

# Parcial-2-S.pdf



**mariiapt02**



**Sistemas Operativos**

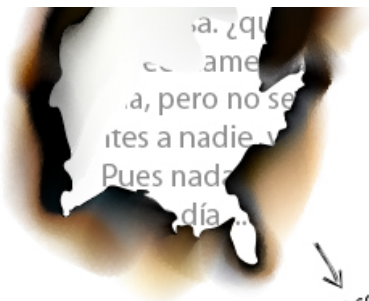


**2º Grado en Ingeniería de Computadores**



**Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga**

NO  
QUEMES  
TUS  
APUNTES



si juegas con fuego  
te fuegas

**GANAR  
0,25 €**

por subir tus apuntes  
en PDF a Wuolah

QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME



## PREGUNTAS BLOQUE 2 S.OP

### TEORÍA

Todas estas preguntas están en:

<https://www.daypo.com/s-operativos-bq2.html#test>

1.El cilindro de un disco magnético consta de:

- a) Todas las pistas ubicadas a diferentes alturas que haya como consecuencia de los múltiples cabezales que posee el brazo del disco
- b) Tantos platos paralelos ubicados a diferentes alturas como cabezales tiene el brazo del disco
- c) Todos los platos y cabezales que hay en cada plano horizontal del disco
- d) Tantas pistas paralelas ubicadas a diferentes alturas como sectores tiene el brazo del disco

2.¿Dónde se almacena la tabla de páginas del sistema operativo?

- a) En la TLB (Translation Look-Aside Buffer)
- b) En el espacio de memoria principal (DRAM)
- c) En la MMU (Memory Management Unit)
- d) En el espacio de disco (memoria secundaria)

3.El modelo del conjunto de trabajo (working set) persigue como objetivo primordial

- a) Mantener en memoria física un número de páginas crítico para cada proceso que evite su congestión en el uso de la memoria
- b) Lograr que las páginas de memoria de un proceso tarden menos en llegar a disco
- c) Optimizar la organización del sistema de ficheros
- d) Todas las otras respuestas son correctas

4.El espacio de direcciones de un proceso se compone de áreas o segmentos destinados a almacenar

- a) El nombre del proceso padre que lo generó (0 en su defecto)
- b) El código de su programa, los datos de su programa, la pila y el heap
- c) Los metadatos del usuario y los metadatos del kernel
- d) El estado de los registros de la CPU, el estado del sistema operativo y el estado del hardware

5.La tabla de páginas invertida del Sistema Operativo en un sistema de memoria virtual tiene una entrada por cada

- a) Dirección lógica de memoria
- b) Página lógica de memoria virtual
- c) Marco de página (o frame) de memoria principal
- d) Dirección física de memoria

6.En un sistema de memoria con direcciones virtuales de 16 bits montado sobre una memoria física de 4 páginas de 8 palabras de un byte, ¿Cuánto valen las longitudes de los campos p para el direccionamiento de la página lógica, f para el direccionamiento de la página física y d para el desplazamiento de la dirección dentro de la página?

- a) p=16, f=2, d=3
- b) p=13, f=2, d=3
- c) p=13, f=3, d=2
- d) p=16, f=3, d=2



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

**7. La elección de un tamaño pequeño para el sector de disco favorece**

- a) El rendimiento del sistema de ficheros
- b) Un menor número de accesos a los sectores de disco para leer cada fichero
- c) Un menor número de metadatos
- d) Un mejor aprovechamiento del espacio de disco**

**8. La jerarquía organizativa de un disco contempla, sucesivamente,**

- a) Directorios, ficheros, sectores y cilindros
- b) Directorios, particiones, ficheros y pistas
- c) Cabezales, motores, pistas y sectores
- d) Particiones, directorios, ficheros y sectores.**

**9. Si un proceso no crea al menos 4 hilos**

- a) No podrá aprovechar él sólo toda la potencia de una CPU de 4GHz
- b) No podrá aprovechar él sólo toda la potencia de una placa base con 4 CPUs
- c) No podrá aprovechar él sólo toda la potencia de una CPU de 4 cores**
- d) No podrá aprovechar él sólo toda la potencia de una memoria quad-channel (de 4 canales)

**10. Cuando un proceso padre crea un proceso hijo**

- a) El hijo amplía el espacio de direcciones del padre
- b) El hijo hereda el espacio de direcciones del padre, copiándolo**
- c) El hijo genera su propio espacio de direcciones en blanco
- d) El hijo reduce el espacio de direcciones del padre

**11. En un sistema de memoria con paginación, la dirección de memoria se descompone en los campos p (página) y d (desplazamiento). ¿Qué relación guardan las longitudes de estos dos campos?**

- a) El campo p siempre es más grande que el campo d
- b) El campo d siempre es más grande que el campo p
- c) Ninguna de las otras tres respuestas es correcta**
- d) El campo p debe ser un múltiplo del campo d

**12. Las señales en el sistema operativo Unix se envían**

- a) Desde el sistema operativo a sus procesos
- b) Desde un proceso a otro proceso
- c) Desde el sistema operativo a sus procesos y desde un proceso a otro proceso**
- d) Desde el hardware al sistema operativo

**13. Una FAT32 nos indica**

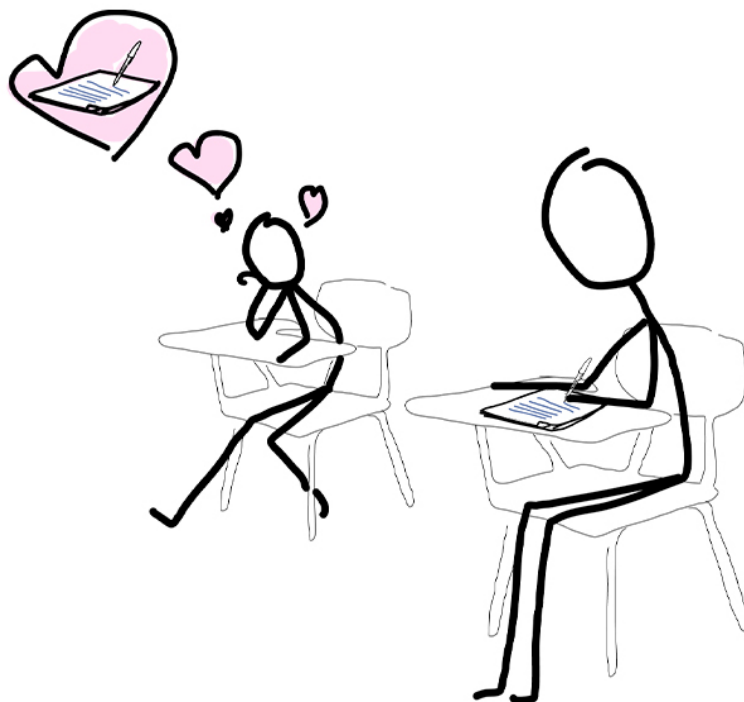
- a) Que la tabla correspondiente a la FAT se implementa en un Sistema Operativo de 32 bits
- b) Que el disco podrá tener como máximo 4 Giga-clusters**
- c) Que las pistas del disco pueden tener un máximo de 32 clusters
- d) Que los clusters del disco pueden ser de un máximo de 32 bytes

**14. ¿Cuántas entradas (o número de filas) tiene una TLB (Translation Look-Aside Buffer)?**

- a) Tantas como palabras de memoria quepan dentro de una página
- b) Tantas como marcos o páginas de memoria física
- c) Tantas como páginas lógicas de memoria virtual
- d) El límite está impuesto por el coste de su implementación**

# QUIERES CONSEGUIR 15€ ??

TRÁENOS A TU  
**CRUSH DE APUNTES**  
ANTES DE QUE  
LOS QUEME



si consigues que suba apuntes, te llevas 15€ + 5  
Wuolah Coins para los sorteos

**WUOLAH**

**15. La función de la TLB (Translation Look-Aside Buffer) consiste en**

- a) Traducir las direcciones lógicas de memoria a direcciones físicas.
- b) Acelerar el acceso a las instrucciones más recientemente utilizadas, manteniendo en la caché aquellas que más utiliza el programa
- c) Acelerar la traducción de las direcciones lógicas de memoria a direcciones físicas, manteniendo en una caché las traducciones que se han realizado más recientemente**
- d) Acelerar el acceso a los datos más recientemente utilizados, manteniendo en la caché aquellos que son más utilizados por el programa.

