

Trabajo Práctico N° 8

Concurrencia

Recomendaciones Generales e Información

Comandos para poder ejecutar los ejercicios

- **g++** -std=c++11 -pthread <archivo_c++> -o <nombre_programa>
- **perl** <archivo_perl>
- **python** <archivo_python>, python2 <archivo_python> o python3 <archivo_python>
- **go** build <archivo_go>
- Compilador en Línea de **C++**: https://www.onlinegdb.com/online_c++_compiler

Ejercicio 1

Dado el siguiente programa escrito en **C++**

```
#include <iostream>
#include <thread>
using namespace std;
void f(int id){
    static int j=4;
    printf("id : %d, j: %d, &j: %d\n",id, j, &j);
    for (int i=0;i<1000;i++);
    j--;
    if (j>0)
        f(id+1);
}
int main (int argc, char *argv[]){
    thread th1(f,100);
    thread th2(f,200);
    thread th3(f,300);
    thread th4(f,400);
    th1.join();
    th2.join();
    th3.join();
    th4.join();
    return 0;
}
```

- Explique brevemente el comportamiento del programa, analizando por qué produce la salida correspondiente.
- Idem a para la siguiente declaración de j: `int j=4;`
- Idem a para la siguiente declaración de j: `static thread_local int j=4;`

Ejercicio 2

```
#include <iostream>
#include <thread>
using namespace std;
static int m;
void f(int id){
    int j=4;
    static thread_local int k=4;
    static int l = 4;
    printf("id : %d, &j: %d, &k: %d, &l: %d, &j-&k: %d, &j-&l: %d, &k-&l: %d\n", id, &j, &k, &l, (&j-&k), (&j-&l), (&k-&l));
    printf("&l-&m : %d, \n", (&l-&m));
    k--;
    if (k>0)
        f(id+1);
}
int main (int argc, char *argv[]){
    thread th1(f,100);
    thread th2(f,200);
    thread th3(f,300);
    thread th4(f,400);
    th1.join();
    th2.join();
    th3.join();
    th4.join();
    return 0;
}
```

- Realizar un diagrama de la pila de ejecución del programa
- Explicar brevemente la diferencia entre el comportamiento de las impresiones (&j-&k) y (&j-&l)
- Explicar qué sucedería si se agregara una declaración en el ámbito global: static int m; y se imprimiera (&l-&m) en f

Ejercicio 3

Dado el siguiente programa escrito en C++

```
#include <iostream>
#include <thread>
#include <mutex>
using namespace std;
mutex mtx;
void f(int id){
    static int j = 0;
    printf("id : %d, j: %d \n",id,j);
    j++;
}
int main (int argc, char *argv[]){
    thread th1(f,100);
    thread th2(f,200);
    thread th3(f,300);
    thread th4(f,400);
    th1.join();
    th2.join();
    th3.join();
    th4.join();
    return 0;
}
```

- Explique brevemente el comportamiento del programa, analizando por qué produce la salida correspondiente.
- ¿Por qué termina el programa luego de 4 incrementos de la variable j si en algunas ejecuciones no se muestra el valor 4 para esta variable?
- Proponer una solución para que las salidas se realicen de acuerdo al orden sucesivo con el cual se incrementa la variable j.
- Responder y fundamentar si es necesaria la solución propuesta del inciso c en el caso que la variable j es local a la función f
- Responder y fundamentar si es necesaria la solución propuesta al inciso b si la variable j es local a cada pila de hilos de ejecución.

Ejercicio 4

Dado el siguiente programa escrito en **C++**

```
#include <iostream>
#include <thread>
#include <mutex>
using namespace std;
mutex mtx;
void f(int id){
    static int j = 0;
    if (j<10)
    {
        mtx.lock();
        printf("id : %d, j: %d \n",id,j);
        j++;
        mtx.unlock();
        f(id+1);
    }
}
int main (int argc, char *argv[]){
    thread th1(f,100);
    thread th2(f,200);
    thread th3(f,300);
    thread th4(f,400);
    th1.join();
    th2.join();
    th3.join();
    th4.join();
    return 0;
}
```

- Explique brevemente el comportamiento del programa, analizando por qué se produce la salida correspondiente.
- Realizar un diagrama que muestre las diferentes pilas de hilos de ejecución del programa.
- Si se quita el semáforo explique brevemente el comportamiento del programa, analizando por qué se produce la salida correspondiente.
- Idem inciso a) cambiando la declaración de la variable j para que sea local a cada pila de hilos de ejecución. ¿Cuáles cambios se producen en el diagrama del inciso b)?
- Idem c) para la modificación del inciso d)

Ejercicio 5

Dado el siguiente código con Threads en Java

```
public class ej5 {
    private static int MY_INT = 0;
    public static void main(String[] args) {
        new Consumidor().start();
        new Productor().start();
    }

    static class Consumidor extends Thread {
        @Override
        public void run() {
            int local_value = MY_INT;
            while ( local_value < 5){
                if( local_value!= MY_INT){
                    System.out.println("Listo el cambio para MY_INT : "+MY_INT);
                    local_value= MY_INT;
                }
            }
        }
    }

    static class Productor extends Thread{
        @Override
        public void run() {
            int local_value = MY_INT;
            while (MY_INT <5){
                System.out.println("Cambiando MY_INT a "+(local_value+1));
                MY_INT = ++local_value;
                try {
                    Thread.sleep(500);
                } catch (InterruptedException e) { e.printStackTrace(); }
            }
        }
    }
}
```

Analice su comportamiento según lo que debería realizar el programa.

