

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.
N2	Aviso 3: Solución del examen y fecha de revisión en http://www.uned.es/71902048/

1. Explique **razonadamente** si las siguientes afirmaciones son verdaderas o falsas:

- I) (1 p) Un cambio de proceso solo se puede producir si existe un proceso B en el estado preparado de mayor prioridad que el proceso A que se encuentra actualmente en ejecución.
- II) (1 p) Para que se produzca un interbloqueo únicamente y necesariamente se deben cumplir las siguientes tres condiciones: exclusión mutua, retención y espera, y espera circular.
- III) (1 p) La implementación de directorios con entradas de tamaño variable no genera fragmentación externa a nivel de directorio.
- IV) (1 p) La matriz de acceso es un objeto dinámico del sistema operativo que contiene las contraseñas de los usuarios dados de alta en el sistema.
2. (2 p) Describir el funcionamiento de las operaciones `wait_mon` y `signal_mon` de un monitor. Indicar las diferencias que tienen estas operaciones con respecto a las operaciones `wait_sem` y `signal_sem` de un semáforo. Explicar qué sucede con un proceso que invoca una operación `signal_mon` según la solución de Hansen.
3. En un computador con un único procesador el sistema operativo debe planificar para ejecución el conjunto de procesos que se muestra en la siguiente tabla:

Proceso	Tiempo de llegada (ut)	Tiempo de servicio (ut)
A	0	5
B	1	4
C	2	6
D	3	5

El planificador del sistema operativo implementa un *algoritmo de turno rotatorio* con un cuanto de 3 ut y un tiempo de cambio de proceso despreciable. Si el cuanto del 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 preparados para ejecución antes que el proceso que termina su cuanto. Se pide:

- a) (1 p) Dibujar el diagrama de uso de la CPU.
- b) (1 p) Determinar el tiempo de retorno y el tiempo de espera de cada proceso.

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.
N2	Aviso 3: Solución del examen y fecha de revisión en http://www.uned.es/71902048/

4. (2 p) En un sistema x86-64 bits la unidad direccionable es un byte y se suelen usar 48 bits para las direcciones lógicas. Supuesto que la memoria principal tiene un tamaño de 8 GiB y un tamaño de página de 1 KiB:
- a) (1 p) Determinar el tamaño en bits de cada uno de los campos en que se descompone una dirección física y una dirección lógica.
 - b) (1 p) Supuesto que una entrada de una tabla de páginas tiene un tamaño de 6 bytes, determinar el tamaño que tendría una tabla de páginas completa.

SISTEMAS OPERATIVOS (Cód. 71902048)

Solución Examen Febrero 2022

Solución Ejercicio 1

- I) Esta afirmación es **FALSA** ya que un *cambio de proceso* o *cambio de contexto* se puede producir por diferentes causas, no solo por la que se indica en la afirmación. Otras causas que motivan un cambio de proceso son las siguientes:
- *El proceso en ejecución pasa al estado bloqueado.* Un proceso A pasa al estado bloqueado cuando debe esperar por la aparición de algún evento. Mientras llega a producirse dicho evento se puede ejecutar otro proceso B.
 - *La terminación (voluntaria o forzada) del proceso en ejecución.* Obviamente si un proceso termina se puede pasar a ejecutar otro.
 - *El proceso A en ejecución ha excedido el tiempo máximo de ejecución ininterrumpida.* En los sistemas de tiempo compartido, a cada proceso se le asigna un tiempo máximo de ejecución ininterrumpida con objeto de poder atender las peticiones de todos los usuarios conectados al sistema. Superado dicho tiempo se produce un cambio de proceso, el proceso A en ejecución pasa al estado preparado y un proceso B en dicho estado pasa al estado ejecutándose.
- II) Para que se produzca un interbloqueo se deben cumplir necesariamente las siguientes cuatro condiciones:
1. *Exclusión mutua.* Cada instancia de un recurso solo puede ser asignada a un proceso como máximo.
 2. *Retención y espera.* Cada proceso retiene los recursos que le han sido asignado mientras espera por la adquisición de los otros recursos que necesita.
 3. *No existencia de expropiación.* Si un proceso posee un recurso, éste no se le puede expropiar.
 4. *Espera circular.* Existe una cadena circular de dos o más procesos, de tal forma que cada proceso de la cadena se encuentra esperando por un recurso retenido por el siguiente proceso de la cadena.

