

Ingeniería Informática. Curso 3º.
Sistemas Operativos
Examen Final. PROBLEMAS.
2 de Febrero de 2010

1. Considérese un sistema multitarea en el que se pretende que la planificación del procesador cumpla con los siguientes principios:

- **Los procesos que se ejecutan en modo “sistema”** tienen prioridad absoluta sobre los que se ejecutan en modo usuario y deben ejecutarse lo antes posible. Estos procesos se gestionan priorizando a aquél que se estime que consumirá menos tiempo de procesador. Además, estos procesos se dividen en dos clases: los que se deben ejecutar sin que deban ser interrumpidos, que identificamos como tipo A, y los que pueden ser interrumpidos durante su ejecución, que identificamos como de tipo B. Los procesos que no deben ser interrumpidos tienen prioridad absoluta sobre el resto de procesos del sistema y deben ejecutarse lo antes posible.
- **Los procesos que se ejecutan en modo “usuario”** no deben acaparar el procesador más de 3 unidades de tiempo consecutivas, y se planifican penalizando a aquellos que hayan consumido más tiempo de procesador. Los procesos que se ejecutan en modo “usuario” los identificamos como de tipo C.

Para la secuencia de llegadas que se da a continuación se pide:

- a) [2 puntos] Obtener la traza de ejecución temporal de los procesos (diagrama de Gantt), indicando las razones que motivan las conmutaciones de proceso que se producen.
- b) [1 punto] Indicar los tiempos medio de retorno y espera.

Proceso	Tipo	Tiempo de llegada	Duración ráfaga de procesador
0	C	0	7
1	B	1	6
2	A	2	5
3	A	3	2
4	C	4	4

2. Partimos de dos programas con los siguientes códigos:

P1	P2
<pre>... fd1 = open("datos", O_RDONLY); read(fd1, buf, 100); pause();</pre>	<pre>... fd1 = open("datos", O_RDWR); if (fork() == 0) { fd2 = open("textos", O_WRONLY O_APPEND); } write(fd1, "abc", 3); pause();</pre>

Se lanzan a ejecutar ambos programas P1 y P2 y, eventualmente, quedan bloqueados en `pause ()`. Para ese instante,

- (a) [2 puntos] indicar la información relevante al estado de las tablas de descriptores de ficheros, intermedia y de i-nodos de los procesos existentes, y
- (b) [1 punto] mostrar qué contienen los ficheros “datos” y “textos” teniendo en cuenta que, inicialmente, el fichero “datos” contiene “AA...A” - 12 A's, y el fichero “textos” contiene “BB...B” - 50 B's.

3. Resolver el siguiente problema de concurrencia que podemos formular brevemente como “un productor de M-en-M y varios consumidores de 1-en-1”.

La historia podría contarse de este modo. Estamos en una pizzería de servicio a domicilio atendida por un cocinero (productor) que normalmente está dentro ocupado cocinando, y que cuando le avisen colocará M pizzas en una mesa para que se sirvan los repartidores. Los repartidores (consumidores) llegan a la pizzería con un pedido pendiente cada uno, y si quedan pizzas en la mesa, tomarán una y se marcharán a entregarla; pero si la mesa está vacía el primero que llegue avisará al cocinero para que reponga M pizzas más y esperará a poder llevarse una.

Las pizzas se modelarán como caracteres; la mesa de intercambio entre cocinero y repartidores se modelará mediante una tubería (pipe), y la sincronización (avisos y esperas) y exclusión mutua se resolverá con semáforos.

(Opcionalmente se puede emplear una llamada **trywait(s)** que actúa sin bloqueos, devolviendo -1 en caso de que el semáforo-argumento valga 0, es decir, sea causa de bloqueo, y devolviendo 0 en caso de que el semáforo tenga un valor mayor que cero, es decir, no cause bloqueo)

- a) (1 punto) Codificar las creaciones, aperturas e inicializaciones de los recursos de CyS en los procesos Cocinero y Repartidor.
- b) (1,5 puntos) Codificar el comportamiento de los Repartidores:
En un bucle sin fin:
 - Si aun hay pizzas en la mesa (caracteres en el pipe), tomar una y seguir
 - Si no quedan pizzas en la mesa (pipe vacío), avisar al Cocinero y esperar a que ponga M más.
- c) (1,5 puntos) Codificar el comportamiento del Cocinero
En un bucle sin fin
 - Esperar hasta que un Repartidor avise que ya no quedan pizzas en la mesa
 - Depositar M pizzas en la mesa (enviar M nuevos caracteres al pipe)