

recopilacion-preguntas-tipo-test...



ingenieriadura



Sistemas Operativos



2º Grado en Ingeniería Informática



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

TEMAS 1 Y 2

¿Cuál de los siguientes NO se considera un requisito hardware esencial en la implementación de un sistema operativo moderno?

Selecciona una:

- a. El sistema de interrupciones
- b. La protección de la memoria
- c. La DMA
- d. **El bus PCI**

A la acción mediante la cual un programa de usuario invoca un servicio del sistema operativo se denomina:

- a. **System call (llamada a sistema)**
- b. Shell script
- c. Batch processing (procesamiento por lotes)
- d. Multiprocesamiento

Básicamente, el sistema operativo es un software que actúa como capa entre:

- a. **El hardware y las aplicaciones de usuario**
- b. Los usuarios y las aplicaciones
- c. El código máquina y el ensamblador
- d. El hardware y el ensamblador

Cuál de los siguientes NO es objetivo de un sistema operativo:

- a. Gestionar eficientemente los recursos de la máquina
- b. Proporcionar un interfaz a los usuarios para el uso de la máquina
- c. Proteger los espacios de memoria de las aplicaciones
- d. **Convertir los códigos de alto nivel en lenguaje máquina**

Cuando se genera una interrupción, se deberá almacenar necesariamente:

- a. **El contador de programa (PC)**
- b. Los registros de la DMA
- c. El segmento pila
- d. El segmento de datos

El modo de funcionamiento del procesador destinado a la ejecución de código del sistema operativo se denomina:

- a. Modo organizador
- **b. Modo supervisor o kernel**
- c. Modo maestro
- d. Modo esclavo

El procesador en modo núcleo, permite principalmente:

- a. Aumentar la velocidad de procesamiento
- b. Consultar la hora de sistema
- c. Poner a ejecutar trabajos por lotes
- **d. Ejecutar instrucciones privilegiadas**

La DMA es un circuito fundamental para poder implementar:

- a. Las interrupciones
- **b. La multiprogramación**
- c. La ejecución de trabajos por lotes
- d. Las instrucciones privilegiadas

La multiprogramación permite solapar en el tiempo:

- a. Interrupciones y excepciones
- b. Excepciones y operaciones de entrada salida
- **c. Operaciones de entrada salida y computación en la CPU**
- d. Instrucciones y datos de la CPU

La técnica de tiempo compartido (time-sharing) permite:

- **a. Compartir un recurso de computación entre varios usuarios/aplicaciones**
- b. Compartir el reloj entre varios computadores
- c. Compartir una interrupción entre varios procesadores
- d. Compartir la E/S por medio de la DMA

Las aplicaciones de usuario se ejecutarán sobre la CPU:

- a. En modo supervisor
- b. En modo hipervisor
- **c. En modo usuario**
- d. En modo kernel

Puede afirmarse que el sistema operativo es:

- **a. Un programa o conjunto de programas**
- b. Un hardware o circuitería
- c. Un conjunto de bibliotecas y librerías
- d. Un repertorio de instrucciones

Un intérprete de comandos ó CLI (command line interface):

- **a. Analiza una orden ó comando, y si es correcta la ejecuta**
- b. Proporciona una máquina extendida
- c. Gestiona la memoria del computador
- d. Gestiona el procesador del computador

Un programa de usuario realiza una llamada a sistema a través de:

- a. una excepcion
- b. una interrupción no enmascarable
- **c. una interrupción software**
- d. una interrupción hardware

Un sistema operativo organizado por capas diríamos que es:

- a. Monolítico
- **b. Estructurado**
- c. Cliente-servidor
- d. No estructurado

Un código máquina binario cargado en la memoria principal:

- a. Necesariamente es un proceso del sistema
- b. Necesariamente es una tarea
- c. Necesariamente ha debido ser lanzado desde un shell
- **d. No tiene porque ser un proceso, puede ser una librería o una rutina de tratamiento de interrupción**

Cuando un proceso crea a otro, al proceso creador lo denominamos 'proceso hijo' y al proceso que ha sido creado 'proceso padre'

- Verdadero
- **Falso**

En una plataforma multiprocesador, con el término Multiprocesamiento Simétrico (SMP) hacemos referencia a un sistema operativo en el que el kernel:

- A. Es un kernel específico para multicores heterogéneos
- B. Sigue una estructura cliente/servidor simétrica
- **C. Puede ejecutarse en cualquier procesador indistintamente**
- D. Sólo puede ejecutarse en un procesador en particular

La transición time-out por quantum de tiempo es una acción expropiativa, que se dará:

- A. Entre los estados Blocked y Ready
- **B. Entre los estados Running y Ready**
- C. Entre los estados Running y Block
- D. Entre los estados Blocked y Suspend

Con respecto a procesos y threads (procesos ligeros):

- A. Los procesos no pueden compartir información mientras que los threads si
- **B. Compartir información entre threads es más fácil que compartir información entre procesos**
- C. Los threads no pueden compartir información mientras que los procesos si
- D. Compartir información entre procesos es más fácil que compartir información entre threads

¿Cuál es la información que no comparten los procesos ligeros (threads) de un mismo proceso?

