



Tema 1_7

PLANIFICACIÓN DEL PROCESADOR

Juan Pozo | SI | 23_24

En la planificación del procesador se decide cuánto tiempo de ejecución se le asigna a cada proceso del sistema y en qué momento y cada proceso va a estar en un estado según sus propiedades en el SO.

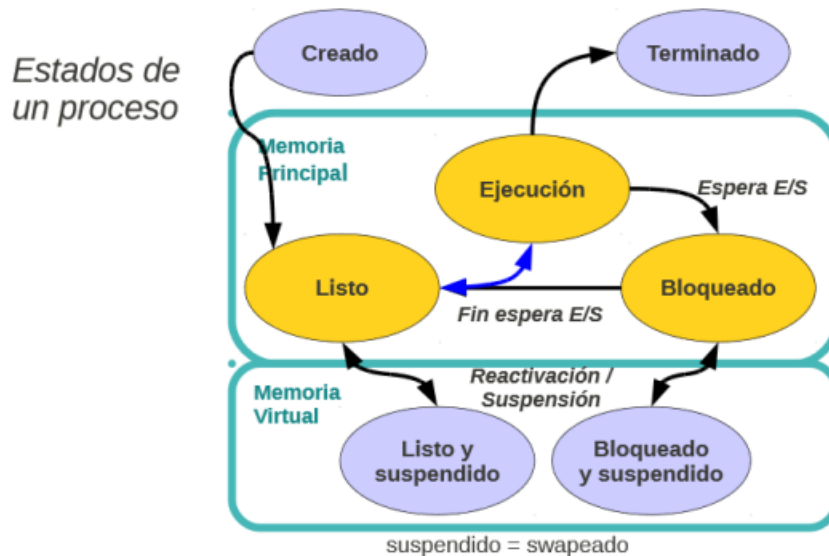


TABLA DE CONTROL DEL PROCESO

El sistema operativo almacena en una tabla denominada tabla de control de procesos con la información relativa a cada proceso que se está ejecutando en el procesador. Este es:

Nombre	Descripción
Imagen	Nombre del fichero ejecutable que dio lugar al proceso
PID	Numero de Identificación del proceso.
PPID	Numero de Identificación del proceso padre.
Nombre de usuario	Información sobre el usuario y grupo que lo han lanzado.
Estado	Estado del proceso en el procesador.
Datos de estado	El contenido de los registros internos, contador de programa, etc. Es decir, el entorno volátil del proceso.
Tiempo de CPU	Información de control de proceso.
Uso de memoria	Segmentos de memoria asignados.
Recursos asignados	Descriptores de ficheros, dispositivos o sockets de red asignados al proceso.
Nombre de sesión y Numero de sesión	Datos sobre la sesión en la que se está ejecutando
Título de ventana	Nombre del interfaz Grafico del proceso en el escritorio

ESTRATEGIA DE PLANIFICACIÓN

Una estrategia de planificación debe buscar que los procesos obtengan sus turnos de ejecución de forma apropiada (momento en que se le asigna el uso de la CPU), junto con un buen rendimiento y minimización de la sobrecarga (overhead) del planificador mismo. En general, se buscan cinco objetivos principales:

- Todos los procesos en algún momento obtienen su turno de ejecución o intervalos de tiempo de ejecución hasta su terminación con éxito.
- El sistema debe finalizar el mayor número de procesos por unidad tiempo.
- El usuario no percibirá tiempos de espera demasiado largos.
- Evitar el aplazamiento indefinido, los procesos deben terminar en un plazo finito de tiempo. Esto es, el usuario no debe percibir que su programa se ha parado o “colgado”.

Tipos de Procesos

La carga de trabajo de un sistema informático a otro puede variar considerablemente, esto depende de las características de los procesos. Nos podemos encontrar:

Características	Descripción
Uso de CPU	Procesos que hacen un uso intensivo de la CPU. Por ejemplo calculos matemáticos
E/S	Procesos que realizan una gran cantidad de operaciones de Entrada/Salida .Esto es acceso a disco duro o memoria
Tipos de Procesos	<ul style="list-style-type: none"> • Procesos por lotes (el usuario no hace nada) • Procesos interactivos (el usuario interactua mucho) • Procesos en tiempo real(Tienen que ejecutarse inmediatamente).
Duración	Procesos de menor o mayor duración.

En función de cómo sean la mayoría de los procesos habrá algoritmos de planificación que den un mejor o peor rendimiento al sistema.

PLANIFICACIÓN APROPIATIVA Y NO APROPIATIVA

La planificación no apropiativa (en inglés, no preemptive) es aquella en la que, cuando a un proceso le toca su turno de ejecución, ya no puede ser suspendido; es decir, no se le puede arrebatar el uso de la CPU, hasta que el proceso no lo determina no se podrá ejecutar otro proceso. Este esquema tiene sus problemas, puesto que, si el proceso contiene ciclos infinitos, el resto de los procesos pueden quedar aplazados indefinidamente. Otro caso puede ser el de los procesos largos que penalizarían a los cortos si entran en primer lugar.

La planificación apropiativa (en inglés, preemptive) supone que el sistema operativo puede arrebatarse el uso de la CPU a un proceso que esté ejecutándose. En la planificación apropiativa existe un reloj que lanza interrupciones periódicas en las cuales el planificador toma el control y se decide si el mismo proceso seguirá ejecutándose o se le da su turno a otro proceso. En ambos enfoques de planificación se pueden establecer distintos algoritmos de planificación de ejecución de procesos.

ALGORITMOS

Algunos de los algoritmos para decidir el orden de ejecución de los procesos en el sistema son:

FIFO

En el algoritmo **"Primero en llegar, primero en ejecutarse"**: Su tiempo de respuesta puede ser alto, especialmente si varían mucho los tiempos de ejecución. La sobrecarga del sistema es mínima. Penaliza los procesos cortos y los procesos con operaciones de Entrada/Salida.

Algoritmo basado en prioridades

- A cada proceso en el sistema se le asigna un entero basándose en algún criterio
- Generalmente, un valor numérico bajo indica alta prioridad
- Se favorece a los procesos de alta prioridad al momento de asignar la CPU
- No apunta a reducir el tiempo de espera promedio general. Los procesos importantes serán atendidos sin caer en retardos innecesarios. Las prioridades pueden ser estáticas o dinámicas **Produce Muerte por inanición.**

Ejemplos de algoritmos basados en prioridades son:

Shortest Job First (SJF)

Es non-preemptive, **se ejecutan los procesos en orden ascendente según el tiempo de ejecución de cada uno**. Se asigna la CPU al proceso más corto, en caso de "empate" se usa FCFS.

Es óptimo con respecto al tiempo de espera promedio mínimo para un conjunto de procesos. Adecuado para procesos batch donde los tiempos de ejecución se conocen de antemano. Ofrece un límite teórico (para el tiempo de espera promedio) frente al cual pueden compararse otros algoritmos.

¿Quieres saber cómo se comunican y sincronizan entre sí los procesos? Visita el siguiente enlace:

<http://www.redes-linux.com/apuntes/so1/teoria/tema4.pdf>

Round Robin

Algoritmo Preemptive, apto para procesos interactivos, simples, justos y ampliamente usados:

- A cada proceso se le asigna un intervalo de tiempo llamado quantum. Si un proceso continúa su ejecución superando su quantum, es preempted y la CPU es dada a otro proceso.
- Si un proceso se bloquea o finaliza antes de que expire el quantum, la CPU se reasigna en ese momento.
- Cuando un proceso usa todo su quantum se lo coloca al final de la lista de procesos listos – suposición: todos los procesos tienen la misma importancia.