

PARCIAL 2020

1. **Mediante los modos de ejecución se controla que un proceso no pueda, por sí mismo, ejecutar instrucciones de CPU relacionadas a E/S**

Verdadero

La entrada salida (E/S) requiere la interacción con un dispositivo. En esta interacción la comunicación debe ser adecuada para que el dispositivo "entienda" lo que se le pide, encolar los sucesivos requerimientos, devolver avisos de estado (códigos de retorno). Por eso se hace imprescindible que las E/S se trabajen mediante las system calls, que son módulos de sw que interactúan adecuadamente, en el "idioma" que entienden los controladores de disco.

2. **En un mismo sistema operativo pueden convivir la modalidad Batch y la Interactiva**

Verdadero

Un sistema operativo puede trabajar en forma interactiva procesos que necesitan interacción y tener la posibilidad de trabajar en forma batch para los momentos de baja interacción. Por ejemplo, puede estar trabajando en modalidad interactiva en los horarios de atención al público y "arrancar" procesos batch cuando la demanda baja.

3. **En un diseño monolítico del kernel, todos los componentes de un SO se deben ejecutar en modo Kernel.**

Falso

4. **En una instrucción del tipo:**

read(referencia_archivo, puntero/variable, Cantidad_bytes)

considerando que "read" es una rutina que implementa esa llamada al sistema...

Sería más rápida la ejecución si los parámetros referencia_archivo, puntero/variable, Cantidad_bytes, se pasan a la rutina mediante registros de la CPU o mediante la pila?

a. Mediante registros

b. Mediante pila

5. **¿Cuál de las siguientes opciones es correcta acerca del concepto de Sistemas Operativos?**

Seleccione una o más de una:

a. Busca abstraer el hardware de los usuarios

b. Buscan que el agregado de nuevas funciones al mismo no interfieran con las anteriores

c. Delega completamente en el hardware la administración de la memoria

d. Tiene como funciones principales la administración de CPU, memoria y E/S

e. Utilizan intérpretes de comandos para implementar llamadas al sistema y ejecutar funciones que requieren elevación

6. Con el fin de limitar el conjunto de instrucciones que puede ejecutar cada proceso, se define el bit de modo (usuario / supervisor), el cual es almacenado en la PCB de cada proceso que se va a ejecutar.

Falso

El bit de modo se encuentra en el procesador, es el procesador quien limita.

7. La PCB de un proceso en estado "Swapeado" se encuentra almacenada en

a. RAM (Mem. Principal)

b. Área de Swap (Mem. Secundaria)

c. Un proceso en ese estado no cuenta con un PCB

8. Para poder llevar adelante la planificación, el sistema operativo utiliza diversos algoritmos y colas en las cuales incluye el espacio de direcciones de cada proceso

Falso

Las colas de planificación enlazan las PCB de los procesos

9. Para poder implementar el algoritmo de planificación de CPU FCFS (primero que llega es el primero en ser atendido, First Come First Served) es indispensable contar con la Interrupción por Clock.

Falso

El algoritmo de planificación de CPU FCFS se basa en el momento de llegada de los procesos y es NO Apropiativo. La intervención de la interrupción por clock es cuando tengo restricciones acerca del tiempo de ejecución de los procesos, para evitar que un proceso se apropie de la CPU.

10. En modo usuario no se puede interactuar con el hardware, por ejemplo, leer de un disco

Verdadero

Cuando un proceso quiere acceder a HW para a modo supervisor: nunca accede directamente a los dispositivos

11. De las siguientes opciones, indicar en orden cuáles de ellas se dan cuando a un proceso que se encuentra en estado de ejecución se le termina su quantum asignado, tomando como referencia el algoritmo de planificación

de CPU Round Robin. Tenga en cuenta que no todas las opciones disponibles son válidas.

Ejecución de Short Term Scheduler

Resguardo de los datos del contexto del proceso saliente

Carga de los datos del contexto del proceso entrante

Cambio a Modo Usuario

Salto a la instrucción del proceso entrante

12. La PCB de un proceso que ejecuta una SysCall bloqueante, por ejemplo un write() a un archivo, será colocada en la cola de waiting o blocked hasta que la SysCall termine la cola de ready

13. Los algoritmos Apropiativos (Preemptive) para la planificación de la CPU son convenientes en procesos interactivo

Verdadero

El criterio de apropiación le permite al SO decidir si hay que ejecutar un proceso de mayor prioridad y darle la CPU rápidamente. No obstante, debe hacerse adecuadamente el cambio de contexto con el proceso que está en ejecución, para garantizar su integridad cuando se le vuelva a dar la CPU. Los procesos interactivos necesitan dar respuestas rápidas y por eso son más adecuados los criterios de apropiación.

