



Introducción a los Sistemas Operativos

Examen final de teoría (temas 1–4) — Problemas

24 de noviembre de 2023

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

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.
- El valor de cada pregunta aparece entre paréntesis, al principio de la misma.
- Esta parte vale 3 puntos y, por tanto, representa el 30 % de la nota de este examen de teoría.
- Esta parte solo será calificada cuando la nota de la parte test sea igual o superior a la nota mínima establecida.

1. (0,75 puntos) En un sistema se ejecutan los siguientes procesos (tiempo en tics):

Proceso	Llegada	CPU	E/S	CPU
P1	0	3	1	2
P2	1	2	1	1
P3	2	2	3	1

Para cada uno de los planificadores que se indican a continuación, rellena el diagrama de Gantt en el cronograma correspondiente, donde el tiempo sobre cada columna indica el instante en el que comienza el tic representado por la celda. Has de utilizar las letras E (en ejecución), B (bloqueado) y L (en la cola de listos) para indicar el estado de cada proceso. En todos los casos, los empates se resuelven dándole mayor prioridad al proceso con menor número.

a) (0,25 puntos) El planificador utilizado es SJF oráculo.

[illegible]

b) (0,25 puntos) El planificador utilizado es SRTF con estimación, donde se utiliza un factor $\alpha=0.5$ y la estimación inicial para cada proceso es el resultado de restar 4 y el número del proceso. Añade a los estados E y L un subíndice que indique la estimación actual de cada proceso.

[illegible]

c) (0,25 puntos) El planificador utilizado es Round-Robin con un quantum de 2 tics de duración.

[illegible]

2. (2,25 puntos) Considera un sistema de ficheros UNIX con las zonas vistas en clase: bloque de arranque, superbloque (una sola copia), mapa de bits de nodos-i, tabla de nodos-i, mapa de bits de datos y zona de datos. El sistema de ficheros emplea bloques lógicos de tamaño 1 KiB, las direcciones de disco son de 4 bytes y los nodos-i, que tienen un tamaño de 64 bytes, son como los vistos en clase, con 10 punteros directos, un puntero a un BSI, un puntero a un BDI y un puntero a un BTI. Vemos que en este momento quedan 64 566 nodos-i libres y 86 717 bloques libres en la zona de datos. Se pide:

- a) (0,75 puntos) Sabiendo que en total hay creados 984 010 ficheros de cualquier tipo y que, en media, cada fichero necesita 10,5 bloques de la zona de datos, calcula el tamaño de cada una de las zonas del disco.
- b) (0,5 puntos) Considerando los bloques libres de la zona de datos, ¿cuál sería el tamaño máximo que podría tener un fichero regular?
- c) (0,4 puntos) Si creamos el directorio `/home/alumno/ExamenNoviembre`, y a continuación, creamos el fichero `/home/alumno/ExamenNoviembre/problemas.pdf` con un tamaño de 128 KiB, ¿cuántos nodos-i y bloques de la zona de datos quedarían libres? Recuerda que antes de la creación del directorio y el fichero hay 86 717 bloques libres en la zona de datos y 64 566 nodos-i libres. Además, supón que el directorio `/home/alumno` ya existe y que dispone de espacio suficiente para crear varias entradas de directorio.
- d) (0,6 puntos) Dado un fichero de tamaño 1 GiB, indica qué punteros de las estructuras del nodo-i y de los bloques indirectos (y de cuáles) habría que seguir para llegar a poder leer el bloque que contiene el byte $1\,073 \times 10^6$ del fichero. Recuerda que los bytes se empiezan a contar a partir del 0. Ejemplo de respuesta:

Para poder leer el byte 123 456 789, habría que leer el bloque de datos 123 456, que se encontraría usando primero el puntero X del BTI, el cual nos llevará a un BDI del que usaremos su puntero Y, que a su vez nos llevará a un BSI del que usaremos su puntero Z.

Soluciones

Ejercicio 1:

La solución del ejercicio de desarrolla en los siguientes apartados:

a) La solución para este apartado se muestra en el cronograma siguiente:

Tic	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
P_1	E	E	E	B	L	E	E									
P_2	–	L	L	E	E	B	L	E								
P_3	–	–	L	L	L	L	L	L	E	E	B	B	B	E		

b) La solución para este apartado se muestra en el cronograma siguiente:

Tic	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
P_1	E_3	E_2	E_1	B	L_3	L_3	L_3	E_3	L_2	E_2						
P_2	–	L_2	L_2	L_2	L_2	E_2	E_1	B	L_2	L_2	E_2					
P_3	–	–	L_1	E_1	E_0	B	B	B	$E_{1,5}$							

c) La solución para este apartado se muestra en el cronograma siguiente:

Tic	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
P_1	E	E	L	L	E	B	L	L	E	E						
P_2	–	L	E	E	B	L	L	E								
P_3	–	–	L	L	L	E	E	B	B	B	E					

Ejercicio 2:

La solución del ejercicio de desarrolla en los siguientes apartados:

- a) Nos dicen que se han creado un total de 984 010 ficheros, con lo que sabemos que hay ocupados ese mismo número de nodos-i. Además, nos dicen que quedan 64 566 nodos-i libres, con lo que el total de nodos-i es $984\,010 + 64\,566 = 1\,048\,576$. Dado que cada nodo-i tiene un tamaño de 64 bytes, el tamaño total de la **tabla de nodos-i** es 67 108 864 bytes, o lo que es lo mismo **65 536 bloques lógicos**.

En cuanto al **mapa de nodos-i**, este requiere 1 bit por nodo-i y, por lo tanto, tiene un tamaño total de 1 048 576 bits, o lo que es lo mismo **128 bloques lógicos**.