- A. registros de la CPU (estado del procesador)
- B. Espacio de pila
- C. Identificación y estado del thread
- **D. Todas las respuestas son correctas**

En los SO tipo UNIX, ¿cuándo un proceso A pasa a tener el estado "zombie"?

- A. Cuando A hace un wait a un proceso hijo que aún no ha terminado
- **B. Cuando termina el proceso A y su padre no le hace un wait**
- C. Cuando A hace un wait a un proceso hijo justo en el instante en el que finaliza el hijo
- D. Cuando A hace un wait a un proceso hijo que ya ha terminado

Cuando un thread termina, siempre debe haber otro thread del proceso que le realice un 'pthread_join'

- Verdadero

- **Falso**

En un sistema con tres estados posibles para los procesos (Ejecución, Listo y Bloqueado), ¿Qué transiciones de estado son posibles para un proceso?

Selecciona una o más de una:

- A. Un proceso puede pasar de bloqueado a ejecución
- **B. Un proceso puede pasar de ejecución a bloqueado**
- **C. Un proceso puede pasar de bloqueado a listo**
- **D. Un proceso puede pasar de ejecución a listo**

Respecto a la multiprogramación:

- A. Un S.O. multiprogramado debe ser de tiempo compartido
- B. Todas las respuestas son correctas
- **C. Un S.O. de tiempo compartido debe ser multiprogramado**
- D. El tiempo compartido permite realizar multiprogramación

Bajo el supuesto de que el sistema operativo asignara los identificadores de proceso en orden creciente a medida que los procesos se van creando, y que el número de identificadores es ilimitado, se verificaría que:

- a. PID + PPID = 1
- **b. PPID < PID**
- c. PPID > PID
- d. PPID = PID

Siempre que se produce un cambio del nivel de privilegio (usuario/núcleo) de un proceso se debe realizar un cambio de contexto

- Verdadero
- **Falso**

Tiempo compartido es una técnica para:

- **A. proporcionar concurrencia de procesos**
- B. coordinar las llamadas a sistema (syscalls)
- C. asignar recursos HW siguiendo determinada política
- D. solapar las operaciones de E/S con la ejecución en la CPU

¿Qué característica exhibe un proceso demonio (daemon)?

- A. tiene prioridad de tiempo real
- **B. se ejecuta generalmente en segundo plano**
- C. se ejecuta en una cola batch
- D. carece de espacio reservado para pila

Los threads a nivel de usuario:

- a. Podrán disponer de un espacio de memoria más amplio que los de kernel
- b. Se planifican según la política del sistema operativo
- c. Nunca cambian de contexto
- **d. Podrán conmutar de contexto más rápidamente que los de kernel ya que no hace falta conmutar a modo kernel**

La creación de procesos en un sistema unix se realiza por medio de la llamada a sistema:

- a. create()
- **b. fork()**
- c. exec()
- d. spawn()

Hablando de multithreading, cuál de las siguientes afirmaciones es correcta:

- a. El multithreading es una tecnología anticuada, los SS.OO. de hoy en día casi no la soportan
- b. Elimina por completo la necesidad de tener procesos, puesto que todos los programas se ejecutarán como threads
- **c. El multithreading facilita la programación con respecto a la multitarea pura (procesos de un sólo thread)**
- d. Históricamente se desarrolló antes que la multiprogramación

Un proceso que no puede ejecutarse hasta que se produzca un cierto evento se dice que está en el estado:

- a. New
- b. Ready
- **c. Blocked**
- d. Suspend

Con relación a las señales POSIX:

- A. Todas las respuestas son correctas
- **B. Es posible cambiar el comportamiento por defecto de un proceso al recibir algunas señales**

- C. Las señales que finalizan un proceso siempre son enviadas por el SO
- D. La recepción de cualquier señal por parte de un proceso siempre provoca la finalización del mismo

¿Como influirá un cambio de contexto en el comportamiento de las memorias cachés?

- A. De forma negativa, disminuyendo el número de reemplazos
- B. De forma positiva, aumentando el número de aciertos
- C. No afecta en absoluto
- **D. De forma negativa, aumentando el número de fallos**

En un sistema de tiempo compartido con varios procesos ready, si uno realiza una llamada a `_exit()`:

- A. se producirá un cambio de contexto a nivel de thread
- B. aumentará el grado de multiprogramación
- **C. se producirá un cambio de contexto**
- D. en ocasiones no se producirá un cambio de contexto

La acción de swapping (sacar un proceso de memoria principal) es una causa:

- **A. De suspensión de un proceso**
- B. Ninguna de los anteriores
- C. De bloqueo de un proceso
- D. De terminación de un proceso

La llamada de Posix con la que un proceso le puede enviar una señal a otro proceso es:

- **a. kill()**
- b. signal()
- c. sigsend()
- d. signalrise()

En un sistema con un planificador basado en el algoritmo primero el trabajo más corto (SJF) puede producirse un cambio de contexto:

- A. Cuando se bloquea, se termina el proceso o se acaba su cuento de ejecución
- **B. Cuando se bloquea o se termina el proceso**
- C. Cuando se bloquea, se termina el proceso o existe otro proceso de más prioridad listo para ejecución
- D. Cuando se bloquea, se termina el proceso o llegue a listos un proceso más corto que el tiempo restante de ejecución del actual

La planificación afecta al rendimiento del sistema puesto que determina que proceso esperará y cuál progresará en su ejecución.

