



Calificación
1
2
3

Nombre	Grupo
SOLUCIONES	

Dispones de dos horas y media para completar el examen

1 (3 puntos. Todas las cuestiones puntúan lo mismo) Contestar de forma precisa a las siguientes cuestiones:

1. Defina con precisión los siguientes términos:
 - a) sistema operativo
 - b) núcleo
 - c) llamada al sistema

Solución: consultar esos términos en la bibliografía de la asignatura.

2. ¿Qué acciones realiza típicamente una CPU cuando recibe una señal de interrupción, en un sistema con interrupciones vectorizadas?

Solución: lo habitual es que suspenda la ejecución de la instrucción actual y cambiará el contador de programa al valor apuntado por la entrada del vector de interrupciones correspondiente con el número asociado a la interrupción. Al variar el contador de programa, es habitual que conmute a modo supervisor y que guarde el estado del proceso interrumpido (ej. valores de los registros).

3. En la gestión de procesos, ¿qué función tiene el planificador de largo plazo, también llamado de alto nivel?

Solución: el planificador de largo plazo se encarga de alimentar de procesos recién llegados al planificador de corto plazo. Es decir, actúa como un filtro de entrada inicial para los procesos que pretenden ejecutarse, para lograr objetivos de eficiencia como impedir que se supere un máximo de procesos en ejecución, un cierto nivel de consumo de recursos, cuotas de procesos por usuarios, etc.

4. En la planificación de procesos con Round Robin, ¿qué efecto se produce si el cuanto de tiempo se vuelve demasiado largo?

Solución: el algoritmo tiende a parecerse a un FCFS, con lo cual el tiempo de respuesta aumenta, se va perdiendo interactividad, se penaliza a los procesos intensivos en E/S y puede observarse el efecto convoy por acaparamiento de los procesos intensivos en CPU.

5. ¿Por qué las operaciones de un semáforo deben de ser necesariamente atómicas?

Solución: deben serlo para evitar que la manipulación concurrente del valor asociado al semáforo dé lugar a inconsistencias en el comportamiento esperado, por ejemplo que dos procesos realicen una operación WAIT sobre el mismo semáforo con valor uno, y ambos procesos decidan no bloquearse al no producirse el decremento del valor de forma secuenciada.

6. ¿En qué consiste el problema de la fragmentación externa en la gestión de archivos?

Solución: consiste en la existencia de zonas de espacio libre, pero que no pueden ser asignadas a procesos que lo solicitan, debido a que las zonas no están contiguas (están fragmentadas) y los algoritmos de gestión no permiten conceder varios fragmentos para conceder una petición.

7. En las políticas de reemplazo de páginas, ¿qué ventaja tiene el algoritmo LRU sobre el FIFO?

Solución: la ventaja principal es que el algoritmo LRU es una mejor aproximación que el FIFO a la política óptima: la página menos recientemente usada es más probable que sea la que más tiempo se tarde en acceder, comparado con la página más vieja (que puede ser una página frecuentemente accedida). Teniendo en cuenta esto, el LRU tenderá con el tiempo a darnos menos fallos de página que el FIFO.

8. Tenemos un sistema de ficheros que trabaja con bloques de datos de 1Kbyte y utiliza asignación de espacio indexada, con un solo bloque para los índices. Cada enlace ocupa 16 bits. Con estos datos, obtenga el tamaño máximo que puede tener un archivo y cuál es el espacio máximo

Solución: Como cada enlace son 16 bits = 2 bytes, en el único bloque de índices caben $1024/2 = 512$ enlaces. Por tanto un archivo puede llegar a tener 512 bloques = 512 Kbytes.
El espacio máximo direccionable viene dado por el tamaño del enlace, que nos da 2^{16} combinaciones = 65.536 bloques = 64 Mbytes.

9. Tenemos estos dos procesos que están continuamente accediendo a una variable compartida x, cuyo valor inicial es cero:

<pre>loop x := x + 1; end loop;</pre>	<pre>loop x := x - 1; end loop;</pre>
---	---

a) ¿Cuáles son los posibles valores que puede adquirir x a lo largo del tiempo?

