

Wrong answer

Pregunta 1

0 / 1 pts

Suponga que tiene una función recursiva que no tiene una adecuada condición de parada.

Al correr ese programa, ¿qué sucedería?

- El sistema operativo marcaría el programa como no-responsivo
- Una excepción de Segmentation Fault
- El programa correría indefinidamente
- Una excepción de Stack Overflow

Correct answer

Pregunta 2

1 / 1 pts

¿Cuántas secciones tiene el modelo de memoria estilo C?

- 4
- 3
- 1
- 5

Wrong answer

Pregunta 3

0 / 1 pts

Cuando se reserva un nuevo segmento de memoria para un programa, este se enlaza en el process control block para protegerlo de ser escrito por otros programas

- Verdadero, cuando un programa trata de acceder a esta memoria se dispara una excepción de desborde de pila
- Verdadero, cuando un programa trata de acceder a esta memoria se dispara una excepción de escritura prohibida
- Verdadero, cuando un programa trata de acceder a esta memoria se dispara un fallo de segmentación
- Falso, la memoria no se protege en el process control block

Wrong answer

Pregunta 4

0 / 1 pts

Durante la ejecución de una función, todas las variables que se crean se almacenan en cuál segmento

- En el segmento de texto
- En el segmento de monte
- En el segmento de datos
- En el segmento de pila

Correct answer

Pregunta 5

1 / 1 pts

¿En cuál sección de memoria se almacena el código fuente del programa?

Segmento de Pila

Segmento de Datos

 Segmento de Texto

Segmento de Monte

Wrong answer

Pregunta 6

0 / 1 pts

En el Modelo de Memoria Estilo C existen varias secciones de sólo lectura

 Verdadero, hay dos secciones de sólo lectura

 Verdadero, hay tres secciones de sólo lectura

Falso, todas las secciones pueden modificarse

Verdadero, el código fuente es de sólo lectura

Correct answer

Pregunta 7

1 / 1 pts

Seleccione la opción que mejor describa qué es la **tabla de símbolos** de un programa.

- Es una estructura de datos del entorno de ejecución que asocia todos los tipos de datos con su tamaño
- Es una sección de memoria que guarda todos los valores literales usados en el programa
- Es una lista que relaciona todos los nombres de variables con sus respectivas direcciones de memoria
- Es una lista de todos los segmentos de memoria protegidos por el programa

Wrong answer

Pregunta 8

0 / 1 pts

Considere este código fuente

```
#include<stdio.h>

int* func()
{
    int v[10] = {1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10};
    return v;
}

int main() {
    int* a;
    a = func();
    printf("Value of 5 is %d", a[5]);
}
```

Indique cuál es el resultado de ese programa.

Pregunta 8

0 / 1 pts

Considere este código fuente

```
#include<stdio.h>

int* func()
{
    int v[10] = {1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10};
    return v;
}

int main() {
    int* a;
    a = func();
    printf("Value of 5 is %d", a[5]);
}
```

Indique cuál es el resultado de ese programa.

→ SegFault, porque la memoria a la que apunta `a` ya fue liberada

5

✗ SegFault, porque la memoria a la que apunta `a` está definida en el segmento de datos

No compila porque no se puede guardar un `int[]` en un `int*`

Correct answer

Pregunta 9

1 / 1 pts

¿En cuál sección de memoria se almacenan los valores literales que se definen durante el programa?

- En el segmento de Texto
- En el segmento de Pila
- En el segmento de Datos
- En el segmento de Monte

Wrong answer

Pregunta 10

0 / 1 pts

Suponga que tiene esta línea de código y que su programa está compilado para arquitectura de 64 bits.

```
char* str = 'holá';
```

Ignorando el peso del código fuente mismo

¿Cuánta memoria se asigna relacionada con esa línea?

- 4B en el segmento de datos y 4B en el segmento de texto
- 8B en el segmento de datos y 4B en el segmento de Monte
- 8B en el segmento de pila y 4B en el segmento de datos
- 4B en el segmento de pila y 8B en el segmento de datos

Wrong answer

Pregunta 11

0 / 1 pts

Suponga que tiene la siguiente secuencia de accesos a memoria, en un ambiente con sólo 4 marcos que inicia con el estado [1 9 3 8], suponga que cada página de las que ya fueron cargadas se ha usado una sola vez:

5 2 1 4 5 2 1 4 5 1 5 1

¿Cuántos fallos de página presenta esta secuencia usando el algoritmo Óptimo?