- **Verdadero**
- Falso

En los planificadores de corto plazo actuales de sistemas operativos genéricos como Linux o Windows XP, se usan prioridades fijas para procesos de tiempo real y así permitir la predictibilidad de su planificación

- **Verdadero**
- Falso

En términos de la frecuencia de ejecución, el planificador de corto plazo es el que se ejecuta:

- a. Ninguna de las demás respuestas es cierta
- b. Más o menos lo mismo que los demás planificadores
- c. Menos frecuentemente
- **d. Más frecuentemente**

En el planificador de Linux, la prioridad de los procesos de tiempo real es dinámica.

- Verdadero
- **Falso**

Sean 4 procesos W, X, Y, Z con tiempos de servicio (ráfaga de CPU) 12, 10, 2, 7 unidades de tiempo respectivamente. ¿Cuál es el orden de finalización con planificación FCFS/FIFO, asumiendo que todos fueron creados en el mismo instante y en este orden?

- a. Con los datos proporcionados no se conoce el orden de finalización
- b. Z, Y, X, W
- **c. W, X, Y, Z**
- d. X, Y, W, Z

¿En cuál de los siguientes algoritmos de planificación de procesos el tratamiento de una interrupción que indica el final de una operación de E/S bloqueante puede producir cambio de proceso?

- **A. Menor tiempo restante (Shortest-Remaining-Time First)**
- B. FIFO
- C. Turno rotatorio (Round-Robin)
- D. Primero el proceso más corto (Shortest-Job First o Shortest-Process Next)

Una forma usual de solucionar el problema de la inanición (starvation) de los procesos de menor prioridad en un sistema con planificación basada en las prioridades es:

- **a. Cambiar la prioridad de un proceso de acuerdo con su antiguedad**
- b. La rotación de los procesos con un round-robin en una cola con prioridad
- c. Cambiar al azar las prioridades de los procesos
- d. Todas las demás respuestas son válidas

El planificador de corto plazo puede limitar el grado de multiprogramación para proporcionar un servicio satisfactorio al conjunto actual de procesos.

- Verdadero
- **Falso**

Un problema asociado con la técnica de planificación Shortest Job First (SJF) es:

- **a. Todas las demás respuestas son válidas**
- b. La ausencia de expropiación (preemption)
- c. La inanición (starvation) de los procesos de mayor duración
- d. La necesidad de conocer o estimar los tiempos de procesamiento requeridos por cada proceso

¿Cuál de las siguientes políticas de planificación es más adecuada para un sistema de tiempo compartido con carga mixta de procesos interactivos y procesos de tiempo real?

- A. Primero el trabajo más corto (SJF)
- **B. Prioridades**
- C. FIFO
- D. Round-Robin (turno circular)

Se lanzan 3 procesos en un sistema Round-Robin de una sola cola en el mismo instante. Los tres procesos constan de una única ráfaga de CPU de duración 10 unidades de tiempo. ¿Cuál de los siguientes valores de quantum darán lugar a un tiempo medio de retorno menor?

- A. 2
- **B. 10**
- C. 1
- D. 5

¿Cuál de las siguientes políticas de planificación requiere una estimación o conocimiento previo de las duraciones de los procesos o de sus ráfagas de CPU?

- a. Todas las demás respuestas son válidas
- b. Round Robin (RR)
- c. First Come First Served(FCFS)

- **d. Shortest Remaining Time (SRT)**

El tiempo de retorno (Tr) de un proceso se define como:

- A. Tiempo en el que el proceso ha estado ejecutando código del SO
- B. Tiempo límite para la pulsación de la tecla "return"
- **C. El tiempo que pasa desde que se lanza el proceso hasta que finaliza su ejecución**
- D. El tiempo que pasa desde que se lanza el proceso hasta que toma la CPU por primera vez

En un sistema de planificación por prioridades expropiativo, despertar a un proceso obliga a valorar cuál es el próximo proceso a despachar (asignarle la CPU).

- **Verdadero**
- Falso

El objetivo principal del planificador de corto plazo es distribuir el tiempo de uso del procesador de tal manera que se optimicen uno o más aspectos del comportamiento del sistema.