Solución: en general, la variable x contendrá la diferencia de iteraciones entre los dos procesos, tanto en positivo (ventaja para el proceso de la izquierda) como en negativo (ventaja para el de la derecha). Cada uno de los dos procesos puede adelantarse al otro un número arbitrario de iteraciones, ya que el enunciado no establece ninguna especificación sobre cómo se están repartiendo el tiempo los procesos. Por tanto la variable x puede adquirir cualquier valor entero.

b) ¿Varía su respuesta dependiendo del número de procesadores que tenga el sistema?

Solución: El número de procesadores no influye en la respuesta anterior.

10. Mencione alguna diferencia significativa entre el proceso de traducción de direcciones de un sistema basado en segmentación, respecto al proceso de traducción de un sistema paginado.

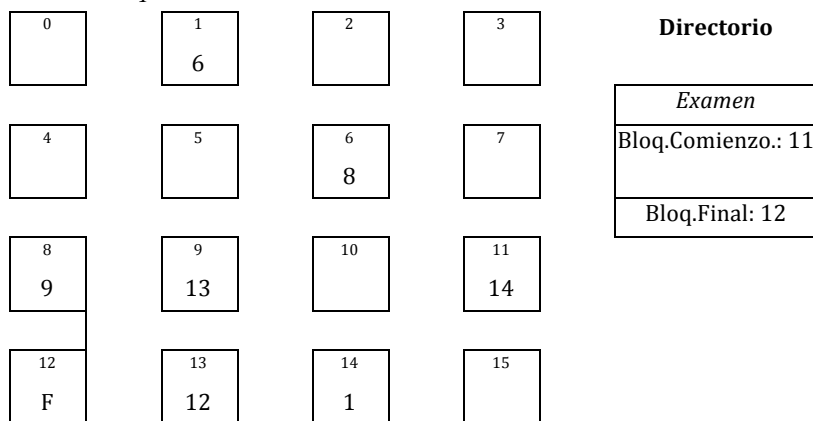
Solución: Algunas de las diferencias son:

1. Como un segmento puede iniciarse en una dirección arbitraria de la memoria, hay que realizar una suma de la dirección base del segmento y del desplazamiento. En la paginación, las páginas se ubican en marcos de posiciones fijas y la suma se realiza implícitamente con una mera sustitución de bits.
2. Como un segmento puede tener un tamaño variable, durante la traducción se debería comprobar que el desplazamiento dentro del segmento se encuentra dentro del rango permitido. Esta comprobación no se hace en un sistema paginado simple.

Nombre

2 (5 puntos. Todos los problemas puntúan lo mismo)

1. En la figura se representa los 16 primeros bloques de un dispositivo de almacenamiento secundario (disco) de capacidad 1 Gb. El método que se utiliza para la asignación de espacio en disco es el encadenamiento (enlazado). Cada bloque tiene 4Kb. En la figura también se representa un fichero llamado *examen*. El último bloque del fichero se indica con el símbolo F:



- a) Calcular el tamaño máximo (en bytes) de los datos almacenados en el fichero *examen*. (0.5p)

Solución: Los bloques empleados en el fichero "*Examen*" son: 11-14-1-6-8-9-13-12. En total, 8 bloques. Por otro lado, el disco tiene una capacidad de 1Gb (2^{30} bytes). Además cada bloque tiene 4Kbytes (2^{12} bytes). Por tanto, dividiendo $2^{30}/2^{12}$, tenemos que el número de bloques del disco sería 256 (2^8 bloques). Por tanto, de los 4096 bytes de cada bloque, 4095 se dedicarían a datos y 1 para almacenar el enlace al siguiente bloque. Para el fichero "*Examen*", la cantidad de datos que se podrían almacenar es $8 \cdot 4095 = 32760$ bytes.

- b) ¿Es posible el acceso directo a los archivos de este sistema? ¿es rentable con esta organización de disponer de este servicio? (0.5p)

Solución: No es posible ni rentable. Sería muchísimo mejor emplear la asignación indexada.

2. Deducir las expresiones y calcular el tiempo que es necesario para leer 5 bloques consecutivos de un archivo en un sistema con:

- a) Asignación contigua. (0.33p)

Solución:

$$t = t_b + t_r + (N \cdot t_t) = 15 + 10 + (5 \cdot 1.2)$$

- b) Asignación mediante listas enlazadas. (0.33p)

Solución:

$$t = N \cdot (t_b + t_r + t_t) = 5 \cdot (15 + 10 + 1.2)$$

- c) Asignación mediante indexación de un solo nivel. (0.33p)

Solución:

$$t = (N + 1) * (tb + tr + tt) = 6 * (15 + 10 + 1.2)$$

Considerar que el tiempo de búsqueda es $t_b=15\text{ms}$, el retardo rotacional $t_r=10\text{ms}$, y el tiempo de transferencia de los datos de un bloque $t_t=1,2\text{ms}$.

