

Material permitido: Solo calculadora no programable	Aviso 1: Todas las respuestas deben estar debidamente razonadas.
Tiempo: 120 minutos	Aviso 2: Escriba con buena letra y evite los tachones.
N1	Aviso 3: Solución del examen y fecha de revisión en http://www.uned.es/71902048/

1. Conteste **razonadamente** a las siguientes preguntas:

- a) (1 p) Enumerar y describir las principales causas que originan la creación de un proceso.
 - b) (1 p) Describir el algoritmo de planificación basado en múltiples colas de prioridad y realimentación.
 - c) (1 p) Cuando se usa paginación, ¿en qué campos se descompone una dirección física y una dirección lógica? ¿cómo se puede determinar el tamaño de dichos campos?
 - d) (1 p) Describir los elementos en qué se descompone un sector de un disco duro.
2. (2 p) Describir el funcionamiento de las operaciones `wait_mon` y `signal_mon` de un monitor. ¿Qué diferencias tienen estas operaciones con respecto a las operaciones `wait_sem` y `signal_sem` de un semáforo? ¿Qué sucede con un proceso que invoca una operación `signal_mon` según la solución de Hansen?
3. (2 p) Supóngase un sistema con cuatro procesos (P0, P1, P2 y P3) y cuatro recursos diferentes (R0, R1, R2 y R3). En un instante determinado el sistema se encuentra en el siguiente estado:

Proceso	Asignados				Adicional Necesitados				Disponibles			
	R0	R1	R2	R3	R0	R1	R2	R3	R0	R1	R2	R3
P0	0	1	1	0	0	1	1	0	3	2	1	3
P1	2	0	0	0	2	0	0	2				
P2	0	0	0	0	1	0	1	0				
P3	1	0	0	2	2	0	0	2				

Detectar la posible existencia de interbloqueos usando el algoritmo de Coffman. En el caso de que exista interbloqueo indicar que procesos P_i se quedan bloqueados.

4. Un sistema con memoria virtual mediante demanda de página utiliza el algoritmo LRU para la sustitución de páginas. Un proceso genera la siguiente secuencia de referencia a páginas de memoria:

1 3 2 4 1 5 7 4 3 2 8 9 4 5 4 9 1 8 3 2

- a) (1.5 p) Determinar cuántos fallos de página se producen cuando se dispone de 4 marcos de página para este proceso.
- b) (0.5 p) ¿Cuál es el número de fallos mínimo que siempre produce esta secuencia, independientemente del número de marcos de página que exista?

SISTEMAS OPERATIVOS (Cód. 71902048)

Solución Examen Enero 2024

Solución Ejercicio 1

a) Existen diferentes causas que originan la creación de un proceso por parte del sistema operativo, algunas de las más importantes son las siguientes:

- *Arranque del sistema operativo.* Al arrancar el sistema operativo se crean diversos procesos. Algunos son procesos que se ejecutan en segundo plano, como es el caso de los procesos demonio y los procesos del sistema. Otros son procesos que se ejecutan en primer plano, como por ejemplo, los intérpretes de comandos y los gestores de ventanas.
- *Interacción del usuario con un intérprete de comandos o un entorno de ventanas.* En sistemas operativos interactivos, cuando un usuario teclea una orden en un intérprete de comandos o hace clic con el ratón sobre un icono de una ventana se crea un nuevo proceso para atender la petición del usuario.
- *Inicio de un trabajo por lotes.* Cada vez que se inicia un trabajo por lotes se crea un proceso asociado a la ejecución de dicho trabajo.
- *Un proceso en ejecución invoca a una llamada al sistema para crear otro proceso.* Una determinada tarea puede plantearse para ser realizada en varias fases independientes, cada una de las cuales puede ser realizada por un proceso. Así un proceso puede realizar llamadas al sistema para solicitar la creación de otros procesos que le ayuden en la realización de una tarea.

b) En el algoritmo de planificación basada en múltiples colas de prioridad y realimentación existen varias colas de prioridad en las que se colocan los procesos.

Cuando hay que planificar un proceso se toma uno perteneciente a la cola de procesos preparados de mayor prioridad utilizando algún algoritmo de planificación (FCFS, SJF, turno rotatorio, etc). Sólo cuando la cola de mayor prioridad está vacía se puede planificar otro proceso perteneciente a una cola de procesos preparados de menor prioridad.

