

CAPÍTULO VIII: GESTIÓN DE PROCESOS

GESTIÓN DE LA CPU. GESTIÓN DE PROCESOS.

Existen dos formas básicas de trabajar con un computador: **por lotes** (o cola de trabajos) y en forma **interactiva**. En esta última forma la CPU está constantemente atendiendo al usuario y, en cualquier caso, aunque la CPU solape el trabajo del usuario con otros, se tiene la impresión de que se está trabajando directamente y en exclusiva con el computador.

La gestión de la CPU por parte del sistema operativo se centra en el concepto de proceso. Un proceso es un programa en ejecución o en el que se ha iniciado la misma, y que es tratado por el sistema operativo como un todo. Un programa por sí es un ente pasivo, mientras que un proceso es un ente activo.

Un proceso puede generar la llamada a otro, denominándose a ambos proceso padre y proceso hijo, respectivamente. Los procesos hijo acaban con una llamada al sistema operativo (suele denominarse EXIT) que cuando se ejecuta el núcleo del sistema operativo pasa el control al padre y libera la zona de memoria que ocupaba el hijo.

Multiprogramación.

Los primeros sistemas operativos se denominaban de monotarea o serie; en ellos, hasta que no finaliza la ejecución de un programa no empieza a ejecutarse otro.

Un sistema operativo multitarea aprovecha los tiempos muertos de la CPU, los tiempos muertos en periféricos y los espacios de memoria principal no ocupados por el proceso. Consiste, en esencia, en cargar en la memoria principal varios procesos e ir asignando la CPU sucesivamente a los distintos procesos en memoria, de forma que se aproveche al máximo la CPU y varios procesos vayan avanzando en su ejecución, sin necesidad de que finalice completamente uno para iniciar la ejecución de otro.

Se dice que un proceso entra en estado de bloqueo cuando la CPU no puede continuar trabajando con él, a causa de tener que esperar la realización de un operación de entrada/salida o a algún otro evento de naturaleza similar. Se dice que un proceso está en estado preparado o ejecutable cuando la CPU puede iniciar o continuar su ejecución. Un proceso se dice que está en estado activo o de ejecución, cuando la CPU está ejecutando instrucciones del mismo.

En los sistemas multitarea puros o clásicos, cuando un proceso entra en estado de bloqueo, un módulo del sistema operativo denominado "distribuidor" (dispatcher) pasa el turno de ejecución a uno de los procesos en memoria principal que esté en estado preparado.

Un sistema multitarea se denomina de multiprogramación si dispone de técnicas apropiadas de protección de memoria y de control de concurrencia para permitir el acceso compartido a dispositivos de E/S y archivos. Si el sistema de multiprogramación prevé el uso concurrente de distintos usuarios (con identificación, autenticación y control de los mismos) se denomina sistema multiusuario.

La idea de tiempo compartido (time sharing) es una forma de gestionar la multiprogramación para obtener sistemas multiusuario, que requieren tiempos de respuesta adecuados, dando la ilusión a cada usuario que está trabajando en exclusiva con la máquina. En realidad, no se trabaja en exclusiva con el computador ya que una CPU sólo puede, en un instante dado, estar dedicada a un proceso (ejecutando una instrucción). Con rigor se dice que la CPU está ejecutando **concurrentemente** (en vez de simultáneamente) varios procesos: en un intervalo de tiempo determinado se han estado ejecutando alternativamente a "trozos" varios procesos ubicados en la memoria central.

El tiempo compartido puede llevarse a cabo eligiendo adecuadamente un algoritmo de planificación de multitarea. Por ejemplo, si existen varios procesos preparados, un problema que debe resolver el "distribuidor" es elegir a cuál de ellos darle el turno (pasar a activo). El módulo del sistema operativo que se encarga de solventar este problema se denomina planificador (scheduler) que implementa un algoritmo de planificación.

Uno de los algoritmos de planificación de mayor interés -por su antigüedad, sencillez y amplio uso- es el de petición circular [round robin] también denominado cooperativo, usado por el Windows- MS.DOS y los computadoras Macintosh. Con este algoritmo, a cada uno de los procesos en memoria se le asigna un intervalo de tiempo fijo, o periodo T , o "cuantum", y se cambia de contexto de un proceso a otro, de forma rotatoria, conforme se van consumiendo los cuanta.

