

Apellidos:	Nombre:
Grupo: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> PCEO	DNI:

Instrucciones:

- Todas las respuestas deben escribirse con **bolígrafo**.
- Se valorará la **exactitud, completitud y brevedad** de todas las respuestas, que deberán ser **RAZONADAS**. No tendrán validez las respuestas no justificadas.
- Esta parte vale 3 puntos y, por tanto, representa el 30 % de la nota del examen final de teoría.

1. (1,2 puntos) Tenemos un sistema donde la planificación de procesos se hace utilizando múltiples colas sin realimentación. En concreto, existen dos colas. La cola 0 es para los procesos interactivos y, dentro de ella, la planificación se hace según un algoritmo circular (o *round robin*) con un *quantum* de 1 tic de tiempo. La cola 1 es para los procesos por lotes y, dentro de ella, la planificación se hace según un algoritmo SJF con estimaciones de tiempo para las ráfagas de CPU, donde la estimación inicial es 2 y el coeficiente de credibilidad α es 0,2. La planificación entre colas es no apropiativa por prioridad, siendo la cola 0 la de mayor prioridad.

En este sistema se tienen que ejecutar los siguientes procesos interactivos (I_1 e I_2) y por lotes (Lt_1 y Lt_2) (los tiempos vienen dados en tics):

Proceso	Llegada	CPU	E/S	CPU
I_1	3	3	4	3
I_2	3	1	4	3
Lt_1	0	5	1	2
Lt_2	1	1	3	2

Con estos datos, se pide:

- a) (0,6 puntos) Dibujar el diagrama de Gantt en el cronograma que sigue, donde el tiempo indica el instante en el que comienza el tic representado por la celda. Marca el estado de cada proceso utilizando las letras E (en ejecución), B (bloqueado) y L (cola de listos). En el caso de los procesos por lotes, añade a los estados E, B y L un subíndice (p. ej., E_1 , L_2) para indicar el tiempo de ejecución estimado en cada caso. Nota: en caso de empate entre procesos, supón que coge la CPU el proceso de menor número.

Tic	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
I_1																							
I_2																							
Lt_1																							
Lt_2																							

- b) (0,4 puntos) Explicar detalladamente las decisiones que se han tomado en los instantes 3, 5 y 9 del diagrama de Gantt anterior a la hora de decidir a qué proceso asignarle (o a qué proceso mantenerle) la CPU.
- c) (0,2 puntos) Calcular el tiempo medio de respuesta y de espera de los trabajos interactivos por un lado, y el tiempo medio de retorno y de espera de los trabajos por lotes por otro.

2. (1,8 puntos) Tenemos un disco duro con 16 cabezas, 8 192 cilindros y 512 sectores por pista de 512 bytes cada uno. Sobre este disco, que es mostrado por el sistema operativo al resto de componentes como un array de bloques físicos, se ha creado una partición de 16 GiB que empieza en el sector 2048 de dicho disco. En esa partición se ha creado un sistema de ficheros UNIX estándar con bloques lógicos de 2 KiB. Tenemos un bloque para el bloque de arranque, otro para el superbloque y a continuación una zona ocupada por el mapa de bits de

nodos-i, seguida de una zona para nodos-i (de 128 bytes cada uno), una zona para el mapa de bits de bloques de datos seguida de una última zona para bloques de datos que llega hasta el final de la partición. El nodo-i es el estándar, con 10 entradas directas de bloque, 1 BSI, 1 BDI y 1 BTI. Para guardar un número de bloque se usan 4 bytes. Con estos datos, se pide:

- a) (0,5 puntos) La posición exacta en disco (cilindro, cabeza y sectores) del bloque lógico 1 044 067 **de la partición**.
- b) (0,3 puntos) El número de bloques ocupados en disco por un fichero de 1 064 KiB, teniendo en cuenta tanto sus bloques de datos como de metadatos.
- c) (0,6 puntos) El tamaño, en bloques lógicos, de cada una de las zonas descritas del sistema de ficheros si se sabe que sería posible almacenar 15 650 ficheros de 1 064 KiB sin que sobre ningún bloque de datos. Calcula también cuantos ficheros de cualquier tipo se podrían llegar a crear en este sistema de ficheros.
- d) (0,4 puntos) Número de accesos a disco para escribir el byte 4 299 202 560 de un fichero de 5 GiB del que solo está en memoria su nodo-i en el momento de hacer la escritura. El byte debe quedar almacenado en disco, es decir, aunque podamos suponer la existencia de una caché de disco, la modificación no puede quedar en memoria, sino que debe reflejarse finalmente en disco.