14. Cómo detecta el HW que un proceso está intentando acceder a una dirección ilegal? Seleccione una:

- a. Porque no está dentro del rango delimitado por el registro base y el registro límite
- b. Porque el HW le solicita al Kernel que le confirme la validez o no de cada dirección
- c. Porque cuando quiere acceder en la memoria, esa dirección está marcada como no accesible
- d. Porque la dirección es mayor a la cantidad de bytes que tiene el proceso

15. El kernel se asegura que un proceso no se apropie de la CPU mediante: Seleccione una o más de una:

- a. La protección del vector de interrupciones
- b. La interrupción por clock
- c. El uso de llamadas al sistema

- d. Un proceso no se puede programar para que no se apropie de la CPU
- e. La existencia de los modos de ejecución (Kernel / Usuario)

16. Cuando un proceso ejecuta la System Call fork() tanto padre e hijo compartirán el mismo espacio de direcciones hasta que es ejecutada la System Call execve()

Falso

La system call fork() crea un nuevo espacio de direcciones para el proceso hijo. Esta system call devuelve dos resultados: Un resultado lo devuelve como resultado de la ejecución del fork en el espacio de direcciones original, y el otro resultado lo devuelve en el nuevo espacio de direcciones (del proceso nuevo) donde quedó la instrucción fork en esa "copia". El resultado en el nuevo espacio es lo que habilita que se ejecute la System Call execve() y se "cargue" allí, código, datos y stack del nuevo proceso

17. Windows muestra la pantalla azul o BSOD, cuando: Seleccione una:

- a. Siempre que haya un error que no permite que un proceso se pueda ejecutar
- b. Cuando ocurre una interrupción por SW (trap) estando en modo supervisor
- c. Cuando quiere acceder a un archivo que no existe o no está disponible

18. Un proceso es una entidad abstracta que incluye una sección de Código, una sección de datos y 1 o más stacks

Verdadero

19. En los algoritmos de planificación de CPU no apropiativos, el proceso en ejecución dejara la CPU cuando él decide

Verdadero

Los criterios no apropiativos tienen el riesgo de que un proceso puede apropiarse de la CPU por mucho tiempo. Es recomendable cuando hay procesos batch solamente, por ejemplo. Pero no debería optar por esa modalidad en ambientes donde necesito darle la CPU a procesos con mayor prioridad que puedan llegar.

20. Mientras que en un SO el KERNEL es el responsable de la interacción con el Hardware, el SHELL es el responsable de la interacción del Usuario

PARCIAL 2021

1. En un diseño monolítico del kernel, todos los componentes de un SO deben ejecutar en modo Kernel

Falso

2. En las técnicas de administración de memoria por paginación y segmentación la resolución de direcciones es realizada por el HW. No así la técnica de particiones (fijas o dinámicas), donde la resolución es realizada por el Kernel

Falso

3. Cuando un proceso quiere llamar a una System Call, la instrucción que se ejecuta es la que permite cargar en el contador de programa del procesador la dirección de la rutina que corresponde

Falso

4. En la paginación pura unique

- a. Es responsabilidad del Kernel el subir las páginas correspondientes de cada proceso
- b. Es responsabilidad del Kernel el mantener el estado de ocupación de los frames
- c. Es responsabilidad del HW determinar la validez o no de una dirección lógica
- d. Es responsabilidad del Kernel el armar las tablas de página

5. Cuando hay un cambio de contexto, el valor del registro PSW del proceso que se estaba ejecutando es guardado en la PCB de ese proceso

Verdadero

6. Todas las interrupciones por HW generar un cambio de contexto entre procesos

Falso

7. La PCB es una estructura de datos única que utiliza el Kernel para almacenar información de todos los procesos. Contiene una lista encadenada con los atributos de los procesos creados

