Comenzado el	Thursday, 25 de November de 2021, 14:00
Estado	Finalizado
Finalizado en	Thursday, 25 de November de 2021, 15:42
Tiempo empleado	1 hora 42 minutos
Puntos	13,80/14,00
Calificación	**9,86** de 10,00 (**99**%)

Pregunta **1**

Correcta

Puntúa 1,00 sobre 1,00

Indicar si el siguiente multiprograma termina.
Suponga atomicidad línea a línea.
Inicialmente x,y=25,25.

```
while(0<x+y<100){
    x=x+1
    y=y+1
}
```

```
while (0<x+y<100) {
    x=x-1
    y=y+1
}
```

```
while (0<x+y<100) {
    x=x+1
    y=y-1
}
```

```
while (0<x+y<100) {
    x=x-1
    y=y-1
}
```

Seleccione una:

- a. Nunca
- b. A veces ✔
- c. Siempre

Respuesta correcta

La respuesta correcta es: A veces

Pregunta **2**

Correcta

Puntúa 1,00 sobre 1,00

Para el siguiente multiprograma de dos componentes, **suponga atomicidad línea a línea**.
Inicialmente **x=4**.

```
x=x+2          x=x-1
```

Indique que valores finales puede tomar **x**.

Seleccione una o más de una:

- [ ] a. x=0
- [ ] b. x=1
- [ ] c. x=2
- [ ] d. x=3
- [ ] e. x=4
- [x] f. x=5
- [ ] g. x=6
- [ ] h. x=7

Respuesta correcta

La respuesta correcta es: x=5

Pregunta **3**

Correcta

Puntúa 1,00 sobre 1,00

El siguiente es multiprograma de dos componentes.
No suponga atomicidad, es decir cada incremento o decremento es: leer la memoria, operar, escribir en la memoria.
Inicialmente **x=4**.

```
x=x+2          x=x-1
```

Indique que valores finales puede tomar **x**.

Seleccione una o más de una:

- [ ] a. x=0
- [ ] b. x=1
- [ ] c. x=2
- [x] d. x=3 ✔
- [ ] e. x=4
- [x] f. x=5 ✔
- [x] g. x=6 ✔
- [ ] h. x=7

Respuesta correcta

Las respuestas correctas son: x=3, x=5, x=6

Pregunta **4**

Correcta

Puntúa 1,00 sobre 1,00

Suponga atomicidad línea a línea. La variable **i** y el arreglo **a** son compartidos.
Inicialmente i=2 y a=[2,2,2,2].

```
while(0<=i<4) {     while(0<=i<4) {
    a[i]=0              a[i]=1
    i++                 i--
}                   }
```

Indique si el multiprograma termina.

Seleccione una:

- ( ) a. Siempre
- (•) b. A veces ✔
- ( ) c. Nunca

Respuesta correcta

La respuesta correcta es: A veces

Pregunta **5**

Parcialmente correcta

Puntúa 0,80 sobre 1,00

Suponga atomicidad línea a línea. La variable **i** y el arreglo **a** son compartidos.
Inicialmente i=2 y a=[2,2,2,2].

```
while(0<=i<4) {        while(0<=i<4) {
   a[i]=0                  a[i]=1
   i--                     i++
}                      }
```

Indique que valores puede tomar el arreglo **a** del multiprograma cuando termina

Seleccione una o más de una:

- [x] a. [2,2,1,1]
- [ ] b. [2,2,0,0]
- [ ] c. [2,2,2,2]
- [x] d. [0,0,0,0]
- [ ] e. [1,1,0,0]
- [x] f. [0,0,0,2]
- [ ] g. [0,0,1,1]
- [x] h. [1,1,1,1]

Respuesta parcialmente correcta.

Ha seleccionado correctamente 4.
Las respuestas correctas son: [0,0,0,2], [2,2,1,1], [0,0,0,0], [1,1,1,1], [0,0,1,1]

Pregunta **6**

Correcta

Puntúa 1,00 sobre 1,00

Suponga atomicidad línea a línea. La variable **i** y el arreglo **a** son compartidos.
Inicialmente i=2, a=[2,2,2,2], s=1 y t=0

```
while(0<=i<4) {        while(0<=i<4) {
    sem_wait(s)            sem_wait(t)
    a[i]=0                 a[i]=1
    i--                    i++
    sem_post(t)            sem_post(s)
}                      }
```

Indique que valores puede tomar el arreglo a del multiprograma cuando termina

Seleccione una o más de una:

- [x] a. No termina ✔
- [ ] b. a=[1,1,1,1]
- [ ] c. a=[0,...,0]
- [ ] d. a=[1,0,1,0]

Respuesta correcta

La respuesta correcta es: No termina

Pregunta **7**

Correcta

Puntúa 1,00 sobre 1,00

Se tiene la siguiente implementación de locks.

```
typedef struct __lock_t {
  int flag;
} lock_t;
```

```
void init(lock_t *mutex) {
  // 0 -> disponible, 1 -> tomado
  mutex->flag = 0;
}
```

```
void lock(lock_t *mutex) {
  for(int i=N; 0<i; i--) {
    while (mutex->flag == 1); // testeo bien la bandera
  }
  mutex->flag = 1; // finalmente la tengo
}
```

```
void unlock(lock_t *mutex) {
  mutex->flag = 0; // devuelvo la bandera
}
```

Decir si funciona:

Seleccione una:

- a. Nunca
- b. Siempre, si N es lo suficientemente grande.
- c. A veces ✔
- d. Siempre

Respuesta correcta

La respuesta correcta es: A veces

Pregunta **8**

Correcta

Puntúa 1,00 sobre 1,00

Para estos tres discos indicar cual es el más rápido en **lecturas aleatorias de bloques de 128 KiB**.