En los sistemas operativos de tiempo compartido, se cambia de contexto, no sólo cuando se le acaba un quantum a un proceso sino también en el instante en que quede bloqueado, es decir, el distribuidor provoca una conmutación de contexto de un proceso P_i a uno P_j , dando el turno a P_j , siempre que se cumpla una de las siguientes condiciones:

- ✍ el proceso P_i agote su quantum,
- ✍ el proceso P_i se bloquee,

donde P_j es el siguiente proceso preparado que le corresponde el turno.

La asignación de cuanta suele hacerse con ayuda de un generador de pulsos sincronizado con la frecuencia de la red de suministro de energía eléctrica (50 Hz en Europa, 60 Hz en USA), transcurrido el período de tiempo correspondiente (20 ms o 16,7 ms) se produce una interrupción que lanza a ejecución el módulo gestor del reloj de tiempo real, que entre otros objetivos sirve de referencia para generar el tiempo asociado al quantum.

La elección del período del quantum es problemática, ya que si es demasiado pequeño (10 ms) hay que hacer muchas conmutaciones de contexto (cada 15 ms se pierden 5 ms en cambio de contexto) y la eficiencia de la CPU es baja (66.6%), pero si es grande (500 ms) los tiempos de respuesta para cada usuario pueden llegar a ser muy bajos (si hubiese ejecutándose concurrentemente 20 procesos, al pulsar una simple tecla podríamos obtener respuestas tan lentas como 5 segundos).

Otro algoritmo de planificación es el de asignación de prioridades, usado por los sistemas operativos OS/2, UNIX y NT-Windows, en los que las prioridades de los procesos no son iguales, pudiendo el planificador modificarlas dinámicamente. El distribuidor da el turno al proceso preparado al que el planificador le haya asignado mayor prioridad. Existen varios criterios de asignación de prioridades. Así, para que el de mayor prioridad (y por tanto activo) no monopolice el uso de la CPU, a cada interrupción de reloj de tiempo real (cada 20 ms) se le baja su prioridad. Se conmuta al siguiente proceso en el momento en que alguno en cola supere la prioridad del proceso activo.

Pueden existir prioridades estáticas (dadas en función de la relevancia de los usuarios) o dinámicas. Es un buen criterio, para obtener un aprovechamiento equilibrado de todos los recursos del computador y obtener mayor **productividad** (número de programas ejecutados en un período determinado de tiempo), dar mayor prioridad a los procesos que tienen más tiempo de E/S

Otros tipos de sistemas operativos de interés son los de multiprocesamiento y de tiempo real.

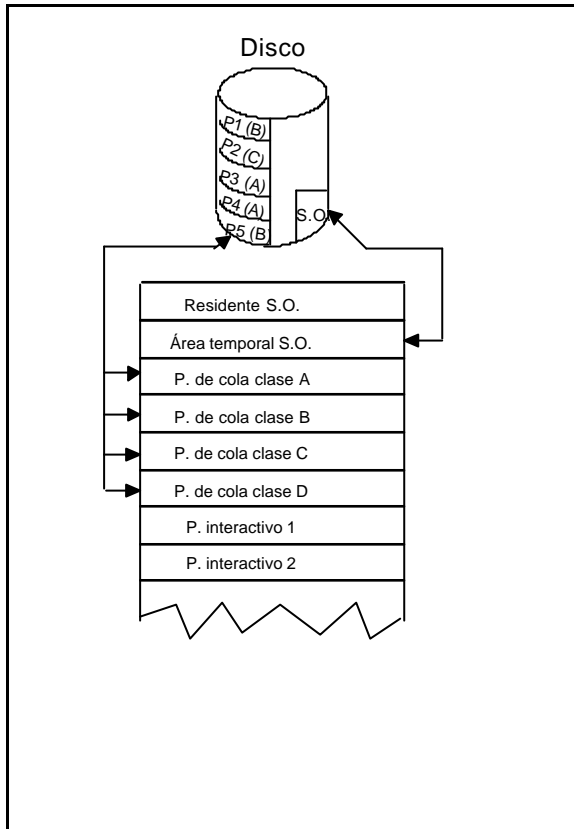
Los sistemas operativos de multiprocesamiento están ideados para actuar con sistema que contienen dos o más procesadores trabajando simultáneamente.

