

Wrong answer

Pregunta 1

0 / 1 pts

La generación de procesamiento por lotes permitía la siguiente funcionalidad

- ☐ Correr programas de varios usuarios distintos en forma concurrente
- ☒ Correr varios programas en secuencia, cada programa de principio a fin
- ☐ Correr varios programas de un mismo usuario, todos en forma concurrente
- ☐ Correr programas críticos en modo privilegiado y programas normales en modo restringido

Wrong answer

Pregunta 2

0 / 1 pts

¿Cuál es el propósito del Kernel en un Sistema Operativo Multitarea?

- ☒ Permitir que varios programas corran al mismo tiempo previniendo conflictos entre ellos
- ☐ Garantizar que el procesador esté ocupado siempre
- ☐ Prevenir que el usuario tenga que administrar de forma directa los dispositivos de hardware
- ☐ Evitar condiciones de carrera entre procesos paralelos

Pregunta 3

1 / 1 pts

Esta capa de los sistemas operativos multitarea modernos incorpora programas básicos para manipulación de archivos y otras tareas menores.

☒ Util☐ Shell☐ Kernel☐ Bash

Pregunta 4

0 / 1 pts

En un sistema operativo Windows, el error produce una pantalla azul requiere...

☒ Una excepción no capturada del shell☐ Una excepción no capturada de un programa del kernel☐ Una excepción no capturada de acceso ilegal de memoria☐ Un error de hardware catastrófico

Wrong answer

Pregunta 5

0 / 1 pts

El shell siempre corre en modo kernel

- ☐ Verdadero, el shell es parte del kernel
- ☒ Falso, el shell corre comúnmente en modo usuario
- ☒ Falso, el shell sólo corre en modo kernel si el usuario tiene privilegios de administrador
- ☐ Verdadero, el shell no es parte del kernel, pero debe correr en modo kernel para operar

Wrong answer

Pregunta 6

0 / 1 pts

El Shell es una de interfaz entre el usuario y el kernel, siempre se presenta como una línea de comandos

- ☒ Verdadero, la interfaz que provee el shell siempre es una línea de comandos
- ☐ Verdadero, el manejador de ventanas siempre corre sobre un shell en modo texto
- ☒ Falso, existen los shell gráficos y los shell en modo texto
- ☐ Falso, el shell tiene como propósito administrar los archivos de la computadora

Pregunta 7

1 / 1 pts

El concepto de Sistema Operativo nace por la necesidad de...

- ☐ Poder acceder a los recursos de hardware de forma más sencilla
- ☐ Poder acelerar la ejecución de programas pesados
- ☐ Poder administrar sistemas con cada vez más dispositivos de entrada y salida
- ☒ Poder correr varios programas en una misma computadora

Pregunta 8

1 / 1 pts

Cada generación de sistemas operativos responde a una generación diferente de computadoras, empezando por la generación de Tubos al Vacío.

- ☐ Falso, las generaciones de sistemas operativos responden a las iniciativas de Ritchie, Gates y Jobs
- ☐ Verdadero, el sistema operativo de la ENIAC estaba especializado en cálculo de trayectorias balísticas
- ☐ Verdadero, en la generación de tubos al vacío se usaban sistemas operativos por lotes
- ☒ Falso, en la generación de tubos al vacío no había sistemas operativos

Pregunta 9

1 / 1 pts

Es posible hacer módulos personalizados para el kernel

- ☐ Falso, sólo el fabricante del sistema operativo puede programar módulos de kernel
- ☒ Verdadero, muchas aplicaciones comerciales de terceros tienen sus propios módulos
- ☐ Verdadero, pero sólo en Linux por su filosofía de software libre y código abierto
- ☐ Falso, sólo fabricantes de hardware tienen autorización para hacer módulos de kernel

Pregunta 10

1 / 1 pts

Esta características propuesta por Dennis Ritchie y Kenneth Thompson cambió por completo la forma en que se manipulan los almacenamientos secundarios, acercando el sistema operativo a usuarios finales y a la computación doméstica.