**16. El swapping se produce**

- a) Siempre en los sistemas multiprocesador
- b) Cuando un proceso requiere más memoria física de la que hay disponible**
- c) En cuanto tengamos una unidad de disco instalada en el sistema
- d) Cada vez que el contenido de una dirección virtual es solicitado desde un programa

**17. La tabla de páginas del Sistema Operativo en un sistema de memoria virtual nos dice**

- a) La página lógica que contiene una página física
- b) La página física que contiene una página lógica
- c) La dirección lógica que contiene una dirección física
- d) La dirección física que contiene una dirección lógica**

**18. Sea un espacio de direcciones lógico de 1024 páginas de 4 bytes cada una, sobre una memoria física de 32 bytes direccionable a nivel de byte en la que conviven 3 procesos de 32 bytes cada una, representados respectivamente por las direcciones A0,...,A31,B0,...,B31,...,C0,...,C31. Si el algoritmo de reemplazo es FIFO, se pide determinar las direcciones de los datos alojados en los 32 bytes de memoria física cuando se solicita la secuencia de direcciones lógicas A0, A8, A16, A24, B0, B8, B16, B24, C0, C8, C16, C24.**

- a) B0 a B3, B8 a B11, B16 a B19, B24 a B27, C0 a C3, C8 a C11, C16 a C19 y C24 a C27.**
- b) La segunda mitad de los procesos B y C (esto es, B16 a B31 y C16 a C31), y no queda ninguna posición vacía
- c) Ninguna de las otras tres respuestas es correcta
- d) C0 a C31 (esto es, el proceso C completo), y no queda ninguna posición vacía

**19. A diferencia de un proceso, un hilo (thread)**

- a) No necesita apropiarse de recursos, ya que hereda los que utiliza el proceso que lo creó**
- b) No necesita llevar un registro del contador de programa, ya que utiliza el del proceso padre
- c) No necesita salvar el contexto cuando detiene su ejecución
- d) Ninguna respuesta es correcta

**20. ¿Cuál es el espacio máximo ocupado en disco por una FAT32?**

- a) 16Kbytes
- b) 1 Gbyte
- c) 16 Mbytes
- d) 16 Gbytes**

**21. En la gestión de memoria, el proceso de compactación de memoria se realiza para reducir**

- a) La fragmentación interna
- b) La fragmentación externa**
- c) La fragmentación interna y la fragmentación externa
- d) La fragmentación interna o la fragmentación externa

22. Sea un espacio de direcciones lógico de 1024 páginas de 8 bytes cada una, sobre una memoria física de 32 bytes direccionable a nivel de byte en la que conviven 3 procesos de 32 bytes cada uno, representados respectivamente por las direcciones A0,...,A31,B0,...,B31,...,C0,...,C31. Si el algoritmo de reemplazo es FIFO, se pide determinar las direcciones de los datos alojados en los 32 bytes de memoria física cuando se solicita la secuencia de direcciones lógicas A0, A4, A8, A12, A16, A20, A24, A28, B0, B4, B8, B12, B16, B20, C0, C4, C8, C12, C16, C20, C24, C28.

a) La segunda mitad de los procesos B y C (esto es, B16 a B31 y C16 a C31), y no queda ninguna posición vacía

b) Ninguna de las otras tres respuestas es correcta.

c) A0, A4, A8, A12, A16, A20, A24, A28, B0, B4, B8, B12, B16, B20, C0, C4, C8, C12, C16, C20, C24, C28 y el resto de posiciones quedan vacías.

d) C0 a C31 (esto es, el proceso C al completo), y no queda ninguna posición vacía

23. En el proceso de traducción de dirección virtual a física, ¿puede la TLB tener una anchura inferior a la tabla de páginas?

a) Sí, cuando la tabla de páginas es multinivel

b) Sí, cuando el tamaño del marco de página es mayor que el tamaño de la página física

c) Sí, y no es necesario ningún prerrequisito para que esto sea posible

d) No, en ningún caso

24. ¿Puede acceder un usuario a los metadatos que contiene un disco magnético?

a) No

b) Sí, de forma directa

c) Sí, a través del compilador

d) Sí, a través de llamadas al sistema operativo

25. Sea un espacio de direcciones lógico de 1024 páginas de 4 bytes cada una, sobre una memoria física de 64 bytes direccionable a nivel de byte en la que conviven 3 procesos de 32 bytes cada uno, representados respectivamente por las direcciones A0,...,A31,B0,...,B31,C0,...,C31. Si el algoritmo de reemplazo es FIFO, se pide determinar las direcciones lógicas A0,A8,A16,A24,B0,B8,B16,B24,C0,C8,C16,C24.

a) La segunda mitad de los procesos B y C (esto es, B16 a B31 y C16 a C31), y no queda ninguna posición vacía

b) B0 a B3, B8 a B11, B16 a B19, B24 a B27, C0 a C3, C8 a C11, C16 a C19 y C24 a C27

c) C0 a C31 (proceso C completo), y no queda ninguna posición vacía

d) Hay espacio en memoria física para todas las páginas referenciadas. Si la memoria estuviera inicialmente vacía, sobrarían 16 bytes y no sería necesario reemplazar ninguna página.

26. Formateamos un disco duro de 4 Terabytes con una sola partición de i-nodos Linux que usa clusters de 1Kbyte. ¿Cuántos punteros o índices a clusters contiene el i-nodo de un fichero que ocupa 1 Gbyte?

a) Es necesario utilizar hasta el puntero indirecto triple

b) Es necesario utilizar hasta el puntero indirecto doble

c) Con los 10 punteros directos del i-nodo es suficiente para direccionar al fichero de 1 Gbyte

d) Es necesario utilizar hasta el puntero indirecto simple

27. En un esquema de memoria virtual. ¿Cuándo se genera la dirección física que corresponde a una dirección lógica?

a) Durante la ejecución del programa

b) Cuando se está depurando el programa

c) En tiempo de compilación

d) En el momento en el que se carga el programa en memoria



QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME



28.El swapping de un proceso conlleva su traslado

- a) De memoria secundaria (disco) a memoria principal (DRAM)
- b) De memoria principal (DRAM) a memoria secundaria (disco)**
- c) De memoria principal (DRAM) a memoria caché (SRAM)
- d) De memoria principal (DRAM) a los registros del procesador (CPU)

29.Un sistema operativo lleva a cabo la compactación de la memoria para reducir

- a) La fragmentación interna
- b) La fragmentación externa**
- c) Ni la fragmentación externa ni la fragmentación interna
- d) Tanto la fragmentación externa como la interna

30.Cuando paginamos la memoria

- a) El espacio de direcciones lógico de los procesos ya no es consecutivo
- b) El espacio de direcciones físico de los procesos ya no es consecutivo
- c) Ni el espacio de direcciones lógico ni el espacio de direcciones físico de los procesos es consecutivo**
- d) Ninguna de las otras respuestas es correcta

31.En un sistema de memoria paginado de 4 Gbytes con páginas de 4 Kbytes y palabras de memoria de 4 bytes, la dirección de memoria se descompone en los campos p (página) y d (desplazamiento). ¿Qué longitudes tienen estos dos campos?

- a) p tiene 22bits y d tiene 10bits
- b) p tiene 20bits y d tiene 10bits
- c) p tiene 20bits y d tiene 12 bits**
- d) p tiene 22 bits y d tiene 12 bits

32.Diferentes estrategias para registrar la memoria que está sin usar en el sistema son

- a) Mapa de bits
- b) Listas enlazadas
- c) Índices
- d) Las otras tres respuestas son correctas**

33.La función de la TLB (Translation Look-Aside Buffer) consiste en

- a) Acelerar el acceso a los datos más recientemente utilizados, manteniendo en la caché aquellos que son más utilizados por el programa
- b) Traducir las direcciones lógicas de memoria a direcciones físicas
- c) Acelerar el acceso a las instrucciones más recientemente utilizadas, manteniendo en la caché aquellas que más utiliza el programa
- d) Acelerar la traducción de las direcciones lógicas de memoria a direcciones físicas, manteniendo en una caché las traducciones que se han realizado más recientemente**

34.¿Dónde se almacena la tabla de páginas del sistema operativo?

- a) En el espacio de disco (memoria secundaria)
- b) En la TLB (Translation Look-Aside Buffer)
- c) En el espacio de memoria principal (DRAM)**
- d) En la MMU (Memory Management Unit)

35.¿Qué es LRU?