- **Verdadero**
- Falso

Sean 4 procesos W, X, Y, Z con tiempos de servicio (ráfaga de CPU) 7, 13, 9, 11 unidades de tiempo respectivamente. ¿Cuál es el orden de finalización con planificación SJF, asumiendo que todos fueron creados en el mismo instante?

- a. Z, Y, X, W
- b. W, X, Y, Z
- **c. W, Y, Z, X**
- d. Con los datos proporcionados no se conoce el orden de finalización

¿Con qué tipo de planificación el Sistema Operativo puede interrumpir al proceso que está actualmente en ejecución para pasarlo al estado Ready?

- a. Ninguna de las demás respuestas es cierta
- **b. Expropiativa (Preemptive)**
- c. No expropiativa (non-preemptive)
- d. First-come-first-served

La invocación del planificador de largo plazo se produce con cada evento que pueda causar el bloqueo o la expropiación del proceso que está actualmente en ejecución.

- Verdadero
- **Falso**

Un problema asociado con la técnica de planificación Shortest Remaining Time (SRT) es:

- a. Todas las demás respuestas son válidas
- b. **La necesidad de conocer o estimar los tiempos de procesamiento requeridos por las ráfagas de CPU de cada proceso**
- c. Aumenta los tiempos de espera medios disminuyendo la productividad
- d. La ausencia de expropiación (preemption)

Podemos considerar que el sistema de planificación FIFO FCFS es un caso particular de otro tipo de planificación que es:

- A. ... Round-Robin, con quantum
- B. ... SJF no preemptive
- C. ... SJF preemptive
- D. ... Round-Robin, con quantum

Señala una situación que refleje el "convoy effect" que puede producirse en el contexto del uso compartido de la CPU por parte de los procesos

- a. **Un proceso que apenas necesita la CPU espera demasiado antes de usarla porque otro proceso que la necesita más acaba monopolizando su uso.**
- b. Un proceso que necesita mucho la CPU no puede monopolizarla frente a otros que apenas la necesitan.
- c. Un proceso que necesita mucho la CPU espera demasiado antes de usarla porque su cambio de contexto es muy lento.
- d. Un proceso apenas puede usar la CPU porque existen multitud de procesos pequeños que comutan de forma excesivamente rápida.

Sea un sistema por prioridad expropiativa en el que comienzan a ejecutarse dos procesos: P1 con prioridad 1 (más alta) y P2 con prioridad 2 (más baja), y ambos necesitan el mismo tiempo de CPU para concluir su ejecución. ¿En qué situación puede acabar P2 antes que P1?

- a. Si P1 es un código escrito en lenguaje Fortran y P2 es un código escrito en lenguaje C.
- b. En ningún caso.
- c. Si P1 realiza operaciones con cadenas de caracteres y P2 sólo realiza operaciones aritméticas.
- d. **Si P1 realiza una operación de entrada/salida y P2 no.**

¿Cómo se asocian los procesos y sus hilos dentro de un sistema operativo?

- a. Un proceso puede tener un hilo, que puede pertenecer a varios procesos.
- b. Un proceso puede tener muchos hilos y un hilo puede pertenecer a muchos procesos.
- c. **Un proceso puede tener muchos hilos, pero un hilo sólo puede pertenecer a un proceso.**

- d. Un hilo puede tener un proceso, que puede pertenecer a varios hilos.

Sea un sistema por prioridad NO expropiativa en el que comienzan a ejecutarse dos procesos: P1 con prioridad 1 (más alta) y P2 con prioridad 2 (más baja), y ambos necesitan el mismo tiempo de CPU para concluir su ejecución. ¿En qué situación puede acabar P2 antes que P1?

- **a. Si P1 realiza una operación de entrada/salida y P2 no.**
- b. Si P1 se ha compilado después que P2.
- c. Si P1 realiza operaciones aritméticas en doble precisión y P2 realiza operaciones aritméticas en simple precisión.
- d. En ningún caso.

En un PC dotado de 2 CPUs, cada una con un procesador quad-core, para aspirar a ocuparlo al 100% necesitamos al menos

- a. Un proceso que tenga al menos ocho hilos.
- b. Ocho procesos aunque no tengan hilos.
- c. Cuatro procesos con 2 hilos cada uno.
- **d. Dos procesos con 4 hilos cada uno**

Señala tres acciones que normalmente desembocan en una llamada al sistema.

- **a. Abrir un fichero, imprimir un mensaje, finalizar con un error.**
- b. Leer una variable de memoria, introducirla en la caché, almacenarla en un registro del procesador.
- c. Realizar una operación aritmética, esperar una entrada de teclado, saltar al inicio de un bucle del programa.
- d. Verificar los permisos de un proceso, denegarle el acceso a los recursos, autorizar el uso de un dispositivo.

¿Cuál es la aportación fundamental de la multiprogramación en un sistema operativo?

- a. Facilidad de uso.
- b. Seguridad.
- **c. Eficiencia.**
- d. Fiabilidad.

