

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

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

I) (1 p) ¿Qué son las *llamadas al sistema*?

II) (1 p) ¿Qué se entiende por *sobrecarga del sistema*?

III) (1 p) ¿Qué es y en qué consiste la *paginación simple*?

IV) (1 p) ¿Qué es un *nodo índice*?

2. (2 p) Describir el funcionamiento de las operaciones `wait_sem` y `signal_sem` cuando se aplican sobre un semáforo general y cuando se aplican sobre un semáforo binario.

3. (2 p) Un sistema tiene cuatro procesos y cinco recursos asignables. En un cierto instante de tiempo la matriz **M** de recursos máximos necesitados adicionalmente, la matriz **A** de recursos asignados y el vector de recursos disponibles **R<sub>D</sub>** son:

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

En cada matriz se ha asociado la fila  $i$  al proceso  $P_i$  ( $i = 1, 2, 3,$  y  $4$ ) y la columna  $j$  al recurso  $R_j$  ( $j = 1, 2, 3, 4$  y  $5$ ). 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.

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.

N

**Aviso 3:** Solución del examen y fecha de revisión en <http://www.uned.es/71902048/>

4. Supóngase que un determinado sistema operativo asigna cinco marcos de página para la ejecución de un determinado proceso. Además utiliza el **algoritmo de reemplazamiento del reloj** y para implementarlo utiliza una lista enlazada o cola circular de cinco entradas. Cada entrada contiene el número de una página del proceso cargada en memoria. En la Figura 1 se muestra el estado de la cola circular en el instante de tiempo  $t_0$  y un puntero a una página de la cola. Además se muestra el estado del bit referenciada  $r$  de la tabla de páginas del proceso para cada página de la cola. Supóngase que en el instante de tiempo  $t_0$  una referencia a la página  $i = 7$  produce un fallo de página. Se pide:

- (1 p) Explicar el funcionamiento del algoritmo de reemplazamiento del reloj.
- (0.5 p) Determinar **razonadamente** la página que sería seleccionada para ser reemplazada al aplicar este algoritmo.
- (0.5 p) Dibujar el estado final de la cola circular y del puntero tras aplicarse el algoritmo y realizarse el reemplazamiento.

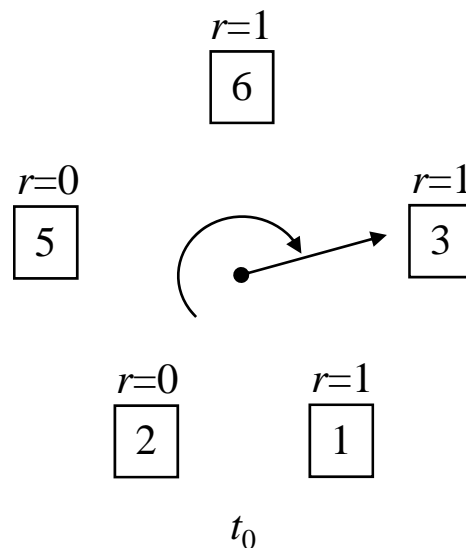


Figura 1

## SISTEMAS OPERATIVOS (Cód. 71902048)

### Solución Examen Septiembre 2022

#### Solución Ejercicio 1

- D) La mayoría de los procesadores disponen de dos modos de ejecución: *modo núcleo o supervisor* y *modo usuario*. Un programa ejecutándose en *modo núcleo* puede ejecutar cualquier instrucción del repertorio de instrucciones del procesador y, en consecuencia, acceder a todas las características del hardware. Por el contrario, un programa ejecutándose en *modo usuario* solo puede ejecutar un conjunto limitado de instrucciones del repertorio del procesador, con lo que tiene limitado el acceso al hardware. Típicamente las instrucciones privilegiadas no se pueden ejecutar en modo usuario, solo en modo supervisor.

