




30 DE OCTUBRE DE 2019

Programación de Servicios y Procesos

IAGO OTERO
ESICA
VIGO



1- ¿Qué es la programación concurrente? ¿Cuáles son sus características?

La programación concurrente es la simultaneidad en la ejecución de varias tareas (programas en ejecución) las cuales pueden funcionar gracias a que las operaciones se realizan en tan poco tiempo que da la sensación de que se están ejecutando paralelamente.

Se puede ejecutar en una única CPU o sino en varias, también podrían estar conectadas a una red donde se distribuyen los cómputos.

2- ¿Cuáles son las diferencias entre programación concurrente y programación paralela?

La programación concurrente es aquella habiendo dos programas A y B estos no son procesados a la vez, sino que uno lleva su procesamiento antes que el otro a diferencia de la programación paralela, que en este si es posible que dos procesos se lleven a cabo al mismo tiempo, es decir que mientras el proceso A se está calculando el proceso B también está siendo procesado.

3- Visualiza los siguientes vídeos y contesta:

¿Qué entiendes por contexto de un proceso?

Es la información que se conoce sobre un proceso que está en ejecución.

¿A qué se refiere el concepto de multiprogramación?

En la ejecución de distintos procesos parece que se ejecutan a la vez sin embargo solo se ejecuta uno y se va intercambiando, todo esto ocurre en tiempos muy pequeños.

¿Qué entiendes por demonio o daemon?

Son procesos no interactivos ejecutándose en segundo plano

¿Cuál es la diferencia principal entre UNIX y Windows a la hora de organizar los procesos?

UNIX crea bifurcaciones (hijos) a partir de un programa (padre) para poder realizar una tarea distinta sin que se cierre la tarea padre, la tarea padre y la tarea hijo son iguales donde lo único que cambia es el identificador del proceso. Windows no crea bifurcaciones.

¿Cuántos padres como máximo puede tener un proceso en UNIX?

Únicamente uno.

¿Cuántos hijos puede tener un proceso en UNIX?

Hasta los límites del hardware.

¿Cuáles son los estados básicos de un proceso? ¿Cuándo se encuentra un proceso en cada estado?

Nuevo: Se crea un proceso y entra al estado de listo.

Listo: El proceso está listo y cargado para poder ser ejecutado.

Bloqueado: El proceso está esperando a que ocurra un evento.

Ejecutando: El proceso está siendo procesado.

Finalizado: El proceso ha sido realizado y se libera de memoria.

¿Qué es un cambio de contexto? Pon un ejemplo.

Para que un ordenador este ejecutando todo el tiempo un programa y no se vea afectado por los tiempos de lectura de disco o simplemente no existan tiempos muertos se necesitan generar interrupciones intervenidas por el hardware, es aquí donde se crea un cambio de contexto para aprovechar el tiempo muerto.

Cuando esto ocurre se realizan las siguientes operaciones:

- 1- Guardar el estado del programa que se estaba ejecutando.
- 2- Seleccionar otro programa para ejecutar.
- 3- Restaurar el estado del programa seleccionado.
- 4- Ejecutar el programa seleccionado.

4- ¿Qué es un interbloqueo (Deadlock)? ¿Cuáles son las condiciones necesarias para que se produzca? Explícalas brevemente.

Un deadlock, interbloqueo o deadlock tiene lugar cuando varios procesos en un sistema concurrente compiten por los recursos del sistema. Cuando un proceso entra en ejecución el resto debe esperar a que este termine y así sucesivamente.

5- ¿Qué clase utilizarías en Java para crear un proceso? Pon un ejemplo de como la emplearías.

Utilizaría un ProcessBuilder

//Ejemplo

```
Process processBuilder = new ProcessBuilder(file.getAbsolutePath(),  
nombre).start();
```

```
BufferedReader br2 = new BufferedReader(new  
InputStreamReader(processBuilder.getInputStream()));
```

```
while ((linea2 = br2.readLine()) != null) {
```

```
System.out.println(linea2);
```

```
}
```

6- ¿Qué es un algoritmo de planificación (AP)? ¿Cuántos tipos de AP conoces? Describe cada tipo de AP.

Un algoritmo de planificación será dirigido por un planificador, parte del sistema operativo. Consiste en un conjunto de políticas y mecanismos incorporados al Sistema Operativo y por los que se rige el orden en que se completa el trabajo que hay que realizar. Su objetivo primordial es optimizar el rendimiento del sistema.

Aparecen dos tipos de planificación, la apropiativa y la no apropiativa:

-Apropiativa: Esta disciplina será apropiativa si permite que se le pueda retirar la CPU a un proceso mientras que la está usando.

-No apropiativa: Esta disciplina será no apropiativa si una vez que se le ha asignado la CPU a un proceso, ya no se le puede retirar. Como consecuencia no se podrá ejecutar otro programa si el anterior no ha terminado.

Ejemplos:

-Planificación por orden de llegada (FIFO – First Input, First Output, primero en entrar-primero en salir.

-Planificación del trabajo más corto primero (SJF).

-Planificación del tiempo restante más corto (SRT).