



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



PRÁCTICA - SEMÁFOROS

• Ejercicio #0:

Dados los siguientes thread, resolver, utilizar semáforos para garantizar las siguientes condiciones::

❑ A:

Simultáneamente, 'A' se muestra antes que 'F', y 'F' se muestra antes que 'C'.

```
thread {  
    print("A");  
    print("B");  
    print("C");  
}
```

```
thread {  
    print("E");  
    print("F");  
    print("G");  
}
```

❑ B:

Las únicas salidas posibles sean "ACERO" y "ACREO".

```
thread {  
    print("C");  
    print("E");  
}
```

```
thread {  
    print("A");  
    print("R");  
    print("O");  
}
```

❑ C:

El único resultado impreso sea "No lo sé Rick. Parece Falso" (asumimos que print es atómico)

```
thread {  
    print("No");  
    print(" Rick.");  
}
```

```
thread {  
    print(" lo");  
    print(" Parece");  
}
```

```
thread {  
    print(" sé");  
    print(" falso.");  
}
```

❑ D:

¿Se acuerdan de los pitufos? Volvieron. En forma de ejercicio de concurrente:

```
thread {  
    print("Papa PituFo");  
    print("Pitufo");  
}
```

```
thread {  
    print("Pitufina");  
    print("Pitufo");  
}
```

```
thread {  
    print("Pitufo");  
    print("filósofo");  
}
```



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



```
alquimista");           arqueólogo");           print("Pitufito
}                           }                       robot");
}                           }                       }
```

- Garantizar que Papá Pitufito vaya siempre primero.
- Pitufina vaya siempre después de Papá pitufito y antes de todos los demás.
- Pitufito filósofo va antes que el Pitufito alquimista.

❑ E:

```
thread {                  thread {
    print("Hola Horacio");    print("Hola Damian");
    print("Adiós Horacio");   print("Adiós Horacio");
}                            }
```

- A) Garantizar que las únicas salidas posibles sean "Hola Horacio" "Hola Damian" "Adiós Horacio" "Adiós Damian".
- B) Garantizar que las únicas salidas posibles sean "Hola Damian" "Hola Horacio" "Adiós Damian" "Adiós Horacio".
- C) Garantizar que las únicas salidas sean Hola XXXX, Hola YYYY, Adiós XXXX, Adiós YYYY, sin importar quien salude primero a quien.

❑ F:

Recordando al inolvidable Comandante Fort garantizar que la única salida posible sea "MA EA M E E" (asumimos que **print** es atómico)



```
thread {                  thread {                  thread {
    print("MA");           print("EA");           print("M");
    print("E");            print("E");            print("E");
}                          }                          }
```

❑ G:

```
thread {                  thread {                  thread {
    while(true) {          while(true) {          while(true) {
        print("A");         print("E");         print("H");
        print("B");         print("F");         print("I");
        print("C");         print("G");
        print("D");
    }
}                          }                          }
```



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



Agregar semáforos para garantizar que simultáneamente se den las siguientes condiciones:

- La cantidad de 'F' es menor o igual a la cantidad de 'A'
- La cantidad de 'H' es menor o igual a la cantidad de 'E'.
- La cantidad de 'C' es menor o igual a la cantidad de 'G'.

❑ H:

```
global int x = 0;
```

```
thread{
    while(true){
        print("Damian");
    }
}
```

```
thread{
    while(true){
        print("Horacio");
    }
}
```

- A) Utilice semáforos para garantizar que en todo momento la cantidad de **Damian** y **Horacio** difiera como máximo en 1.
- B) Dar otro solución para que la salida se siempre **Damian Horacio Damian Horacio ...**

Nota: Este bucle es infinito, deben cortarlo apretando **stop**

• Ejercicio #1:

Garanticen, por medio del uso de semáforos las siguientes condiciones:

❑ A:

```
global int x = 0;
```

```
thread{
    x = x + 1;
}
```

```
thread{
    x = x + 2;
}
```

```
thread{
    x = x + 3;
}
```

Luego de la ejecución del programa la variable **x** posea el valor 6. Procure que su solución no se bloquee incluso si sólo se ejecutarán dos (o uno) de los tres threads.

❑ B:

Siempre se imprima que **x** vale 2.

```
global int x = 0;
```



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



```
thread{  
    x++;  
}
```

```
thread{  
    ++ x;  
}
```

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

❑ C:

Siempre se imprima que **x** vale 2.

```
global int x = 0;
```

```
thread{  
    x = x + 1;  
}
```

```
thread{  
    x = x + 1;  
}
```

❑ D:

Siempre se imprima que **x** vale 2.

```
global int y = 0, z = 0;
```

```
thread{  
    int x;  
    x = y + z;  
}
```

```
thread{  
    y = 1;  
    z = 2;  
}
```

- A) ¿Cuáles son los posibles valores finales de **x**? Justifiquen.
- B) Para cada uno de los posibles valores finales de **x**, de una solución para garantizar con el uso de semáforos que **x** termine teniendo únicamente ese valor.