Con el multiprocesamiento asimétrico una CPU principal controla el comportamiento global de todas las CPU's, pudiendo utilizarlas como si fuesen dispositivos colocados en su bus. Con el multiprocesamiento simétrico, se consigue un incremento lineal de prestaciones según el número de CPU's, pero requieren sistemas operativos y compiladores muy sofisticados para poder aprovechar las ventajas del multiprocesamiento. Hay versiones de UNIX para procesamiento asimétrico, y se están desarrollando versiones para procesamiento simétrico. Windows NT admite procesamiento simétrico.

Otros sistemas operativos de gran interés son los sistemas operativos de tiempo real. El concepto de tiempo real hace referencia a que el computador debe dar imprescindiblemente la respuesta dentro de un límite de tiempo preestablecido. Este tiempo puede ser pequeño o grande, dependiendo de la aplicación. Los sistemas en tiempo real se usan ampliamente en control industrial, equipos de conmutación telefónicos, control de vuelo, aplicaciones militares (guía de misiles, etc...) y simuladores de tiempo real.

Secuencias múltiples.

El procesamiento por lotes o en cola se denomina lotes serie o cola-serie, a causa de que con él van pasando a ejecución de uno a uno los procesos en espera. Hay sistemas operativos de **secuencias múltiples** (multi-streams) o lotes-paralelo o colas paralelas, con los que coexisten en memoria, en ejecución concurrente, varios *tipos de trabajos de la cola*.



Para asignar la memoria hay distintas "puertas"; cuatro, por ejemplo: A, B, C y D. Al especificar los parámetros concretos de funcionamiento del sistema operativo, se fijan las características (tiempos de CPU, prioridades, etc..) que deben reunir los trabajos para poder pasar por cada una de las puertas. En ejecución concurrente habrá de tipo A, otro de tipo B, otro de tipo C, y por último uno de tipo D. Cuando acabe la ejecución de uno de dichos trabajos, el de tipo B, por ejemplo, un proceso del sistema operativo denominado **planificador de trabajos** (job scheduler) selecciona de la cola, para introducir en memoria, el trabajo de mayor prioridad que reúna las características del tipo del que acaba (el de tipo B de mayor prioridad).

Además de los procesos del sistema operativo y de la cola, en memoria pueden estar compitiendo por la CPU varios trabajos de tipo interactivo, dados mediante terminales de este tipo. La CPU debe ir procesando estos trabajos según vaya recibiendo las instrucciones tecleadas por los usuarios, y en el momento que el distribuidor les dé la vez. Los sistemas que permiten la utilización aparentemente simultánea del computador por diversos terminales interactivos se denominan **sistemas multiacceso** (multiaccess). Estos sistemas utilizan multiprogramación de tiempo compartido. Ahora bien, un sistema de tiempo compartido no ha de ser necesariamente de acceso múltiple, ya que puede estar dedicado únicamente a procesar trabajos de cola.

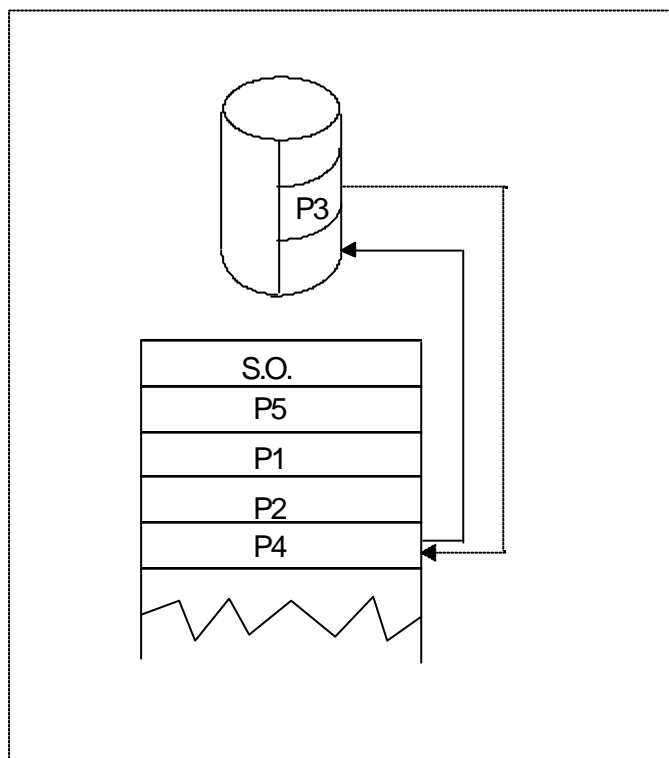
Intercambiabilidad memoria principal/disco.

