



# ITESO

**Universidad Jesuita  
de Guadalajara**

Axel Roberto Orozco Hernández  
Edgar Alan Torres Tovar

Fundamentos de Sistemas Operativos  
Profesor: Leonardo Sandoval. Gonzalez

1.- Considere un programa que crea los dos procesos P y Q, definidos a continuación. el proceso P muestra en pantalla la salida "A", "B", y "C"; el proceso Q muestra en pantalla la salida "D" y "E".

Indicar todas las posibles intercalaciones en las salidas a pantalla de los dos procesos anteriores (indicarlo por medio de trazas de ejecución dadas en términos de sentencias atómicas).

1. Los procesos se pueden intercalar de forma concurrente, lo que significa que las salidas pueden intercalarse de distintas maneras.

- P antes que Q:

1. A, B, C, D, E

- Q antes que P:

1. D, E, A, B, C

- Intercalaciones posibles

1. A, D, B, C, E

2. A, B, D, C, E

3. A, D, E, B, C

4. D, A, E, B, C

5. D, A, B, E, C

6. D, E, A, C, B

2.- Considere el siguiente programa que crea dos hilos que incrementan la variable global cuenta

Determinar los valores mínimos y máximos que puede tomar la variable cuenta.  
Suponga que incrementar una variable en memoria requiere 3 operaciones

1.

1. Carga la variable en un registro

2. Incrementa el registro

3. Carga el registro en la variable

**-El valor maximo de cuenta:**

Si no hubiera ningún problema de concurrencia cada hilo incrementaría la variable 100,000, por lo que el valor máximo de la cuenta sería:

$2 \times 100,000 = 200,000$

**- Valor maximo de cuenta:**

Debido a la sincronización de la cuenta, es posible que se sobrescriba todos los datos, lo que causaría

que el numero final sea menor que el valor esperado.  
cuenta minima = 100,000 (si hay interferencias maximas entre los hilos).

**3.-** El código que se muestra en la Figura 1 es el intento donde dos hilos T(0) y T(1) se sincronizan a través de un algoritmo por software empleando variables en memoria compartida.

Haga un análisis de la ejecución concurrente de los dos hilos en una tabla donde se muestra la ejecución intercalada de instrucciones de ambos hilos, y muestre si esta solución funciona o no funciona, en caso de no funcionar explique el por qué.

| Instrucción T(0) | Instrucción T(1) | Descripcion                                   |
|------------------|------------------|---|
| Turno = 1        |                  | T(0) asigna el turno a T(1)                   |
|                  | Turno = 0        | T(1) asigna el turno a T(0)                   |
| Flag [0] = true  |                  | T(0) indica que quiere entrar a la SC         |
|                  | Flag [1] = true  | T(1) indica que quiere entrar a la SC         |
|                  |                  | Ambos hilos estann esperando indefinidamente. |

En este caso, ninguno de lo shilos puede progresar por que ambos esperen que el otro hilo ceda el turno, causando **interbloqueo**.

Este algoritmo **no funciona** debido a condiciones de carrera y la falta de exclusion mutua. Una posible solucion seria usar un mecanismo de sincronizacion adecuado, como mutex o semaforos, que garantice la exclusion mutua entre los hilos.

#### **4.- ¿Qué aprendiste?**

Intercalacion de procesos: Como procesos concurrentes pueden intercalarse de diferentes maneras en funcion del planificador del SO.

Condiciones de carrera: La falta de control sobre acceso concurrente a una variable compartida puede llevar a resultados inesperados, como se vio en el problema del incremento de la variable cuenta con hilos.

Sincronizacion de hilos: Algoritmos de sincronizacion basados en banderas y turnos que puede fallar si no se manejan cuidadosamente de carrera.