

Sistemas Operativos

UNIDAD 3 – SISTEMA DE PROCESOS
CONTROL, MODOS Y EJECUCION

Índice

- Modos de Ejecución
- Creación de procesos
- Cambio de proceso
 - Cuándo se realiza
 - Interrupción de un proceso
 - Cambio de modo
 - Cambio de estado del proceso
- Ubicación del Sistema Operativo en Ejecución

Modos de ejecución

- Las aplicaciones no deben poder usar todas las instrucciones de la CPU.
- El Sistema Operativo, tiene que poder utilizar todo el juego de instrucciones de la CPU.

Por ello, una CPU debe tener (al menos) dos modos de operación diferentes para la ejecución de las instrucciones:

Modo Privilegiado (Privileged Mode – kernel space)

- Tiene acceso completo a los recursos hardware.
- Se puede ejecutar cualquier instrucción del repertorio de la CPU.
- Normalmente está reservado sólo para las capas más internas del propio S.O.
- Un error en ejecución en modo kernel es catastrófico y provoca el paro completo del sistema.
- Gracias a esto el kernel mantiene el control del sistema

Modo Usuario (User Mode – user space)

- Acceso limitado a los recursos hardware
- Si el código intenta ejecutar una instrucción privilegiada el microprocesador avisa al kernel quien normalmente mata al proceso, no cae el sistema.
- Tiene que pedirle al kernel, via APIs de llamadas al sistema, que ejecute la operación por el, por ejemplo para:
 - Acceso a ficheros y a la red, Crear y destruir procesos, Apropiarse de más memoria (allocating memory)

Algunas instrucciones sólo modo Privilegiado

Lectura y modificación de registros de control (PSW)
Instrucciones de E/S primitivas (kernel)
Instrucciones de gestión de memoria

Algunas zonas de memoria quedan restringidas

Gestión de procesos

- Creación y terminación de procesos
- Planificación y activación de procesos
- Intercambio de procesos
- Sincronización de procesos y soporte para IPC
- Gestión de los PCB

Gestión de Memoria

- Reserva del espacio de direcciones para los procesos
- Swapping
- Gestión de páginas y Segmentos

Gestión de E/S

- Gestión de buffers
- Reserva de canales de E/S y dispositivos para los procesos

Funciones de Soporte

- Gestión de interrupciones
- Auditoría de procesos y sistema
- Monitorización del sistema (uso de recursos)

Funciones típicas del kernel

Modos de ejecución

¿Cómo sabe el procesador en qué modo se está ejecutando?

¿Cómo se puede cambiar el modo de ejecución?

- El modo de ejecución lo indica un bit (o varios) del PSW (Program Status Word).
 - Con varios bits puedo tener distintos niveles de ejecución → 0 para kernel mode, > 0 User mode
- Este modo se cambia a Kernel Mode en base a ciertos eventos:
 - Interrupciones
 - Llamadas al sistema
- Tras la llamada al sistema o el fin de la rutina de interrupción, vuelve a User Mode.

¿Cómo se cambia a Kernel Mode?

Ejemplo, Sucede una interrupción:

El hardware (procesador):

1. Guarda el estado
2. Establece el bit indicando Kernel Mode
3. A la vez que coloca en el PC la dirección de la rutina de tratamiento de interrupciones.
4. El controlador de interrupciones ha colocado en el bus de datos la entrada de la tabla de interrupciones.

La rutina de tratamiento captura del bus de datos la entrada de la tabla de interrupciones.

Se salta a esta rutina ya en Kernel Mode.

Retorno de interrupción

¿Cómo se vuelve a User Mode?

- El/los bits de modo se establecen de nuevo a User Mode con una instrucción (privilegiada) de cambio de modo como última instrucción de la rutina de tratamiento de la interrupción.
- O simplemente se cambiará al restaurar el PSW del programa en ejecución

Creación de procesos

Una vez que el S.O. decide crear un proceso, lo crea siguiendo estos pasos:

1. Asignar un identificador ID único

Se añade la nueva entrada en la tabla principal de procesos

2. Reservar espacio para el proceso

Reserva de espacio para todos los elementos de la imagen del proceso

Código + Datos + Pila → Tamaños por defecto, por la petición de creación o los toma el S.O. del padre.

