

## Relación de problemas - Sistemas Operativos

### Temas 1 y 2: Estructuras de sistemas operativos, y Procesos y hebras

1. Cuestiones sobre procesos, y asignación de CPU:
  - a) ¿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?
  - b) Cuando un proceso se bloquea, ¿deberá encargarse él directamente de cambiar el valor de su estado en el descriptor de proceso o PCB?
  - c) ¿Qué debería hacer el planificador a corto plazo cuando es invocado pero no hay ningún proceso en la cola de ejecutables?
  - d) ¿Qué algoritmos de planificación quedan descartados para ser utilizados en sistemas de tiempo compartido?
2. La representación gráfica del cociente  $[(tiempo\_en\_cola\_ejecutables + tiempo\_de\_CPU) / tiempo\_de\_CPU]$  frente a  $tiempo\_de\_CPU$  suele mostrar valores muy altos para ráfagas muy cortas en casi todos los algoritmos de asignación de CPU. ¿Por qué?
3. Para cada una de las llamadas al sistema siguientes, especificar y explicar si su procesamiento por el sistema operativo requiere la invocación del planificador a corto plazo:
  - a) Crear un proceso.
  - b) Abortar un proceso, es decir, terminarlo forzosamente.
  - c) Suspender o bloquear un proceso.
  - d) Reanudar un proceso (inversa al caso anterior).
  - e) Modificar la prioridad de un proceso.
4. Sea un sistema multiprogramado que utiliza el algoritmo Por Turnos (*Round-Robin*). Sea **S** el tiempo que tarda el despachador en cada cambio de contexto. ¿Cuál debe ser el valor de quantum **Q** para que el porcentaje de uso de la CPU por los procesos de usuario sea del 80%?
5. Sea un sistema multiprogramado que utiliza el algoritmo Por Turnos (*Round-Robin*). Sea **S** el tiempo que tarda el despachador en cada cambio de contexto, y **N** el número de procesos existente. ¿Cuál debe ser el valor de quantum **Q** para que se asegure que cada proceso “ve” la CPU al menos cada **T** segundos?
6. ¿Tiene sentido mantener ordenada por prioridades la cola de procesos bloqueados? Si lo tuviera, ¿en qué casos sería útil hacerlo?
7. ¿Puede el procesador manejar una interrupción mientras está ejecutando un proceso si la política de planificación que utilizamos es no apropiativa?
8. Suponga que es responsable de diseñar e implementar un sistema operativo que va a utilizar una política de planificación apropiativa. Suponiendo que tenemos desarrollado el algoritmo de planificación a tal efecto, ¿qué otras partes del sistema operativo habría que modificar para implementar tal sistema? y ¿cuáles serían tales modificaciones?
9. En el algoritmo de planificación FCFS, la **penalización**  $((t + t^o \text{ de espera}) / t)$ , ¿es creciente, decreciente o constante respecto a  $t$  (tiempo de servicio de CPU requerido por un proceso)? Justifique su respuesta.

10. En la tabla siguiente se describen cinco procesos:

Proceso	Tiempo de creación	Tiempo de CPU
A	4	1
B	0	5
C	1	4
D	8	3
E	12	2

Si suponemos que tenemos un algoritmo de planificación que utiliza una política FIFO (primero en llegar, primero en ser servido), calcula:

- Tiempo medio de respuesta
- Tiempo medio de espera
- La penalización, es decir, el cociente entre el tiempo de respuesta y el tiempo de CPU.

11. Utilizando los valores de la tabla del problema anterior, calcula los tiempos medios de espera y respuesta para los siguientes algoritmos:

- Por Turnos con quantum  $q=1$
- Por Turnos con quantum  $q=4$
- El más corto primero (SJF). Suponga que se estima una ráfaga igual a la real.

12. Calcula el tiempo de espera medio para los procesos de la tabla utilizando el algoritmo: el primero más corto apropiativo (o primero el de tiempo restante menor, SRTF).

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

13. Utilizando la tabla del ejercicio anterior, 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.

14. Suponga que debe maximizar la eficiencia de un sistema multiusuario y que está recibiendo quejas de muchos usuarios sobre los pobres tiempos de respuesta (o tiempos de vuelta) de sus procesos. Los resultados obtenidos con una herramienta de monitorización del sistema nos muestran que la CPU se utiliza al 99'9% de su tiempo y que los procesadores de E/S están activos solo un 10% de su tiempo. ¿Cuales pueden ser las razones de estos tiempos de respuesta pobres y por qué?
- a) El quantum en la planificación Round-Robin es muy pequeño.
  - b) La memoria principal es insuficiente.
  - c) El sistema operativo tiene que manejar mucha memoria principal por lo que las rutinas de gestión de memoria están consumiendo todos los ciclos de CPU.
  - d) La CPU es muy lenta.
  - e) El quantum en la planificación Round-Robin es muy grande.
15. Compare el rendimiento ofrecido al planificar el conjunto de tareas multi-hebras descrito en la tabla y bajo las siguientes configuraciones:
- a) Sistema operativo multiprogramado con hebras de usuario. En este sistema se dispone de una biblioteca para la programación con hebras en el espacio de usuario. El algoritmo de planificación de CPU utilizado por el SO es *Round-Robin* con un *quantum* de 50 u.t. (unidades de tiempo). El planificador de la biblioteca de hebras reparte el *quantum* del proceso (tarea) entre las hebras utilizando *Round-Robin* con un *quantum* para cada hebra de 10 u.t. Suponga que no existe coste en el cambio de contexto entre hebras ni entre procesos.
  - b) Sistema operativo multiprogramado con hebras kernel. El SO planifica las hebras usando *Round-Robin* con un quantum de 10 u.t. Como en el apartado anterior, suponga que no existe coste en la operación de cambio de contexto. Considere además que las operaciones de E/S de un proceso únicamente bloquean a la hebra que las solicita.

Suponga en ambos casos que los dos procesos están disponibles y que el planificador entrega la CPU al proceso P1. Para realizar la comparación represente en cada caso el diagrama de ocupación de CPU y calcule el grado de ocupación de la CPU (tiempo CPU ocupada \ tiempo total).

Proceso	Hebras	Comportamiento		
		Ráfaga de CPU	Tiempo de E/S	Ráfaga de CPU
P1	Hebra 1	20	30	10
	Hebra 2	30	-	-
	Hebra 3	10	-	-
P2	Hebra 1	30	30	10
	Hebra 2	40	-	-

16. ¿El planificador CFS de Linux favorece a los procesos limitados por E/S (cortos) frente a los procesos limitados por CPU (largos)? Explique cómo lo hace.
17. ¿Cuál es el problema que se plantea en Linux cuando un proceso no realiza la llamada al sistema `wait` para cada uno de sus procesos hijos que han terminado su ejecución? ¿Qué efecto puede producir esto en el sistema?
18. La orden `clone` sirve tanto para crear un proceso en Linux como una hebra. a) Escriba los argumentos que debería tener `clone` para crear un proceso y una hebra. b) Dibuje las principales estructuras de datos del kernel que reflejan las diferencias entre ambas.