

ACTIVIDADES 1, 7, 13

1. Cuestiones sobre procesos, y asignación de CPU:

- *¿Es necesario que lo último que haga todo proceso antes de finalizar sea una llamada al sistema para finalizar? ¿Sigue siendo esto cierto en sistemas monoprogramados?*

Si es necesario, puesto que al finalizar un proceso, este debe informar al SO de que su ejecución ha acabado con el fin de que libere el espacio reservado y el PCB de dicho proceso para posteriormente llamar a otro proceso en estado preparado.

Y en sistemas monoprogramados también es necesario puesto que igualmente (y con más razón) el SO necesita conocer cuando termina el proceso para liberar su espacio de memoria e introducir un nuevo proceso en espera.

- *Cuando un proceso se bloquea, ¿deberá encargarse él directamente de cambiar el valor de su estado en el descriptor de proceso o PCB?*

Cuando un proceso se bloquea, el sistema operativo es el encargado de cambiar su estado en el PCB. Al ser datos del sistema operativo, se debe hacer desde el modo kernel, por eso el proceso no

- *¿Qué debería hacer el planificador a corto plazo cuando es invocado pero no hay ningún proceso en la cola de ejecutables?*

Este problema es resuelto en muchos sistemas operativos con el proceso NULO que es creado por el sistema en el momento de arranque. El proceso nulo nunca termina, no tiene E/S y tiene la prioridad más baja en el sistema. En consecuencia la cola de listos nunca está vacía, además la ejecución del planificador puede hacerse más rápida al eliminar la necesidad de comprobar si la cola de listos está vacía o no. Algunas de las tareas que se le pueden dar al proceso nulo, por ejemplo, es realizar estadísticas de uso de procesador, o asistencia de vigilancia de la integridad del sistema, etc.

- *¿Qué algoritmos de planificación quedan descartados para ser utilizados en sistemas de tiempo compartido?*

Tenemos que usar aquellos algoritmos que favorezcan procesos cortos (ya que estamos en tiempo compartido) generalmente los que no están basados en quantum de tiempo. Descartamos FCFS Y el más corto primero no apropiativo.

7. ¿Puede el procesador manejar una interrupción mientras está ejecutando un proceso si la política de planificación que utilizamos es no apropiativa?

Una interrupción debería de ser procesada independientemente del algoritmo de planificación que se esté utilizando. Si solo tenemos un procesador no puede estar haciendo las dos cosas a la vez. Si llega una interrupción, se le quitaría la CPU al proceso y se le da a dicha interrupción y cuando se despache la interrupción, y volvemos al proceso.

13. Utilizando la tabla del ejercicio anterior,

Proceso	Tiempo de creación	Tiempo de CPU
A	0	3
B	1	1
C	3	12
D	9	5
E	12	5

dibuja el diagrama de ocupación de CPU para el caso de un sistema que utiliza un algoritmo de colas múltiples con realimentación con las siguientes colas:

Cola	Prioridad	Quantum
1	1	1
2	2	2
3	3	4

y suponiendo que:

- *Todos los procesos inicialmente entran en la cola de mayor prioridad (menor valor numérico). Cada cola se gestiona mediante la política Por Turnos.*
- *la política de planificación entre colas es por prioridades no apropiativo.*

- *un proceso en la cola i pasa a la cola $i+1$ si consume un quantum completo sin bloquearse.*
- *cuando un proceso llega a la cola de menor prioridad, permanece en ella hasta que finalice.*