Falso

8. **Cómo detecta el hardware que un proceso en MU está intentando ejecutar una instrucción privilegiada y qué ocurre?**
- a. Un proceso en MU puede ejecutar una instrucción privilegiada, por lo tanto no ocurre nada
 - b. Se detecta por el código de la instrucción. El HW produce un trap al SO
 - c. Se detecta por el código de la instrucción entonces el HW finaliza el proceso
 - d. Lo detecta porque esa instrucción no está en el espacio de direcciones del proceso. Da error y finaliza el proceso
9. **En modo usuario no se puede interactuar con el HW, por ejemplo mandar comandos de impresión a la impresora**
Verdadero
10. **La System Call fork() devuelve al proceso hijo el PID del proceso padre**
Falso
11. **De las siguientes opciones, indicar en orden cuáles de ellas se dan cuando a un proceso que se encuentra en estado de ejecución se le termina su quantum asignado, tomando como referencia el algoritmo de planificación de CPU Round Robin. Tenga en cuenta que no todas las opciones disponibles son válidas.**

Ejecución de Short Term Scheduler

Resguardo de los datos del contexto del proceso saliente

Carga de los datos del contexto del proceso entrante

Cambio a Modo Usuario

Salto a la instrucción del proceso entrante

12. **En la técnica de particiones dinámicas el Kernel es el responsable de fijar los valores de los registros base y límite de cada proceso, no así en la técnica de particiones fijas cuyos valores son definidos por la MMU**
Falso

- 13. El control para impedir el acceso indebido a memoria por parte de un procesador es realizado por:**
- a. El compilador al momento de crear el programa
 - b. El kernel
 - c. El HW
 - d. No es posible que un proceso haga un acceso indebido a memoria

- 14. Dadas las siguiente técnicas de administración de memoria principal indique que tipo de fragmentación pueden generar**
- a. **Particiones Fijas:** fragmentación interna
 - b. **Particiones Dinámicas:** fragmentación externa
 - c. **Segmentación:** fragmentación externa
 - d. **Paginación:** fragmentación interna
 - e. **Segmentación paginada:** no genera interna

- 15. En la paginación pura las páginas de un proceso se deben cargar en marcos (frames) consecutivos**
- Falso

- 16. En la técnica de segmentación de procesos son divididos en diferentes porciones, todas de igual tamaño, denominadas segmentos**
- Falso, los segmentos pueden ser de diferentes tamaños

- 17. La PCB de un proceso se almacena en el espacio de direcciones del mismo**
- Falso

- 18. Un proceso deberá realizar el cambio de modo de ejecución a MK, cambiando el bit de modo del procesador, antes de realizar la invocación a una llamada al sistema**
- Falso, se realiza la llamada y ahí se cambia

- 19. Las rutinas de atención de interrupciones, referenciadas en el vector de interrupciones, no requieren de CPU**
- Falso

- 20. Un cambio de modo de ejecución siempre causa que se realice un cambio de contexto**
- Falso

PARCIAL 2022

1. La ejecución de las rutinas de atención de interrupciones, referenciadas en el vector de interrupciones, no requieren de CPU.

Falso

2. En modo usuario no se puede interactuar con el hardware, por ejemplo mandar comandos de impresión a la impresora

Verdadero

3. Cuando un proceso quiere llamar a una system call, la instrucción que se ejecuta es la que permite cargar en el contador de programa (program counter, PC) del procesador la dirección de la rutina que corresponde a dicha la system call

Falso

Cuando un proceso intenta llamar a una syscall lo que se realiza es indicar el número correspondiente a la misma y luego generar una interrupción por soft (o trap)

4. El control para impedir el acceso indebido a memoria por parte de un proceso es realizado por:
 - a. No es posible que un proceso haga un acceso indebido a memoria
 - b. El Hardware
 - c. El Kernel
 - d. El compilador al momento de crear el programa

5. Una llamada al sistema (system call) genera la creación de un nuevo proceso del sistema operativo para atender la llamada.

Falso

6. En un diseño monolítico del kernel, todos los componentes de un SO se deben ejecutar en modo Kernel

Falso

7. En paginación pura las páginas de un proceso se deben cargar en marcos (Frames) consecutivos

Falso

8. En paginación pura indique:
 - a. Es responsabilidad del:
 - i. KERNEL el subir las páginas correspondientes de cada proceso.
 - ii. KERNEL el mantener el estado de los frames (libre u ocupado)

- iii. HW determinar la validez o no de una dirección lógica.
- iv. KERNEL el generar la tabla de página de cada proceso.