3. La siguiente tabla recoge información de cinco procesos que se van a ejecutar en un sistema. Se pide calcular el tiempo de retorno de cada uno de los trabajos y representar la ejecución en diagramas de Gantt para los siguientes algoritmos:

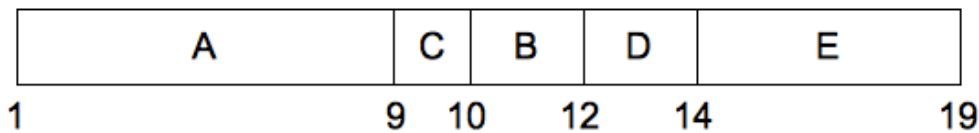
- a) SJF. (0.2p)
- b) Round Robin con cuanto de 2 u.t. considerando que si el cuanto de un proceso en ejecución expira a la vez que la llegada de un nuevo proceso, entonces el nuevo proceso se añade a la cola de procesos en espera de ejecutarse antes que el proceso termine. (0.4p)
- c) Ídem que el apartado anterior, pero considerando un Round Robin con prioridades, siendo 4 la máxima prioridad y 1 la mínima. (0.4p)

Trabajo	Tiempo de llegada	Tiempo de ejecución (u.t.)	Prioridad
A	1	8	2
B	2	2	4
C	3	1	3
D	4	2	4
E	5	5	1

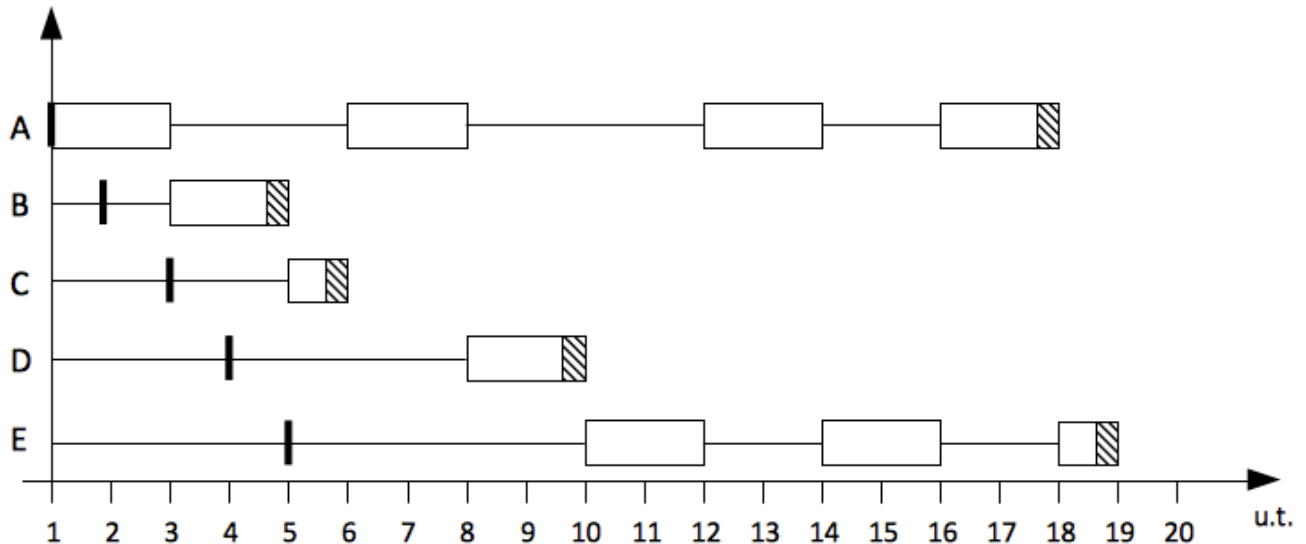
Solución:

a) SJF

El diagrama de Gantt simplificado es:

