

Centro Asociado Cadiz.

Francisco Javier de la Fuente Rodriguez

DNI, 48969959-S

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

- . I) (1 p) Una de las principales ventajas de los enlaces simbólicos en comparación con los enlaces duros es que ocupan menos espacio en disco.

Falso. Una de las desventajas de los enlaces simbólicos frente a los duros es que estos primero consumen más espacio que los enlaces duros, por que los enlaces duros solo ocupan solo entrada de directorios

- . II) (1 p) La paginación simple permite compartir entre varios procesos un código común.

Verdadero. A este código se le denomina código recurrente y se mantiene en memoria principal. Es un código que no se puede modificar. Y solo se pueden compartir en modo lectura

- . III) (1 p) La capacidad de un disco duro disminuye cuando se formatea a bajo nivel.

Verdadero. Una vez que se realiza el formateo a bajo nivel se pierde cerca de un 20% debido a las siguientes características.

- 1- Cabezera
- 2- Área de datos
- 3- Código de recuperación de errores.

- . IV) (1 p) Uno de los principales inconvenientes del uso de la memoria virtual es que disminuye el grado de multiprogramación del sistema.

Verdadero. Debido a las técnicas de gestión de memoria, particionamiento fijo, particionamiento dinámico, paginación simple y segmentación simple, el grado de multiprogramación se ve reducido, ya que el programa debe estar cargado totalmente en memoria principal.

2. (2 p) Un cierto sistema de archivos utiliza un tamaño de bloque de 16 bytes y su área de datos consta de 256 bloques. La asignación de espacio se realiza mediante el método de asignación indexada. Además en el nodo-i asociado a un archivo, entre otros datos, se almacenan las direcciones físicas de los ocho primeros bloques de datos del archivo y la dirección física de un bloque de indirección simple. Calcular el tamaño máximo en bytes que puede tener un archivo en este sistema de archivos.

$$256 = 2 \text{ elevado } 8$$

8 bits para direcciones

$$128/8 = 16 \text{ direcciones físicas}$$

$$8 + 16 = 24 \text{ bits}$$

$$24/8 = 3$$

$$16 - 3 = 13$$

en cada bloque se puede almacenar un tamaño o máximo de 13 bytes

3. (2 p) Supóngase un computador que utiliza como unidad direccionable la palabra y un tamaño de palabra de 16 bits. En un cierto instante de tiempo el sistema operativo pasa al controlador de DMA los siguientes datos para que realice una operación de DMA:

- Tipo de operación sobre la memoria: lectura.
- Dirección de memoria inicial: AA0016.
- Número de bytes a leer: 1000.
- Dispositivo (destino): disco 1.

Determinar la dirección (expresada en hexadecimal) de la última palabra que se lee en la memoria principal.

La última dirección leída será la dirección máxima

Paso los Bytes a Bits

$$1000 \times 8 = 8000 \text{ Bits}$$

Divido la totalidad entre el tamaño de palabra

$$8000/16 = 500 \text{ bits}$$

Paso los Bits a Bytes

$$500 \text{ bits} = 62 \text{ Bytes}$$

Paso los Bytes a Hexadecimal

$$62 \text{ Bytes} = 3E$$

Hago la suma en Hexadecimal, la dirección inicial más lo anterior.

$$3E + AA00 = AA3E, \text{ última posición a leer.}$$

4. (2 p) Un cierto sistema operativo gestiona la memoria principal usando la técnica de paginación simple con un tamaño de página de 4 KiB. La memoria principal del computador tiene una capacidad de 200 KiB. Para determinar qué marcos de la memoria están libres y cuáles están ocupados el sistema operativo mantiene un mapa de bits, si un bit está a 1 el marco correspondiente está ocupado si está a 0 el marco está libre. En un cierto instante de tiempo T el estado del mapa de bits es el siguiente:

1010111111111111111111111111111100100000101010010001000010000

El primer bit del mapa (el situado más a la izquierda) está asociado al primer marco ($j = 0$). La asignación de marcos se realiza inspeccionando el mapa de bits de forma secuencial comenzando desde el bit asociado al primer marco. Si en el instante de tiempo T se requiere cargar en la memoria principal un proceso A de 50 KiB:

- . a) Determinar cuántas entradas tiene la tabla de páginas del proceso A y cuál será su contenido cuando se haya cargado el proceso en memoria.
- . b) Determinar el tipo de fragmentación que se produce y su magnitud.
- . c) El estado del mapa de bits después de haber cargado en la memoria el proceso A.

a)

Si el proceso es de 50 KiB y cada marco puede tener 4 KiB, dividimos 50/4 para saber la cantidad de marcos que necesita para ser cargado en memoria

$$50/4 = 12 \text{ Necesita para ser cargado en memoria principal.}$$