- a) Un esquema de protección de memoria
- b) El nombre de una popular implementación de un sistema de ficheros
- c) Un algoritmo para el reemplazo de páginas en un sistema de memoria virtual**
- d) Una caché para las traducciones más recientes de memoria virtual a física



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

**36.El fenómeno de thrashing se produce cuando**

- a)Un proceso consume la mayor parte de su tiempo interacmbiando páginas entre memoria virtual y física en lugar de avanzando en su ejecución**
- b)El grado de multiprocesamiento compartiendo páginas es bajo
- c)Una página de memoria física se sustituye por otra procedente de disco
- d)Un fallo de página no puede ser atendido por el sistema operativo

**37.¿Quién lleva a cabo la gestión del espacio libre en el disco?**

- a)El usuario
- b)El sistema operativo**
- c)El Master Boot Record (MBR)
- d)El superusuario o usuario root

**38.La elección de un tamaño pequeño para el sector de disco favorece**

- a)Un menor número de accesos a los sectores de disco para leer cada fichero
- b)Un mejor aprovechamiento del espacio de disco**
- c)Un menor número de metadatos
- d)El rendimiento del sistema de ficheros

**39.¿Qué valores de la variable status y ground guardan una mayor relación? (esto es, un proceso tiene una mayor probabilidad en coincidir en ellos)**

- a)status=REANUDADO y ground=PRIMERPLANO
- b)status=FINALIZADO y ground=DETENIDO
- c)status=SUSPENDIDO y ground=DETENIDO**
- d)status=SUSPENDIDO y ground=SEGUNDOPLANO

**40.Por PTBR (Page Table Base Register) se entiende:**

- a)Un registro ubicado en la MMU que apunta al comienzo de la tabla de páginas del proceso**
- b)En realidad es el registro puntero de pila del programa (SP)
- c)Un tipo específico de MMU
- d)En realidad es el registro contador del programa (PC)

**41.En ausencia de TLB, el fallo de página dará lugar a:**

- a)Un reset de la CPU
- b)Nada en particular; es un evento que el sistema suele ignorar
- c)Una interrupción o excepción**
- d)Un bug de sistema

**42.Si el tamaño de página es de 4KB y el número de marcos de página en memoria principal es 256K marcos ¿cuántas entradas podemos afirmar que tiene la TLB?**

- a)4 entradas
- b)24 entradas
- c)El tamaño de la TLB no tiene ninguna relación con la información proporcionada en esta cuestión**
- d)14 entradas

**43.Se referencia la siguiente secuencia de páginas lógicas: pag. 3, pag. 5, pag. 6, pag. 7, pag. 6, pag. 7, pag. 5, pag. 6, pag. 8. La cardinalidad (número de elemtnos) máxima del “working set” tomando una ventana de 5 referencias es:**

- a)3
- b)6
- c)4**
- d)5



**45. La traducción inversa de páginas permite:**

- a) En realidad no aporta ninguna ventaja con respecto a la traducción convencional directa
- b) Tener una mayor capacidad de direccionamiento (memorias de mayor capacidad)
- c) Reducir la cantidad de memoria ocupada para gestionar la traducción de direcciones**
- d) Reducir considerablemente el tiempo de acceso a memoria

**46. Si la dirección física es de 38 bits, la dirección lógica de 52 bits y el tamaño de página 16KB, el número de marcos en memoria principal es:**

- a)  $2^{14}$
- b)  $2^{24}$**
- c)  $2^{16}$
- d)  $2^{38}$

**47. Si el tamaño de página es 16KB, el número de bits del campo desplazamiento de la dirección lógica (offset) tendrá un tamaño:**

- a) 12 bits
- b) 10 bits
- c) 14 bits**
- d) 16 bits

**48. Un sistema multiprogramado con asignación de particiones variables (residente, inmóvil, contiguo y entero) gestiona el espacio libre con vectores de bits. Si existen 256 bloques y el vector de ocupación tiene el valor hexadecimal 0x00...09012, el algoritmo "Worst-Fit" asignará:**

- a) Un hueco de tamaño 4 bloques
- b) Un hueco de tamaño 3 bloques
- c) Un hueco de tamaño mayor a 4 bloques**
- d) Un hueco de tamaño 2 bloques

**49. Con el cambio de contexto:**

- a) Se ha de notificar a la MMU la tabla de páginas que debe usar (la del proceso entrante)**
- b) Se mantiene una lista de marcos libres
- c) Se resetean todos los macros de página ocupados
- d) Se resetean las tablas de página

**50. Toda página de un proceso residente en un marco de memoria física tendrá activo necesariamente:**

- a) El bit de válido (V)
- b) El bit de presencia (P)**
- c) El bit de modificación (M)
- d) El bit de escritura

**51. En caso de fallo de página, el proceso que provoca el fallo:**

- a) Está zombie durante el swap-in/out
- b) Está running durante el swap-in/out
- c) Está bloqueado durante el swap-in/out**
- d) Está suspendido durante el swap-in/out

**52. ¿Qué tipo de fragmentación sufre la paginación de memoria?**

- a) Media
- b) Interna**
- c) La memoria paginada carece por completo de fragmentación
- d) Externa

**53. La paginación y segmentación**

- a) Son modelos de memoria contiguos
- b) Son modelos de memoria no-contiguos**
- c) Son modelos de memoria monoprogramados
- d) Son modelos de memoria inmóvil

**54. La política de asignación "Next-Fit", en un sistema de particiones de tamaño variable, es una política derivada de la asignación:**

- a) "Mid-Fit"
- b) "First-Fit"**
- c) "Best-Fit"
- d) "Worst-Fit"

**55. De los siguientes algoritmos de reemplazo de páginas, ¿qué opción aprovecha mejor la localidad temporal?**

- a) Random
- b) FINUFO
- c) FIFO
- d) LRU**

**56. En un modelo multiprogramado con asignación de particiones (residente, inmóvil, contiguo y entero), ¿qué estrategia de asignación presentará una mayor posibilidad de fragmentación interna?**

- a) Particiones de tamaño variable
- b) Particiones de tamaño fijo con particiones grandes**
- c) Particiones de tamaño fijo con particiones pequeñas

**57. En un sistema de memoria paginado, si se disminuye el tamaño de página, manteniendo igual los tamaños de los espacios físico y lógico, aumentará:**

- a) El tamaño de las entradas de la tabla de página, pero no su número
- b) Nada, la tabla de páginas mantendrá su tamaño inicial
- c) El número de entradas de la tabla de páginas, pero cada entrada será del mismo tamaño
- d) El número de entradas de la tabla de páginas, y también el tamaño de cada entrada**

**58. El cargador (loader):**

- a) Resuelve las referencias intramodulo
- b) Asigna direcciones iniciales a los segmentos de programa**
- c) Resuelve las referencias intermodulo
- d) Traduce direcciones lógicas a físicas

**59. De las siguientes opciones, ¿cuál no es un requerimiento hardware imprescindible de un sistema de memoria virtual?**

- a) La unidad de punto flotante (FPU)**
- b) Bits específicos en la tabla de páginas, como el bit de presencia
- c) El mecanismo de interrupción de fallo de página
- d) La zona de swap en el disco duro (memoria secundaria)

**60. ¿Qué modelo de memoria, de los siguientes, hace uso de la memoria secundaria?**

- a) El modelo entero
- b) El modelo móvil
- c) El modelo no contiguo
- d) El modelo no residente**

QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME



61. En un modelo de memoria multiprogramado, los límites permitidos del espacio de memoria de un proceso están almacenados en:

- a) La FPU
- b) El PCB**
- c) El PC
- d) La pila (SP)

62. El acceso aleatorio a un fichero en disco ...

- a) es más rápido en asignación enlazada que en contigua
- b) es más rápido en asignación contigua que en enlazada**
- c) es más rápido en asignación enlazada que en indexada

63. ¿Qué son los atributos de un fichero?

- a) metadatos asociados a cada fichero**
- b) los ficheros no tienen atributos, solo los directorios
- c) metadatos asociados a la partición en la que reside el fichero
- d) son los tres últimos caracteres del nombre del fichero que vienen precedidos por un punto.

64. Un cluster ...

- a) siempre es varios ordenes de magnitud mayor que un sector
- b) siempre está formado por un solo sector
- c) puede estar formado por 1 o más sectores**
- d) en ocasiones puede ser más pequeño que un sector (para minimizar la fragmentación interna)

65. El número de bytes por entrada de la tabla de asignación de archivos en FAT16 es:

- a) 2 bytes**
- b) 16 bytes
- c) 8 bytes
- d) 4 bytes

66. En un HD con 64 sectores por pista rotando a 2500 r.p.m, sin mover el cabezal lector, podremos leer a lo sumo:

- a) 160000 sectores por minuto**
- b) 39(1/16) sectores por minuto
- c) 150000 sectores por minuto
- d) 41(2/3) sectores por minuto

67. Un disco duro gira a una velocidad angular constante de 6000 r.p.m. ¿Cuanto tiempo a lo sumo se tardará en encontrar un dato en una pista determinada, una vez que el cabezal ya está ubicado en esa pista?

- a) Depende de la pista en cuestión
- b) 6ms
- c) 1ms
- d) 10ms**

68. Un dispositivo de almacenamiento externo de 256MB de capacidad se formatea en FAT16 con el tamaño mínimo de cluster. ¿Cuál es el tamaño de cada FAT?