- ☐ Pipe
- ☒ Sistema de Archivos
- ☐ Shell
- ☐ Ventanas

Wrong answer

Pregunta 11

0 / 1 pts

Los sistemas por lotes utilizan algoritmos de tipo

- ☐ Aprehensivos
- ☒ No aprehehensivos
- ☐ Cíclicos
- ☒ Basados en colas

Wrong answer

Pregunta 12

0 / 1 pts

Seleccione la opción que tenga sólo algoritmos aprehehensivos

- ☒ Colas de Prioridad, Lotería, FIFO
- ☒ Shortest Job Next, Colas de Prioridad, Round Robin
- ☐ Shortest Job First, Colas de Prioridad, Lotería
- ☐ FIFO, Round Robin, Random

Correct answer

Pregunta 13

1 / 1 pts

Cuando un programa es invocado automáticamente se carga en este estado del ciclo de vida:

- ☐ Iniciado
- ☒ Listo
- ☐ En Ejecución
- ☐ Bloqueado

Wrong answer

Pregunta 14

0 / 1 pts

El algoritmo Round Robin es...

- ☐ Un algoritmo usado en sistemas multitarea
- ☒ La versión no aprehensiva de FIFO
- ☐ La versión aprehensiva de FIFO
- ☐ Un algoritmo usado en sistema por lotes

Pregunta 15

1 / 1 pts

En un ambiente multitarea el mejor algoritmo favorece procesos de ráfagas cortas...

- ☐ Falso, en ambientes multitarea se deben favorecer primero los procesos CPU-Bound
- ☐ Verdadero, porque los procesos de ráfagas cortas se interrumpen menos y aprovechan más el procesador
- ☐ Falso, eso produce que procesos grandes sufran de inanición
- ☒ Verdadero, porque aumenta la probabilidad de que todos los procesos puedan usar el procesador

Pregunta 16

0 / 1 pts

Este tipo de interrupción produce que el programa regrese el program counter y trate de volver a correr la línea que disparó la interrupción.

- ☒ Trap
- ☐ Interrupt
- ☒ Fallo
- ☐ System Call

Pregunta 17

1 / 1 pts

Sólo el algoritmo de colas de prioridad soporta el concepto de prioridades en calendarización



- ☒ Falso, el algoritmo de Lotería también permite prioridades
- ☐ Falso, el algoritmo de Round Robin también soporta prioridades
- ☐ Falso, el algoritmo de Shortest Job Next también soporta prioridades
- ☐ Verdadero, el único otro algoritmo es el de envejecimiento que es una versión especial de colas de prioridad

Pregunta 18

0 / 1 pts

Cuando un programa solicita recursos de entrada y salida pasa lo siguiente...



- ☐ El programa se bloquea y reanuda cuando el recurso esté disponible para su uso
- ☒ El programa accede al recurso transparentemente
- ☐ El programa se bloquea hasta que el proceso solicitado al recurso de E/S sea resuelto
- ☐ Se le entrega el recurso de inmediato y el programa continúa su ejecución

Wrong answer

Pregunta 19

0 / 1 pts

Cuando una interrupción se dispara, se crea una estructura de memoria especial que almacena el estado de la interrupción, esta se llama...

- ☐ Process Control Block
- ☒ Interrupt Vector Table
- ☐ Interrupt Handler
- ☐ Process Table

Wrong answer

Pregunta 20

0 / 1 pts

El siguiente algoritmo se introdujo para sistemas de tiempo compartido

- ☐ Shortest Job Next
- ☐ FIFO
- ☐ Round Robin
- ☒ Lotería


Wrong answer

Pregunta 21

0 / 1 pts

Suponga que tiene una función que ejecuta el siguiente código:

```
int* mat = malloc(sizeof(int) * 1000);  
t_pid pid = fork();
```



- ☒  ambos procesos comparten 4000B de memoria para mat
- ☐  cada proceso tiene reservados 4000B de memoria para mat
- ☐ el proceso padre tiene reservados 4000B de memoria para mat, el puntero mat es de sólo lectura para el hijo
- ☐ el proceso padre tiene reservados 4000B de memoria para mat, el puntero mat es inválido para el hijo

Wrong answer

Pregunta 22

0 / 1 pts

El operador Sleep bloquea un proceso que se desbloquea cuando...



