

## SISTEMAS OPERATIVOS

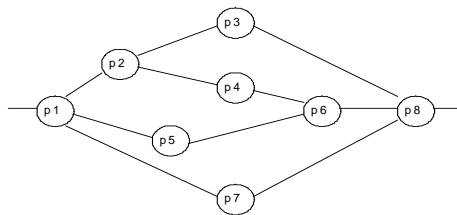
### PROBLEMAS DE GESTION DE PROCESOS

1. Dado el siguiente programa, ejecutado bajo UNIX,

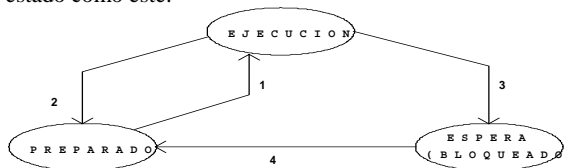
```
void main(int argc, char *argv[]) {
    int i;

    for (i=1; i<=argc; i++)
        fork();
    ...
}
```

- a) Dibuje el esquema jerárquico de procesos que se genera para  $argc = 3$
  - b) ¿Cuántos procesos se crean para  $argc = n$ ?
2. a) Use las llamadas `fork()`, `exec()`, `exit()` y `wait()` de UNIX para describir la sincronización de los ocho procesos cuyo grafo general de precedencia es el siguiente:



- b) Alternativamente, use señalización para forzar las precedencias anteriores, en una situación donde los ocho procesos coexisten simultáneamente.
3. Podemos describir gran parte de la gestión del procesador en términos de diagramas de transiciones de estado como éste:



- a. ¿Qué "evento" causa cada una de las transiciones marcadas?
  - b. Si consideramos todos los procesos del sistema, podemos observar que una transición de estado por parte de un proceso podría hacer que otro proceso efectuara una transición también. ¿Bajo qué circunstancias podría la transición 3 de un proceso provocar la transición 1 inmediata de otro proceso?
  - c. ¿Bajo qué circunstancias, si las hay, podrían ocurrir las siguientes transiciones causa-efecto?  
 $2 \rightarrow 1$ ;  $3 \rightarrow 2$ ;  $4 \rightarrow 1$
  - d. ¿Bajo qué circunstancias, si las hay, las transiciones 1, 2, 3 y 4 NO producirían ninguna otra transición inmediata?
4. Suponga que los siguientes trabajos llegan a procesarse en los instantes indicados. ¿Cuáles son los tiempos de retorno (*turnaround*) y de espera para cada uno de ellos, los tiempos de retorno y de espera promedios, así como la productividad (*throughput*) del sistema, aplicando las diferentes estrategias de planificación listadas? Aplique los algoritmos de planificación sin expropiación y base las decisiones en la información de que se dispone en el momento de tomarlas. El desglose de los tiempos de los trabajos se refiere a ráfagas de UCP y de E/S alternativamente.

**Comentario [ABP1]:** (MD4.10)

Trabajo	Llegada	Tiempo	(Desglose)		
1	0.0	8	[3	4	1]

2	0.5	4	[1	1	1	1]
3	1.0	2	[0.5	0.5	1]	
4	3.0	6	[0.5	5	0,5]	

- a) FCFS (Primero en llegar, primero en pasar)
- b) SPN (Primero el de menor tiempo de UCP siguiente)
- c) RR (Prioridad circular con cuanto = 1)
- d) SPN, pero dejando la CPU inactiva durante la primera unidad de tiempo

5. Cinco trabajos de lotes, de A a E, llegan a un centro de cálculo casi al mismo tiempo. La estimación de sus respectivos tiempos de ejecución es de 10, 6, 2, 4, y 8 minutos. Sus prioridades, determinadas externamente, son 3, 5, 2, 1 y 4, respectivamente, siendo 5 la prioridad superior. Para cada uno de los siguientes algoritmos de planificación determine el tiempo medio de retorno de los procesos. (Ignorar el tiempo de conmutación de procesos).

- (a) Por turnos (*round-robin*) con  $q \approx 0$ , es decir, empleando la política de compartición del procesador
- (b) Por prioridad estricta
- (c) Por orden de llegada (*FCFS*)
- (d) Por menor tiempo de ejecución (*SPN*)

6. Considere un sistema con una política de planificación de procesos de 3 niveles con realimentación. Cada nivel usa a su vez una política de planificación circular (*round robin*) cuyos cuantos de tiempo son 2, 4 y 8, respectivamente. Al principio hay 3 procesos en la cola del nivel 1 (máxima prioridad). Los patrones de ejecución de los procesos son los siguientes:

P<sub>1</sub> (3-CPU,5-E/S)      P<sub>2</sub> (8-CPU,5-E/S)      P<sub>3</sub> (5-CPU,5-E/S)

Las colas de los otros dos niveles están vacías. Cuando acaba una operación de E/S, los procesos entran en la cola de mayor prioridad.