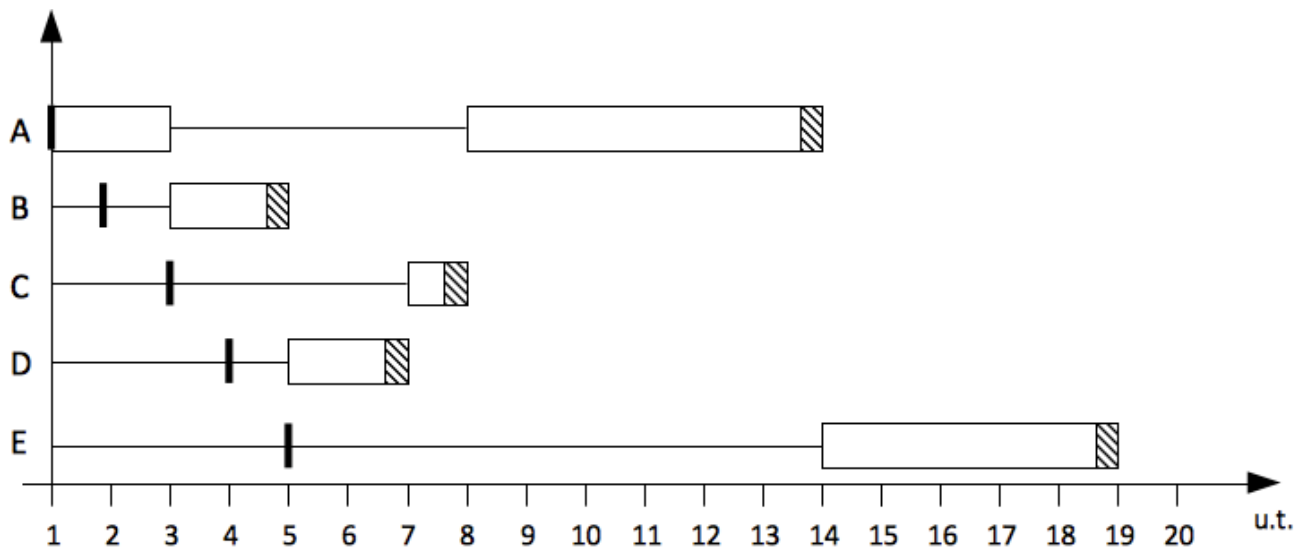


b) El diagrama de Gantt correspondiente al apartado b es el siguiente:



Nombre

c) El diagrama de Gantt correspondiente al apartado c es el siguiente:



Con la información de los diagramas pueden calcularse los tiempos de retorno:

	Tiempo de retorno (u.t.)		
Trabajo	SJF	RR (b)	RR (c)
A	8	13	17
B	7	3	3
C	10	5	3
D	10	3	6
E	14	14	14

4. Supongamos un sistema de gestión de memoria virtual con paginación. Existe un proceso al que se le asignan 4 marcos durante su ejecución y que hace referencia a la siguiente lista de páginas: 4 8 9 7 3 8 4 8 4 6 8 5 8. Indique cuantos fallos de página se originarán si se aplica:

a) El algoritmo de reemplazo de la segunda oportunidad. (0.5p)

Solución: Se generan 8 fallos de página.

4	8	9	7	3	8	4	8	4	6	8	5	8
* 4 ₁	* 4 ₁	* 4 ₁	* 4 ₁	3 ₀	3 ₀	3 ₀	3 ₀	3 ₀	* 3 ₀	* 3 ₀	5 ₁	5 ₁
	8 ₁	8 ₁	8 ₁	* 8 ₀	* 8 ₁	8 ₀	8 ₁	8 ₁	8 ₁	8 ₁	* 8 ₁	* 8 ₁
		9 ₁	9 ₁	9 ₀	9 ₀	4 ₁	4 ₁	4 ₁	4 ₁	4 ₁	4 ₁	4 ₁
			7 ₁	7 ₀	7 ₀	* 7 ₀	* 7 ₀	* 7 ₀	6 ₁	6 ₁	6 ₁	6 ₁
F	F	F	F	F		F			F		F	

- b) Alguien afirma que aplicando el algoritmo LRU, tanto para esta cadena de referencia como para cualquier otra, se generaría un número de fallos de página menor o igual que el obtenido al aplicar el algoritmo de la segunda oportunidad. ¿Es cierta la afirmación anterior? Justifique su respuesta. (0.5p)

Solución: Sí, el algoritmo de la segunda oportunidad es una aproximación a la política LRU en la que para evitar sustituir páginas usadas recientemente, se emplea un bit de referencia en lugar de mantener una ordenación de las páginas en función del momento en el que han sido accedidas, política que sería mucho más costosa de implementar.