- ☐ todos los hilos que están esperando se despiertan
- ☒  el recurso compartido es liberado por el proceso que lo tiene cautivo
- ☐ otro hilo o proceso invoca el operador Signal
- ☐  pasa el tiempo que se le indicó por parámetro

Wrong answer

Pregunta 23

0 / 1 pts

La comunicación entre procesos sólo puede darse entre proceso padre e hijo, nunca entre procesos hermanos

- ☒  Verdadero, porque sólo el padre conoce a los hijos, no se conocen entre hermanos
- ☐ Falso, la comunicación sólo puede darse entre hermanos, no entre padre e hijo
- ☒  Falso, nada impide la comunicación entre procesos hermanos
- ☐ Verdadero, porque un proceso sólo conoce al padre, no a los hermanos

Wrong answer

Pregunta 24

0 / 1 pts

La función `fork()` retorna

0: para el proceso hijo recién creado

N: para el proceso padre, donde N es el PID del hijo

Esto significa que es imposible para un proceso hijo conocer el PID de su padre

- ☐ Falso, hay una función que permite a un hijo pedir el PID de su padre
- ☒  Verdadero, por eso es más conveniente usar hilos que procesos
- ☒  Falso, el padre puede almacenar el PID antes de llamar a `fork()`
- ☐ Verdadero, por eso en algoritmos paralelos se monitorean los hijos, pero nunca se monitorean los padres

Wrong answer

Pregunta 25

0 / 1 pts

Suponga que tiene un sistema multinúcleo con 8 núcleos, dos hilos por núcleo, sin sincronización automática de caché

- ☐ En ese entorno es prohibido crear más de 16 hilos a la vez
- ☒ En ese entorno se pueden tener hasta 16 hilos simultáneos, u 8 procesos simultáneos
- ☐ En ese entorno es prohibido crear más de dos hilos a la vez
- ☒ En ese entorno se puede tener un máximo de 2 hilos, pero hasta 16 procesos corriendo en simultáneo

Wrong answer

Pregunta 26

0 / 1 pts

Cuando un hilo invoca la operación mutex.lock, le pasa lo siguiente

- ☐ El hilo siempre se bloquea hasta que el mutex le permita capturar el recurso
- ☒ El hilo dispara un trap que se libera cuando el mutex esté disponible
- ☒ Si el mutex está disponible, el hilo continúa su ejecución normal
- ☐ El hilo entra en modo privilegiado hasta desbloquear el mutex

Wrong answer

Pregunta 27

0 / 1 pts

¿Cuál de las siguientes opciones es una forma inválida de comunicar procesos entre sí?

- ☐ Escribir en un archivo virtual en memoria
- ☒ Enviar un mensaje por red
- ☒ Escribir en una variable definida en memoria dinámica
- ☐ Escribir en un archivo de disco duro

Wrong answer

Pregunta 28

0 / 1 pts

El mutex es una estructura de sincronización que permite controlar el acceso de dos hilos a una sección de memoria compartida

- ☐ Verdadero, si se quiere controlar el acceso de más de dos hilos se debe usar un semáforo
- ☒ Falso, el mutex permite controlar el acceso de múltiples hilos, no está limitado a dos
- ☐ Verdadero, el mutex es la base de las estructuras de sincronización para procesos
- ☒ Falso, el mutex permite controlar el acceso a una condición de carrera del programa

Pregunta 29

1 / 1 pts

¿Cuál de las afirmaciones es correcta?

- ☒ ☐ Concurrencia hace referencia a la simulación de paralelismo por medio de la rápida conmutación de recursos
- ☐ Paralelismo hace referencia a la simulación de concurrencia por medio de la rápida conmutación de recursos
- ☐ En ambientes multinúcleo no existe el paralelismo, sólo la concurrencia
- ☐ En ambientes multinúcleo no existe la concurrencia, sólo el paralelismo

Pregunta 30

0 / 1 pts

A los hilos también se les llama subprocesos ligeros porque...

- ☒ ☐ Comparten memoria entre sí, haciéndolos más fáciles de manipular que los procesos
- ☐ No permiten paralelizar tanto como los subprocesos creados con fork
- ☒ ☐ No clonan la memoria del proceso padre, y no tienen su propio process control block
- ☐ No clonan la memoria del proceso padre, su process control block comparte los rangos de memoria con el padre