Por lo tanto, esta afirmación es **FALSA**.

- III) En la implementación de directorios con entradas de tamaño variable cada entrada de un directorio tiene un determinado tamaño. En cada entrada primero se almacena el tamaño que ocupa la entrada. A continuación se almacenan las informaciones asociadas al archivo (atributos o número de nodo-i) y finalmente se almacena el nombre del archivo que no puede superar un cierto tamaño máximo. Al nombre del archivo se le añade un carácter especial, por ejemplo un 0, para marcar el final del nombre. Además se añaden algunos caracteres de relleno para que el tamaño de una entrada sea un número entero positivo de palabras de memoria principal. De este modo se evita que dentro de una misma palabra haya información sobre dos entradas de directorios distintas, lo que complicaría la administración de los directorios.

La principal ventaja de esta implementación es que se reduce el espacio desperdiciado dentro de un directorio, como máximo el tamaño desperdiciado por entrada puede ser de casi una palabra. Por otra parte un inconveniente de esta organización es que cuando se eliminan archivos de un directorio se crean huecos de longitud variable, lo que puede generar un problema de fragmentación externa a nivel de directorio, es decir, que exista espacio en el directorio pero cuando se desea añadir una nueva entrada no se pueda hacer porque ningún hueco tiene el tamaño suficiente.

Por lo tanto, esta afirmación es **FALSA**.

IV) Un sistema informático dispone de un conjunto de recursos hardware y software, también denominados *objetos*, que son utilizados por los procesos de los usuarios. Cada objeto posee un identificador o nombre que lo distingue de los demás. Estos objetos son utilizados por un conjunto de procesos, también denominados *sujetos*.

Un *dominio de protección* (o simplemente *dominio*) establece para un subconjunto de objetos las operaciones permitidas o derechos de acceso sobre cada uno de ellos. De esta forma un dominio queda definido por un conjunto de pares de la forma

[objeto, derechos].

Un mismo objeto puede estar en varios dominios simultáneamente con los mismos o con distintos derechos de acceso.

La *matriz de acceso* o *matriz de protección* A es un objeto dinámico del sistema operativo que contiene todos los dominios de protección existentes en el sistema. Cada fila i de la matriz está asociada a un dominio D_i . Mientras que cada columna j de la matriz está asociada a un objeto O_j . En definitiva, cada elemento A_{ij} de la matriz de acceso contiene los derechos de acceso asociados al objeto O_j dentro del dominio D_i .

Por lo tanto, esta afirmación es **FALSA**.

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

a) En la Figura 1 se muestra el diagrama de uso de la CPU.

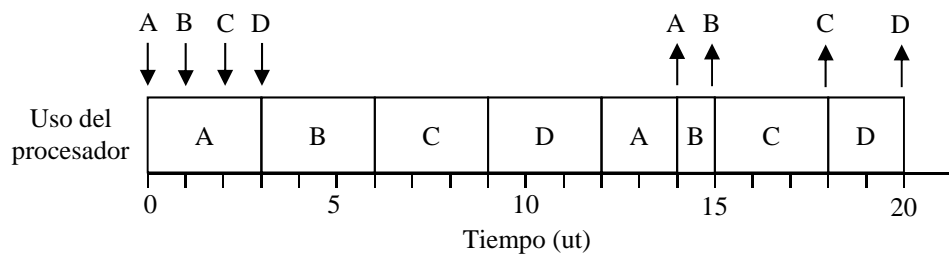


Figura 1

b) En Tabla 1 se muestran los tiempos de llegada T_{LL} , finalización T_F , retorno T_R , servicio T_S y espera T_E de cada trabajo.