Se asignan enlaces a zonas compartidas.

Reserva de espacio para el PCB

3. Inicializar el PCB

La mayoría de las entradas a cero excepto:

PC, Punteros de pilas, PPID, PID, estado (Listo o Listo/Suspendido), prioridad, etc.

No tendrá descriptores de ficheros asignados, excepto los heredados del padre (si procede).

4. Establecer los enlaces apropiados

Insertar referencia al proceso en la cola enlazada de procesos (listo o listo/suspendido)

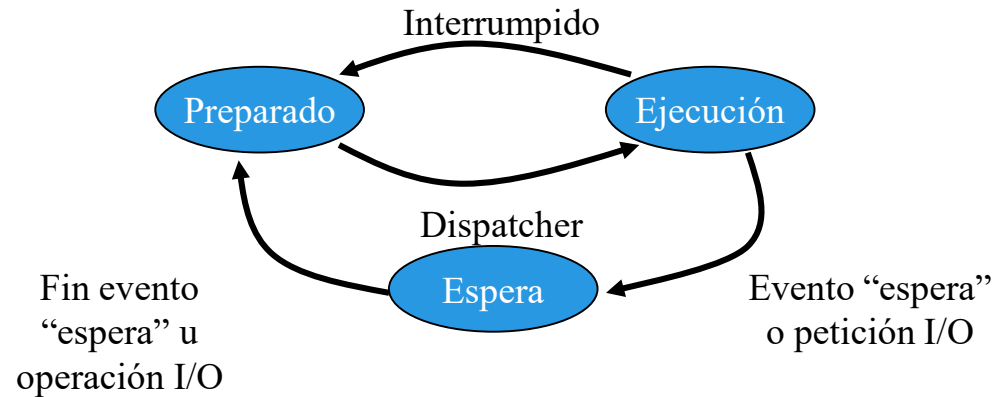
Ajustar las referencias cruzadas en las estructuras de datos.

5. Crear o ampliar otras estructuras

Creación de estructuras para auditoría

Cambio de proceso

El S.O. realiza el cambio de proceso para mantener la multiprogramación



¿Qué evento/s disparan un cambio de proceso?

Hay que diferenciar entre Process Switching y Mode Switching (cambio de proceso y modo)

¿Qué hace el S.O. con las estructuras de datos ante los cambios de proceso y modo?

Cambio de proceso – Cuándo se realiza

En cualquier momento en que el S.O. tome el control de ejecución.

Se interrumpe la ejecución de un proceso → El S.O. toma el control de ejecución

Tabla 3.8. Mecanismos para la interrupción de la ejecución de un proceso.

Mecanismo	Causa	Uso
Interrupción	Externa a la ejecución del proceso actualmente en ejecución.	Reacción ante un evento externo asíncrono.
<i>Trap</i>	Asociada a la ejecución de la instrucción actual.	Manejo de una condición de error o de excepción.
Llamada al sistema	Solicitud explícita.	Llamada a una función del sistema operativo.

Cambio de proceso – Interrupción de un proceso

Interrupciones

- **Ordinarias (Interrupciones Hardware)**

1- El control pasa a la rutina de gestión de interrupciones.

2- Luego se salta a la rutina de tratamiento específico en función del tipo de interrupción.

Ejemplos:

Interrupción de reloj

Quantum. El proceso debe pasar a Listos o Listos Suspendido

E/S

El SO determina el evento.

Si hay procesos a la espera los traslada de Bloqueado a Listo.

Puede reanudar el mismo proceso o cambiar.

Fallo de página

El SO debe cargar la página o segmento en memoria principal

El proceso actual debe pasar a Bloqueado.

Se produce un cambio de proceso.

Cuando la página o segmento estén disponibles (fin E/S) el proceso pasará a Listo

- **Traps (Cepos o Interrupciones Software)**

El SO determina si el error es fatal

→ Lo es → Termina el proceso,

→ No lo es → Depende del diseño del S.O. (rutinas de recuperación o código error al proceso)

Llamadas al sistema

El sistema pasa a modo kernel → rutina de tratamiento → el proceso puede bloquearse o no.