- a) 256MB
- b) 256KB
- c) 128KB**
- d) 255MB



si  
consigues  
que suba  
apuntes, te  
llevarás 15€ +  
5 Wuolah  
Coins para  
los sorteos

**69. La tabla de páginas mantiene, para cada proceso:**

- a) **La localización del marco para cada página del proceso**
- b) Ninguna de las anteriores
- c) La localización en memoria física del proceso
- d) La localización de la página para cada marco del proceso

**70. En un sistema de asignación contigua de espacio en disco los tipos posibles de acceso a los ficheros son:**

- a) solo aleatorio
- b) solo secuencial
- c) ninguno de ellos
- d) **secuencial y aleatorio**

**71. ¿Que método de asignación de espacio en disco puede provocar fragmentación interna?**

- a) **asignación enlazada**
- b) asignación contigua
- c) ninguno de ellos
- d) asignación indexada

**72. ¿Qué sistema de ficheros es el más habitual en sistemas operativos tipo Unix?**

- a) FAT (File Allocation Table)
- b) MFT (Master File Table)
- c) NTFS (NT File System)
- d) **Basado en i-nodes**

**73. ¿Cuántas operaciones de disco puede implicar como máximo un fallo de página en un sistema de memoria virtual simple (sin buffering y sin precarga)?**

- a) Ninguna, nunca se opera con el disco a causa de un fallo de página, se genera una excepción o error y se aborta el proceso que la causó
- b) Una, para leer la página que hay que traer de memoria secundaria
- c) Ninguna de las anteriores es correcta
- d) **Dos, si no hay marcos libres y la página a reemplazar ha sido modificada**

**74. En un sistema que utiliza segmentación para la gestión de memoria, un proceso se divide en:**

- a) Un segmento por thread
- b) Ninguna de las anteriores
- c) Un número de segmentos variable que deben ser del mismo tamaño
- d) **Un número de segmentos variable que no tienen que ser del mismo tamaño**

**75. En un sistema que combine la paginación y la segmentación, el espacio de direcciones del usuario se descompone en una serie de:**

- a) Páginas de tamaño fijo, que se descomponen a su vez en segmentos de tamaño variable
- b) Todas las anteriores
- c) **Segmentos de tamaño variable, que se dividen a su vez en páginas de tamaño fijo**
- d) Segmentos o páginas, a elección del programador

**76. Cuando la entrada de la tabla de páginas solicitada no se encuentra en la TLB se dice que se produce:**

- a) Un fallo de caché
- b) **Un fallo de TLB**
- c) Un fallo de página
- d) Un fallo de MMU

**77.El denomida “cilindro” de un disco ya formateado contiene:**

- a)tantas pistas como clusters
- b)tantas pistas como platos
- c)tantas pistas como cabezas**
- d)tantas pistas como sectores

**78.¿Qué tipos de ficheros debe “entender” (conocer su estructura interna) un sistema operativo?**

- a)ningún tipo de fichero
- b)todos los tipos de ficheros
- c)ficheros ejecutables**
- d)ficheros de text

**79.El tipo de memoria que permite una multiprogramación muy efectiva, liberando al usuario de las restricciones ocasionadas por el tamaño de la memoria, se denomina:**

- a)Memoria paginada
- b)Memoria transaccional
- c)Memoria física
- d)Memoria virtual**

**80.En la técnica de gestión de memoria basada en el particionamiento dinámico, la compactación consiste en desplazar los procesos en memoria para que ocupen un bloque contiguo, de forma que toda la memoria libre queda ubicada en un único bloque:**

- a)Verdadero**
- b)Falso

**81.La dirección física de una palabra en memoria se traduce a partir de las siguientes porciones de una dirección virtual:**

- a)Número de página y número de marco
- b)Número de marco y desplazamiento
- c)Ninguna de las anteriores
- d)Número de página y desplazamiento**

**82.Con respecto a FAT12, el formato FAT16 permite:**

- a)Clusters de menor tamaño**
- b)Sectores de menor tamaño
- c)Sectores de tamaño mayor
- d)Clusters de mayor tamaño

**83.La mínima cantidad de información a la que un controlador de disco puede acceder se denomina:**

- a)cilindro
- b)cluster
- c)sector**
- d)pista

**84.¿Cuál de las siguientes ternas son las coordenadas utilizadas para localizar la información en un disco duro a bajo nivel?**

- a)Cabeza, cilindro y sector**
- b)Cluster, pista y cilindro
- c)Plato, cluster y pista
- d)Plato, cilindro y cluster

**85. Un sistema de memoria que utilice paginación puede sufrir, en pequeño grado, el problema de la fragmentación interna, pero no va a sufrir de ninguna fragmentación externa**

- a) Verdadero
- b) Falso

**86. En un sistema que gestiona la memoria utilizando solamente segmentación, el desperdicio de memoria se debe a:**

- a) Ninguna de las anteriores
- b) La fragmentación externa
- c) Segmentos del mismo tamaño
- d) La fragmentación interna

**87. El uso de tablas de página multinivel provoca**

- a) Acelerar el proceso de traducción de direcciones
- b) Ahorrar espacio de memoria consumido por la tabla de páginas
- c) Todas las respuestas son correctas
- d) Simplificar el hardware de la MMU

**88. En relación al sistema de ficheros, por cada proceso existirá:**

- a) Una tabla maestra de archivos (MTF)
- b) Un i-nodo
- c) Una tabla de ficheros abiertos
- d) Una FAT

**89. En un sistema de memoria virtual paginada:**

- a) El tamaño del espacio lógico posible de un proceso puede ser mayor que el espacio físico disponible
- b) El número de operaciones de E/S causadas por cada fallo de página no es siempre el mismo
- c) Las páginas de un proceso no siempre están presentes en la memoria RAM del sistema
- d) Todas las respuestas son correctas

**90. Hemos estudiado como fragmentación:**

- a) El mecanismo que permite dividir el espacio físico de memoria de un sistema procesador en marcos de página
- b) La división en niveles de una tabla de páginas
- c) La pérdida efectiva de memoria provocada por fragmentos no asignables o no utilizables
- d) El mecanismo que permite dividir el espacio lógico de memoria de un proceso en páginas

**91. En Unix, suponga un proceso que ha establecido una máscara de señales usando la función `sigprocmask()` y a continuación invoca `fork()` sin error.**

- a. Se mantiene la misma máscara de señales del proceso PARENT y se establece a un valor por defecto en el proceso CHILD
- b. La máscara de señales de ambos procesos se restablece a un valor por defecto
- c. Se mantiene la misma máscara de señales del proceso PARENT y el proceso CHILD hereda la máscara del proceso PARENT
- d. Se mantiene la misma máscara de señales del proceso PARENT y el proceso CHILD hereda la máscara del proceso PARENT, salvo las señales relacionadas con el terminal (SIGINT, SIGTSTP, SIGPIPE, SIGTTIN, SIGTTOU) que se resetean a valores por defecto.
- e. Ambos procesos comparten los manejadores, si un proceso establece un nuevo manejador de señal el comportamiento del otro proceso también cambiará puesto que comparten código.



QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME



92. En Unix, suponga un proceso que tiene señales pendientes (que están enmascaradas) e invoca `execve(...)` sin error, comenzando a ejecutar el programa indicado en los parámetros de `execve()`.

- a. Las señales pendientes son eliminadas y la máscara de señales es reseteada
- b. Las señales que estaban pendientes seguirán pendientes hasta que se desenmascaren
- c. Las señales pendientes son eliminadas, pero se mantiene la máscara de señales

93. En Unix, suponga un proceso que ha establecido una serie de manejadores de señales usando la función `signal()` y a continuación invoca `execve(...)` sin error, comenzando a ejecutar el programa indicado en los parámetros de `execve()`

- a. Todos los manejadores de señales son reseteados a valores por defecto salvo los que estén ignorados (SIG\_IGN) que seguirán ignorados
- b. Todos los manejadores de señales son reseteados a valores por defecto (SIG\_DFL)
- c. Se mantienen los manejadores de señales establecidos previamente

94. En UNIX, qué valor obtiene en la variable `ret_val` un proceso que ejecuta la sentencia C: `pid_t ret_val = fork();`

- a. Siempre obtiene en la variable `ret_val` el identificador del proceso CHILD generado.
- b. Obtiene en la variable `ret_val` un cero si no hubo error durante la ejecución de `fork()`.
- c. Obtiene en la variable `ret_val` un entero positivo que indica el tipo de error y que puede ser usado por `perror(retval)` para producir un mensaje de error.
- d. Ninguna de la anteriores es cierta

