

## UF2. [PAC03] Repaso UF2

Para la realización de las siguientes cuestiones deberéis ayudaros del libro de texto, capítulos 1 y 2, y consultar, si lo creéis necesario, internet.

Debéis subir un único archivo comprimido que contenga UN ÚNICO DOCUMENTO PDF con las respuestas del test.

Todas las preguntas valen lo mismo, cada pregunta tiene un valor de 0,5 puntos.

Las preguntas tipo test se corregirán sólo en la caja que veréis después de las preguntas.

Recordad que la fecha límite para la entrega de la PAC es el día 6 de ABRIL.

## Actividades

### Parte teórica

Responde a este test en la matriz de respuestas que se encuentra al final de las preguntas:

1. Señala la respuesta correcta:
  - a. Se tarda mucho menos tiempo en crear un nuevo hilo en un proceso existente que en crear un nuevo proceso.
  - b. Se tarda mucho menos tiempo en terminar un hilo que un proceso.
  - c. Se tarda mucho menos tiempo en conmutar entre hilos de un mismo proceso que entre procesos.
  - d. Todas las respuestas son correctas
2. Señala la respuesta correcta:
  - a. Un hilo dentro de un proceso se ejecuta secuencialmente.
  - b. Cada hilo tiene su propia pila y contador de programa.
  - c. Pueden crear sus propios hilos hijos.
  - d. Todas las respuestas son correctas.

3. ¿En qué se diferencian los hilos de los procesos?
  - a. No comparten la CPU.
  - b. Los hilos no son independientes entre sí, mientras que los procesos son independientes.
  - c. Los procesos no son independientes entre sí, mientras que los hilos son independientes.
  - d. Los hilos no crean sus propios hijos mientras que los procesos sí.
4. Los hilos no comparten:
  - a. Contador del programa.
  - b. Instrucciones.
  - c. Variables globales.
  - d. Ficheros abiertos.
5. Con la función `fork()`:
  - a. Creamos un hilo.
  - b. Creamos un proceso.
  - c. Ejecutamos un proceso.
  - d. Ejecutamos un hilo.
6. ¿Qué es un proceso zombie?
  - a. Aquel que está en ejecución.
  - b. Es lo mismo que un proceso huérfano.
  - c. Un proceso que ha terminado pero que sus recursos no han sido liberados.
  - d. Un proceso que hemos revivido.
7. ¿Cuál es la sentencia para que el proceso padre espere a que finalice el hijo?
  - a. `Wait(NULL)`
  - b. `Wait(PID_HIJO)`
  - c. `Pipe(NULL)`
  - d. `Pipe(PID_HIJO)`
8. ¿Para qué utilizamos la función `kill()`?
  - a. Para matar un proceso.
  - b. Para pausar un proceso.
  - c. Para esperar por un proceso.
  - d. Para enviar una señal.

9. ¿Cuál de las siguientes sentencias utilizarías para que un proceso se suspenda 1 segundo?
- Sleep(1)
  - Sleep(1000)
  - Pause(void)
  - Sleep(void)
10. Tenemos un proceso ABUELO, un proceso HIJO y un proceso NIETO, ¿cuál de estas comunicaciones no se puede hacer directamente?
- Del HIJO al NIETO.
  - DEL HIJO al PADRE.
  - DEL ABUELO al NIETO.
  - Todas se pueden realizar directamente.
11. Señala la respuesta correcta:
- La programación concurrente aprovecha mejor la CPU.
  - La programación concurrente reduce la velocidad de ejecución.
  - Ambas respuestas son correctas.
  - Ambas respuestas son incorrectas.
12. Tenemos las siguientes instrucciones:
- $$I1 = a = x + 1;$$
- $$I2 = b = x + z;$$
- $$I3 = c = x - 1;$$
- ¿Cuál de las siguientes respuestas es la correcta?
- Las instrucciones I1 e I3 son las únicas que se pueden ejecutar concurrentemente.
  - Las instrucciones I1 e I2 son las únicas que se pueden ejecutar concurrentemente.
  - Las tres instrucciones se pueden ejecutar concurrentemente.
  - Todas las instrucciones se tienen que ejecutar de forma secuencial porque todas leen la variable x.
13. ¿Cuáles son los problemas inherentes a la programación concurrente?
- Exclusión mutua y región crítica.
  - Región crítica y condición de sincronización.
  - Condición de sincronización y exclusión mutua.
  - Exclusión mutua, región crítica y condición de sincronización.

14. ¿Qué función utilizamos para iniciar un hilo?

- a. Run()
- b. Start()
- c. Sleep()
- d. Stop()

15. Tenemos la siguiente clase definida:

```
Public class Reloj extends Applet implements Runnable {}
```

¿Cuál es la interface?:

- a. Class
- b. Applet
- c. Runnable
- d. Ninguna de ellas es una interface.

16. En una ejecución tenemos dos hilos que han realizado un wait(), ¿Cómo despertamos a los dos hilos?

- a. Con un notify().
- b. Con un notifyAll().
- c. No los tenemos que despertar porque se despiertan solos.
- d. No hay forma de despertarlos.

17. ¿Qué significa el valor 0 que devuelve un fork()?

- a. Se ha producido un error.
- b. Es el PID del proceso hijo.
- c. Es el PID del proceso padre.
- d. Nos encontramos en el proceso hijo.

18. ¿Qué significa el valor -1 que devuelve un fork()?

- a. Se ha producido un error.
- b. Es el PID del proceso hijo.
- c. Es el PID del proceso padre.
- d. Nos encontramos en el proceso hijo.

19. ¿Qué significa el valor 678 que devuelve un fork()?

- a. Se ha producido un error.
- b. Es el PID del proceso hijo.
- c. Es el PID del proceso padre.
- d. Nos encontramos en el proceso hijo.

20. Tenemos las siguientes instrucciones:

$I1 = a = x + 1;$

$I2 = b = a + z;$

$I3 = x = z - 1;$

¿Cuál de las siguientes respuestas es la correcta?

- a. Las instrucciones I1 e I2 son concurrentes, pero I3 no.
- b. Las tres instrucciones son secuenciales.
- c. La instrucción I3 no entra en conflicto con I1 ni con I2.
- d. Las instrucciones I2 e I3 son concurrentes, pero I1 no.

**Matriz de respuestas**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

**Entrega:** Entregar la solución en una carpeta comprimida con el nombre nombre\_apellido1\_apellido2\_NúmeroUF\_NumeroPAC donde se incluya todos los archivos y documentos generados necesarios para la solución de estos ejercicios.