Usando un diagrama de tiempos muestre:

- a) qué proceso está ejecutándose y
  - b) qué procesos hay en cada nivel,
- durante las 30 primeras unidades de tiempo de ejecución.

Calcule además la utilización de CPU y los tiempos de espera de cada proceso.

Comentario [ABP2]:

7. Considere un sistema con una planificación MLF (multinivel con realimentación), donde  $n = 10$ ,  $T_i = 2 * q$  ( $q$  es el valor del intervalo básico de planificación o cuanto) y, en cada nivel de prioridad,  $T_i$  es dos veces mayor que el del nivel inmediatamente superior. Hay tres procesos en la cola de máxima prioridad: sus tiempos respectivos hasta la siguiente petición de E/S son 3, 8, y 5 unidades de tiempo (1 unidad de tiempo =  $q$ ). Todas las otras colas están vacías. Cuando un proceso alcanza su petición de E/S, permanece suspendido (sin competir por la CPU) durante 5 unidades de tiempo, tras las cuales vuelve a entrar en la cola de máxima prioridad. Los tiempos de CPU requeridos hasta la petición de E/S siguiente son de nuevo 3, 8, y 5. Usando un diagrama de tiempos, muestre

- (a) qué proceso estará ejecutándose,
- (b) qué procesos estarán en qué cola durante cada una de las primeras 30 unidades de tiempo de ejecución, y
- (c) cuáles son los valores de las siguientes medidas de rendimiento: productividad, tiempos retorno (*turnaround*) individual y promedio, y tiempos de espera individual y promedio.

Comentario [ABP3]:

8. Repita el ejercicio anterior pero suponiendo que  $T_i = 2 * q$  (es decir, constante), para todos los niveles de prioridad.

9. Sea un sistema operativo que sigue el modelo cliente-servidor en el que existe un proceso servidor SF (Sistema de Ficheros) y un proceso CD (Controlador de Disco). La prioridad del proceso SF es mayor que la de los procesos de usuario y la prioridad del proceso CD es superior a la de SF. En un determinado instante la cola de PREPARADOS de este sistema contiene dos procesos de usuario A y B, en dicho orden. Las características de ejecución de A y B son las siguientes:

Proceso A: 160 ms. CPU; 50 ms. E/S de disco; 50 ms. CPU

Proceso B: 20 ms. CPU; 50 ms. E/S de disco; 60 ms. CPU

Se pide construir un diagrama de tiempos donde se muestre, a partir del instante en que aparecen los procesos A y B en el sistema, los estados de estos dos procesos (EJECUCIÓN; PREPARADO;

BLOQUEADO). Tener en cuenta que la E/S de disco requiere usar el proceso SF y el proceso CD, aunque se supone que el tiempo que tardan es despreciable. El cuanto de planificación aplicado a los procesos de usuario es de 100 ms.

10. Dados tres procesos de tiempo real con comportamiento periódico cuyos valores de Período y Uso de CPU son respectivamente [100;20], [145;30]; y [150; X], averiguar para qué valores de X la planificación RMS no cumple la condición suficiente de ejecución sin pérdidas y, sin embargo, resulta ser correcta la ejecución efectiva obtenida.
11. Determinar el rendimiento (%uso de CPU y %sobrecarga del S.O.) obtenido al planificar dos procesos multi-hilo en un sistema donde los hilos se gestionan en espacio de usuario. El algoritmo de planificación del S.O. es RR con  $q=100$ , y el cambio de contexto entre procesos tarda 20. El planificador de la biblioteca de hilos reparte el  $q$  del proceso entre los hilos aplicando RR con  $q=10$  y sin coste de cambio de contexto entre sus hilos. Los dos procesos tienen el siguiente comportamiento:

<u>Proceso A</u>	<u>Proceso B</u>
Hilo 1 (30-CPU; 110-E/S; 40-CPU)	Hilo 1 (20-CPU; 50-E/S; 60-CPU)
Hilo 2 (50-CPU)	Hilo 2 (40-CPU; 110-E/S; 20-CPU)
Hilo 3 (30-CPU)	
Hilo 4 (20-CPU; 60-E/S; 40-CPU)	

- b) ¿Sería diferente la planificación final si los hilos fuesen gestionados por el kernel?