

Sistemas Operativos

Tema 3 Procesos

Contenido

- 3.1 – Conceptos
- 3.2 – Estado de un proceso
 - 3.2.1 – Creación de procesos
 - 3.2.2 – Terminación de procesos
 - 3.2.3 – Modelo de 5 estados
 - 3.2.4 – Procesos suspendidos
 - 3.2.5 – Modelo de 7 estados
 - 3.2.6 – Razones para suspender procesos
- 3.3 – Descripción de procesos
 - 3.3.1 – Estructuras de control del SO
 - 3.3.2 – Estructuras de control de procesos

Contenido

- 3.4 – Control de procesos
 - 3.4.1 – Modos de ejecución
 - 3.4.2 – Creación de procesos
 - 3.4.3 – Cambio de proceso
 - 3.4.4 – Cambio de proceso VS cambio de contexto
- 3.5 – Ejecución del SO
- 3.6 – Procesos e hilos
- 3.7 – Planificación
- 3.8 – Algoritmos de planificación

3.1 - Conceptos

- Recordar definición de proceso:
 - *Programa en ejecución*
 - *Unidad de trabajo asignable al procesador para su ejecución*
- Los procesos permiten trabajar con varios programas a la vez: pseudoparalelismo
 - *Multiprogramación*: rápida conmutación entre procesos.
 - Para tener *paralelismo* es necesaria una máquina multiprocesador.

3.2 - Estado de un proceso

- Para conseguir la máxima eficiencia se intercala la ejecución de los procesos.
- Se pueden definir varios estados en los que se pueden encontrar los procesos.

3.2.1 – Creación de procesos

TABLA 3.1 Razones para la Creación de Procesos

Nuevo trabajo por lotes	El sistema operativo está provisto de un flujo de control de trabajos por lotes, generalmente para cinta o disco. Cuando el sistema operativo se prepara para tomar un nuevo trabajo, leerá la próxima secuencia de órdenes de control de trabajos.
Conexión interactiva Creado por el SO para dar un servicio	Un usuario entra en el sistema desde un terminal. El sistema operativo puede crear un proceso para llevar a cabo una función de parte de un programa usuario, sin que el usuario tenga que esperar (por ejemplo, imprimir).
Generado por un proceso existente	Con afán de modularidad o para aprovechar el paralelismo, un programa de usuario puede ordenar la creación de una serie de procesos.

3.2.2 – Terminación de procesos

Terminación normal	El proceso ejecuta una llamada a un servicio del SO que indica que ha terminado de ejecutar.
Tiempo límite excedido	El proceso ha ejecutado por más del tiempo límite total especificado. Hay varias posibilidades para la clase de tiempo que se mide. Entre éstas se incluyen el tiempo total transcurrido ("tiempo de reloj"), el tiempo que ha estado ejecutando y, en el caso de un proceso interactivo, el tiempo transcurrido desde que el usuario realizó su última entrada de datos.
No hay memoria disponible	El proceso necesita más memoria de la que el sistema le puede proporcionar
Violación de límites	El proceso trata de acceder a una posición de memoria a la que no le está permitido acceder.
Error de protección	El proceso intenta utilizar un recurso o un archivo que no le está permitido utilizar, o trata de utilizarlo de forma incorrecta, como escribir en un archivo que es sólo de lectura.
Error aritmético	El proceso intenta hacer un cálculo prohibido, como una división por cero, o trata de almacenar un número mayor del que el hardware acepta.
Tiempo máximo de espera rebasado	El proceso ha esperado más allá del tiempo máximo especificado para que se produzca cierto suceso.

3.2.2 – Terminación de procesos

Fallo de E/S	Se produce un error en la entrada o la salida, tal como la incapacidad de encontrar un archivo, un fallo de lectura o escritura después de un número máximo de intentos (cuando, por ejemplo, hay un región defectuosa en una cinta), o una operación ilegal (como intentar leer de una impresora)
Instrucción inválida	El proceso intenta ejecutar una instrucción inexistente (a menudo como resultado de un salto a una zona de datos para intentar ejecutar los datos).
Instrucción privilegiada	El proceso intenta usar una instrucción reservada para el sistema operativo.
Mal uso de los datos	Un elemento de dato es de un tipo equivocado o no está inicializado.
Intervención del operador o del SO	Por alguna razón el operador o el sistema operativo termina con el proceso (por ejemplo, si existe un interbloqueo).
Terminación del padre	Cuando un proceso padre finaliza, el sistema operativo puede diseñarse para terminar automáticamente con todos sus descendientes.
Solicitud del padre	Un proceso padre tiene normalmente la autoridad de terminar con cualquiera de sus descendientes.