95. En Unix, que valor obtiene en la variable `ret_val` un proceso que ejecuta la sentencia C: `pid_t ret_val = wait();`

- a. En caso de error retorna -1
- b. Cero en caso de no haya ningún cambio en los procesos CHILD
- c. Siempre obtiene el identificador del proceso CHILD que ha terminado.
- d. Cero en el proceso CHILD y el identificador del proceso CHILD en el proceso PARENT, o bien, -1 si hay error.
- e. Cero en el proceso CHILD y el identificador del proceso CHILD en el proceso PARENT

96. La gestión de la creación de threads requiere más tiempo que la gestión de la creación del mismo número de procesos.

- a) Verdadero
- b) Falso

97. Para que la señal SIGCONT llegue a todo un grupo de procesos con `pgrp=2120`, tendremos que utilizar en nuestro programa: Selecciona una o más de una:

- a. `killpg(2120, SIGCONT)`
- b. `sigsend(2120, SIGCONT)`
- c. `kill(-2120, SIGCONT)`
- d. `signal(2120, SIGCONT)`

98. En Unix, suponga un proceso que ha establecido una máscara de señales usando la función `sigprocmask()` y a continuación invoca `execve(...)` sin error, comenzando a ejecutar el programa indicado en los parámetros de `execve()`

- a. Se mantiene la misma máscara de señales del proceso, salvo las relacionadas con la terminal (SIGINT, SIGTSTP, SIGPIPE, SIGTTIN, SIGTTOU) que se desenmascaran
- b. Se mantiene la misma máscara de señales del proceso
- c. La máscara de señales del proceso se restablece a un valor por defecto

99. En la llamada a `pthread_create` uno de los parámetros es el nombre del fichero ejecutable cuyo código va a empezar a ejecutar el thread una vez sea creado

- a) Verdadero
- b) Falso



si  
consigues  
que suba  
apuntes, te  
llevarás 15€ +  
5 Wuolah  
Coins para  
los sorteos

**100. Cuando se pulsa Control+Z mientras se ejecuta una tarea en primer plano, el shell debe suspender a la tarea de primer plano enviándole la señal correspondiente**

- a) Verdadero
- b) Falso

**101. Cual es la forma correcta de enmascarar un conjunto de señales en Unix:**

- a. Hacer la operación de OR bit a bit de las señales, por ejemplo: `procsigmask(SIG_BLOCK, SIGINT | SIGTSTP );`
- b. Hacer la operación de AND bit a bit de las señales, por ejemplo: `procsigmask(SIG_BLOCK, SIGINT & SIGTSTP );`
- c. Hacer la operación de OR lógico de las señales, por ejemplo: `procsigmask(SIG_BLOCK, SIGINT || SIGTSTP );`
- d. Hacer la operación de AND lógico de las señales, por ejemplo: `procsigmask(SIG_BLOCK, SIGINT && SIGTSTP );`
- e. Ninguna de la anteriores

**102. En UNIX, qué valor obtiene en la variable `ret_val` un proceso que ejecuta la sentencia `C: pid_t ret_val = waitpid(-1, &status, WOHANG);`**

- a. Si no hay error, el identificador de un proceso CHILD que ha terminado
- b. Si no hay error, el identificador de un proceso CHILD que se ha reanudado (previamente suspendido)
- c. -1 en caso de error
- d. Si no hay error, el identificador de un proceso CHILD que se ha suspendido (stopped)
- e. Cero si no hay ningún cambio de estado en sus procesos CHILD ni se ha producido un error.

**103. Cuando se crea un proceso CHILD con `fork()`, automáticamente el sistema operativo le asigna su propio grupo de procesos, diferente al del proceso PARENT.**

- a) Verdadero
- b) Falso

**104. En Unix, suponga un proceso que tiene señales pendientes (que están enmascaradas) e invoca `fork()` sin error. Selecciona una:**

- a. Las señales pendientes seguirán pendientes en el proceso PARENT. El proceso CHILD solo mantiene pendientes las señales que el proceso PARENT se envió a sí mismo para dejárselas pendiente al CHILD, ya que esta es la técnica habitual para pasar señales de PARENT a CHILD.
- b. Las señales pendientes son eliminadas del proceso PARENT de forma que ambos procesos están libres de señales pendientes tras `fork()`
- c. Las señales que estaban pendientes seguirán pendientes en ambos procesos.
- d. Las señales pendientes seguirán pendientes en el proceso PARENT, pero el proceso CHILD comienza sin señales pendientes

**105. En Unix, suponga un proceso que ha establecido una serie de manejadores de señales usando la función `signal()` y a continuación invoca `fork()` sin error.**

- a. Todos los manejadores de señales son reseteados a valores por defecto salvo los que estén ignorados (`SIG_IGN`) que seguirán ignorados en el proceso CHILD y se mantienen en el proceso PARENT
- b. Todos los manejadores de señales son reseteados a valores por defecto (`SIG_DFL`) en el proceso CHILD y se mantienen en el proceso PARENT
- c. Todos los manejadores de señales son reseteados a valores por defecto (`SIG_DFL`) en ambos procesos
- d. Se mantienen los manejadores de señales establecidos previamente en ambos procesos

**106. Cuando en nuestro shell ejecutamos: `> cat &`**

- a. Como hemos lanzado a ejecutar `cat` en segundo plano e intenta leer del terminal, se suspende.
- b. `Cat` obtiene el terminal, ya que lo utiliza como entrada estándar
- c. `Cat` comienza a ejecutarse en primer plano
- d. El proceso creado para la ejecución del `cat` finalizará de forma abrupta.

107.Considérese el siguiente fragmento de código C, donde "ls" corresponde con el nombre de un ejecutable correcto, que podemos lanzar sin errores:

```
execlp("ls", "ls", NULL);
```

Si el ejecutable "ls" no estuviera en la ruta de búsqueda (PATH), la llamada a execlp ...: Selecciona una:

- A. intentaría ejecutar el comando "ls" interno al shell
- B. provocaría que el proceso llamante muera inmediatamente
- C. provocaría que el proceso hijo muera inmediatamente
- D. fallaría, esto es, devolvería un código de error

108.En UNIX, qué valor obtiene en la variable ret\_val un proceso que ejecuta la sentencia C: pid\_t ret\_val = waitpid(pid,&status,WHOHANG|WUNTRACED);

- a. Si no hay error, obtendrá el valor pid si el procesocon identificadordpidha terminado
- b. -1 en caso de error**
- c. Si no hay error, obtendrá el valor pid si el procesocon identificadordpidse ha suspendido (stopped)
- d. Cero si no hay ningún cambio de estado en el proceso con identificador pid ni se ha producido un error.
- e. Si no hay error, obtendrá el valor pid si el procesocon identificadordpidse ha reanudado (previamente suspendido)

109.Los threads pertenecientes a un mismo proceso tienen acceso compartido a las variables globales de su proceso, lo que facilita el trabajo colaborativo entre ellos

- a)Verdadero**
- b)Falso

110.¿Qué formato de los siguientes implica un menor grado de fragmentación interna?

- FAT32 con tamaño de cluster 32KB
- FAT16 con tamaño de cluster 64KB
- FAT32 con tamaño de cluster 16KB**
- FAT16 con tamaño de cluster 128 KB

111.A la hora de transferir datos desde un disco duro a memoria principal, ¿cuál de las siguientes latencias es la que más limite la velocidad?

- La latencia asociada a la DMA
- La latencia de rotación
- La latencia de posicionamiento de la cabeza lectora**
- La latencia de acceso a bits consecutivos en un mismo "track"

112.El cluster (también denominado bloque o unidad de asignación) es un múltiple de:

- 512 KB
- La pista
- El cilindro
- El sector**

113.El File Control Block (FCB) es un bloque de datos ubicado en:

- El propio disco**
- El PCB (Process Control Block)
- Un registro del procesador
- El sector de arranque (MBR)

114.En un sistema basado en i-nodes , el nombre del fichero (filename) es un atributo ubicado en:

- El i-nodo
- La FAT
- El superbloque
- El directorio**

**115.En un sistema de ficheros tipo Unix, una entrada de directorio (directory entry) correspondiente a un fichero regular apunta a:**

**El i-nodo del fichero**

El primer bloque de datos del fichero

El PCB del proceso que ha abierto el fichero

El directorio padre

**116.La gestión del espacio libre en FAT32 se realiza:**

En la tabla maestra (MTF)

En el MBR

En el superbloque

**En la propia FAT**

**117.La organización de la tabla de asignación de archivos del formato utilizado por SO de la familia Windows, conocido como vFAT es básicamente:**