¿Qué aporta el acceso directo a memoria o DMA?

- **a. Que la CPU quede liberada de realizar transferencias a memoria y pueda dedicarse a otras tareas.**
- b. Una forma segura de acceder a memoria por parte de los dispositivos.
- c. Que los usuarios puedan compartir el espacio de memoria.

- d. Que los dispositivos puedan compartir el espacio de memoria.

Los algoritmos de planificación del uso de la CPU de un sistema operativo que no desalojan un proceso presentan las siguientes ventajas y desventajas respecto a los que sí permiten desalojarlo (preemptive).

- a. Ventaja: Favorecen a los procesos memory-bound (aquellos que tienen el cuello de botella en el acceso a memoria), que por otra parte son los más habituales. Inconveniente: Perjudican a los procesos compute-bound (aquellos que tienen el cuello de botella en el uso de la CPU).
- b. Ventaja: Predicen mejor el comportamiento futuro del proceso. Inconveniente: Son más difíciles de implementar.
- c. Todas las respuestas son correctas.
- **d. Ventaja: Minimizan los cambios de contexto. Inconveniente: Algún proceso puede tener un trato de favor y perjudicar al resto más de lo debido.**

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.

El bloque de control de un proceso se utiliza para

- a. Liberar la información que ocupa cuando finaliza el proceso.
- b. Ocupar la información que necesita cuando el sistema conmuta a ese proceso.
- c. Planificar la ejecución del proceso y registrar su actividad.
- **d. Las otras tres respuestas son correctas.**

El kernel del Sistema Operativo Linux proporciona

- **a. El sistema de ficheros, la gestión de la memoria y la planificación de la CPU.**
- b. El API, el CLI y el GUI.
- c. Las ALUs, los bancos de registros y las memorias cachés para la ejecución de los programas.
- d. El software, el firmware y el hardware

Cuando un proceso padre crea un proceso hijo

- a. Si el hijo progresá, el padre estará siempre esperándole.
- b. Si el padre progresá, el hijo no puede hacerlo.
- **c. Cuando el hijo progresá, el padre puede también hacerlo.**
- d. Los dos procesos se ejecutan de forma síncrona

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.

En los sistemas operativos reales (tomemos Windows y Linux), los algoritmos de planificación son todos

- a. Con colas pero sin prioridad.
- b. Con prioridad pero sin colas.
- **c. Preemptive (permiten desalojar un proceso).**
- d. Non-preemptive (no permiten desalojar un proceso).

Una interrupción puede ser generada por la capa

- a. Software.
- b. Hardware.
- **c. Software o hardware.**
- d. Ni software ni hardware.

¿Cuál es el principal objetivo del algoritmo de planificación Shortest Remaining Time First)?

- **a. Minimizar el tiempo medio de compleción de los procesos (Tr).**
- b. Maximizar el tiempo medio de compleción de los procesos (Tr).
- c. Maximizar el tiempo medio de espera de los procesos (Te).
- d. Maximizar el tiempo medio de compleción (Tr) y espera (Te) de los procesos.

En la pescadería de Carrefour, el cliente toma un número a su llegada y se va atendiendo a los clientes según el número más bajo. ¿Qué algoritmo de planificación estamos usando?

- a. Tiempo compartido.
- b. Colas multnivel.
- c. SJF (Shortest Job First).
- **d. FCFS.**

La función básica de una API es

- a. Ocultar la complejidad del hardware subyacente.
- b. Optimizar las llamadas al sistema.

- c. Ejecutar el código en modo superusuario.
- d. **Proporcionar una interfaz común a los servicios y las aplicaciones de usuario.**

¿Qué peculiaridad presenta el siguiente programa?

```
int main() {
    pid_t pid;
    pid = fork();
    if (pid > 0)
        exit();
    else {
        wait(NULL);
        fork();
    }
}
```

- a. El segundo fork() (la última instrucción que aparece) nunca se ejecuta.
- b. El proceso hijo trata de recoger al padre, no al contrario.
- c. **Las otras tres respuestas son correctas.**
- d. La llamada a exit() no es necesaria.

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. Los metadatos del usuario y los metadatos del kernel.
- c. El estado de los registros de la CPU, el estado del sistema operativo y el estado del hardware.
- d. **El código de su programa, los datos de su programa, la pila y el heap**

¿Qué sincronización puede establecerse entre dos procesos padre e hijo?

- a. El hijo puede esperar a que acabe el padre.
- b. Tanto el padre como el hijo pueden esperar a que acabe el otro.
- c. Ni el padre ni el hijo pueden esperar a que acabe el otro.
- d. **El padre puede esperar a que acabe el hijo.**

Si un proceso no crea al menos 4 hilos

- a. No podrá aprovechar él sólo toda la potencia de una memoria quad-channel (de 4 canales).

- 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 CPU de 4 GHz

El bloque de control de un proceso (PCB) es una estructura de datos que crea y mantiene

- **a. El sistema operativo.**
- b. El usuario.
- c. El usuario administrador del sistema (root).
- d. El hardware de la máquina.

