



Arquitectura y Sistemas Operativos – Plan 2024 – Profesor Agustin Martinez

Segundo parcial – Comisión Nº5 – Fecha 27/05/2025

- **Condición mínima de aprobación: 60% del examen correctamente resuelto. (6 puntos)**
- **Todo el contenido debe ser escrito en lapicera para ser corregido.**
- **Todas las respuestas deben estar justificadas para ser tenidas en cuenta**

Nombre y apellido.....DNI.....

1 (2 Puntos)	2 (1 Puntos)	3 (2 Puntos)	4 (3 puntos)	5 (2 puntos)	Calificación

1) Indique si las siguientes sentencias son verdaderas o falsas. JUSTIFICAR LAS FALSAS

- a) El algoritmo de Peterson y Dekker es una estrategia para la sincronización de procesos. **F**
Para resolver el problema de la exclusión mutua.
- b) Al módulo de E/S se le da control directo de la memoria mediante DMA. **Verdadero**
- c) Si el valor del semáforo mutex/binario es 2 quiere decir que 2 procesos pueden acceder simultáneamente a la Sección Crítica. **Falso**

2) Seleccione con una cruz la opción correspondiente según el enunciado:

La función pthread_join (pthread, null)

- Cierra el hilo pasado por parámetro.
- Devuelve el identificador del hilo pasado por parámetro.
- Garantiza que el hilo pasado por parámetro finalice siempre. **X**
- Crea un nuevo hilo y lo ejecuta.

¿Cuál de los siguientes dispositivos de E/S entra en la categoría de “Legibles para el usuario”?

- Disco.
- Modem.
- Mouse. **X**
- Ninguno de los anteriores.

3) Explique BREVEMENTE los siguientes conceptos:

- Deadlock.
- Livelock.



4) Evalúe cuidadosamente el siguiente código y responda:

```
0 int x = 7;
1
2 void thread_code1() {
3     for(int i=0 ; i<2 ; i++){
4         x+=2;
5         printf("El valor de x es: %d \n", x);
6         pthread_exit(0);
7     }
8 }
9 void thread_code2() {
10
11     if(x==5){
12         sleep(1);
13         printf("Valgo %d\n",x);
14     }
15     pthread_exit(0);
16 }
17
18 int main() {
19     pthread_t hilo1, hilo2;
20
21     pthread_create(&hilo2, NULL, thread_code1, NULL);
22     pthread_create(&hilo1, NULL, thread_code2, NULL);
23
24     x=5;
25     pthread_join(hilo1, NULL);
26 }
27 }
28
29
```

- A. ¿Siempre terminan todos los threads? ¿Por qué? **NO**
- B. ¿Puede pasar que se imprima el mensaje "Valgo 9" por consola?, ¿y "Valgo 7"? **Justifique si es necesario. Valgo 9 no, valgo 7 si.**



5) Se tienen los siguientes tres hilos: p1, p2 y p3, que son ejecutados concurrentemente. Escribe las instrucciones lógicas restantes e inicialice los semáforos que considere necesarios de tal forma que X (variable global) muestre: 2, 1, 11, 12, 6, 16, 17, 27, 28... siendo x = 1

SemA=0
SemB=1
SemC=0

<pre>void p1() { while (true) { SemWait(A) if(x%2==0) { x=x/2; print(x); } SemPost(C) } }</pre>	<pre>void p2() { while (true) { SemWait(B) x+=1; print(x); SemPost(A) } }</pre>	<pre>void p3() { while (true) { SemWait(C) x+=10; print(x); SemPost(B) } }</pre>
--	---	--