FAT12

FAT14

**FAT16**

FAT32

**118.Sobre los i-nodos de un sistema de ficheros tipo Unix, podemos decir:**

Hay muchos más i-nodos usados que ficheros

Hay muchos menos i-nodos usados que ficheros

**Hay tantos i-nodos usados como ficheros**

El número de i-nodos y de ficheros usados no guarda ninguna relación

**119.Un i-nodo de UNIX (4.1) contiene índices indirectos:**

Solo simples

Hasta de dos niveles

**Hasta de tres niveles**

Hasta de cuatro niveles

**120.La FAT es un sistema de asignación:**

Contiguo

**Enlazado**

Indexado

Volátil

**121.En un instante dado, la cardinalidad (número de elementos) del conjunto activo (working set) de un proceso depende:**

**De la localidad del proceso**

Del número de marcos de memoria física

Del algoritmo de reemplazo de paginas

Del tamaño de la zona de swap

**122.La TLB es un buffer de traducción anticipada, que se encuentra en**

**El hardware de la CPU**

La memoria principal

La zona de swap

El PCB

QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME



123. En general, se asume que la latencia de acceso de la zona de swap es:

**Mucho mayor que la de memoria principal**

Mucho menor que la de memoria principal

Del mismo orden aproximadamente que la memoria principal

Depende del tamaño de dicha zona de intercambio

124. La anomalía de Belady es una situación que se presenta en ciertos algoritmos de reemplazo de páginas como:

FIFO y LRU

LRU pero no FIFO

**FIFO pero no LRU**

FINUFO y LRU

125. Con respecto a la tabla de páginas convencional (traducción directa), la tabla de páginas invertida, es una solución que permite:

**Ahorrar espacio de memoria**

Realizar la traducción más rápidamente

Disminuir la tasa de fallo de páginas de un proceso

Facilitar la compartición de memoria entre procesos

126. El hardware de traducción (MMU)

Se encarga de traducir las instrucciones máquina del procesador a microinstrucciones que entiende la memoria

**Traduce las direcciones del espacio lógico de un proceso a direcciones físicas en memoria principal**

Gestiona la memoria libre y asigna memoria a las peticiones de los procesos

Traduce el código objeto de los procesos a código máquina

127. Cuando el procesador invierte la mayor parte de su tiempo haciendo swapping de partes del proceso, en vez de ejecutar sus instrucciones, se dice que se produce:

Metapaginación

Hiperlocalidad

**Trasiego (thrashing)**

Fragmentación

128. La política de reemplazo que escoge solo entre las páginas residentes del proceso que genere el fallo de página, para decidir cuál es la página que va a ser reemplazada se denomina:

Política de reemplazo global

**Política de reemplazo local**

Política de reemplazo dinámica

Política de reemplazo aleatoria

129. ¿Qué algoritmo/s de reemplazo no controla/n la hiperpaginación, si no coopera/n con un algoritmo de control de páginas?

**Reemplazo global, aunque sea el óptimo.**

"Buffering" de páginas

Demonio de paginación

Reemplazo local con algoritmo FINUFO

130. En un gestor de memoria con paginación de 2 niveles sin TLB, con respecto a la ausencia de paginación, en media, los accesos a memoria serán:

2 veces más rápidos

2 veces más lentos

3 veces más rápidos

**3 veces más lentos**



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

**131. A mayor probabilidad de fallo de página**

- Mayor aprovechamiento de la localidad espacial
- Mayor latencia media de los accesos a memoria**
- Tablas de página más reducidas
- Menor latencia media de los accesos a memoria

**132. En un sistema de memoria virtual de 1 Terabyte para cada proceso, que se monta sobre un espacio físico de 1 Gbyte, ¿qué probabilidad hay de que una entrada de la tabla de páginas me devuelva un marco físico de memoria donde se encuentra la página virtual? Considera que el proceso referencia a todas sus direcciones virtuales con igual probabilidad**

- Una entre mil**
- Una entre mil millones
- Una entre diez
- Una entre un millón

**133. Implementaciones populares del sistema de ficheros son**

- Logical Block Addressing (LBA) y Master Boot Record (MBR)
- LILO y GRUB
- FAT e i-nodos**
- Superbloque y Master Boot Record (MBR)

**134. ¿Cuál es el espacio máximo ocupado en disco por una FAT16?**

- 128 KB**
- 1 MB
- 16 KB
- 1 KB

**135. El programa de un usuario utiliza**

- Direcciones lógicas en los accesos a datos y direcciones físicas en los accesos a instrucciones.
- b. Direcciones lógicas en los accesos a instrucciones y direcciones físicas en los accesos a datos.
- c. Siempre direcciones lógicas.**
- d. Siempre direcciones físicas.

**136. ¿Qué es un frame o marco de página?**

- a. Ninguna de las otras tres respuestas es correcta.
- b. Una dirección de memoria (puntero).
- c. Una traza de compilación.
- d. Un bloque de memoria del tamaño de una página.**

**137. En la gestión de memoria aparece el fenómeno de fragmentación interna y externa. ¿Interna y externa a qué? Selecciona una:**

- a. A la CPU.
- b. Al sistema operativo.
- c. A la partición de memoria asignada a cada proceso.**
- d. Al disco que actúa de memoria secundaria.

**138. El cilindro de un disco magnético consta de Selecciona una:**

- a. Tantas pistas paralelas ubicadas a diferentes alturas como cabezales tiene el brazo del disco.**
- b. Tantos sectores ubicados a diferentes alturas como pistas tiene cada brazo del disco.
- c. Tantas pistas paralelas ubicadas a diferentes alturas como sectores tiene cada anillo del disco.
- d. Tantos platos ubicados a diferentes alturas como pistas tiene el brazo del disco.



**139. Una FAT de disco aloja los sectores de forma Selecciona una:**

- a. Consecutiva.
- b. Enlazada.**
- c. Contigua.
- d. Indexada.

**140. Las principales desventajas de un sistema de tablas de páginas multinivel son**

- a. Las otras tres respuestas son correctas.
- b. Desperdiciamos un montón de espacio en memoria para almacenar metadatos.
- c. Se necesitan varios accesos a memoria por cada petición a memoria.**
- d. No pueden implementar ningún mecanismo de seguridad.

**141. Durante el proceso de traducción de dirección virtual a física Selecciona una:**

- a. Primero se consulta a la TLB y luego a la tabla de páginas.**
- b. Se consulta simultáneamente a la TLB y a la tabla de páginas.
- c. Primero se consulta a la tabla de páginas y luego a la TLB.
- d. Nunca se consulta a la TLB si la tabla de páginas está implementada en varios niveles.

**142. Sectores de disco grandes favorecen**

- a. Una reducción de la fragmentación interna del espacio de almacenamiento del disco.
- b. Una reducción de la fragmentación externa del espacio de almacenamiento del disco.
- c. El rápido posicionamiento del cabezal en la superficie del disco donde comenzará la lectura/escritura de datos.
- d. Tasas de transferencia elevadas (ancho de banda)**

**143. En un disco magnético, el sistema de posicionamiento se define por las siguientes coordenadas:**

- a. Pista y sector.
- b. Cilindro, pista, sector y dirección.
- c. Cabezal, plato, cilindro y pista.
- d. Cabezal, plato, cilindro y sector**

**144. En un sistema de paginación multinivel, ¿qué anchura (o número de bits en cada fila de la tabla) tiene cada una de las tablas de páginas que se necesitan para realizar la traducción de dirección virtual a física? (considera el nivel más alto numéricamente como aquel más cercano a los marcos de páginas en que acaba la traducción)**

- a. Depende del tamaño de la memoria principal, las tablas podrían ser cada vez más estrechas o más anchas.
- b. Las tablas van siendo más estrechas a medida que nos acercamos al nivel más alto numéricamente.
- c. Todas las tablas tienen la misma anchura.**
- d. Las tablas van siendo más anchas a medida que nos acercamos al nivel más alto numéricamente.

**145. En una tabla de páginas invertida, se habilita una entrada por cada**

- a. Entrada en la TLB.
- b. Fichero del sistema.
- c. Página virtual alojada en memoria secundaria.
- d. Página real de memoria principal.**

**146. La FAT de disco es una tabla que tiene tantas entradas o filas como**

- a. Clusters de datos tiene el disco.**
- b. Pistas de datos tiene el disco.
- c. Ficheros de datos haya en el disco.
- d. Directorios pueda haber en el disco.

**147. Cuando una página de memoria física se elige para ser reemplazada por otra de memoria lógica o virtual procedente del disco**

- a. Se guarda su contenido en la página lógica que tiene asociada en disco y después se colocan ceros en los bytes que tiene asignados en memoria principal.
- b. Se guarda su contenido en la página lógica que tiene asociada en disco sólo si el bit dirty está activo.**
- c. No se guardan sus contenidos, simplemente son reescritos por aquéllos procedentes de la nueva página lógica que ocupará su lugar en memoria principal.
- d. Se actualiza su contenido en memoria virtual (disco) por los de la página lógica que la reemplaza en memoria física.