Si se permite que un proceso pueda ir cambiando de cola de prioridad durante su existencia, es decir, que su prioridad pueda ser modificada dinámicamente, se dice que el algoritmo posee *realimentación*.

La implementación de la realimentación requiere definir un mecanismo que regule cómo se modifica dinámicamente la prioridad de los procesos durante su tiempo de existencia. O visto de otra forma, cuándo se produce la transición de un proceso de una cola de prioridad a otra.

c) En la técnica de paginación una *dirección física* consta de dos campos: el *número de marco de página* de f bits y el *desplazamiento* dentro del marco de d bits.

Marco j	Desplazamiento d
f bits	d bits

El tamaño f del campo número de marco de página se obtiene a partir del número N_{MP} de marcos de página existentes, resolviendo la siguiente desigualdad:

$$\min_f \{N_{MP} \leq 2^f\} \quad (1)$$

Mientras que el tamaño d del campo desplazamiento se obtiene a partir del tamaño S_P de una página o de un marco de página expresado en unidades direccionables (usualmente palabras) resolviendo la siguiente desigualdad:

$$\min_d \{S_P \leq 2^d\} \quad (2)$$

Por su parte, una *dirección lógica* consta de los siguientes dos campos: el *número de página* de p bits y el *desplazamiento* dentro de la página de d bits.

Página i	Desplazamiento d
p bits	d bits

El tamaño p del campo número de página se obtiene a partir del número total de páginas N_P de que consta un proceso X resolviendo la siguiente desigualdad:

$$\min_p \{N_P \leq 2^p\} \quad (3)$$

Por su parte, el tamaño del campo desplazamiento es igual que el de una dirección física.

- d) Un *sector de un disco duro* es una estructura de datos que consta principalmente de tres elementos:
- *Cabecera o preámbulo*. Contiene, entre otra informaciones, un cierto patrón de bits que permite reconocer al controlador del disco el comienzo del sector, el número de sector y el número de cilindro.
 - *Área de datos*. En ella se almacena la información accesible para el sistema operativo y los usuarios. Su tamaño típico suele ser 512 bytes. Aunque es posible usar tamaños de 256 bytes o 1024 bytes.
 - *Código de corrección de errores* (Error Correcting Code, ECC). Contiene información que permite al controlador del disco detectar y corregir posibles errores de lectura. El ECC suele ocupar 16 bytes.

Solución Ejercicio 2

Sobre una variable de condición X de un monitor es posible realizar dos posibles operaciones:

- `wait_mon(X)`. El proceso dentro del monitor que realiza esta operación queda suspendido en una cola de procesos asociada al cumplimiento de la condición X . En consecuencia otro proceso puede entrar en el monitor. Nótese que a diferencia de la operación `wait_sem` de los semáforos la operación `wait_mon` de los monitores siempre produce el bloqueo del proceso que la invoca.
- `signal_mon(X)`. Comprueba si la cola de procesos asociada a la condición X contiene algún proceso bloqueado. En caso afirmativo se desbloquea a un proceso. Si la cola estaba vacía esta operación no tiene ningún efecto. Nótese que a diferencia de la operación `signal_sem` de los semáforos, la operación `signal_mon` de los monitores no incrementa ningún contador, en consecuencia si no existe ningún proceso en la cola de condición la señal de aviso se pierde.

Según la solución de B. Hansen el proceso invocador de la operación `signal_mon` sale del monitor inmediatamente. Luego esta operación aparecería como la sentencia final del procedimiento de un monitor.

