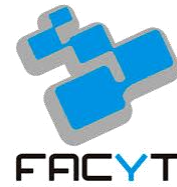




Universidad de Carabobo
Facultad de Ciencias Experimentales y Tecnología
Departamento de Computación
Sistemas Operativos



Taller II – II-2015: Comunicación y sincronización entre procesos

Objetivos:

- Comprender el concepto de concurrencia.
- Conocer las herramientas disponibles en los sistemas operativos para realizar sincronización y comunicación entre procesos.

Pre-laboratorio

Investigar el contenido referente a las siguientes preguntas:

1. ¿Qué es un hilo?
2. ¿Cuál es la diferencia entre un hilo y un proceso?
3. Indique la importancia de los hilos en un Sistema Operativo.
4. ¿Qué es para y para que se utiliza POSIX?
5. ¿Qué métodos y servicios ofrece POSIX para la sincronización y comunicación entre proceso

Laboratorio

Parte 1. Empezando con hilos en POSIX.

Descargar el archivo correspondiente al Taller II de nombre “hilos.c”. Una vez visualizado el archivo, compílelo y ejecútelo. Para compilar debes linkear la librería libpthread utilizando `-lpthread`. Basado en la investigación del pre-laboratorio: ¿Tiene sentido lo que se muestra en la terminal? Analice el porqué de los resultados.

Parte 2. Sincronización y comunicación entre procesos.

Descargar el archivo correspondiente al Taller II de nombre “semaforos.c”. Una vez visualizado el archivo, compílelo y ejecútelo. Para compilar de igual forma debe incluir `-lpthread`. Basado en la investigación del pre-laboratorio: ¿Qué método de sincronización se está utilizando? ¿Son correctas las salidas? Analice los resultados.

Evaluación

1 . Se desea desarrollar una aplicación que debe realizar dos tareas que se pueden ejecutar de forma independiente. Los códigos de estas dos tareas se encuentran definidos en dos funciones cuyos prototipos en lenguaje C son los siguientes:

```
void tarea_A(void);  
void tarea_B(void);
```

Se pide:

Programar la aplicación anterior utilizando tres modelos distintos: un programa secuencial ejecutado por un único proceso, un programa que realiza las tareas anteriores utilizando threads, y un programa que crea procesos para desarrollar cada una de las tareas.

2 . El objetivo de este ejercicio es simular la ejecución de penales de un partido de fútbol. Los tres componentes necesarios para llevar a cabo una ronda de penales son: el árbitro, el arquero y el jugador.

Para realizar la simulación se tienen que tener en cuenta las siguientes consideraciones:

- Existen cinco (5) procesos jugadores, un (1) proceso árbitro y un (1) proceso arquero.
- Los jugadores no pueden patear si el árbitro no lo indica.
- El arquero no puede atajar si el jugador no patea.
- El árbitro solo puede dar la orden cuando el jugador y el arquero están posicionados.
- Una vez que se valide el penal, se le pasará el turno al próximo jugador.
- Los jugadores siempre patean en un orden definido (Ejemplo: j1,j2,...,j5,j2,...,j5).

Consideraciones

- Las soluciones de los problemas deben ser entregados en un archivo .c y debe llevar el número del ejercicio por nombre.
- Si el código no compila no será evaluado.
- El tiempo aproximado de realización es de dos (2) horas.