3.2.3 – Modelo de 5 estados

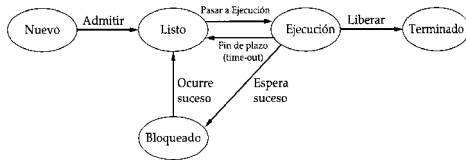


FIGURA 3.5 Modelo de procesos de cinco estados

3.2.3 – Modelo de 5 estados

Definición de los estados:

- **Ejecución:** El proceso que está actualmente en ejecución. En computadores con un único procesador sólo un proceso, a lo sumo, puede estar en este estado en un instante dado.
- **Listo:** Proceso que está preparado para ejecutar, en cuanto se le dé la oportunidad.
- **Bloqueado:** Proceso que no puede ejecutar hasta que se produzca cierto suceso, como la terminación de una operación de E/S.
- **Nuevo:** Proceso que se acaba de crear, pero que aún no ha sido admitido por el sistema operativo en el grupo de procesos ejecutables.
- **Terminado:** Un proceso que ha sido excluido por el sistema operativo del grupo de procesos ejecutables, bien porque se detuvo o porque fue abandonado por alguna razón.

3.2.3 – Modelo de 5 estados

Descripción de las transiciones:

- **Nulo → Nuevo:** Se crea un nuevo proceso para ejecutar un programa. Este suceso se produce por algunas de las razones enumeradas anteriormente.
- **Nuevo → Listo:** El sistema operativo pasará un proceso del estado Nuevo al estado Listo cuando esté preparado para aceptar un proceso más. La mayoría de los sistemas ponen un límite en función del número de procesos existente o en la cantidad de memoria virtual dedicada a los procesos existentes. El motivo de este límite es asegurar que no haya tantos procesos activos como para degradar el rendimiento.
- **Listo → Ejecución:** Cuando es hora de seleccionar un nuevo proceso para ejecutar, el sistema operativo elige a uno de los procesos del estado Listo. La cuestión de qué proceso se escoge es la planificación.
- **Ejecución → Terminado:** El proceso que se está ejecutando es finalizado por el sistema operativo si indica que terminó o si se abandona.

3.2.3 – Modelo de 5 estados

Descripción de las transiciones:

- **Ejecución → Listo:** La razón más común de esta transición es que el proceso que está en ejecución ha alcanzado el tiempo máximo permitido de ejecución ininterrumpida; casi todos los sistemas operativos con multiprogramación imponen este tipo de norma de tiempo. Hay otras causas alternativas para esta transición que no están implementadas en todos los sistemas operativos.
- **Ejecución → Bloqueado:** Un proceso se pone en el estado Bloqueado si solicita algo por lo que debe esperar.
- **Bloqueado → Listo:** Un proceso que está en el estado Bloqueado pasará al estado Listo cuando se produzca el suceso que estaba esperando.
- **Listo → Terminado:** Por razones de claridad, esta transición no se muestra en el diagrama de estados de la figura 3.5. En algunos sistemas, un padre puede terminar con un proceso hijo en cualquier momento. Además, si el padre termina, todos los procesos hijos asociados con él pueden ser finalizados.
- **Bloqueado → Terminado:** Se aplica el mismo comentario que en el caso anterior.

3.2.3 – Modelo de 5 estados

Implementación:

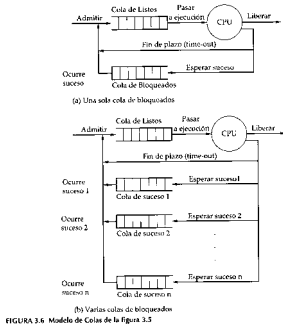
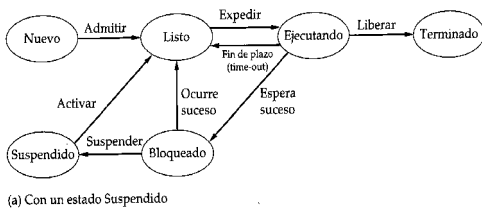


FIGURA 3.4. Modelo de Colas de la figura 3.3

3.2.4 – Procesos suspendidos

- El modelo de 5 estados es suficiente, pero según este, todos los procesos están completamente en MP
- Aunque haya varios procesos en MP, el procesador es tan rápido (vs ops de E/S) que todos suelen estar bloqueados.
- Hay dos soluciones:
 - Ampliar la memoria principal.
 - Intercambio: mover una parte del proceso (o todo) al disco duro