Trabajos	T_{LL} (ut)	T_F (ut)	$T_R = T_F - T_{LL}$	T_S (ut)	$T_E = T_R - T_S$
A	0	14	14	5	9
B	1	15	14	4	10
C	2	18	16	6	10
D	3	20	17	5	12

Tabla 1

Solución Ejercicio 4

- a) Cuando se utiliza la técnica de paginación una dirección física se descompone en los campos número de marco de página de f bits y desplazamiento dentro del marco de d bits. Por su parte una dirección lógica se descompone en los campos número de página de p bits y desplazamiento dentro de la página de d bits.

- *Dirección Física:*

Para obtener el tamaño del campo número de marco de página se va a calcular en primer lugar el número de marcos N_{MP} en que se descompone la memoria principal si se considera un tamaño de página $S_P = 1 \text{ KiB}$:

$$N_{MP} = \text{floor}\left(\frac{C_{MP}}{S_P}\right) = \text{floor}\left(\frac{8 \text{ GiB}}{1 \text{ KiB}}\right) = \text{floor}\left(\frac{2^{33}}{2^{10}}\right) = 2^{23} \text{ marcos}$$

Conocido N_{MP} , el tamaño de este campo se obtiene resolviendo la siguiente desigualdad:

$$\min_f \{N_{MP} \leq 2^f\}$$

Luego como $N_{MP} = 2^{23}$ entonces $f = 23$ bits.

Para obtener el tamaño del campo desplazamiento, puesto que la unidad direccionable es un byte, hay que resolver la siguiente desigualdad:

$$\min_d \{S_P \leq 2^d\}$$

Luego como $S_P = 2^{10}$ entonces $d = 10$ bits.

Marco j	Desplazamiento
$f = 23$ bits	$d = 10$ bits

- *Dirección lógica:*

El tamaño del campo desplazamiento de una dirección lógica es igual al de una dirección física, en este caso, $d = 10$ bits. Por otra parte, en el enunciado se indica que una dirección lógica tiene un tamaño de 48 bits, luego la suma del tamaño p del campo número de página y del tamaño d del campo desplazamiento debe ser igual a 48 bits:

$$p + d = 48$$

Luego $p = 48 - d = 48 - 10 = 38$ bits.

Página i	Desplazamiento
$p = 38$ bits	$d = 10$ bits

- b) La unidad direccionable es un byte y se suelen usar 48 bits para las direcciones lógicas. En ese caso, el espacio de direcciones lógicas de un proceso puede tener un tamaño máximo de

$$C_X = 2^{48} \text{ B} = 256 \text{ TiB}$$

Supuesto un proceso de este tamaño, si el tamaño de página es $S_P = 1 \text{ KiB} = 2^{10} \text{ B}$, entonces el número de páginas N_P en que se divide el espacio de direcciones lógicas de este proceso es:

$$N_P = \text{ceil}\left(\frac{C_X}{S_P}\right) = \text{ceil}\left(\frac{2^{48}}{2^{10}}\right) = \frac{2^{48}}{2^{10}} = 2^{38} \text{ páginas}$$

La tabla de páginas de este proceso tiene una entrada por cada página, es decir, 2^{38} entradas. Como cada entrada de la tabla de páginas ocupa $S_E = 6$ bytes, entonces la tabla de páginas completa tendría un tamaño de

$$C_{TP} = S_E \times N_P = 6 \times 2^{38} = 6 \times 2^8 \times 2^{30} = 1536 \text{ GiB}$$

Nótese que como la memoria principal tiene un tamaño de $C_{MP} = 8 \text{ GiB}$, entonces la tabla de páginas del proceso no podría almacenarse en la memoria principal.