9. Dadas las siguientes técnicas de administración de memoria principal indique que tipo de fragmentación pueden generar:

- a. **Particiones Fijas:** fragmentación interna
- b. **Particiones Dinámicas:** fragmentación Externa
- c. **Paginación:** fragmentación interna

10. La PCB es una estructura de datos única que utiliza el Kernel para almacenar información de todos los procesos. Contiene una lista encadenada con los atributos de los procesos creado

Falso

11. Un cambio de modo de ejecución siempre causa que se realice un cambio de contexto

Falso

12. Indique que imprime el siguiente programa:

```
print 'Anda a rendir el Primer Parcial de Promo!'

newpid = fork()

if newpid == 0
    print 'Estoy comenzando el Examen'
    execv('ps ')
    print 'Termine el Examen'
endif

print '¿Como te fue?'

exit(0)

print 'Ahora anda a descansar'
```

En el padre imprime	En el hijo imprime
<ul style="list-style-type: none"> Anda a rendir el Primer Parcial de Promo! 	<ul style="list-style-type: none"> Estoy comenzando el Examen Listado de procesos en ejecución de linux Termine el Examen

i)

En el padre imprime	En el hijo imprime
<ul style="list-style-type: none"> Anda a rendir el Primer Parcial de Promo! ¿Como te fue? 	<ul style="list-style-type: none"> Listado de procesos en ejecución de linux

ii)

En el padre imprime	En el hijo imprime
<ul style="list-style-type: none"> Anda a rendir el Primer Parcial de Promo! ¿Como te fue? Ahora anda a descansar 	<ul style="list-style-type: none"> Estoy comenzando el Examen Listado de procesos en ejecución de linux

iii)

En el padre imprime	En el hijo imprime

--	--

--	--

<ul style="list-style-type: none"> Anda a rendir el Primer Parcial de Promo! ¿Como te fue? 	<ul style="list-style-type: none"> Estoy comenzando el Examen Listado de procesos en ejecución de linux
--	---

iv)

esta

En el padre imprime	En el hijo imprime
<ul style="list-style-type: none"> • Anda a rendir el Primer Parcial de Promo! • ¿Como te fue? • Ahora anda a descansar 	<ul style="list-style-type: none"> • Estoy comenzando el Examen • Listado de procesos en ejecución de linux • Termine el Examen

v)

13. Un proceso es una entidad abstracta que incluye una sección de Código, una sección de datos y 1 o más stacks

Verdadero

14. Las direcciones de retorno de las rutinas invocadas durante la atención de una SysCall son apiladas en:

Seleccione una:

- a. Stack en modo Usuario
- b. Stack en modo Kernel**
- c. Se usa una u otra indistintamente, según la que venía usando el proceso cuando llamo a la SysCal

15. Cuando un proceso quiere invocar una System Call, es el mismo proceso el que realiza el cambio a modo kernel para luego hacer un CALL a la rutina correspondiente.

Falso

16. Cuando hay un cambio de contexto, el valor del registro PSW (program status word, palabra de estado de programa) del proceso que se estaba ejecutando es guardado en la PCB de ese proceso

Verdadero

La PSW es un registro donde está almacenada información acerca de estado en ese momento de la ejecución, por ejemplo, entre otros datos, el modo de ejecución del proceso.

17. Durante un Cambio de Contexto (Context Switch), el despachador (dispatcher) cambia a Modo Usuario luego de realizar el salto a la instrucción del proceso que se va a ejecutar

Falso, antes de realizar el salto?

El pasaje a modo usuario debe ser ANTES de que comience a ejecutarse el proceso entrante. El dispatcher se ejecuta en modo supervisor y debe cuando se le da el control al nuevo proceso restringirse al modo usuario, para limitarlo a que se ejecute dentro de su espacio de direcciones.

18. La System Call fork() devuelve al proceso hijo el PID del proceso padre

Falso

19. Un proceso existe desde que el programa es cargado en el espacio de direcciones de memoria del mismo.

Falso, desde que se ejecuta

20. Si se aplica la planificación apropiativa por prioridades, cuando un proceso que llega a la cola de listos tiene más prioridad que el proceso que se está ejecutando, indique qué ocurre con el proceso que se está ejecutando:

Seleccione una:

- a. Es Swapeado
- b. Ninguna de las otras opciones
- c. Continúa su ejecución
- d. Pasa a estado de listo
- e. Pasa a estado de Espera