPU podría tener unidades separadas de obtención, decodificación y ejecución, de manera que mientras se encuentra ejecutando la instrucción n, también podría estar decodificando la instrucción n 1 y obteniendo la instrucción n 2. A dicha organización se le conoce como **canalización (pipeline).**

la CPU puede ejecutar cualquier instrucción de su conjunto de instrucciones y utilizar todas las características del hardware. **El sistema operativo opera en modo kernel, lo cual le da acceso al hardware completo.**

Para obtener servicios del sistema operativo, un programa usuario debe lanzar una **llamada al sistema** (*system call*), la cual se atrapa en el kernel e invoca al sistema operativo. La instrucción TRAP cambia del modo usuario al modo kernel e inicia el sistema operativo.

MULTIHILAMIENTO

multihilamiento (*multithreading*) o **hiperhilamiento** (*hyperthreading*) (el nombre que puso Intel al multihilamiento). Para una primera aproximación, lo que hace es permitir que la CPU contenga el estado de dos hilos de ejecución (*threads*) distintos y luego alterne entre uno y otro con una escala de tiempo en nanosegundos (un hilo de ejecución es algo así como un proceso ligero, que a su vez es un programa en ejecución.

• El multihilamiento tiene consecuencias para el sistema operativo, debido a que cada hilo aparece para el sistema operativo como una CPU separada.

• Una CPU con multihilamiento puede cambiar a otro hilo.

• El multihilamiento no ofrece un verdadero paralelismo.

Sólo hay un proceso en ejecución a la vez, pero el tiempo de cambio entre un hilo y otro se reduce al orden de un nanosegundo.

Más allá del multihilamiento, tenemos chips de CPU con dos, cuatro o más procesadores completos, o **núcleos** (*cores*) en su interior.

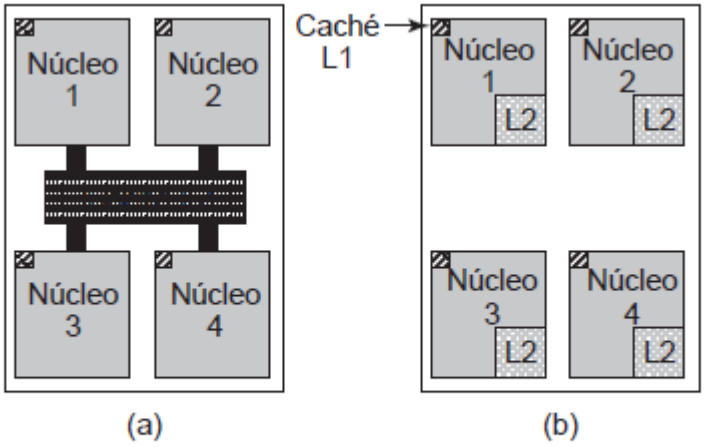


Figura 1-8. (a) Un chip de cuatro núcleos (*quad-core*) con una caché L2 compartida. (b) Un chip de cuatro núcleos con cachés L2 separadas.

La memoria debe ser en extremo rápida (más rápida que la velocidad de ejecución de una instrucción, de manera que la memoria no detenga a la CPU), de gran tamaño y muy económica.

El sistema de memoria está construido como una **jerarquía de capas**.

Las capas superiores tienen mayor velocidad, menor capacidad y mayor costo por bit que las capas inferiores, a menudo por factores de mil millones o más.

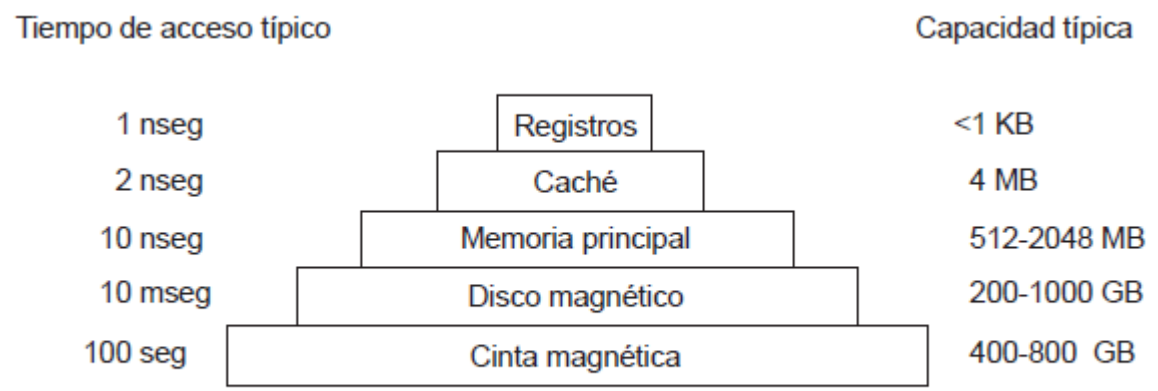


Figura 1-9. Una común jerarquía de memoria. Los números son sólo aproximaciones.