El sistema operativo siempre se ejecuta en modo núcleo lo cual le da un acceso completo al hardware. Por su parte, los programas de los usuarios (programas creados por el propio usuario, programas de aplicación o programas de utilidades distribuidos junto con el sistema operativo) se ejecutan en modo usuario y su acceso al hardware está restringido. Para poder usar los recursos hardware, los programas de usuario deben solicitar su uso al sistema operativo mediante la realización de *llamadas al sistema*, que conforman la interfaz entre los programas de usuario y el núcleo. Por lo tanto, las *llamadas al sistema* son el mecanismo usado por los programas de los usuarios para solicitar los servicios del sistema operativo.

- III) La *paginación* es una técnica de gestión de memoria que consiste en dividir la memoria principal en bloques del mismo tamaño  $S_p$  denominados *marcos de página* o *páginas físicas*. El espacio de direcciones de un proceso también se divide en bloques del mismo tamaño  $S_p$  denominados *páginas* o *páginas lógicas*. Una página de un proceso se carga en un marco de página libre de memoria principal. Además las páginas de un mismo proceso no tienen porque ocupar marcos contiguos.

En la paginación simple para que un proceso X pueda ser ejecutado es necesario que todas sus páginas se encuentren cargadas en memoria principal, aunque no es necesario que estén cargadas en marcos contiguos.

- IV) Un *nodo índice* o *nodo-i* es una estructura de datos utilizada por algunos sistemas operativos para almacenar los atributos de un archivo y la ubicación del archivo en disco. Cada nodo-i tiene asociado un número entero positivo que lo identifica de manera única.

Los nodos-i de un determinado sistema de archivos se almacenan en bloques de datos del dispositivo de memoria secundaria donde se encuentra alojado dicho sistema de archivos, para localizarlos se utiliza una lista de nodos-i. Una entrada de la lista de nodos-i contiene la dirección física o número de bloque del disco donde se encuentra el nodo-i asociado a un determinado archivo.

## Solución Ejercicio 2

El funcionamiento de las operaciones `wait_sem` y `signal_sem` cuando se aplican sobre un *semáforo general*  $S$  es el siguiente:

- `wait_sem(S)`. Esta operación disminuye en una unidad el valor del semáforo, es decir,  $S=S-1$ . Si el valor resultante es un número negativo entonces el proceso que ha invocado esta operación es añadido a la cola de procesos bloqueados asociada al semáforo y se bloquea.
- `signal_sem(S)`. Esta operación incrementa en una unidad el valor del semáforo, es decir,  $S=S+1$ . Si el valor resultante es menor o igual a 0 entonces se elimina de la cola asociada al semáforo uno de los procesos bloqueados, y se le despierta, lo que hace que pase al estado preparado para ejecución.

En el caso de un *semáforo binario*  $S$  las operaciones `wait_sem` y `signal_sem` funcionan de la siguiente manera:

- `wait_sem(S)`. Esta operación comprueba el valor del semáforo. Si  $S=0$ , entonces el proceso es colocado en la cola de procesos bloqueados asociada al semáforo y se bloquea. Si  $S=1$ , entonces pone el semáforo a 0 y el proceso puede continuar su ejecución.
- `signal_sem(S)`. Esta operación comprueba si la cola de procesos bloqueados asociada al semáforo  $S$  está vacía. En caso afirmativo, pone el semáforo a 1, y continúa su ejecución. En caso negativo, es decir, hay procesos bloqueados en el semáforo, entonces el sistema operativo elimina de la cola asociada al semáforo a uno de los procesos bloqueados, y le despierta lo que hace que pase al estado preparado para ejecución.