La comunicación entre procesos puede realizarse por pase de mensajes o memoria compartida. ¿Qué ventajas presenta cada método?

- a. La memoria compartida requiere una mayor intervención del kernel del sistema operativo.
- **b. El pase de mensajes resulta más fácil de implementar, pero la memoria compartida es más rápida.**
- c. El pase de mensajes es más útil y efectivo para grandes volúmenes de datos.
- d. El pase de mensajes es más rápido, pero la memoria compartida es más fácil de implementar

¿Qué es un vector de interrupción?

- a. Una señal o evento.
- **b. Un puntero a memoria.**
- c. Una llamada al sistema.
- d. El driver de un dispositivo

El sistema operativo tiene como función primordial

- a. Transformar el código fuente en código máquina directamente ejecutable por el microprocesador o CPU.
- **b. Controlar y coordinar el uso del hardware por parte de las aplicaciones y los usuarios.**
- c. Definir las formas en que los recursos del sistema se utilizan para resolver los problemas computacionales de los usuarios.
- d. Proporcionar los recursos de computación básicos.

¿Qué métrica para evaluar la idoneidad de un planificador de procesos es más efectiva para sistemas operativos en tiempo real?

- a. El tiempo de espera.
- b. El throughput o ritmo en la compleción de procesos.
- c. El porcentaje de ocupación de la CPU.

- **d. El tiempo de respuesta.**

¿Qué les sucede a los procesos hijo en Unix cuando muere su padre?

- a. El padre no puede morir hasta que no acaben todos los hijos que ha creado.
- **b. Siguen vivos y se les asigna como padre el proceso Init.**
- c. Se quedan en estado zombie.
- d. También mueren.

Señala los dos objetivos de un sistema operativo que más entran en conflicto, es decir, si nos dedicamos excesivamente a uno de ellos podemos comprometer el otro

- **a. Seguridad y eficiencia.**
- b. Seguridad y facilidad de uso.
- c. Facilidad de uso y eficiencia.
- d. Todos los objetivos son 100% compatibles.

En la planificación de procesos por prioridad expropiativa (*preemptive*), cuando un proceso abandona la CPU para completar ciclos de E/S, retoma inmediatamente el uso de la CPU al concluir dichos ciclos de E/S, si tiene una prioridad

- a. Igual que el proceso que ocupa la CPU en ese momento.
- b. Más baja (número más alto) que el proceso que ocupa la CPU en ese momento.
- c. No tiene opción de volver a ocupar la CPU hasta que termine de usarla el proceso que ocupa la CPU en ese momento.
- **d. Más alta (número más bajo) que el proceso que ocupa la CPU en ese momento.**

El tiempo que el sistema operativo dedica al cambio de contexto de un proceso

- a. Aumenta con la complejidad del sistema operativo.
- b. Aumenta con el tamaño del PCB de los procesos involucrados en dicho cambio de contexto.
- **c. Aumenta con la complejidad del sistema operativo y con el tamaño del PCB de los procesos involucrados en dicho cambio de contexto.**
- d. Ninguna respuesta es correcta.

¿Cómo se consigue que un algoritmo de planificación de procesos por prioridades se asemeje lo máximo posible a un algoritmo SRTF (Shortest Remaining First Time)?

- a. Asignando las prioridades más altas (números más bajos) a los procesos más largos.
- **b. Asignando las prioridades más altas (números más bajos) a los procesos más cortos.**
- c. Asignando la misma prioridad a todos los procesos.
- d. No es posible conseguir que se parezcan lo más mínimo, son estrategias completamente diferentes

En la planificación de procesos por prioridad NO expropiativa (*non-preemptive*), cuando un proceso abandona la CPU para completar ciclos de E/S, retoma inmediatamente el uso de la CPU al concluir dichos ciclos de E/S, si tiene una prioridad

- a. Igual que el proceso que ocupa la CPU en ese momento.
- b. Más baja (número más alto) que el proceso que ocupa la CPU en ese momento.
- c. **No tiene opción de ocupar la CPU hasta que no termine de usarla el proceso que ocupa la CPU en ese momento.**
- d. Más alta (número más bajo) que el proceso que ocupa la CPU en ese momento.

¿A quién penaliza más el algoritmo SJF?

- a. A los procesos que menos tiempo utilizan la CPU.
- b. A ninguno. Es el algoritmo más justo que hemos estudiado.
- c. A los procesos que más utilizan la memoria.
- d. **A los procesos que más tiempo utilizan la CPU.**

En un sistema de planificación de colas multinivel, un proceso B ubicado en la cola 2 puede arrebatar la CPU a un proceso A que está usando la CPU habiendo entrado desde la cola 1 cuando

- a. El proceso B acaba su *quantum* de tiempo o finaliza una operación de entrada/salida.
- b. El proceso B acaba su *quantum* de tiempo.
- c. **En ninguno de los casos contemplados en las otras tres opciones.**
- d. El proceso B finaliza una operación de entrada/salida.