MEMORIA

Memoria Cache

Las líneas de caché que se utilizan con más frecuencia se mantienen en una caché de alta velocidad, ubicada dentro o muy cerca de la CPU. Cuando el programa necesita leer una palabra de memoria, el hardware de la caché comprueba si la línea que se requiere se encuentra en la caché. Si es así (a lo cual se le conoce como **acierto de caché**), la petición de la caché se cumple y no se envía una petición de memoria a través del bus hacia la memoria principal.

La memoria caché está limitada en tamaño debido a su alto costo. Algunas máquinas tienen dos o incluso tres niveles de caché, cada una más lenta y más grande que la anterior.

Las cachés son una idea tan útil que las CPUs modernas tienen dos de ellas.

La **caché L1 o de primer nivel** está siempre dentro de la CPU, y por lo general alimenta las instrucciones decodificadas al motor de ejecución de la CPU.
La mayoría de los chips tienen una segunda caché L1 para las palabras de datos que se utilizan con frecuencia.

caché L2 o de segundo nivel, que contiene varios megabytes de palabras de memoria utilizadas recientemente.

La diferencia entre las cachés L1 y L2 está en la velocidad. El acceso a la caché L1 se realiza sin ningún retraso, mientras que el acceso a la caché L2 requiere un retraso de uno o dos ciclos de reloj.

Todas las peticiones de la CPU que no se puedan satisfacer desde la caché pasan a la memoria principal.

Por lo general a la memoria principal se le conoce como **RAM** (*Random Access Memory*, Memoria de Acceso Aleatorio).

Además de la memoria principal, muchas computadoras tienen una pequeña cantidad de memoria de acceso aleatorio no volátil. A diferencia de la RAM, la memoria no volátil no pierde su contenido cuando se desconecta la energía.

La **ROM** (*Read Only Memory*, Memoria de sólo lectura) se programa en la fábrica y no puede modificarse después. Es rápida y económica. En algunas computadoras, el cargador de arranque (*bootstrap loader*) que se utiliza para iniciar la computadora está contenido en la ROM.

La **EEPROM** (*Electrically Erasable PROM*, PROM eléctricamente borrable) y la **memoria flash** son también no volátiles, pero en contraste con la ROM se pueden borrar y volver a escribir datos en ellas. Sin embargo, para escribir en este tipo de memorias se requiere mucho más tiempo que para escribir en la RAM, por lo cual se utilizan en la misma forma que la ROM, sólo con la característica adicional de que ahora es posible corregir los errores en los programas que contienen.

MEMORIA RAM/ROM

Otro tipo más de memoria es **CMOS**, la cual es volátil. Muchas computadoras utilizan memoria CMOS para guardar la fecha y hora actuales. La memoria CMOS y el circuito de reloj que incrementa la hora en esta memoria están energizados por una pequeña batería, por lo que la hora se actualiza en forma correcta aun cuando la computadora se encuentre desconectada. La memoria CMOS también puede contener los parámetros de configuración, como el disco del cual se debe iniciar el sistema.

PROCESOS E HILOS

El concepto más importante en cualquier sistema operativo es el de proceso.

Cuando se arranca el sistema se inician muchos procesos en forma secreta, lo que a menudo el usuario desconoce.

En cualquier sistema de multiprogramación, la CPU conmuta de un proceso a otro con rapidez, ejecutando cada uno durante décimas o centésimas de milisegundos, hablando en sentido estricto, en cualquier instante la CPU está ejecutando sólo un proceso, y en el transcurso de 1 segundo podría trabajar en varios de ellos, dando la apariencia de un paralelismo.

Un proceso no es más que una instancia de un programa en ejecución, incluyendo los valores actuales del contador de programa, los registros y las variables.

la CPU real conmuta de un proceso a otro. Esta conmutación rápida de un proceso a otro se conoce como multiprogramación. Cuando decimos que una CPU puede ejecutar sólo un proceso a la vez, si hay dos núcleos (o CPUs) cada uno de ellos puede ejecutar sólo un proceso a la vez.