**148. ¿Dónde está implementada la TLB en un sistema de memoria virtual?**

- a. En memoria principal (DRAM).
- b. En memoria secundaria (disco).
- c. En los metadatos del proceso en curso.
- d. Dentro de la CPU (por ejemplo, el i7 de Intel).**

**149. Un sistema de memoria virtual tiene 1024 páginas de 8 Kbytes mapeadas sobre una memoria física de 1 Mbyte direccionable a nivel de byte. ¿Qué anchura tiene la tabla de páginas y qué anchura tiene la TLB (Translation Look-Aside Buffer)? (considera la anchura como la longitud en bits de cada fila en cada tabla)**

- a. La tabla de páginas tiene una anchura de 10 bits. La TLB tiene una anchura de 17 bits.
- b. La tabla de páginas tiene una anchura de 23 bits. La TLB tiene una anchura de 20 bits.
- c. La tabla de páginas tiene una anchura de 10 bits. La TLB tiene una anchura de 20 bits.
- d. La tabla de páginas tiene una anchura de 7 bits. La TLB tiene una anchura de 17 bits.**

**150. ¿Dónde es necesario implementar algoritmos de reemplazo?**

- a. En las entradas de la TLB (Translation Look-Aside Buffer).
- b. Ni en los marcos de memoria física ni en las entradas de la TLB.
- c. En los marcos de memoria física y en las entradas de la TLB.**
- d. En los marcos de memoria física.

**151. ¿Puede tener la dirección lógica de memoria virtual una longitud inferior a la dirección física? Selecciona una:**

- a. Sí, aunque sólo en sistemas de paginación multinivel.
- b. Sí, aunque es muy poco habitual.**
- c. Sí, aunque sólo cuando se usa, adicionalmente, una TLB para la traducción de la página lógica a la página física.
- d. No, en ningún caso.

**152. ¿Que método de asignación de espacio en disco puede provocar fragmentación externa?**

- a) asignación enlazada
- b) asignación contigua**
- c) ninguno de ellos
- d) asignación indexada

QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME 🔥



## SHELL

1. ¿Cuántos procesos se crean en el siguiente código? (incluido el proceso inicial)

```
int main() {  
    pid_t pid, pid2;  
    pid = fork();  
    pid2 = fork();  
    if (pid != pid2)  
        fork();  
}
```

7

2. Sabiendo que `sleep(X)` hace esperar a un proceso X segundos, indicar qué palabra escribe en pantalla el siguiente código C

```
void imprime(char *C)  
{  
    printf("%s", C);  
    fflush(stdout);  
}  
  
int main() {  
    pid_t pid, pid2;  
    pid = fork();  
    if (pid > 0) {  
        imprime("X");  
        wait(NULL);  
        fork();  
        imprime("\n");  
    }  
    else {  
        sleep(1);  
        imprime("Z");  
    }  
}
```

- a) XZYY
- b) YZX
- c) YXZZ
- d) XYZ



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

### 3.

Sea un sistema de memoria con direcciones virtuales de 16 bits montado sobre una memoria física de 4 páginas de 8 palabras de un byte, con algoritmo LRU para el reemplazo de páginas. El segmento de código de un programa alojado en esta memoria es el siguiente bucle en lenguaje C:

```
int main()
{
    int i, x[100], y[100], z[100];
    for (i=0; i<100; i++)
        x[i] = i;
    for (i=0; i<100; i++)
        y[i] = i;
    for (i=0; i<100; i++)
        z[i] = i;
}
```

El compilador aloja el contador  $i$  en un registro interno de la CPU y utiliza 2 bytes de memoria por cada entero de los vectores  $x[]$ ,  $y[]$  y  $z[]$ , resultando el siguiente patrón de acceso a las direcciones pares de memoria dentro del segmento de datos del programa ( $A = \text{address}$  - puntero o dirección de memoria):  $Ax0, Ax2, Ax4, \dots, Ax196, Ax198, Ay0, Ay2, Ay4, \dots, Ay196, Ay198, Az0, Az2, Az4, \dots, Az196, Az198$ .

Dado que el programa apenas tiene un par de instrucciones, nos olvidaremos del segmento de código y simplificaremos suponiendo que toda la memoria física se dedica a alojar el segmento de datos del programa, esto es, los vectores  $x[]$ ,  $y[]$  y  $z[]$ , y que ninguna de estas páginas ha sido solicitada previamente, por lo que no existe la posibilidad de encontrarla en memoria física cuando comienza a ejecutarse el programa.

En estas condiciones, se pide indicar cuántas faltas de página se producen durante el acceso a los datos de los vectores  $x[]$ ,  $y[]$ ,  $z[]$  mientras se ejecuta el programa.

75

### 4.

Sea un sistema de memoria con direcciones virtuales de 16 bits montado sobre una memoria física de 8 páginas de 4 palabras de un byte, con algoritmo LRU para el reemplazo de páginas. El segmento de código de un programa alojado en esta memoria es el siguiente bucle en lenguaje C:

```
int main()
{
    int i, x[100], y[100], z[100];
    for (i=0; i<100; i++)
        x[i] = i;
    for (i=99; i>=0; i--)
        y[i] = i;
    for (i=0; i<100; i++)
        z[i] = i;
}
```

El compilador aloja el contador  $i$  en un registro interno de la CPU y utiliza 2 bytes de memoria por cada entero de los vectores  $x[]$ ,  $y[]$  y  $z[]$ , resultando el siguiente patrón de acceso a las direcciones pares de memoria dentro del segmento de datos del programa ( $A = \text{address}$  - puntero o dirección de memoria):  $Ax0, Ax2, \dots, Ax198, Ay198, Ay196, \dots, Ay0, Az0, Az2, \dots, Az198$ .

Dado que el programa apenas tiene un par de instrucciones, nos olvidaremos del segmento de código y simplificaremos suponiendo que toda la memoria física se dedica a alojar el segmento de datos del programa, esto es, los vectores  $x[]$ ,  $y[]$  y  $z[]$ , y que ninguna de estas páginas ha sido solicitada previamente, por lo que no existe la posibilidad de encontrarla en memoria física cuando comienza a ejecutarse el programa.

En estas condiciones, se pide indicar cuántas faltas de página se producen durante el acceso a los datos de los vectores  $x[]$ ,  $y[]$ ,  $z[]$  mientras se ejecuta el programa.

150

## 5.

Sea un sistema de memoria con direcciones virtuales de 16 bits montado sobre una memoria física de 8 páginas de 4 palabras de un byte, con algoritmo LRU para el reemplazo de páginas. El segmento de código de un programa alojado en esta memoria es el siguiente bucle en lenguaje C:

```
int main()
{
    int i, x[100], y[100], z[100];
    for (i=0; i<100; i++)
    {
        x[i] = i;
        y[i] = i;
        z[i] = i;
    }
}
```

El compilador aloja el contador *i* en un registro interno de la CPU y utiliza 2 bytes de memoria por cada entero de los vectores *x*[], *y*[] y *z*[], resultando el siguiente patrón de acceso a las direcciones pares de memoria dentro del segmento de datos del programa (*A* = *address* - puntero o dirección de memoria): *Ax0, Ay0, Az0, Ax2, Ay2, Az2, ..., Ax198, Ay198, Az198*.

Dado que el programa apenas tiene un par de instrucciones, nos olvidaremos del segmento de código y simplificaremos suponiendo que toda la memoria física se dedica a alojar el segmento de datos del programa, esto es, los vectores *x*[], *y*[] y *z*[], y que ninguna de estas páginas ha sido solicitada previamente, por lo que no existe la posibilidad de encontrarla en memoria física cuando comienza a ejecutarse el programa.

En estas condiciones, se pide indicar cuántas faltas de página se producen durante el acceso a los datos de los vectores *x*[], *y*[] y *z*[] mientras se ejecuta el programa.

150

6. Los comandos internos del shell son aquellos interpretados por el shell y que no corresponden a comandos ejecutables. Entre ellos, se encontraría

- a) jobs
- b) clock
- c) ls
- d) cp

7. Si quiero cambiar el aspecto de mi Shell para la recepción de comandos, "COMANDO->", ¿Qué parte del código no debo cambiar?

- a) El manejador de señales
- b) El proceso padre
- c) El proceso hijo
- d) Las otras tres respuestas son correctas

8. Fíjate en los 3 argumentos de las llamadas a `waitpid(pgid, &status, flags)`; ¿por qué `pgid` y `flags` se pasan por valor y `status` se pasa por referencia?

- a) Porque de no ser así, se bloquea la llamada a `waitpid()`, y no nos interesa ese comportamiento para nuestro Shell
- b) Porque `waitpid()` puede recoger señales además de esperar a la finalización de un hijo
- c) Porque `waitpid()` devuelve información en la variable `status`, mientras que los otros dos parámetros le suministran información
- d) Porque `waitpid()` es sensible a multitud de señales

9.

