

RESPUESTAS EXAMEN FUNDAMENTOS DEL SOFTWARE DGIIM

1. Al igual que en un sistema multiprogramado, en un sistema monoprogramado es necesaria una planificación de la CPU.

Falso. Al ser monoprogramado solo va a poder llevar a cabo el procesamiento de un programa, por lo que no requiere reparto de recursos ni nada de lo visto en multiprogramación, pues la CPU solo se dedica a ese programa en concreto.

2. La gestión de memoria utilizando paginación tiene el problema de:

a. Fragmentación interna. Esto se debe a que si la memoria está particionada de forma fija (las páginas tienen tamaño fijo) y el bloque de datos que se carga es menor que la partición, se desperdicia memoria.

3. Cuando un proceso hace una llamada al sistema, se produce necesariamente un cambio de modo.

Verdadero. Una llamada al sistema siempre supone una comunicación con el SO, y toda aquella operación en la que está involucrado el SO requiere pasar a modo superusuario/kernel.

4. Un sistema que permita el acceso simultáneo de varios usuarios es siempre multiusuario.

Verdadero. Esa es la propia definición de un SO multiusuario.

5. Una palabra de memoria tiene una longitud de:

d. Las tres respuestas anteriores son correctas ya que se tratan de múltiplos de 2.

6. Modificar las posiciones de memoria donde se almacena la Tabla de Vectores de Interrupción solo se puede realizar en modo supervisor.

Verdadero. En la Tabla de Vectores de Interrupción encontramos direcciones de interrupciones que atañen a otros programas, y cualquier acción que pueda alterar el correcto funcionamiento de otros programas debe hacerse en modo supervisor.

7. El despachador (o dispatcher) es código del SO que se encarga de realizar el cambio de modo.

Falso. El dispatcher se encarga del cambio de contexto ya que hace que sea posible el cambio de un proceso a otro, pero el cambio de modo tiene que ver con instrucciones de procesos que pueden ser peligrosas para el sistema.

8. En el PCB o Bloque de Control de Proceso se guarda una copia de los valores que contenían los registros de la CPU la última vez que se ejecutó el proceso.

Verdadero. En el cambio de contexto realizado por el dispatcher el primer paso que siempre se da antes de cambiar los registros de la CPU por los del PCB del proceso al que se va a cambiar es guardar los actuales del PCB del proceso que se va a dejar en standby.

9. Dado el esquema de un computador elemental según se ha descrito en el tema, el puntero de pila (SP) indica:

c. La dirección de memoria donde se encuentra la dirección donde debe saltar el programa después de ejecutarse la instrucción de retorno al finalizar una función o procedimiento. En ejercicios como el 9 de la relación del tema 1 se ve claramente con la instrucción CALL.

10. El grado de multiprogramación es un valor variable en un determinado sistema de computación.

Falso. La multiprogramación es una característica de un SO, la cual posee todo SO que sea capaz de controlar y llevar a cabo varios procesos simultáneamente. No hay un sistema que sea más multiprogramado que otro.

11. Las distintas hebras de un proceso se pueden comunicar entre sí mediante memoria compartida.

Verdadero. Las hebras son las particiones de un proceso y por lo tanto residen en un mismo espacio de memoria y tienen acceso a los mismos datos al compartir PCB y un espacio de direcciones de usuario.

12. Cuando el S.O. detiene un proceso ejecutándose porque le va a asignar la CPU a otro, no es necesario salvar los valores actuales de los registros de la CPU porque ya se han ido salvando adecuadamente durante la ejecución de las instrucciones máquina por las que ha ido pasando la ejecución del proceso.

Falso. Dicha acción supondría un cambio de contexto, y para ello siempre se guardan los registros del proceso que se estaba ejecutando en su PCB, por lo que no estaban salvados durante la ejecución. Es decir, el PCB no se actualiza hasta que un proceso se detiene por alguna razón o es desasignado de la CPU.

13. Es imprescindible asignarle memoria principal a un programa para que este se pueda ejecutar.

Verdadero. En los ejercicios 5 y 9 de la relación siempre vamos pasando de una instrucción a otra de un programa porque se encuentran en direcciones de memoria determinadas.

14. El programa bash, intérprete de órdenes de Linux, puede ejecutar todo el repertorio de instrucciones de la CPU.

Falso. No forma parte del SO precisamente con la intención de que no pueda ejecutar órdenes en modo kernel, ya que esto sería un peligro (esto se vio en el ejercicio 11 de la relación del tema 2).

15. Cuando se realiza un cambio de modo, el sistema operativo salva los contenidos de los registros del procesador en el PCB del proceso que está activo.

Falso. Se guardan como mínimo el PC y la palabra de estado (PSW), pero no en el PCB ya que en cuanto se vuelva a modo usuario se va a proseguir con el proceso que estaba en ejecución, es decir, no hay un cambio de contexto entre un proceso y otro.

16. Cuando un controlador produce una interrupción se produce necesariamente un cambio de contexto.

17. En un sistema monoprogramado no es necesario controlar la protección de memoria ya que sólo hay un proceso cargado en memoria.