En cuanto a la zona de datos, sabemos que hay creados un total de 984 010 ficheros, cada uno de los cuales ocupa, de media, 10,5 bloques. De esta forma, el total de ficheros ocupan 10 332 105 bloques lógicos. Además, nos dicen que hay un total de 86 717 bloques libres, con lo que el tamaño total de la **zona de datos es 10 418 822 bloques lógicos**. Para poder cubrir todos estos bloques lógicos se necesita un **mapa de bits para la zona de datos** de tamaño $\lceil \frac{10\,418\,822}{8 \times 1024} \rceil = \mathbf{1272 \text{ bloques lógicos}}$.

Por último, el bloque de arranque y el superbloque requerirían un bloque lógico cada uno.

- b) Vamos a considerar, por el momento, que todos los bloques libres los repartimos entre bloques de datos del ficheros y bloques simplemente indirectos. Si suponemos que el fichero usa B bloques para guardar sus datos, se requerirán un total de B punteros directos. Dado que el nodo-i proporciona únicamente 10 punteros directos, el resto tendrán que ser obtenidos a partir de bloques simplemente indirectos. Dado que el tamaño de bloque lógico es 1 KiB y el tamaño de las direcciones de disco 4 bytes, cada bloque indirecto puede proporcionar $\frac{1024}{4} = 256$ punteros directos adicionales. De esta forma, quedaría que:

$$B + \frac{B-10}{256} = 86\,717$$

Despejando, obtendríamos que $B = 86\,379,618\dots$ bloques lógicos, con lo que, en un principio, tendríamos que dejar 86 379 bloques lógicos para los datos, y el resto, 338, serían para BSI.

Ahora bien, el nodo-i tiene capacidad para almacenar únicamente el puntero a un BSI. El resto tienen que ser enlazados a través de BDIs. Dado que cada BDI proporciona 256 punteros a BSI, necesitamos 2 BDI, y por la misma razón, recurrir al BTI. De esta forma, hay que «quitar» 3 bloques de datos más para dedicarlos a los 2 BDI y al BTI.

De esta forma, el fichero tendría como máximo 86 376, dando un **tamaño máximo total de 86 376 KiB**.

- c) Para crear el directorio se necesitarán 1 nodo-i y 1 bloque lógico de la zona de datos. Para crear el fichero, además del correspondiente nodo-i, y puesto que su tamaño es de 128 KiB, se requerirán 128 bloques lógicos para los datos, y necesitaremos 128 punteros directos para registrarlos. A partir de lo visto en el apartado anterior, nos bastará, por lo tanto, con un bloque simplemente indirecto.

Por lo tanto, en total vamos a consumir 2 nodos-i y 130 bloques de datos, con lo que **quedarían 64 564 nodos-i libres y 86 587 bloques libres en la zona de datos**.

- d) Dividimos el byte que buscamos ($1\,073 \times 10^6$) entre el tamaño de bloque, y el cociente de la división nos indica el bloque en el que está (bloque 1 047 851) del fichero, mientras que el resto de dicha división indica la posición del byte dentro del bloque (es el byte 576 del bloque 1 047 851 del fichero)

Dado que los 10 primeros bloques del fichero están apuntados por los punteros directos del nodo-i, los restamos y buscamos, por lo tanto, el bloque 1 047 841 a partir de los bloques indirectos.

Si dividimos 1 047 841 entre los 256 punteros que tenemos por cada bloque indirecto, el cociente de la división nos indicará el número de BSI que contiene el puntero directo al bloque de datos (BSI 4093) y el resto, la posición de ese puntero dentro del BSI (puntero 33 dentro del BSI).

Dado que el nodo-i contiene el primer puntero a un BSI, lo restamos y dividimos el resultado (4092) entre 256 para buscar el BDI al que hemos de acudir. Al igual que antes, el cociente de la división indica el número del BDI (15) y el resto, el puntero dentro del BDI (252).

Por último, el primer BDI está apuntado en el nodo-i directamente, con lo que tendremos que usar el puntero 14 dentro del BTI.

En resumen, para lograr leer el byte $1\,073 \times 10^6$ del fichero, usaremos primero el **puntero 14 del BTI**, el cual nos llevará a un **BDI del que usaremos su puntero 252**, que a su vez nos llevará a un **BSI del que usaremos su puntero 33**. Este último es el que contiene la dirección de disco del bloque 1 047 851 del fichero, donde está el byte buscado.

Grupo:	DNI:
Apellidos:	Nombre:

Nº	a	b
1	X	
2	X	
3	X	
4	X	
5		X
6	X	
7	X	
8		X
9	X	
10		X
11	X	
12		X
13		X
14	X	
15	X	
16		X
17		X
18		X
19	X	
20		X
21	X	
22	X	
23	X	
24		X
25		X

Respuestas del PROFESOR

Nº	a	b
26	X	
27		X
28		X
29	X	
30	X	
31		X
32		X
33		X
34		X
35	X	
36	X	
37	X	
38	X	
39		X
40		X
41		X
42	X	
43	X	
44	X	
45	X	
46		X
47		X
48	X	
49		X
50	X	

Grupo:	DNI:
Apellidos:	Nombre:

Nº	a	b
1		X
2		X
3		X
4		X
5	X	
6	X	
7	X	
8	X	
9		X
10		X
11		X
12	X	
13	X	
14	X	
15	X	
16		X
17		X
18	X	
19		X
20	X	
21	X	
22		X
23		X
24	X	
25	X	

Respuestas del PROFESOR

Nº	a	b
26	X	
27	X	
28	X	
29	X	
30		X
31	X	
32	X	
33		X
34	X	
35		X
36	X	
37		X
38		X
39	X	
40	X	
41		X
42		X
43		X
44	X	
45		X
46	X	
47	X	
48	X	
49		X
50		X