3.2.4 – Procesos suspendidos

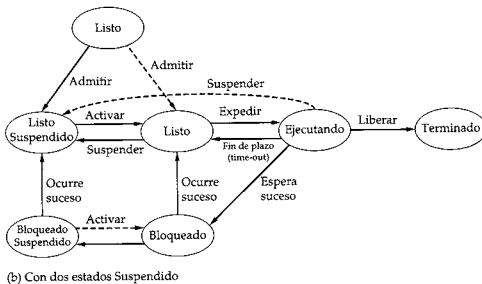


(a) Con un estado Suspendido

3.2.4 – Procesos suspendidos

- Problemas: Es difícil encontrar qué proceso suspendido está listo para ser ejecutado cuando ocurre el suceso que esperaba.
- Solución: Mantener dos colas de suspendidos (modelo de 7 estados)

3.2.5 – Modelo de 7 estados



(b) Con dos estados Suspendido

3.2.5 – Modelo de 7 estados

- Nuevos estados:
 - **Listo**: El proceso está en memoria principal y listo para la ejecución.
 - **Bloqueado**: El proceso está en memoria principal esperando un suceso.
 - **Bloqueado y suspendido**: El proceso está en memoria secundaria esperando un suceso.
 - **Listo y suspendido**: El proceso está en memoria secundaria pero está disponible para su ejecución tan pronto como se cargue en la memoria principal.

3.2.5 – Modelo de 7 estados

• Nuevas transiciones

- **Bloqueado -> Bloqueado y suspendido:** Si no hay procesos Listos, entonces al menos un proceso Bloqueado se expulsa para dar cabida a otro proceso que no esté bloqueado.
- **Bloqueado y suspendido -> Listo y suspendido:** Un proceso en estado Bloqueado y suspendido se pasa al estado Listo y suspendido cuando ocurre el suceso que estaba esperando.
- **Listo y suspendido -> Listo:** Cuando no hay procesos Listos en la memoria principal, el sistema operativo tendrá que traer uno para continuar la ejecución.
- **Listo -> Listo y suspendido:** Generalmente, el sistema operativo prefiere suspender a un proceso Bloqueado en vez de a uno Listo, ya que el proceso Listo podría ejecutarse de inmediato, mientras que el proceso Bloqueado estará ocupando espacio en la memoria principal sin poder ejecutarse. Sin embargo, puede ser necesario suspender un proceso Listo si ésta es la única forma de liberar un bloque lo suficientemente grande de memoria principal.

3.2.5 Modelo de 7 estados

• Otras transiciones interesantes:

- **Nuevo -> Listo y Suspendido y Nuevo -> Listo:** Cuando se crea un nuevo proceso, se le puede añadir a la cola de Listos o a la de Listos y suspendidos.
- **Bloqueado y suspendido -> Bloqueado:** La siguiente situación es posible: Un proceso termina, liberando memoria principal. Hay un proceso en la cola de Bloqueados y suspendidos que tiene una prioridad mayor que la de cualquier proceso de la cola de Listos y suspendidos, así que el sistema operativo tiene razones para suponer que pronto ocurrirá el suceso por el que el proceso está bloqueado.
- **Ejecución -> Listo y suspendido:** Si se está expulsando al proceso porque hay un proceso de prioridad mayor en la lista de Bloqueados y suspendidos que se acaba de desbloquear, entonces el sistema operativo podría pasar el proceso en Ejecución directamente a la cola de Listos y suspendidos, liberando espacio en la memoria principal.
- **Varios -> Terminado.**

3.2.6 – Razones para suspender un proceso

TABLA 3.6 Razones para la Suspensión de procesos

Intercambio	El sistema operativo necesita liberar suficiente memoria principal para cargar un proceso que está listo para ejecutarse.
Otra razón del SO	El sistema operativo puede suspender un proceso de fondo, de utilidad o cualquier proceso que se sospecha sea el causante de un problema.
Solicitud de un usuario	Un usuario puede querer suspender la ejecución de un programa con fines de depuración o en conexión con el uso de un recurso.
Por tiempo	Un proceso puede ejecutarse periódicamente (por ejemplo, un proceso de contabilidad o de supervisión del sistema) y puede ser suspendido mientras espera el siguiente intervalo de tiempo.
Solicitud del proceso padre	Un proceso padre puede querer suspender la ejecución de un descendiente para examinar o modificar el proceso suspendido o para coordinar la actividad de varios descendientes.

3.3 – Descripción de procesos

- El SO ha de administrar el uso de los recursos del sistema por parte de los procesos.
- ¿Qué necesita el SO para poder controlar los procesos y administrar los recursos para ellos?