El número de trabajos ejecutándose concurrentemente depende (entre otros factores) de la capacidad de la memoria principal. En un computador de 16 MB de memoria principal, si el sistema operativo residente en memoria ocupa 2 MB, y cada usuario de terminal interactivo (pantalla/teclado) ocupa 512 KB, como máximo podrán estar trabajando n usuarios, con n dado por:

$$16 = 2 + 0,5 \cdot n$$

es decir, $n = 28$ usuarios. No podrán utilizar el computador concurrentemente más de 28 usuarios.

Para evitar esta limitación, existe un procedimiento consistente en trasvasar a disco (hacia afuera) un proceso, de los que están residiendo en memoria, para dar cabida a otro, que se trasvasa de disco a memoria (hacia adentro). El proceso descargado de memoria será vuelto a cargar en memoria para cuando le llegue su turno.



Este procedimiento, denominado **intercambiabilidad (o trasvase) memoria principal/disco** (swapping) es gestionado por un módulo del sistema operativo, denominado **intercambiador**, y permite al computador realizar concurrentemente un número de procesos que no está limitado por la capacidad de la memoria principal. La eficiencia o velocidad de esta técnica depende de la velocidad de transferencia memoria/disco y de la velocidad de funcionamiento de la unidad de disco (siendo recomendable disponer de DMA y/o procesadores de E/S).

Con la técnica de intercambiabilidad el número de procesos en ejecución no viene impuesto por la capacidad de memoria; ahora bien, si la relación capacidad de memoria/número de procesos es baja, el funcionamiento del sistema puede ser muy lento, ya que de turno a turno a un proceso dado puede transcurrir un tiempo relativamente grande. Este problema puede reducirse fácilmente con una ampliación de memoria principal adecuadamente dimensionada.

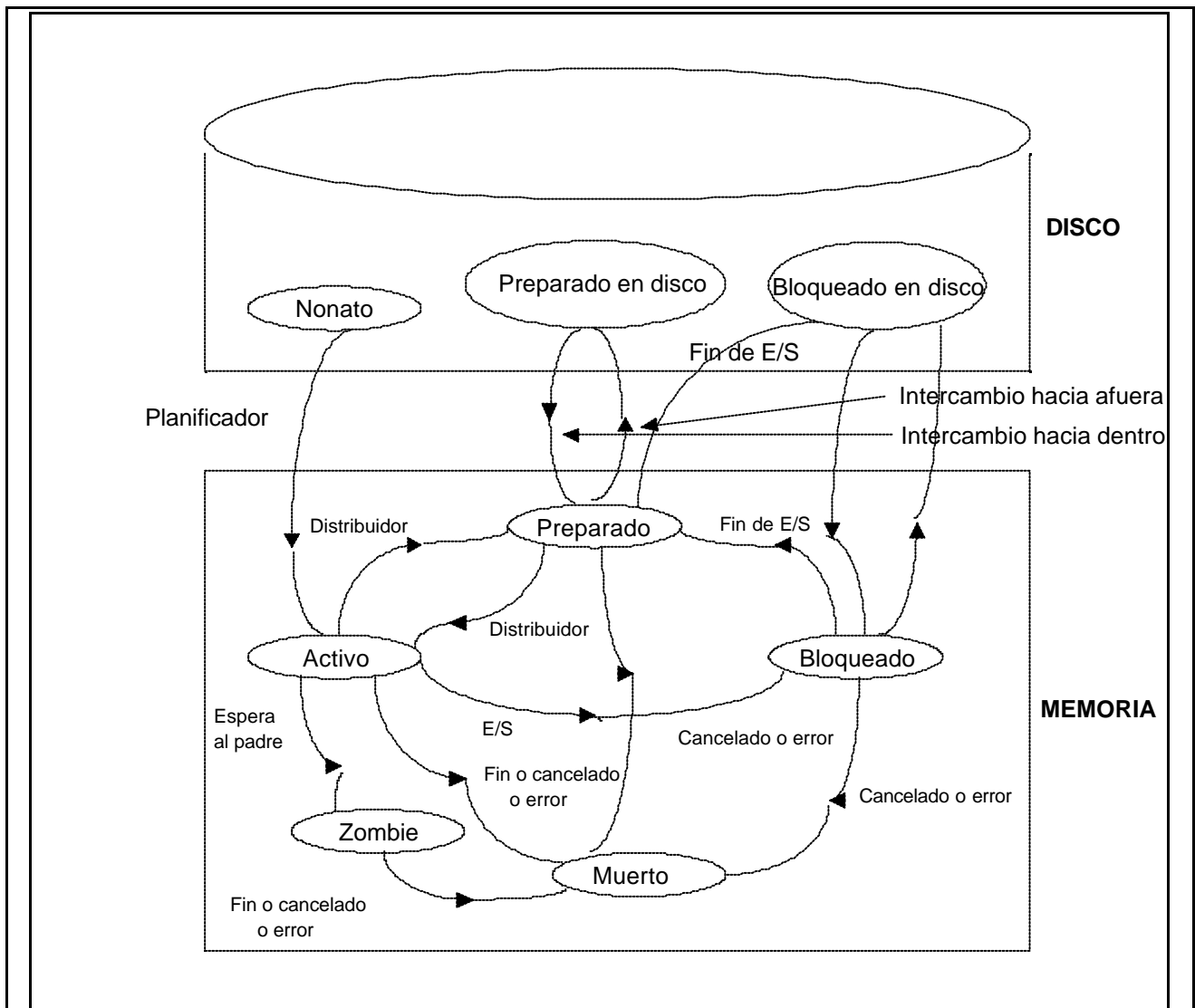
Estados de un proceso.