## Solución Ejercicio 3

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

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

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  $(0 \ 0 \ 0 \ 0 \ 0)$ . Como no existe ninguna, no se marca ningún proceso.
2. Se realiza la asignación

$$\mathbf{X} = \mathbf{R}_D = ( 0 \quad 1 \quad 0 \quad 2 \quad 1 )$$

3. Para cada proceso  $P_i$  no marcado, en este caso los cuatro procesos, se comprueba la condición

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

3.1) Para la primera fila de  $\mathbf{M}$  asociada al proceso  $P_1$ :

$$(1 \ 1 \ 0 \ 2 \ 1) \leq (0 \ 1 \ 0 \ 2 \ 1)$$

Si nos fijamos en el primer elemento de los vectores se observa que la condición no se cumple. Luego el proceso  $P_1$  no se marca.

3.2) Para la segunda fila de  $\mathbf{M}$  asociada al proceso  $P_2$ :

$$(0 \ 1 \ 0 \ 2 \ 1) \leq (0 \ 1 \ 0 \ 2 \ 1)$$

se observa que la condición si se cumple. Luego se marca el proceso  $P_2$ .

3.3) Para la tercera fila de  $\mathbf{M}$  asociada al proceso  $P_3$ :

$$(0 \ 2 \ 0 \ 3 \ 1) \leq (0 \ 1 \ 0 \ 2 \ 1)$$

Si nos fijamos en el segundo elemento de los vectores se observa que la condición no se cumple. Luego el proceso  $P_3$  no se marca.

3.4) Para la cuarta fila de  $\mathbf{M}$  asociada al proceso  $P_4$ :

$$(0 \ 2 \ 1 \ 1 \ 0) \leq (0 \ 1 \ 0 \ 2 \ 1)$$

Si nos fijamos en el segundo elemento de los vectores se observa que la condición no se cumple. Luego el proceso  $P_4$  no se marca.

4. Se supone que el proceso marcado  $P_2$  se ejecuta hasta su finalización y que luego se liberan sus recursos. Hay que realizar la siguiente actualización:

$$\mathbf{X} = \mathbf{X} + \mathbf{A}_2 = (0 \ 1 \ 0 \ 2 \ 1) + (0 \ 1 \ 0 \ 1 \ 0) = (0 \ 2 \ 0 \ 3 \ 1)$$

5. Para cada proceso  $P_i$  no marcado, en este caso  $P_1$ ,  $P_2$  y  $P_4$ , se comprueba la condición

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

5.1) Para la primera fila de  $\mathbf{M}$  asociada al proceso  $P_1$ :

$$(1 \ 1 \ 0 \ 2 \ 1) \leq (0 \ 2 \ 0 \ 3 \ 1)$$

Si nos fijamos en el primer elemento de los vectores se observa que la condición no se cumple. Luego el proceso  $P_1$  no se marca.

5.2) Para la tercera fila de  $\mathbf{M}$  asociada al proceso  $P_3$ :

$$(0 \ 2 \ 0 \ 3 \ 1) \leq (0 \ 2 \ 0 \ 3 \ 1)$$

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

5.3) Para la cuarta fila de  $\mathbf{M}$  asociada al proceso  $P_4$ :

$$(0 \ 2 \ 1 \ 1 \ 0) \leq (0 \ 2 \ 0 \ 3 \ 1)$$

Si nos fijamos en el tercer elemento de los vectores se observa que la condición no se cumple. Luego el proceso  $P_4$  no se marca.

6. Se supone que el proceso marcado  $P_3$  se ejecuta hasta su finalización y que luego se liberan sus recursos. Hay que realizar la siguiente actualización:

$$\mathbf{X} = \mathbf{X} + \mathbf{A}_3 = (0 \ 2 \ 0 \ 3 \ 1) + (0 \ 0 \ 0 \ 0 \ 1) = (0 \ 2 \ 0 \ 3 \ 2)$$

7. Para cada proceso  $P_i$  no marcado, en este caso  $P_1$  y  $P_4$ , se comprueba la condición

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

- 7.1) Para la primera fila de  $\mathbf{M}$  asociada al proceso  $P_1$ :

$$(1 \ 1 \ 0 \ 2 \ 1) \leq (0 \ 2 \ 0 \ 3 \ 2)$$

Si nos fijamos en el primer elemento de los vectores se observa que la condición no se cumple. Luego el proceso  $P_1$  no se marca.