Si el comportamiento no fuese el esperado, ¿Qué propondría para que así fuera?

Ejercicio 6

Dado el siguiente programa escrito en **GO**

```
package main
import (
    "fmt"
)
var msg string
func f() {
    msg = "Hola"
}
func g() {
    msg = "Que tal"
}
func main() {
    var i int
    i = 0
    for i < 10 {
        f()
        fmt.Println(msg)
        g()
        fmt.Println(msg)
        i = i + 1
    }
}
```

- Modificar el programa para que las funciones f y g se ejecuten como goroutines. Explicar lo que sucede.
- Proponer una solución para el problema ocasionado por ejecutar f y g como goroutines.
- Modificar el programa para que las impresiones se realicen en una goroutine.
- Medir el tiempo respecto del código inicial y proporcionar una fundamentación.
- Proponer una idea de solución de este problema en un lenguaje que no posea canales.

Ejercicio 7

Dado el siguiente programa escrito en **PYTHON**

```
import threading
x = 0
CANT = 10000000
def f1(s):
    global x, CANT
    for i in range(CANT):
        x += 1

def f2(s):
    global x, CANT
    for i in range(CANT):
        x -= 1

s = threading.Lock()
t1 = threading.Thread(target=f1, args = (s,))
t2 = threading.Thread(target=f2, args = (s,))
t1.start()
t2.start()
t1.join()
t2.join()
print (x)
```

- Antes de probarlo, describa lo que ocurriría con el programa y qué valores espera que se impriman.
- El desarrollador de la rutina intentó pasar un parámetro llamado `s` pero no llegó a concluir el programa. Como líder de proyecto asignado, antes de derivarla a otro integrante del equipo, deduzca el motivo de la inclusión y, en pseudocódigo, como lo hubiese usado el programador y para qué.
- Luego de analizar b) explique cuál sería la salida esperada y los valores por los que puede pasar la variable global `x` durante la ejecución del programa.

Ejercicio 8

Dado el siguiente programa escrito en **PERL**

C1	C2
<pre>#!/usr/bin/perl use threads; use threads::shared; my \$a: shared = 1; sub sub1 { for (\$i=0;\$i<10;\$i++){ \$a = \$a + 1; sleep(rand()*1.5); printf("Hilo: %2d, a: %2d\n", threads->self->tid,\$a); } } \$th1 = threads->new(\&sub1); \$th2 = threads->new(\&sub1); \$th1->join(); \$th2->join(); print "Hilo principal\n";</pre>	<pre>#!/usr/bin/perl use threads; use threads::shared; my \$a = 1; sub sub1 { for (\$i=0;\$i<10;\$i++){ \$a = \$a + 1; sleep(rand()*1.5); printf("Hilo: %2d,a: %2d\n", threads->self->tid,\$a); } } \$th1 = threads->new(\&sub1); \$th2 = threads->new(\&sub1); \$th1->join(); \$th2->join(); print "Hilo principal\n";</pre>

- a) Asignar cada código a una de las salidas que se encuentran debajo
- b) Proponer una representación de la tabla de símbolos para los programas C1 y C2
- c) Proponer una modificación del programa C1 eliminando una línea de código para que el resultado siga siendo similar al del código C2.

S1	S2
<pre>Hilo, a: 2 3 Hilo, a: 2 4 Hilo, a: 2 5 Hilo, a: 2 6 Hilo, a: 1 7 Hilo, a: 2 8 Hilo, a: 2 9 Hilo, a: 1 10 Hilo principal</pre>	<pre>Hilo, a: 1 2 Hilo, a: 1 3 Hilo, a: 2 2 Hilo, a: 2 3 Hilo, a: 1 4 Hilo, a: 2 4 Hilo, a: 1 5 Hilo, a: 2 5 Hilo, a: 1 6 Hilo, a: 2 6 Hilo, a: 2 7 Hilo, a: 2 8 Hilo, a: 1 7 Hilo, a: 1 8 Hilo, a: 1 9 Hilo, a: 1 10 Hilo, a: 1 11 Hilo, a: 2 9 Hilo, a: 2 10 Hilo, a: 2 11 Hilo principal</pre>

██████

S3	S4
Hilo, a: 2 3 Hilo, a: 2 4 Hilo, a: 2 5 Hilo, a: 2 6 Hilo, a: 1 7 Hilo, a: 2 8 Hilo, a: 2 9 Hilo, a: 1 10 Hilo, a: 2 11 Hilo, a: 1 12 Hilo, a: 1 13 Hilo, a: 1 14 Hilo, a: 1 15 Hilo, a: 1 16 Hilo, a: 1 17 Hilo, a: 1 18 Hilo, a: 2 19 Hilo, a: 2 20 Hilo, a: 1 21 Hilo, a: 2 21 Hilo, a: 1 22 Hilo, a: 2 23 Hilo, a: 1 24 Hilo, a: 2 25 Hilo principal	Hilo, a: 2 3 Hilo, a: 2 4 Hilo, a: 2 5 Hilo, a: 2 6 Hilo, a: 1 7 Hilo, a: 2 8 Hilo, a: 2 9 Hilo, a: 1 10 Hilo, a: 2 11 Hilo, a: 1 12 Hilo, a: 1 13 Hilo, a: 1 14 Hilo, a: 1 15 Hilo, a: 1 16 Hilo, a: 1 17 Hilo, a: 1 18 Hilo, a: 2 19 Hilo, a: 2 20 Hilo, a: 1 21 Hilo, a: 2 21 Hilo principal