Cada interrupción tiene un nivel de prioridad. Las de baja prioridad (p.ej. teclado) deben esperar a que se traten las de mayor prioridad (p.ej. temporizador, disk events)

El S.O. está diseñado para dar mayor prioridad a determinados eventos de forma que la respuesta del Sistema sea suave (sin retardos).

El S.O. debe conocer en startup que IRQ (hard) va a ser atendida por qué driver.

Cambio de proceso – Cambio de Modo

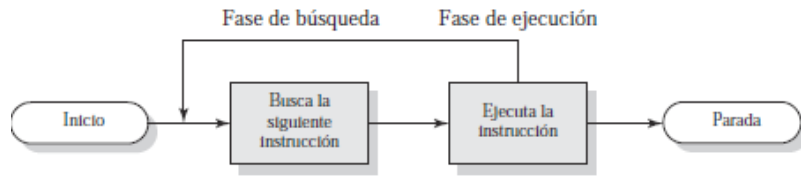


Figura 1.2. Ciclo de instrucción básico.

El ciclo de instrucción es “ininterrumpible”

Tras el ciclo de instrucción tenemos

la fase de interrupción → Donde se comprueba si ha habido interrupciones

Si no hay interrupciones pendientes → Siguiendo ciclo de instrucción

Si hay interrupción, el hardware:

1. $PC \leftarrow$ Dirección de la rutina de tratamiento de interrupción
2. Cambio de modo usuario a modo núcleo
3. Salvaguarda el contexto del procesador

La siguiente instrucción será de la rutina de tratamiento de interrupciones y en modo kernel.

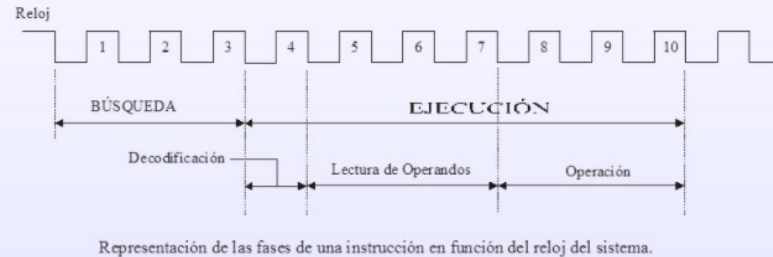
Borra el flag de interrupciones.

Copia el contexto del procesador en el PCB del proceso interrumpido.

La existencia de una interrupción → Provoca un cambio de modo

Pero no necesariamente un cambio de proceso. Eso lo decide el Planificador del S.O.

La culminación de cada una de estas fases necesita de uno a seis ciclos de reloj.



Ubicación del Sistema Operativo en Ejecución

El Sistema Operativo:

Se ejecuta en el procesador (al igual que cualquier software)

Cede el control del procesador para que se ejecute otro software

Depende del procesador para recuperar el control.

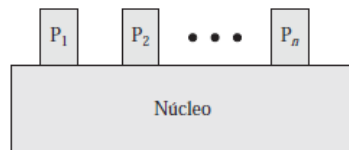
¿Es el S.O. un proceso del sistema?

¿Quién/cómo se controla ese proceso?

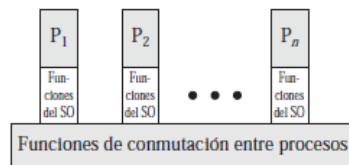
¿Dónde se ubica su código en memoria?

Ubicación del Sistema Operativo en Ejecución

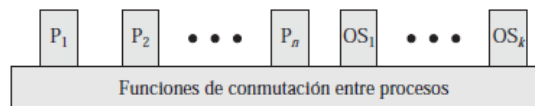
- ¿Es el S.O. un proceso del sistema?
- ¿Quién/cómo se controla ese proceso?
- ¿Dónde se ubica su código en memoria?



(a) Núcleo independiente



(b) Funciones del SO se ejecutan dentro del proceso de usuario



(c) Funciones del SO se ejecutan como procesos independiente

Figura 3.15. Relación entre el sistema operativo y los procesos de usuario.