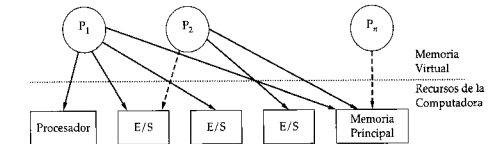


FIGURA 3.8 Procesos y recursos

3.3.1 – Estructuras de control del SO

- Para la administración es necesario mantener información sobre **procesos y recursos**. El SO construye y mantiene unas tablas de información sobre cada entidad administrada.
- Las organizamos en 4 categorías:
 - Tablas de memoria (asignación de MP y MS, atributos de protección, información para la gestión de MV...)
 - Tablas de E/S (dispositivos de E/S libres, asignados, operaciones en marcha, ...)
 - Tablas de archivos (existencia de archivos, ubicación en MS, estado actual, atributos, ...)
 - Tablas de procesos

3.3.1 Estructuras de control del SO

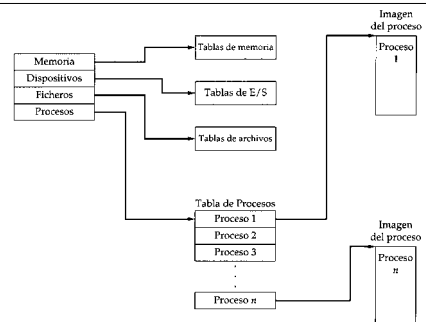


FIGURA 3.9 Estructura general de las tablas de control del sistema operativo

3.3.2 – Estructuras de control de procesos

- Para administrar y controlar los procesos el SO debe conocer datos sobre:
 - **Ubicación:** ¿Dónde está el proceso? Dirección de MP o MS donde se encuentra. Para la administración, parte de la información debe estar en MP.
 - **Atributos:** ¿Cómo es el proceso?
- Imagen de un proceso
 - Programa
 - Datos
 - Pila, parámetros, ...
 - Atributos (contexto) → Bloque de control de proceso

3.3.2 – Estructuras de control de procesos

- El Bloque de control de proceso (BCP) está formado por 3 categorías principales:
 - **Identificación del proceso**
 - **Información de estado del procesador**
 - **Información de control del proceso**
- El BCP es la estructura de datos central de un SO. Los bloques son leídos y/o modificados por muchos módulos del SO.
- Es necesario proteger el acceso a los BCP (errores, cambio de diseño) → Árbitro

3.4 – Control de procesos

3.4.1 – Modos de ejecución

Procesadores → dos modos de ejecución (al menos)

- El menos privilegiado: Modo usuario.
- El más privilegiado: Modo sistema, núcleo o control.
- Ciertas operaciones solo pueden ejecutarse en modo privilegiado (primitivas de E/S, R/W de registros de control, gestión de memoria, ...).
- En la PSW se guarda un bit que indica el modo utilizado.

3.4.2 – Pasos para la creación de procesos

1. Asignar un identificador único. Se añade una nueva entrada a la tabla principal de procesos.
2. Asignar espacio para el proceso
 - Espacio para el BCP
 - Es necesario saber cuanto ocupará (código, datos, ...)
3. Inicializar el BCP
4. Establecer enlaces necesarios (estado, ...)
5. Crear/Modificar otras estructuras de datos (contabilidad, ...)

3.4.3 – Cambio de proceso

Sucesos que pueden dar el control al SO:

Mecanismo	Causa	Uso
Interrupción	Externa a la ejecución de la instrucción en curso	Reacción a un suceso externo
Cepo	Asociada a la ejecución de la instrucción en curso	Tratamiento de un error o de una condición excepcional
Llamada al sistema	Solicitud explícita	Llamada a una función del SO

3.4.4 – Cambio de proceso VS cambio de estado

¿Qué pasa cuando hay una interrupción?

El procesador ha de:

- **Salvar el contexto** del programa en ejecución
- Asignar al CP la dirección de comienzo de un programa de **tratamiento de interrupción**.

¿Qué contexto se salva?

Cualquiera que pudiera alterarse por la ejecución de la rutina de tratamiento de la interrupción.

BCP → Info. Del estado del procesador

- Puede tenerse que guardar más cosa si se ha de hacer un cambio de proceso. ¡Esto no siempre es necesario!

3.4.4 – Cambio de proceso VS cambio de estado

Cambio de contexto != cambio de proceso

- Si el proceso se sigue ejecutando, el HW restaura el contexto
- Si no, se cambia el proceso:
 - Salvar el contexto del procesador.
 - Actualizar BCP del proceso que estaba en ejecución.
 - “Mover” BCP a la cola adecuada.
 - Seleccionar otro proceso para ejecutar.
 - Actualizar BCP del proceso a ejecutar.
 - Actualizar estructuras de datos de gestión de memoria.
 - Restaurar el contexto del procesador al que tenía la última vez que se ejecutó el proceso.