¿Cuándo se conoce el tiempo total que un proceso va a estar suspendido durante su ejecución?

- a. Cuando se compila su programa asociado.
- b. Cuando se construye su bloque de control de proceso (PCB).
- c. **No puede conocerse hasta que no haya terminado la ejecución del proceso.**
- d. Cuando comienza a ejecutarse su programa asociado.

En un cruce de tráfico regulado por dos semáforos que van estableciendo turnos alternativos de paso de un minuto a cada uno de los dos flujos circulatorios, ¿Qué algoritmo de planificación replica mejor la realidad si consideramos que cada flujo circulatorio es un proceso computacional?

- a. FCFS.
- b. Dos colas multinivel, una para cada flujo circulatorio, y FCFS dentro de cada cola.
- c. **Tiempo compartido con *quantums* de un minuto para cada proceso.**
- d. SJF (Shortest Job First).

En un sistema de planificación por dos colas multinivel con el mismo *quantum* de tiempo en las dos colas, se procede de la siguiente forma

- a. Se conceden dos *quantums* de tiempo a cada proceso de la cola 1 por cada *quantum* de tiempo concedido a la cola 2, y así sucesivamente hasta que acaban todos los procesos.
- b. Se van otorgando *quantums* a procesos de la cola 1 y de la cola 2 de forma alterna, aunque si coinciden un proceso de cada cola al salir de una operación de entrada/salida, el desempate se resuelve a favor del proceso que está en la cola 1
- c. Se concede un *quantum* a cada proceso de la cola 1, luego se concede un *quantum* a cada proceso de la cola 2, y se vuelve a la cola 1 reanudando este mismo proceder de forma indefinida.
- d. **Se concede un *quantum* a cada proceso de la cola 1, y luego se siguen repartiendo *quantums* en los procesos que sigan quedando en ella. No se pasará a la cola 2 hasta que no esté vacía la cola 1.**

El sistema operativo

- a. Aloja recursos, gestiona dispositivos y monitoriza actividades.
- b. Transforma el programa fuente y visualiza información en pantalla.
- c. Organiza el código y delimita el uso de los recursos que necesita.
- d. Las otras tres respuestas son correctas.

TEMAS 3 Y 4

¿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

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”

Con respecto a FAT12, el formato FAT16 permite:

- Sectores de menor tamaño
- Sectores de tamaño mayor
- Clusters de mayor tamaño
- **Clusters de menor tamaño**

El cluster es un múltiple de:

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

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)

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**

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

La gestión del espacio libre en FAT32 se realiza:

- En la tabla maestra (MTF)
- En el MBR
- En el superbloque
- **En la propia FAT**

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

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

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

- Solo simples
- Hasta de dos niveles
- **Hasta de tres niveles**
- Hasta de cuatro niveles

¿Qué método de asignación de espacio en disco puede provocar fragmentación externa?

- **Asignación contigua**
- Asignación enlazada
- Asignación indexada
- Ninguno de ellos

La FAT es un sistema de asignación:

- Contiguo
- **Enlazado**
- Indexado
- Volátil

El acceso aleatorio ...

- Es más rápido en asignación enlazada que en contigua
- Es más rápido en asignación enlazada que en indexada
- **Es más rápido en asignación contigua que en enlazada**

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 páginas
- Del tamaño de la zona de swap

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

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

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

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

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 maquina

El cargador (loader):

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

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:

- Metapaginacion
- Hiperlocalidad
- **Trasiego (thrashing)**
- Fragmentación

La política de reemplazo qué escoge solo entre las páginas residentes del proceso que genero 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

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

- **Reemplazo global, aunque sea el óptimo.**
- “Buffering” de paginas
- Demonio de paginación
- Reemplazo local con algoritmo FINUFO

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**

A mayor probabilidad de fallo de pagina

- 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

Con el cambio de contexto:

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

La paginacion y segmentacion:

- Son modelos de memoria contiguos
- Son modelos de memoria monoprogramados
- **Son modelos de memoria no-contiguos**
- Son modelos de memoria inmóvil

Con el cambio de contexto :

- **Se ha de notificar a la MMU la tabla de paginas que debe usar (la del proceso entrante)**
- Se resetean todos los marcos de pagina ocupados
- Se resetean las tablas de pagina
- Se mantiene una lista de marcos posibles.

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

- **El PCB**
- La pila (SP)

- El PC
- La FPU

De los siguientes algoritmos de reemplazo de paginas , ¿Qué opción aprovecha mejor la localidad temporal ?

- **LRU**
- FINUFO
- Random
- FIFO

Por PTBR (Page Table Base Register) se entiende:

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

En un modelo multiprogramado con asignación de particiones (residente, inmóvil, contiguo y entero) ¿Qué estrategia de asignación presentaría una mayor posibilidad de fragmentación interna?