Un proceso desde el punto de vista de su ejecución, puede estar en diversas situaciones o estados.

Un proceso **nonato** es aquel que no ha iniciado su ejecución. Un ejemplo es un programa retenido en cola esperando a que el planificador de trabajos y el distribuidor le den paso a ejecución.

Un proceso está **preparado, listo o ejecutable** cuando se encuentra en memoria principal, sin operaciones de E/S pendientes, y apto para entrar (o continuar) en ejecución, en el instante en que el distribuidor le asigne la CPU.

El estado de **ejecución o activo** corresponde al proceso que en ese momento está siendo atendido por la CPU. Un proceso activo sale del estado de ejecución, en un sistema de tiempo compartido, cuando pasa a estado de **bloqueado** o el distribuidor le interrumpe para atender a otro de mayor prioridad (o que le corresponde el turno).



Del estado de bloqueado puede pasar al estado preparado cuando acaba la operación E/S pendiente o se le asigna el recurso esperado. También, en un sistema en que se gestiona la memoria con intercambiabilidad y con más procesos en ejecución concurrente de los que admite la memoria principal, un proceso puede ser trasvasado a disco (**estado bloqueado intercambiado o bloqueado en disco**). Un ejemplo de proceso bloqueado es aquel que solicita al operador del computador montar una determinada cinta magnética; en este caso, el sistema lo mantiene bloqueado hasta que acaba de montarse la cinta y, si no hay memoria suficiente para los otros trabajos en progreso, lo trasvasa a disco. Si la carga de procesos es grande, incluso los procesos preparados pueden trasvasarse a disco, pasando así al estado de **preparado intercambiado o preparado en disco**.

Cuando finaliza la ejecución de un proceso o se detecta un error grave en él, el proceso pasa a **estado de muerto**, liberando el sistema operativo la zona de memoria que ocupaba.

Un proceso (**P**) puede generar un proceso hijo (**H**). Puede ocurrir que el objetivo del proceso hijo sea generar cierta información para el padre y que, por problemas de sincronización entre ambos procesos, finalice la ejecución de **H** sin que el padre haya captado la información correspondiente. En este caso se dice que el hijo está en **estado zombie**: permanece en memoria, y no debe estar en estado de preparado, ni pasar a listo, ni a bloqueado (ya que finalizó su ejecución). Del estado zombie debe pasar al estado de muerto.

Los procesos más importantes, mientras progresa su ejecución, pueden ser obligados a residir en memoria principal. Estos procesos se dice que son **residentes** o no intercambiables. Tal es el caso del monitor y otros módulos importantes del sistema operativo. Hay computadoras que tienen su residente (o parte de él) grabado en memoria sólo lectura (ROM).