Solución Ejercicio 3

La matriz de recursos máximos necesitados adicionalmente, la matriz de recursos asignados y el vector de recursos disponibles son:

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \end{pmatrix} \quad \mathbf{R}_D = (3 \quad 2 \quad 1 \quad 3)$$

Se va aplicar el algoritmo de Coffman para la detección de interbloqueos:

1. Se examina la matriz \mathbf{A} y se comprueba si alguna de sus filas es igual a (0 0 0 0), ya que un proceso que no tiene ningún recurso asignado no puede producir interbloqueo. Se observa que la tercera fila de \mathbf{A} es igual a (0 0 0 0), luego se marca el proceso P_2 .
2. Se realiza la asignación

$$\mathbf{X} = \mathbf{R}_D = (3 \quad 2 \quad 1 \quad 3)$$

3. Para cada proceso P_i ($i = 0, 1, 2, 3$) no marcado, en este caso los procesos P_0, P_1 y P_3 , se comprueba la condición

$$\mathbf{M}_i \stackrel{e}{\leq} \mathbf{X}$$

donde el símbolo $\stackrel{e}{\leq}$ indica la operación de comparación entre dos vectores elemento a elemento.

- 3.1) Para la primera fila de \mathbf{M} asociada al proceso P_0

$$(0 \quad 1 \quad 1 \quad 0) \stackrel{e}{\leq} (3 \quad 2 \quad 1 \quad 3)$$

se observa que se cumple la condición. Luego se marca el proceso P_0 .

- 3.2) Para la segunda fila de \mathbf{M} asociada al proceso P_1

$$(2 \quad 0 \quad 0 \quad 2) \stackrel{e}{\leq} (3 \quad 2 \quad 1 \quad 3)$$

se observa que se cumple la condición. Luego se marca el proceso P_1 .

- 3.3) Para la cuarta fila de \mathbf{M} asociada al proceso P_3

$$(2 \quad 0 \quad 0 \quad 2) \stackrel{e}{\leq} (3 \quad 2 \quad 1 \quad 3)$$

se observa que se cumple la condición. Luego se marca el proceso P_3 .

4. Como todos los procesos están marcados el algoritmo finaliza.

Al terminar el algoritmo todos los procesos han quedado marcados, por lo tanto **no existe interbloqueo**.

Solución Ejercicio 4

a) Se va a suponer que para implementar el algoritmo LRU se utiliza una lista enlazada. En la Figura 1 se muestra el contenido de la lista y el contenido de los marcos antes y después de cada referencia de la secuencia cuando se dispone de 4 marcos de página. Se indica también si dicha referencia produce un fallo o un acierto.

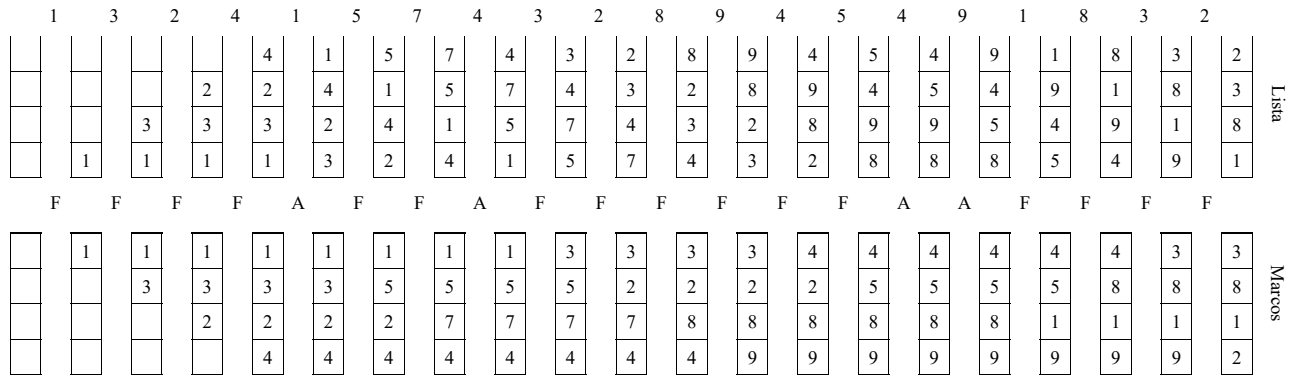


Figura 1

Se observa en la Figura 1 que se producen un total de **16 fallos de página**.

b) Analizando la secuencia de páginas referenciadas se observa que existen 8 páginas distintas:

$$\{1\ 2\ 3\ 4\ 5\ 7\ 8\ 9\}$$

Supuesto que ninguna de estas páginas se encuentra previamente cargada en memoria se producirán siempre **8 fallos de página** independientemente del número de marcos de página de que se disponga.