Soluciones

Ejercicio 1:

a) El diagrama de Gantt que se obtiene es el siguiente:

Tic	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
I_1	–	–	–	L	L	E	L	E	E	B	B	B	B	E	L	E	L	E	–				
I_2	–	–	–	L	L	L	E	B	B	B	B	L	E	L	E	L	E	–					
Lt_1	E ₂	E ₂	E ₂	E ₂	E ₂	B _{4,4}	L _{4,4}	L _{4,4}	L _{4,4}	L _{4,4}	E _{4,4}	E _{4,4}	–										
Lt_2	–	L ₂	L ₂	L ₂	L ₂	L ₂	L ₂	L ₂	L ₂	E ₂	B _{1,2}	B _{1,2}	B _{1,2}	L _{1,2}	L _{1,2}	L _{1,2}	L _{1,2}	L _{1,2}	E _{1,2}	E _{1,2}	–		

b) Veamos lo que ocurre en cada uno de los instantes indicados:

- Instante 3: llega los dos procesos interactivos a la cola 0, la de mayor prioridad. Sin embargo, como la planificación entre colas es no apropiativa, continúa en la CPU el proceso por lotes Lt_1 .
- Instante 5: el proceso por lotes Lt_1 termina su primera ráfaga de CPU. Tenemos 3 procesos listos: los dos interactivos y el proceso por lotes Lt_2 . Como la cola 0 es la de mayor prioridad, toma la CPU un proceso interactivo, en concreto, el proceso I_1 que llegó a la misma vez a la cola de listos que el proceso I_2 , pero tiene un número menor.
- Instante 9: los procesos interactivos han terminado sus primeras ráfagas de CPU y ambos quedan bloqueados. La CPU queda libre. Los dos procesos por lotes están listos. Sin embargo, el proceso Lt_2 sigue teniendo la estimación inicial de ráfaga de CPU de 2 unidades, mientras que el proceso Lt_1 tiene una estimación de duración de su siguiente ráfaga de CPU de 4,4 u.t., ya que, tras su primera ráfaga de CPU, su estimación queda como:

$$E_2 = 0,2 \times 2 + (1 - 0,2) \times 5 = 4,4$$

Por tanto, siguiendo el algoritmo SJF, coge la CPU el proceso Lt_2 al tener una estimación menor.

c) El tiempo medio de espera de los procesos interactivos es $= \frac{5+6}{2} = 5,5$ tics y el de respuesta $= \frac{(18-3)+(17-3)}{2} = \frac{29}{2} = 14,5$ tics.

En el caso de los procesos por lotes, el tiempo medio de espera es $= \frac{4+13}{2} = 8,5$ tics y el de retorno $= \frac{(12-0)+(20-1)}{2} = \frac{31}{2} = 15,5$ tics.

Ejercicio 2:

A continuación se muestran las soluciones de los diferentes apartados.

a) Como los bloques físicos son de 512 bytes y los bloques lógicos son de 2048 bytes, cada bloque lógico ocupa 4 bloques físicos o sectores. Por lo tanto, el bloque lógico 1044067 se encuentra en el sector $1044067 \times 4 = 4176268$ y en los 3 siguientes sectores de la partición. Puesto que la partición comienza en el sector 2048, este primer sector se corresponde con el sector $4176268 + 2048 = 4178316$ del disco.

Veamos ahora la posición exacta en disco de ese sector 4178316. Como el disco tiene 16 cabezas y 512 sectores por pista, cada cilindro tiene un tamaño de $16 \times 512 = 8192$ sectores. Por lo tanto, el cilindro del sector es el:

$$\frac{4\,178\,316}{8\,192} = 510,05 \rightarrow 510.$$

El resto de la división anterior es 396, que nos da el número de sector dentro del cilindro. Al tener cada pista 512 sectores, ya podemos decir que el sector 396 se encuentra en esa misma posición de la pista 0 del cilindro 510. En resumen, el sector 4 178 316 se encuentra en el cilindro 510, cabeza 0 y sector 396 dentro de la pista de esa cabeza. El bloque lógico, por tanto, ocupará ese sector y los tres siguientes, es decir, ocupará los sectores 396, 397, 398 y 399 de la pista 0 del cilindro 510.