Marca/Modelo	RPM	Seek time	Transferencia
A	5500	9 ms	100 MiB/s
B	15000	13 ms	200 MiB/s
C	12500	3 ms	12 MiB/s

A Intermedio ✔

B Más rápido ✔

C Más lento ✔

Respuesta correcta

La respuesta correcta es: A → Intermedio, B → Más rápido, C → Más lento

Pregunta **9**

Correcta

Puntúa 1,00 sobre 1,00

¿Cuántos accesos a disco son necesarios para leer el directorio `/lib/modules/5.14.0-4-amd64/kernel/crypto/`?

Suponga como en el práctico que todo ocupa 1 bloque: i-bitmap, d-bitmap, cada inodo, cada directorio, cada archivo.
Aclaración: no se lee el superbloque.

Respuesta: 12

La respuesta correcta es: 12

Pregunta **10**

Correcta

Puntúa 1,00 sobre 1,00

Se tiene un FS tipo UNIX con bloques de 512 bytes.
En los i-bitmap y d-bitmap, un 0 es libre, un 1 ocupado.
Las estructuras de datos en disco son las siguientes.

```
free inode bitmap | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

free block bitmap | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
```

nombre	inodo
gameover	1
camelot	2
phantis	1
.	0
..	0
phantomas	4
aspar	3

Figura 3: Rootdir

inodo	size	nlink	dblocks
0	512	1	2
1	1027	2	7 8
2	128	1	1
3	721	1	3 4 5
4	512	1	0
5	77	0	10

Figura 4: Tabla de inodos

Indicar que inodos tienen **inconsistencias** en el **tamaño**.

Seleccione una o más de una:

- [ ] a. inodo 0
- [x] b. inodo 1
- [ ] c. inodo 2
- [x] d. inodo 3
- [ ] e. inodo 4
- [ ] f. inodo 5

Respuesta correcta

Las respuestas correctas son: inodo 1, inodo 3

Pregunta **11**

Correcta

Puntúa 1,00 sobre 1,00

Se tiene un FS tipo UNIX con bloques de 512 bytes.
En los i-bitmap y d-bitmap, un 0 es libre, un 1 ocupado.
Las estructuras de datos en disco son las siguientes.

```
                  -------------------------------
free inode bitmap | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
                  -----------------------------------------------------------
free block bitmap | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
                  -----------------------------------------------------------
```

nombre	inodo
gameover	1
camelot	2
phantis	1
.	0
..	0
phantomas	4
aspar	3

Figura 3: Rootdir

inodo	size	nlink	dblocks
0	512	1	2
1	1027	2	7 8
2	128	1	1
3	721	1	3 4 5
4	512	1	0
5	77	0	10

Figura 4: Tabla de inodos

Indicar las **inconsistencias en el i-bitmap**, o sea el free inode bitmap.
Lo inodos se numeran desde 0 en el i-bitmap.

Seleccione una o más de una:

- [ ] a. 0
- [ ] b. 1
- [ ] c. 2
- [ ] d. 3
- [x] e. 4 ✔
- [x] f. 5 ✔
- [ ] g. 6
- [ ] h. 7

Respuesta correcta

Las respuestas correctas son: 4, 5

Pregunta **12**

Correcta

Puntúa 1,00 sobre 1,00

Se tiene un FS tipo UNIX con bloques de 512 bytes.
En los i-bitmap y d-bitmap, un 0 es libre, un 1 ocupado.
Las estructuras de datos en disco son las siguientes.

```
                  -------------------------------
free inode bitmap | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
                  -----------------------------------------------------------------
free block bitmap | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
                  -----------------------------------------------------------------
```

nombre	inodo
gameover	1
camelot	2
phantis	1
.	0
..	0
phantomas	4
aspar	3

Figura 3: Rootdir

inodo	size	nlink	dblocks
0	512	1	2
1	1027	2	7 8
2	128	1	1
3	721	1	3 4 5
4	512	1	0
5	77	0	10

Figura 4: Tabla de inodos

Indicar que inodos tienen **inconsistencia** en el **campo nlink**.

Seleccione una o más de una:

- [x] a. inodo 0
- [ ] b. inodo 1
- [ ] c. inodo 2
- [ ] d. inodo 3
- [ ] e. inodo 4
- [ ] f. inodo 5

Respuesta correcta

La respuesta correcta es: inodo 0

Pregunta **13**

Correcta

Puntúa 1,00 sobre 1,00

En un sistema de archivos de tipo UNIX medio rarito, tenemos los bloques de disco dispuestos dentro del inodo con 4 punteros directos, 2 punteros indirectos y 2 punteros doble indirecto.
Cada bloque es de 4 KiB y los números de bloque ocupan 32 bits.
Calcule la **capacidad máxima de un archivo** en **MiB**.

Respuesta: 8200

La respuesta correcta es: 8200

Pregunta **14**

Correcta

Puntúa 1,00 sobre 1,00

En un sistema de archivos de tipo UNIX medio rarito, tenemos los bloques de disco dispuestos dentro del inodo con 4 punteros directos, 2 punteros indirectos y 2 punteros doble indirecto.
Cada bloque es de 4 KiB y los números de bloque ocupan 32 bits.
Calcule la **sobrecarga máxima de un archivo** en **KiB**.

Respuesta: 8208

La respuesta correcta es: 8200

◄ Persistencia

Ir a...

Consideraciones generales sobre los Laboratorios ►