5. Suponga un sistema operativo multitarea con gestión de memoria virtual que combina la segmentación y la paginación. Las direcciones virtuales son de 32 bits, de los cuales 8 bits indican el número de segmento, 12 bits indican la página lógica y 12 bits el desplazamiento.
- a) ¿Cuál sería la dirección física asociada a la dirección virtual (20,83400) dada en el formato (SEGMENTO, DEZPLAZAMIENTO) si el segmento 20 empieza en la dirección física 327680 y todas sus páginas se encuentran situadas en memoria física de forma consecutiva, siendo el tamaño de dicho segmento 86016 bytes? (0.6p)

Solución: En primer lugar es necesario comprobar que se trata de una dirección lógica válida. Con los 8 bits que se emplean para indicar el número de segmento, se pueden direccionar 256 segmentos. Por tanto, el número de segmento 20 está dentro del rango posible. Por otro lado, el desplazamiento 83400, no supera el tamaño del segmento, que es 86016. Por tanto se trata de una dirección lógica válida.

Dado que el segmento 20 comienza en la dirección física 327680, y todas las páginas que componen dicho segmento se encuentran de forma consecutiva a partir de dicha dirección, para obtener la dirección física será necesario calcular el número de página lógica y desplazamiento correspondiente a la dirección lógica 83400. En este caso sería, página lógica 20, desplazamiento 1480. Por tanto, la dirección física correspondiente a la dirección lógica (20,83400) será $327680 + (20 * 4096) + 1480 = 411080$.

- b) En el sistema de gestión de memoria explicado, ¿tiene sentido hablar de fragmentación interna?, ¿cuánto sería el espacio perdido por fragmentación interna para el segmento lógico 20? (0.4p)

Solución: Efectivamente tiene sentido hablar de fragmentación interna a tratarse de un sistema paginado, entiendo por fragmentación interna el posible desperdicio de memoria de la última página de cada segmento como consecuencia de que el tamaño del segmento no sea un múltiplo exacto del tamaño de página. En el caso del segmento 20, tenemos una fragmentación interna de $4096 - 1480 = 2616$ bytes.

2 (2 puntos) Supongamos que tenemos 5 procesos de dos tipos : A1, A2, A3, B1 y B2. Implementar un programa usando semáforos de forma que los procesos siempre terminen en el orden: A(cualquiera de ellos), B(cualquiera de ellos), A (cualquiera de ellos), A y B. Teniendo en cuenta el código que se muestra a continuación de los dos tipo de procesos, la salida por pantalla debe ser la siguiente:

Un proceso A ha terminado
Un proceso B ha terminado
Un proceso A ha terminado
Un proceso A ha terminado
Un proceso B ha terminado

Nombre

```
Proceso Tipo A() {
...
...
// Protocolo de salida
printf("Un proceso A ha terminado\n")
}

Proceso Tipo B(){
...
...
// Protocolo de salida
printf("Un proceso B ha terminado\n")
}
```

Solución:

```
// Variables globales
//Variable empleada para indicar que un proceso B ha terminado
Bterminado = false;
//El semáforo semA funcionará como cola de espera para los
//procesos de tipo A. Se inicializa a 1 para que un
//proceso A pueda ser el primero en terminar
// El semáforo semB funcionará como cola de espera para
//los procesos de tipo B. Se inicializa a 0 de forma que
//todos los procesos B queden en espera inicialmente hasta
//que un proceso A termine
Semáforo semA=1, semB=0;
```

<pre>Proceso Tipo A() { // Protocolo de salida wait(semA); printf("Un proceso A ha terminado\n") if Bterminado { //Si un proceso B ha terminado //reseteamos Bterminado y despertamos //a un proceso A Bterminado=false; signal(semA); } else signal(semB); }</pre>	<pre>Proceso Tipo B(){ // Protocolo de salida wait(semB); printf("Un proceso B ha terminado\n") //Indicamos que un proceso B termina Bterminado=true; //Despertamos a un proceso A Signal(semA); }</pre>
---	--