- 7.2) Para la cuarta fila de  $\mathbf{M}$  asociada al proceso  $P_4$ :

$$(0 \ 2 \ 1 \ 1 \ 0) \leq (0 \ 2 \ 0 \ 3 \ 2)$$

Si nos fijamos en el tercer elemento de los vectores se observa que la condición no se cumple. Luego el proceso  $P_4$  no se marca.

8. Como no hay ningún proceso marcado el algoritmo finaliza.

Al terminar el algoritmo han quedado los procesos  $P_1$  y  $P_4$  sin marcar, luego **existe interbloqueo**. Los procesos  $P_1$  y  $P_4$  están bloqueados.

### Solución Ejercicio 4

- a) El *algoritmo del reloj* es una implementación del *algoritmo de reemplazamiento de la segunda oportunidad* que utiliza una cola circular y un puntero, los números de páginas contenidos en la cola circular se pueden visualizar situados sobre un círculo como si fueran las horas de un reloj y el puntero se asemejaría a una manecilla. Este algoritmo busca la página que lleva más tiempo cargada en memoria y no ha sido referenciada recientemente.

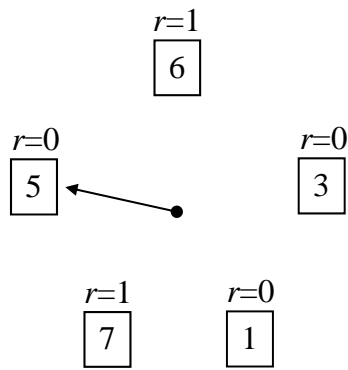
El funcionamiento del algoritmo del reloj es el siguiente: en primer lugar consulta el bit referenciada ( $r$ ) de la página cuyo número  $i$  se encuentra en la entrada de la cola circular apuntada por el puntero. Si  $r = 0$ , entonces la página  $i$  es seleccionada para ser reemplazada por la página  $j$ . En la cola circular la entrada que contenía el número de página  $i$  es sustituido por el número de página  $j$ , el puntero pasa a apuntar a la siguiente entrada de la cola y el algoritmo finaliza. Por el contrario, si  $r = 1$  entonces el algoritmo pone el bit a 0 y el puntero pasa a apuntar a la siguiente entrada (por ejemplo en sentido horario) de la cola circular. A continuación pasa a consultar el bit referenciada de la página cuyo número se encuentra en la entrada de la cola circular apuntada por el puntero. Y así sucesivamente hasta encontrar una página cuyo bit referenciada esté a cero, que será la página  $i$  elegida para ser reemplazada por la página  $j$  que ocasionó el fallo de página.

Nótese que en el caso de que todas las páginas de la cola tengan su bit referenciada a 1, el puntero daría una vuelta completa y volvería a apuntar a la primera entrada que analizó al principio del algoritmo, ahora la página cuyo número se encuentra en dicha entrada tendría su bit referenciada a cero por lo que sería la página escogida.

b) Como la página  $i = 3$  tiene  $r = 1$ , significa que hace poco que fue referenciada, su bit  $r$  se pone a 0 y el puntero pasa a apuntar a la siguiente página. Como la página  $i = 1$  tiene  $r = 1$ , su bit  $r$  se pone a 0 y el puntero pasa a apuntar a la siguiente página. Como la página  $i = 2$  tiene  $r = 0$ , se elige para ser reemplazada. En la posición de la cola ocupada por  $i = 2$  se coloca  $i = 7$  con su bit  $r = 1$ , ya que acaba de ser referenciada. El puntero pasa a apuntar al siguiente número de página de la cola circular.

Luego la **página  $i=2$**  es la elegida para ser reemplazada.

c) En la Figura 1 se muestra el estado de la cola circular y la posición del puntero tras realizarse el reemplazamiento de la página  $i = 2$  por la página  $i = 7$ .



**Figura 1**