PLANIFICACION DE LOS PROCESOS

Objetivos de la planificacion:
Maximizar el uso del CPU.
Minimizar el tiempo de espera.
Minimizar el tiempo de retorno.
Minimizar el tiempo de respuesta

_____

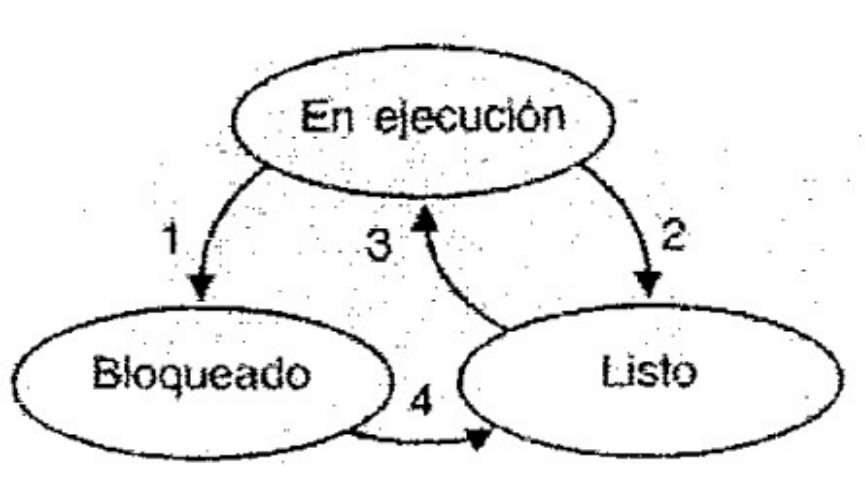
= Eficientizar el uso de los recursos

EL MODELO DEL PROCESO

Todo el software ejecutable en la computadora, se organiza en varios procesos secuenciales.

La idea clave es que un proceso es una actividad de cierto tipo: tiene un programa, una entrada, una salida y un estado.

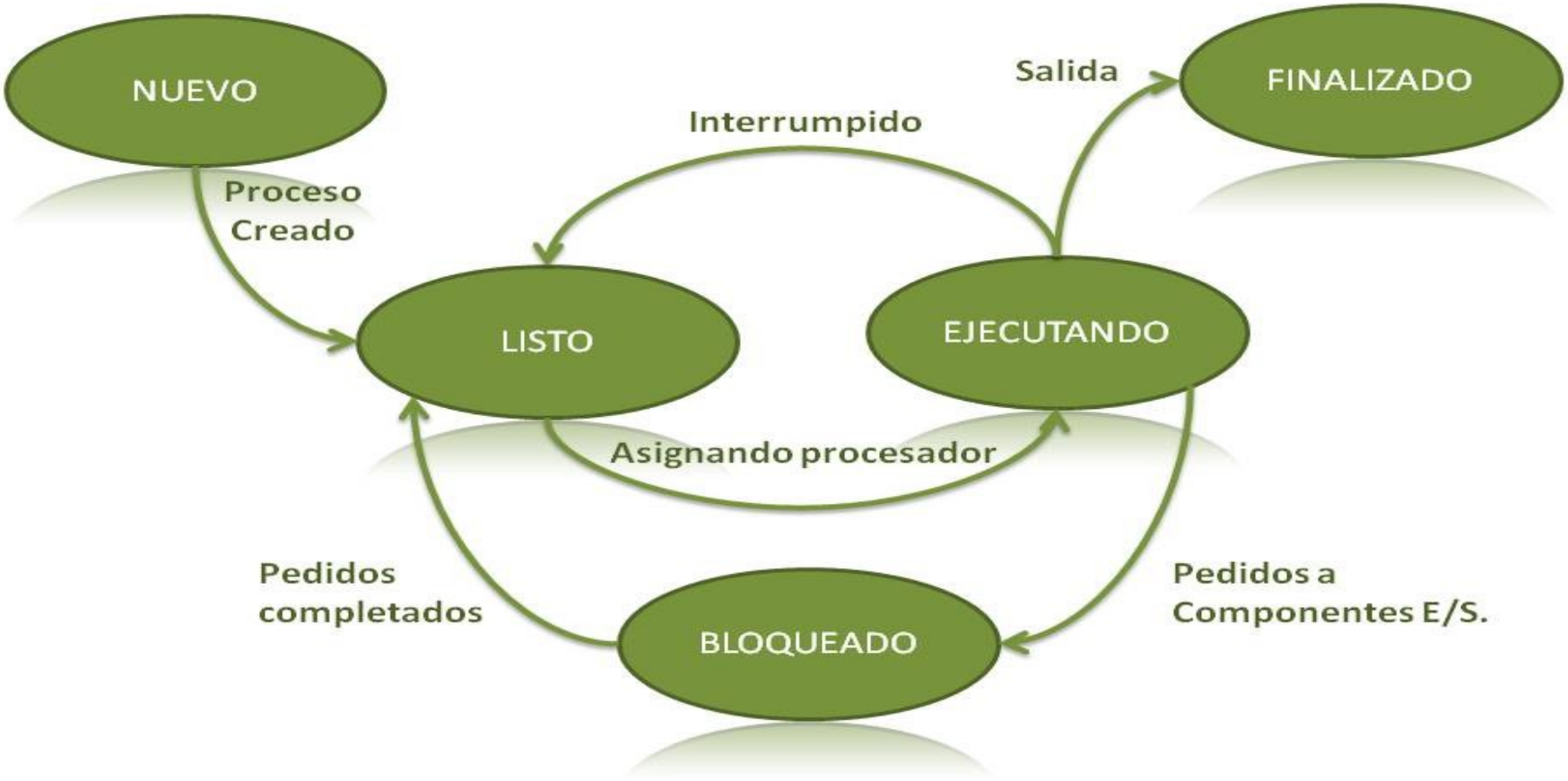
Varios procesos pueden compartir un solo procesador mediante el uso de un algoritmo de planificación para determinar cuándo se debe detener el trabajo en un proceso para dar servicio a otro.