- b) Un fichero de 1 064 KiB ocupa $\frac{1\,064}{2} = 532$ bloques lógicos, ya que estos tienen un tamaño de 2 KiB cada uno. Las 532 direcciones de disco de esos bloques lógicos se almacenarán en el nodo-i (10 direcciones) y en bloques BSI (las 522 restantes). En cada bloque indirecto caben $\frac{2\,048}{4} = 512$ direcciones. Por ello, esas 522 direcciones se almacenarán en el BSI apuntado por el nodo-i (512 direcciones) y en el primer BSI apuntado por el BDI apuntado por el nodo-i (10 últimas direcciones). En total, el fichero ocupa 532 bloques lógicos para datos y 3 bloques lógicos para bloques indirectos (dos BSI y 1 BDI), es decir, 535 bloques en total.
- c) La partición tiene un tamaño de 16 GiB o, lo que es lo mismo, $\frac{16 \times 2^{20}}{2} = 8\,388\,608$ bloques lógicos. Como nos dicen que en la zona de datos sería posible almacenar 15 650 ficheros de 1 064 KiB cada uno, esta zona debe tener un tamaño de $15\,650 \times 535 = 8\,372\,750$ bloques lógicos, ya que, como hemos calculado en el apartado anterior, un fichero de 1 064 KiB ocupa en disco 535 bloques lógicos.

Sabiendo el tamaño de la zona de datos, podemos averiguar el tamaño del mapa de bits de bloques de datos:

$$\frac{8\,372\,750}{8 \times 2\,048} = 511,03 \rightarrow 512 \text{ bloques.}$$

Descontando al tamaño de la partición los bloques lógicos ocupados por el bloque de arranque, el superbloque, el mapa de bits de bloques de datos y la zona de datos, nos quedan los bloques lógicos ocupados por el mapa de bits de nodos-i y la tabla de nodos-i: $8\,388\,608 - (1 + 1 + 512 + 8\,372\,750) = 15\,344$ bloques.

Llamemos X al número de nodos-i (de 128 bytes) que puede haber en el sistema de ficheros. Como cada nodo-i necesita un bit en el mapa de bits de nodos-i, este mapa de bits ocupará, al menos, $\frac{X}{8 \times 2\,048}$ bloques y la tabla de nodos-i ocupará $\frac{X \times 128}{2\,048}$ bloques. La suma de estos tamaños debe ser igual a los 15 344 bloques que quedan por repartir, lo que nos da la siguiente ecuación:

$$\frac{X}{8 \times 2\,048} + \frac{X \times 128}{2\,048} = 15\,344$$

de donde despejando X obtenemos $X = 245\,264,48$. Por lo tanto, es posible tener 245 264 nodos-i, lo que nos da también el número total de ficheros de cualquier tipo que se podrían crear en este sistema de ficheros.

En cuanto al tamaño de las zonas, el mapa de bits de nodos-i ocupará:

$$\frac{245\,264}{8 \times 2\,048} = 14,97 \rightarrow 15 \text{ bloques}$$

y la tabla de nodos-i ocupará los $15\,344 - 15 = 15\,329$ bloques restantes.

- d) Según los datos dados en el enunciado (número de byte a escribir y tamaño del fichero), está claro que el byte a escribir se encuentra en algún bloque intermedio del fichero. Calculemos qué bloque es ese y cómo obtener la dirección de disco del mismo.

El byte 4 299 202 560 se encuentra en el bloque lógico:

$$\frac{4\,299\,202\,560}{2\,048} = 2\,099\,220$$

del fichero. Como la división nos ha salido exacta, se encontrará en el primer byte (es decir, la posición 0) de ese bloque.

La dirección de disco de ese bloque se almacenará en el BSI:

$$\frac{2\,099\,220 - 10}{512} = 4100,02 \rightarrow 4100.$$

Observa que hemos descontado las 10 direcciones de bloques de datos que se almacenan en el nodo-i, y que hemos tenido en cuenta que en cada bloque indirecto se pueden almacenar 512 direcciones, tal y como ya hemos calculado antes.

La dirección de disco del BSI 4100 se almacenará en el BDI:

$$\frac{4100 - 1}{512} = 8,01 \rightarrow 8.$$

Para hacer el cálculo, ahora hemos descontado la dirección de disco del primer BSI, que se almacena en el nodo-i.

Como el BDI es el 8, está claro que la dirección de disco de este BDI se encontrará en el BTI apuntado por el nodo-i.

Por lo tanto, para escribir en el byte 4299202560 del fichero tenemos que leer primero su bloque lógico 2099220 (lo que supone 4 accesos a disco por el BTI, el BDI, el BSI y el propio bloque lógico), modificar el byte de su primera posición y escribir de nuevo el bloque en disco, en la misma dirección, lo que añade un acceso más. En total, son necesarios 5 accesos a disco.