Falso. Siempre debe haber una protección de memoria para todo programa.

18. ¿Cómo se consigue proteger el acceso a la memoria que es parte de otro proceso o del sistema operativo? Justifícalo.

Se protege mediante los modos de ejecución del procesador: usuario y kernel. Si se accede a la memoria de otro programa es necesario hacerlo en modo kernel, por ello la mayoría de programas no pueden acceder a la memoria de los otros ya que se trata de una acción privilegiada que requieren modo kernel. De hecho, si un programa intenta acceder a la memoria de otro, es abortado y se finaliza.

19. Si el espacio de direcciones de memoria es de 32 MB y la longitud de palabra es 16 bits, ¿cuántos bits se necesitan para acceder a cada palabra?

La palabra estará en una posición de memoria, y $32\text{MB} = 2^5 * 2^{20} = 2^{25}$ bytes, por lo que para codificar la posición de memoria serán necesarios 25 bits. Como son 16 bits por palabra, lo cual es igual a 2 bytes, $2^{25}/2 = 2^{24}$ bytes \rightarrow 24 bits se necesitan para acceder a cada palabra.

20. Suponga que un proceso emite una dirección lógica igual a 32530 y que se utiliza la técnica de paginación, con páginas de 2048 palabras ¿Es posible que dicha dirección lógica se traduzca en la dirección física 9300? ¿y en la dirección física 7954? Razónelo.

$$32530/2048 = 15 \text{ Resto} = 1810 = \text{Desplazamiento}$$

Dir.física: $N^{\circ}\text{marco} * \text{Tamaño página} + \text{Desp.}$

$$N^{\circ}\text{marco} * \text{Tamaño de página} = 7490 \rightarrow N^{\circ}\text{marco} = 7490/2^{11} = 3.65722 \text{ (NO)}$$

$$N^{\circ}\text{marco} * \text{Tamaño de página} = 6144 \rightarrow N^{\circ}\text{marco} = 6144/2^{11} = 3 \text{ (SÍ)}$$

21. Suponed un sistema operativo que utiliza paginación como mecanismo de gestión de memoria , y un proceso que tiene la siguiente tabla de páginas donde todos los números son decimales, la numeración comienza en todos los casos desde cero, y todas las direcciones de memoria son direcciones en bytes. El tamaño de página es de 2048 bytes. Calcula las direcciones físicas correspondientes a las siguientes direcciones lógicas: a) 1500 b) 11146 c) 6000.

$$1500/2048 = 0 \text{ Resto} = 1500 \text{ (La página 0 se corresponde con el marco 6)}$$

$$\text{Dir.física} = \text{Marco} * \text{Tamaño pág.} + \text{Desp.} = 6 * 2048 + 1500 = 13788$$

$$11146/2048 = 5 \text{ Resto} = 906 \text{ (La página 5 no existe, no se puede pasar a dirección física)}$$

$$6000/2048 = 2 \text{ Resto} = 1904 \text{ (La página 2 se corresponde con el marco 1)}$$

$$\text{Dir.física} = \text{Marco} * \text{Tamaño pág.} + \text{Desp.} = 1 * 2048 + 1904 = 3952$$

Suponed un sistema operativo que utiliza **paginación** como mecanismo de gestión de memoria, y un proceso que tiene la siguiente tabla de páginas donde todos los números son decimales, la numeración comienza en todos los casos desde cero, y todas las direcciones de memoria son direcciones en bytes. El tamaño de página es de **2048 bytes**. Calcula las direcciones físicas correspondientes a las siguientes direcciones lógicas a) 1500 b) 11146 c) 6000. Justificad las respuestas con las operaciones necesarias y las conclusiones.

Número de página Número de marco de página

0	6
1	17
2	1
3	0
4	100

a) $1500/2048 = 0$, Resto = Desplazamiento = 1500

Como la dirección 1500 se halla en la página 0, y esta se corresponde con el marco de página 6, empleamos la fórmula de las direcciones físicas vista en clase:

$\text{Dir.física} = \text{N}^\circ \text{Marco} \cdot \text{Tamaño página} + \text{Desplazamiento} = 6 \cdot 2048 + 1500 = 13788$

b) $11146/2048 = 5$, Resto = Desplazamiento = 906

La página 5 no existe en este SO, por lo que no podemos realizar la traducción a dirección física que se nos pide.

c) $6000/2048 = 2$, Resto = Desplazamiento = 1904

Como la dirección 6000 se halla en la página 2, y esta se corresponde con el marco de página 1, empleamos la fórmula de las direcciones físicas vista en clase:

$\text{Dir.física} = \text{N}^\circ \text{Marco} \cdot \text{Tamaño página} + \text{Desplazamiento} = 1 \cdot 2048 + 1904 = 3952$

Los otros dos ejercicios fueron uno de la ejecución de un programa usando registros (típico ejercicio de TOC) y un ejercicio teórico aleatorio. En mi caso me cayó la diferencia entre el bloque de control de un proceso y de una hebra.