Núcleo fuera de todo proceso (S.O. antiguos):

- Ejecuta el núcleo del S.O. como una entidad separada que opera en modo kernel, con su memoria protegida y pila propia.
- El concepto de proceso se aplica sólo a los programas de usuario.

Ejecución dentro de los procesos de usuario (PCs y workstations):

- Software del S.O. en el contexto de un proceso de usuario.

Sistema operativo basado en procesos:

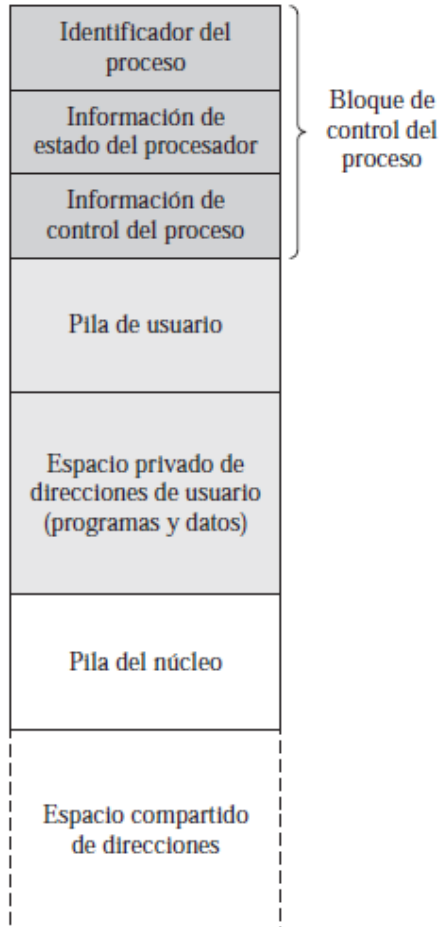
- Software del S.O. como procesos separados.

Ubicación del Sistema Operativo en Ejecución

¿Es el S.O. un proceso del sistema?

¿Quién/cómo se controla ese proceso?

¿Dónde se ubica su código en memoria?



Ejecución dentro de los procesos de usuario (PCs y workstations):

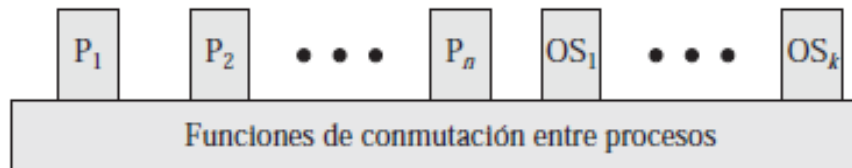
- **El SO es una colección de rutinas** a las que el usuario llama.
- **Se mantiene una pila del kernel en su imagen** para cada proceso
- **Se enlazan (*linkan*) las rutinas del kernel** necesarias para el proceso y se añaden en la imagen del proceso.
- Realmente se accede al conjunto de rutinas **situadas en memoria compartida** por el kernel con los procesos que las usan.
- Estas rutinas **se ejecutan en modo kernel** al que se pasa por interrupción (hw/sw) y **se produce un cambio de modo pero no de proceso**.
- Antes de retornar a user mode se comprueba si puede/debe continuar el mismo proceso, si es así, simplemente **se cambia de modo no de proceso**.

Ubicación del Sistema Operativo en Ejecución

- ¿Es el S.O. un proceso del sistema?
- ¿Quién/cómo se controla ese proceso?
- ¿Dónde se ubica su código en memoria?

Sistema Operativo basado en Procesos:

- Las funciones del sistema operativo más importantes se ejecutan como si fueran procesos.
- El kernel se reduce a lo mínimo, cambio de procesos, seguridad, intercambio, etc...
- Favorece la programación modular del S.O.
- Permite enviar procesos del S.O. a procesadores dedicados en sistemas multiprocesador.



(c) Funciones del SO se ejecutan como procesos independiente

Fin

UNIDAD 3 – SISTEMA DE PROCEOSS
CONTROL, MODOS Y EJECUCION