PAC DESARROLLO UF2

CFGS Desarrollo de Aplicaciones Multiplataforma

Módulo 9: Programación de servicios y procesos

1S 2018/2019

INFORMACIÓN IMPORTANTE

Para la correcta realización de la PAC el alumno deberá consultar los contenidos recogidos en la **Unidad Formativa 2** del material didáctico.

Requisitos que deben cumplirse en vuestros trabajos:

- Todas las PACs de desarrollo se enviarán únicamente a través de la plataforma dentro de los plazos de entrega establecidos en la guía didáctica. En caso de no cumplir dichos plazos, **NO** se podrán enviar de forma posterior.
- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado esta información.
- No se aceptarán copias literales de Internet. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas debe ser de elaboración propia.
- Las respuestas deben estar debidamente argumentadas. No se admiten respuestas escuetas.
- Es responsabilidad del alumno comprobar que el archivo subido en la plataforma es el correcto, ya que en ningún caso el profesor revisará el documento antes del periodo de corrección.
- El día y hora máximo para entregar una PAC de desarrollo es el día especificado en la **guía didáctica**.
- Si no se entrega una PAC, la calificación equivaldrá a un 0.
- Si el proyecto no compila, la puntuación máxima será un 4.
- Si se detecta que dos alumnos presentan dos PAC iguales la nota se dividirá entre dos, aspirando cada alumno a un 50% de la nota como máximo.

CRITERIOS DE CORRECCIÓN

1. Las PAC disponen de una calificación numérica que oscila del **0 al 10**. Respecto a la calificación de cada PAC de desarrollo, el profesor podrá **disminuir hasta 1 punto la nota obtenida en caso de que la PAC contenga errores ortográficos y/o su presentación no se adecúe a los estándares** establecidos por el profesor.
2. Debéis redactar las respuestas en **color azul oscuro o negro**.
3. Podéis utilizar la opción de negrita y subrayado para resaltar palabras clave, enunciados, etc., **NUNCA** para responder la totalidad de la actividad. No se podrá utilizar la función de resaltado.
4. La actividad debe ser redactada en minúsculas siguiendo las normas ortográficas básicas.

ENTREGA DE LA PAC

Se debe entregar una memoria en formato pdf contestando las preguntas siguientes. Si hay un ejercicio de desarrollar un programa poner el código que se vea claramente. Si el profesor lo cree oportuno, se pedirá los archivos para poder ejecutarlo.

ÍNDICE

1. Ejercicio 1.

Realiza un esquema de los estados de un proceso cualquiera, e identifica lo que ocurre en cada uno de ellos.

2. Ejercicio 2.

A partir del siguiente conjunto de instrucciones, indica las que se pueden ejecutar concurrentemente y las que no. Justifica tus respuestas.

Instrucción 1: a = x +y
Instrucción 2: c = z – 1
Instrucción 3: c = a – b
Instrucción 4: w = z + 1

3. Ejercicio 3.

A partir del siguiente conjunto de instrucciones, indica las que se pueden ejecutar concurrentemente y las que no. Justifica tus respuestas.

Instrucción 1: x = z + y
Instrucción 2: z = y + 1
Instrucción 3: c = z + 1
Instrucción 4: x = c + 1

4. Ejercicio 4.

Realiza un esquema de los estados de un hilo, e identifica lo que ocurre en cada uno de ellos.

5. Ejercicio 5.

Cuando varios hilos comparten el mismo espacio de memoria es posible que aparezcan algunos problemas de sincronización, explícalos con ejemplos.

6. Ejercicio 6.

Implemente el problema de los filósofos en Java. Ayúdate de las siguientes indicaciones.
Necesitamos tres clases Java:

- **Filósofo**: donde se van a realizar todas las operaciones relacionadas con el filósofo.
- **Tenedor**: donde se van a realizar todas las operaciones relacionadas con el tenedor.
- **Principal**: el programa principal que lo lanzará todo.

Delante de los métodos de la clase tenedor deberemos poner synchronized, con esto nos aseguramos de que solo va a existir un hilo ejecutándose a la vez en ese método.
La clase filósofo debe extender de Thread, para poder usar el método run dentro de esta clase cuando se llame al método start() desde la clase filósofos.

7. Ejercicio 7.

Contesta el siguiente test.

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?
a. Sleep(1)
b. Sleep(1000)
c. Pause(void)
d. Sleep(void)

¡Buen trabajo!