- Particiones de tamaño fijo con particiones pequeñas
- **Particiones de tamaño fijo con particiones grandes**
- Particiones de tamaño variable

¿Cuál no es un requerimiento hardware imprescindible de un sistema de memoria virtual?

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

En ausencia de TLB, el fallo de página dará lugar a

- **Una interrupción o excepción**
- Un bug del sistema
- Nada en particular, es un evento que el sistema suele ignorar
- Un reset de la CPU

¿Qué tipo de fragmentación sufre la paginación de memoria?

- **Interna**
- Media
- Externa
- La memoria paginada carece por completo de fragmentación

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

- Best-Fit

- Mid-Fit
- Worst-Fit
- **First-Fit**

En un sistema de memoria paginado, si se disminuye el tamaño de la página, manteniendo igual los tamaños de los espacios físico y lógico, aumentara:

- El numero de entradas de la tabla de paginas, pero cada entrada sera del mismo tamaño.
- **El numero de entradas de la tabla de paginas, y tambien el tamaño de cada entrada.**
- Nada, la tabla de paginas mantendra su tamaño inicial.
- El tamaño de las entradas de la tabla de pagina, pero no su numero.

Un sistema multiprogramado con asignacion de particiones variables (residente, inmovil , contiguo y entero) gestiona el espacio libre con vectores de bits. Si existen 256 bloques y el vector de ocupacion tiene el valor hexadeciml 0x00...09012, el algoritmo “Worst-Fit” asignara:

- Un hueco de tamaño de 2 bloques
- Un hueco de tamaño de 3 bloques
- **Un hueco de tamaño mayor a 4 bloques**
- Un hueco de tamaño de 4 bloques

Si el tamaño de pagina es de 16 KB , el numero de bits del campo desplazamiento de la direccion logica (offset) tendra un tamaño:

- 12 bits
- **14 bits**
- 10 bits
- 16 bits

Si el tamaño de pagina es de 4 KB y el numero de marcos de pagina en memoria principal es 256K marcos ¿Cuántas entradas podemos afirmar que tiene la TLB?

- **El tamaño de la TLB no tiene ninguna relación con la información proporcionada en esta cuestión.**
- 4 entradas
- 14 entradas
- 24 entradas

Se referencia la siguiente secuencia de paginas logicas: pag 3 ,pag 5, pag 6, pag 7, pag 6, pag 7, pag 5, pag 6 , pag 8. La cardinalidad (numero de elementos) maxima del working set tomando una ventana de 5 referencias es:

- 6
- **4**
- 3
- 5

Si la direccion fisica es de 38 bits , la direccion logica de 52 bits y el tamaño de pagina 16 KB, el numero de marcos en memoria principal es:

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

Con el cambio de contexto

- **Se ha de notificar a la MMU la tabla de paginas que debe usar (la del proceso entrante)**
- Se mantiene una lista de marcos libres
- Se resetean las tablas de pagina
- Se resetean todos los marcos de pagina ocupados

Toda pagina de un proceso residente en un marco de memoria fisica tendra activo necesariamente:

- El bit de modificacion M
- El bit de valido V
- El bit de escritura W
- **El bit de presencia P**

¿Que modelo de memoria , de los siguientes , hace uso de la memoria secundaria?

- El modelo entero
- **El modelo no residente**
- El modelo no contiguo
- El modelo movil

La traducción inversa de páginas permite:

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

En caso de fallo de pagina, el proceso que provoca el fallo:

- Esta running durante el swap in/out
- **Esta bloqueado durante el swap in/out**
- Esta suspendido durante el swap in/out
- Esta zombie durante el swap in/out

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 millon

El fenómeno de thrashing se produce cuando

- Un fallo de pagina no puede ser atendido por el sistema operativo
- El grado de multiprocesamiento compartiendo paginas es bajo
- Una pagina de memoria fisica se sustituye por otra procedente de disco
- **Un proceso consume la mayor parte de su tiempo intercambiando paginas entre memoria virtual y fisica en lugar de avanzando en su ejecucion.**

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)

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

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

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.

¿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.**

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.

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.

Una FAT de disco aloja los sectores de forma

Selecciona una:

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

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.

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.

Sectores de disco grandes favorecen

- Selecciona una:
- a. Un 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)**

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**

En un sistema de memoria con direcciones virtuales de 16 bits montado sobre una memoria física de 8 páginas de 4 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=14, f=2, d=3.
- **b. p=14, f=3, d=2.**
- c. p=16, f=2, d=3.
- d. p=16, f=3, d=2

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.

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.**

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

- a. Es necesario utilizar hasta el puntero indirecto simple.
- b. Con los 10 punteros directos es suficiente para direccionar a ese fichero.
- c. Es necesario utilizar hasta el puntero indirecto triple.
- d. **Es necesario utilizar hasta el puntero indirecto doble.**

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.

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.

¿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).**

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

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

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)

Selecciona una:

- 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.**

¿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.

¿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.