



Laboratorio de Programación Concurrente



7° Programación - 2020

Valenzuela/Varela

Práctica 0

Teórica:

Ejercicio 0: - Lavarse las manos durante 30 segundos con abundante agua y jabón.

Ejercicio 1: - ¿Qué diferencia hay entre un *programa* y un *proceso*?
- ¿Qué es la *conurrencia*?

Ejercicio 2: - Tomando como ejemplo la analogía de una autopista en hora pico y la concurrencia entre distintos programas, mencionar algunos ejemplos del mundo real que se le asemejen.
- Con lo visto hasta el momento, ¿cuáles son algunos de los problemas que se presentan al considerar la ejecución de un programa concurrente? Mencionar algunos.

Ejercicio 3: - ¿Qué es una *Traza*? ¿Todas son válidas?

Ejercicio 3: - ¿Es válido probar que sí se cumple alguna de estas condiciones mostrando una traza? Justifique su respuesta si cree que **NO** o de un ejemplo si cree que **SI**.

Ejercicio *BONUS*: Con las ideas vistas en clase sobre Concurrencia, paralelismo y demás, haga un meme¹.

¹ Para el mejor hay premio, *el Norte recuerda*.



Laboratorio de Programación Concurrente



7° Programación - 2020

Valenzuela/Varela

Práctica:

Aclaración: Recuerden que salvo que se indique lo contrario sólo se consideran atómicas² la lectura y la escritura de valores primitivos (*int*, *float*, *bool*, *char*) o las instrucciones que asuma o se consideren en el propio ejercicio.

- **Primera parte:**

Asumiendo que cada instrucción `print` es atómica, muestre todas las posibles trazas de ejecuciones del siguiente programa, dibujando el *árbol de posibilidades*:

Ejercicio 1:

```
thread T1: {                      thread T2 : {
    print ("Hola");                print ( " Hola " );
    print ("Horacio");            print("Marce");
}
```

Ejercicio 2:

(Demian dice que *idem anterior*)

```
thread T1 : {                      thread T2 : {
    print ( " Hola " );            print ("Hola" );
    print ( " Marce " );          print ("H");
}
```

Ejercicio 3:

Asumiendo que cada instrucción `print` es atómica, muestre todas las posibles trazas de ejecuciones del siguiente programa, dibujando el *árbol de posibilidades* y diga cuantas son

² Decimos que una instrucción es **atómica** cuando *TODO* lo que hace se ve como una acción que no puede ser interrumpida, propiedad que como ya vimos, no todos los programas/Instrucciones.



Laboratorio de Programación Concurrente



7° Programación - 2020

Valenzuela/Varela

válidas si queremos que `print("nos volvemos a encontrar sr Bond");` se ejecute solo después de que ambos se saluden correctamente³:

```
thread t1:{
    print("Hola");
    print("Horacio");
}

thread T2 : {
    print("Hola");
    print("Marce");
    print( " No te merecemos");
}

thread T3 : {
    print("nos volvemos a
    encontrar sr Bond");
}
```

- **Segunda parte:**

Considerando lo visto hasta el momento y sabiendo que la variable **x** es *global* (común para todos los threads) y que inicialmente **x = 16**, contestar: ¿Cuántas salidas posibles tiene esta ejecución? Cuáles son?

Ejercicio 1:

```
thread T2:{
    print(x);
}

thread T2:{
    print("x");
}
```

Ejercicio 2:

```
thread T1:p
    print(x);
}

thread T3:{
    x = 18;
    print(x);
}
```

Ejercicio 3:

```
thread T1: {
    print (x);
}

thread T1 : {
    x = 32;
}

thread T1 : {
    print(x-1)
}
```

³ Es decir, que empiecen con un "Hola Horacio" "Hola Marce" o "Hola Marce" "Hola Horacio" y terminen con un "Adiós Horacio" "Adios Marce" o "Hola Marce" "Adiós Horacio"