- 4, y el estado final es [5, 2, 1, 4]
- 3, y el estado final es [5, 2, 1, 4]
- 4, y el estado final es [1, 5, 4, 2]
- 3, y el estado final es [1, 5, 2, 4]

Wrong answer

Pregunta 12

0 / 1 pts

El algoritmo Second Chance

- Es un algoritmo off-line
- Es un algoritmo que conserva páginas menos usadas
- Es un algoritmo conservador
- Es un algoritmo que conserva páginas más viejas

Correct answer

Pregunta 13

1 / 1 pts

El algoritmo de FIFO prefiere

- páginas más viejas para conservar
- páginas más viejas para intercambiar
- procesos más viejos para conservar
- procesos más viejos para intercambiar

Correct answer

Pregunta 14

1 / 1 pts

La estrategia de paginación surgió porque

- La estrategia de segmentación dinámica estaba atrasando la ley de Moore
- La arquitectura de 32 bits no podía direccionar más de 4GB de memoria RAM
- El problema de fragmentación externa no permitía crecer en la capacidad de multitarea
- Los sistemas interactivos requieren más memoria de la que se puede segmentar dinámicamente

Wrong answer

Pregunta 15

0 / 1 pts

¿Cuál de estos algoritmos de paginación se considera que opera off-line?

- MRU
- OPT
- FIFO
- MFU

Correct answer

Pregunta 16

1 / 1 pts

¿Cuál de estos algoritmos tiende a conservar páginas más viejas que no se han usado recientemente?

- LFU

En casos en los que una sección de memoria se usa con mucha frecuencia al inicio del programa y es abandonada cuando el programa avanza, el LFU puede conservar las páginas asociadas por su alta frecuencia reportada.

- FIFO
- MRU
- Second Chance

Correct answer

Pregunta 17

1 / 1 pts

Dada una petición de memoria que requiere K páginas y existen H marcos, un algoritmo conservador

- Garantiza que ocurrirán K fallos de página o menos, sólo si $K \geq H$
- Garantiza que ocurrirán K fallos de página o menos, sólo si $H \geq K$
- Garantiza que ocurrirán al menos K fallos de página
- Garantiza que ocurrirán H fallos de página o menos

Wrong answer

Pregunta 18

0 / 1 pts

Según el supuesto de localidad temporal

- Una página usada recientemente será usada pronto
- Una página usada recientemente indica que será liberada pronto
- Una página usada recientemente no será usada pronto
- Una página usada recientemente indica que páginas cercanas serán usadas pronto

Correct answer

Pregunta 19

1 / 1 pts

Suponga que tiene la siguiente secuencia de accesos a memoria, en un ambiente con sólo 4 marcos que inicia con el estado [1 9 3 8], suponga que cada página de las que ya fueron cargadas se ha usado una sola vez:

5 2 1 4 5 2 1 4 5 1 5 1

¿Cuántos fallos de página presenta esta secuencia?

- 3 fallos usando el algoritmo de FIFO
- 5 fallos usando el algoritmo MRU
- 4 fallos usando el algoritmo FIFO
- 7 fallos usando el algoritmo MRU

Wrong answer

Pregunta 20

0 / 1 pts

El algoritmo MRU favorece

- páginas recién creadas para intercambiar
- páginas más usadas para conservar
- páginas recién creadas para conservar
- × páginas más usadas para intercambiar

Correct answer

Pregunta 21

1 / 1 pts

La Fragmentación Externa surge porque

- La memoria no se libera adecuadamente por los procesos zombies
- Las solicitudes de memoria no coinciden con el tamaño estándar de los segmentos de memoria
- Conforme avanza la sesión incrementa la frecuencia de solicitudes de menor tamaño
- Al dividir la memoria quedan restos que no se pueden asignar a ningún proceso

Correct answer

Pregunta 22

1 / 1 pts

La Fragmentación Interna es un problema que se desea evitar porque

- Empeora conforme avanza la sesión
- Desperdicia memoria en cada asignación
- Produce mucho overhead de procesamiento para ubicar segmentos disponibles
- Genera muchos segmentos que no se pueden asignar a procesos nuevos

Correct answer

Pregunta 23

1 / 1 pts

El algoritmo Best Fit tiene los siguientes supuestos

- ④ Es mejor generar fragmentos lo más pequeños posible
- ① Es mejor generar fragmentos lo más cerca posible del inicio de la RAM
- ③ No tiene supuestos
- ② Es mejor generar fragmentos lo más grandes posibles

Wrong answer

Pregunta 24

0 / 1 pts

La compactación es una medida de emergencia que se usa cuando

- ➔ ④ Se rechazan muchas solicitudes de memoria producto de la fragmentación externa
- ① Se cae en estado de Thrashing
- ✗ ③ No se encuentra un segmento de memoria de tamaño adecuado para una solicitud
- ② Se rechazan muchas solicitudes de memoria producto de la fragmentación interna

Correct answer

Pregunta 25

1 / 1 pts

El método de Segmentación Dinámica de Memoria es el que desperdicia menos memoria.

- Falso, porque la memoria genera muchos fragmentos que no se pueden asignar
- Falso, porque la memoria que se desperdicia aumenta conforme avanza la sesión
- Verdadero, porque sólo se asigna la memoria que se solicita y no más.
- Verdadero, porque la memoria desperdiciada se libera cuando acaba el proceso que la tiene capturada