Queremos implementar un comando **verbose** en nuestro Shell que nos desconecte de la terminal de entrada. Desde que teclea "verbose" hasta que teclea "normal", se deberá ignorar cualquier comando que se introduzca desde la terminal. Ejemplo:

```
COMANDO->verbose
COMANDO->clock    (sin efecto)
COMANDO->cp      (sin efecto)
COMANDO->normal
COMANDO->ls      (continúa de forma normal)
/home/usuario/file.txt
```

¿En qué parte del código de tu Shell desarrollarías la implementación para ese comando **verbose**?

- a) En el manejador de señales
- b) En el proceso padre tras la llamada a `fork()`
- c) En el código del `main()`, entre su inicio y la llamada a `fork()`**
- d) En el proceso hijo tras la llamada a `fork()`

10. ¿Qué parte(s) del código del Shell utiliza(n) la llamada a `killpg()` para enviar una señal de reanudación a un proceso detenido?

- a) La implementación del comando `bg`
- b) La implementación del comando `fg`
- c) La implementación de los comandos `fg` y `bg`**
- d) El manejador de señales

11. ¿Qué fase de la construcción del Shell no requiere modificar el código del proceso padre?

- a) La tercera
- b) La segunda
- c) La quinta**
- d) La cuarta

12. ¿Cuál es la diferencia entre el comando `jobs` y el comando `ps`?

- a) `jobs` lista sólo trabajos lanzados desde el shell y `ps` lista procesos en general, incluyendo a los de `jobs`**
- b) `jobs` lista sólo trabajos lanzados desde el shell y `ps` lista procesos en general, excluyendo a los de `jobs`
- c) `ps` lista sólo trabajos lanzados desde el shell y `jobs` lista procesos en general
- d) `ps` lista procesos inactivos y `jobs` lista procesos inactivos

13. En la primera fase de nuestro proyecto Shell, si pulsamos desde el teclado `CTRL+C` se mata tanto al proceso hijo lanzado por el Shell como a éste último. ¿Qué debemos hacer en las fases siguientes para que esto no suceda, y que las señales asociadas a las combinaciones de teclas como `CTRL+Z` y `CTRL+C` le lleguen sólo al proceso lanzado en primer plano?

- a) Utilizar la opción `WUNTRACED` en la llamada a `waitpid()` correspondiente
- b) Instalar un manejador para la señal `SIGCHLD`
- c) Utilizar la macro `ignore_terminal_signals()` proporcionada en las tareas auxiliares
- d) Independizar al nuevo proceso en su propio grupo de procesos y asignarle el terminal**

14. ¿Qué comando interno de tu Shell se implementa de manera más fácil?

- a) `fg`
- b) `jobs`
- c) `bg`
- d) `bg`
- d) `logout`**



QUIERES  
CONSEGUIR  
15€??

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME



15. En un Sistema Operativo Unix, un proceso cuya señal SIGCHLD ha sido atendida por el padre

- a) Finaliza su ejecución de forma normal
- b) Permanece en estado "zombie"
- c) Será resucitado por el padre
- d) Es adoptado por el proceso "init"

16. ¿Cuál de las siguientes llamadas a la función waitpid() no bloquea al proceso que la ejecuta?

- a) `mypid=waitpid(pid,&status,WUNTRACED | WNOHANG);`
- b) `mypid=waitpid(pid,&status,0);`
- c) `mypid=waitpid(pid, &status, WUNTRACED);`
- d) Ninguna de las otras tres respuestas es correcta

17.

Desde el Shell de Linux, utilizamos el comando "vi manulo.txt" para crear un fichero con una sola línea de texto, que contiene los caracteres "Manuel Alcántara". Posteriormente, creamos un enlace soft para manulo.txt y un enlace hard para manulo2.txt, y editamos el archivo manulo2.txt para incorporar una segunda línea de texto que contiene los caracteres "Manuel Bandera". La secuencia completa puede resumirse así:

```
vi manulo.txt (para crear un fichero con una sola línea de texto, "Manuel Alcántara")
ln -s manulo.txt manulo2.txt
ln manulo.txt manulo2.txt
vi manulo2.txt (para añadir una segunda línea de texto, "Manuel Bandera")
```

Si ahora lanzo el comando "vi manulo.txt", ¿Cuántas líneas muestra vi en pantalla?

2

18.

Desde el Shell de Linux, utilizamos el comando "vi manulo.txt" para crear un fichero con una sola línea de texto, que contiene los caracteres "Manuel Alcántara". Posteriormente, creamos un enlace soft para manulo.txt y un enlace hard para manulo2.txt, y editamos el archivo manulo2.txt para incorporar una segunda línea de texto que contiene los caracteres "Manuel Bandera". La secuencia completa puede resumirse así:

```
vi manulo.txt (para crear un fichero con una sola línea de texto, "Manuel Alcántara")
ln -s manulo.txt manulo2.txt
ln manulo.txt manulo2.txt
vi manulo2.txt (para añadir una segunda línea de texto, "Manuel Bandera")
```

Si ahora lanzo el comando "vi manulo.txt", ¿Cuántas líneas muestra vi en pantalla?

2

19.

```
¿Qué muestra en pantalla la ejecución del siguiente programa C?
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
int var_global = 7;
void *do_work(void *null) {
    int i = 21;
    sleep(3);
    var_global = i;
    pthread_exit(NULL);
}
int main(int argc, char *argv[]) {
    int rc;
    pthread_t tid;
    printf("Amanece a las %d ", var_global);
    rc = pthread_create(&tid, NULL, do_work, NULL);
    rc = pthread_join(tid, NULL);
    printf("y anochece a las %d\n", var_global);
}
```

Amanece a las 7 y anochece a las 21



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

WUOLAH

WUOLAH



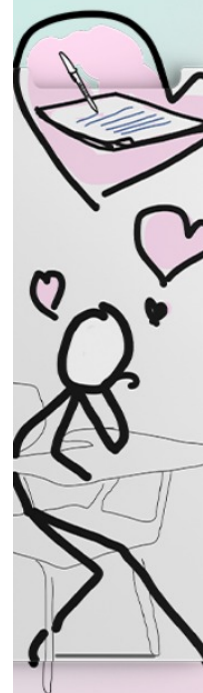




QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME 🔥



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH





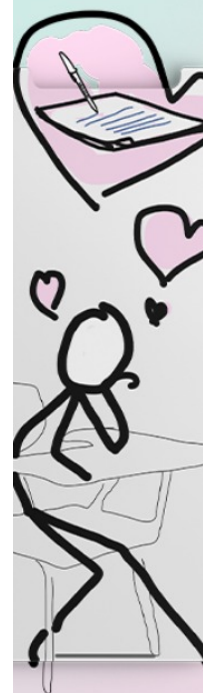




QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME 🔥



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH





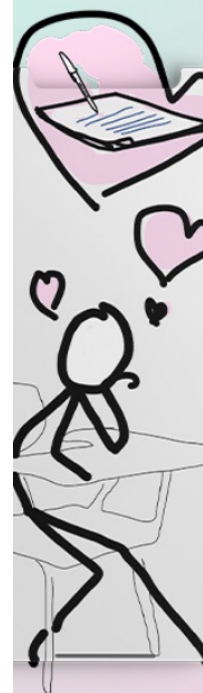




QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME 🔥



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH



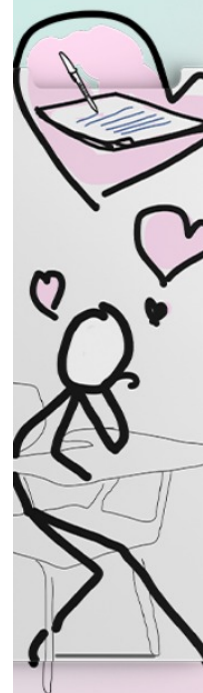




QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME 🔥



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH





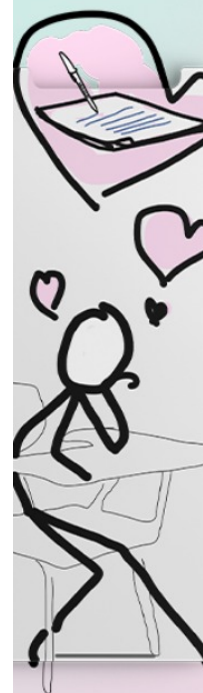




QUIERES  
CONSEGUIR  
15€?? →

TRÁENOS A TU  
CRUSH DE APUNTES

ANTES DE QUE  
LOS QUEME 🔥



si  
consigues  
que suba  
apuntes, te  
llevas 15€ +  
5 Wuolah  
Coins para  
los sorteos

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