EJEMPLO:

a menudo es posible iniciar un procesador de palabras dos veces o imprimir dos archivos al mismo tiempo si hay dos impresoras disponibles. El hecho de que dos procesos en ejecución tengan el mismo programa no importa; son procesos distintos. El sistema operativo puede compartir el código entre ellos de manera que sólo haya una copia en la memoria, pero ése es un detalle técnico que no cambia la situación conceptual de dos procesos en ejecución.

Estados de los Procesos



ANALOGIA

La diferencia entre un proceso y un programa es sutil pero crucial. Aquí podría ayudarnos una analogía: un científico computacional con mente culinaria hornea un pastel de cumpleaños para su hija; tiene la receta para un pastel de cumpleaños y una cocina bien equipada con todos los ingredientes: harina, huevos, azúcar, extracto de vainilla, etcétera.En esta analogía, la receta es el programa (es decir, un algoritmo expresado en cierta notación adecuada), el científico computacional es el procesador (CPU) y los ingredientes del pastel son los datos de entrada. El proceso es la actividad que consiste en que nuestro cocinero vaya leyendo la receta, obteniendo los ingredientes y horneando el pastel.

Ahora imagine que el hijo del científico entra corriendo y gritando que una abeja acaba de picarlo. El científico computacional registra el punto de la receta en el que estaba (el estado del proceso en curso se guarda), saca un libro de primeros auxilios y empieza a seguir las instrucciones que contiene. Aquí el procesador conmuta de un proceso (hornear el pastel) a uno de mayor prioridad (administrar cuidados médicos), cada uno de los cuales tiene un programa distinto (la receta y el libro de primeros auxilios). Cuando ya se ha ocupado de la picadura de abeja, el científico computacional regresa a su pastel y continúa en el punto en el que se había quedado.

Hay cuatro eventos principales que provocan la creación de procesos:

El arranque del sistema.
La ejecución, desde un proceso, de una llamada al sistema para creación de procesos.
Una petición de usuario para crear un proceso.
El inicio de un trabajo por lotes.

Terminación de procesos

Una vez que se crea un proceso, empieza a ejecutarse y realiza el trabajo al que está destinado. Sin embargo, nada dura para siempre, ni siquiera los procesos. Tarde o temprano el nuevo proceso terminará, por lo general debido a una de las siguientes condiciones:

Salida normal (voluntaria).
Salida por error (voluntaria).
Error fatal (involuntaria).
Eliminado por otro proceso (involuntaria).

La mayoría de los procesos terminan debido a que han concluido su trabajo. Cuando un compilador ha compilado el programa que recibe, ejecuta una llamada al sistema para indicar al sistema operativo que ha terminado. Esta llamada es `exit` en UNIX y `ExitProcess` en Windows. Los programas orientados a pantalla también admiten la terminación voluntaria.

Profesor: Kennedy Sanchez

- Ingeniero en Sistemas Informáticos Universidad Central del Este (UCE)
- Profesor Universidad Dominicana O&M
- Maestria en Gerencia y Productividad - Universidad APEC
- Postgrado de Auditoria de Sistemas - Universidad O&M
- Comptia Security+ Certified

@ksanchez_cld

ksanchez_cld