❑ E:

Siempre se imprima *"El sentido de la vida, el universo y todo lo demás es 42"*.

```
global int x= 36;
```

```
thread{  
    x = x + 2;  
}
```

```
thread{  
    x = x + 4;  
    print("El sentido de la vida, el universo y todo lo demás es "  
    + x.toString());  
}
```

❑ F:

Garantizar que siempre se imprima **8**.

```
global int x= 0, y = 1, z = 2;
```



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



```
thread{
  x = x + y;
}
```

```
thread{
  y = y + z;
}
```

```
thread{
  z = z + 5;
}
```

```
thread{
  print(z);
}
```

❑ G:

Siempre se imprima que **x** vale **0**.

```
global int x = 0;
```

```
thread{
  x++;
}
```

```
thread{
  x= x +1
}
```

```
thread{
  print(x)
}
```

- C) ¿Qué cambios deberías hacer para que se imprima siempre **1**?
D) ¿Y para que siempre se imprima **2**?

❑ H:

Haciendo referencia al gran álbum de *Iron Maiden*, *The number of the beast* hacer que siempre se imprima que **666**. *Heavy metal!!!*

```
global int x = 0, y = 0, z = 0;
```

```
thread{
  while(x != 6){
    x++;
  }
}
```

```
thread{
  while(y < 7){
    ++y;
  }
}
```

```
thread{
  while(true){
    z = z +1;
  }
}
```

```
thread{
  print(x);
  print(y);
  print(z);
}
```

- E) Por otro lado, el **777** es considerado como el número de la perfección. Creen otro thread que sume uno a las variables de tal forma que solo se imprima **777**.

● Ejercicio #2

❑ A:

La **FMS Argentina** nos ha contratado para la que programemos un sistema que les permita sincronizar correctamente el orden en que les artistas participan en la batalla de gallos. Cada uno debe rapear sus patrones sin ser interrumpido y como **Wos** es el favorito de todos, quieren que vaya siempre primero. El orden en que canten los otros 2 participantes, no nos importa mientras no sean interrumpidos.



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



```
thread WOS:{  
    print("Voy a entrar en donde quiera,");  
    print("mano arriba toda' la fiera.");  
    print("Tiro rima para el aire,");  
    print("acá no batimos cualquiera.");  
    print("Mandale mecha, rompo brecha, es estrecha.");  
    print("Me dicen Wosito zurdo,");  
    print("es que son todos de derecha, hecha.");  
}
```

```
thread REPLIK: {  
    print("No toca guitarra,");  
    print("tocó arpa, flauta y la gaita.");  
    print("Escribo sonatas, suena campanas compactas.");  
    print("Wacho que te parta, palabras sensatas,");  
    print("hace lo que yo hago y ahí tirame tu data");  
}
```

```
thread PAPO:{  
    print("Yo te tiro si me inspiro,");  
    print("lo castigo, lo domino, ");  
    print("bien tranquilo con el rap ");  
    print("Con el pelo violeta,");  
    print("no me pueden frenar. ");  
    print("Skeereeeeeereeeeeee");  
    print("Pueden decirme Lil Pump");  
}
```

❑ B:

Dada el fragmento de la rima de Replik:

*"Me introduzco lento."
"Un molusco hambriento,"
"en mar abierto, mar adentro. se clavan dentro."
"Son las balas que disparan, en caravana vengo."
"Como me detengo?. Así te reviento."
"Te falta demasiado pa' tener este talento"*

Se pide:

- Dar un escenario donde la rima se distribuye en 1, 2 o 3 thread, que al imprimir los versos puedan dar cualquier cosa.
- Dado los threads definidos en el punto anterior, utilizar semáforos para garantizar que la salida sea correcta.

Nota: Es importante que ningún par de rimas queden en el mismo thread en más de un caso.



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



❑ C:

Dada las (polémicas) rimas de Roma y Dozer:

```
thread DOZER:{
    print("Educación sexual no se merecen");
    print("Para que tener sexo, para que aborten
en 4 meses.");
    print("Y esa es la verdad ");
    print("Les respondo al fondo. Tiró al bombo y
más calidad");

    print("Para vos un bebé, es un esperma");
    print("Para mi un bebé, una familia eterna");
    print("Yo me preocupo por mi familia, para
que se duerma.");
    print("Y vos te preocupas para abortar para
abrir esas 2 piernas");
}

thread ROMA{
    print("No entiende lo que hace.");
    print("Simplemente no comprende.");
    print("No sabe que la guacha posta lo
prende.");
    print("Dice que aborto a los 20 meses, pero
como no tiene útero, no lo entiende");

    print("No sabe que decir, en la instrumental.");
    print("Lo bueno, apuesta ahora en el rap.");
    print("no sabe que decir, aguante la
maternidad.");
    print("pero es deseada o no será.");
}
```

Primer rima de Dozer

Segunda rima de Dozer

Primer rima de Roma

Segunda rima de Roma

Se pide, utilizar semáforos para garantizar que:

- Dozer recite su primer rima primer rima, luego roma y así.
- Roma recite su primer rima primer rima, luego Dozer y así.
- Algune de les 2 recite su primer rima, luego el otre y así.



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



❑ D:

Dado thread que imprime en pantalla un verso de Wos:¹

```
thread WOS: {  
    print("Estoy en la cama de dos plazas,")  
    print("que al final pude comprar.");  
    print("Pero ahora que la tengo,");  
    print("Solo ocupo la mitad.");  
}
```

Buscar una rima de algun otre rapere con alguna rima/patrón que les interese (si es con algún mensaje interesante, mejor), e implementar su rima en un thread y con esto, se pide:

- Que WOS siempre recite su patrón antes que le otre.
- Que le otre vaya antes que WOS.
- Que algune de les 2 empiece, pero que ningune sea interrumpide.

❑ E:

Dado los siguientes threads:²

```
thread WOS:{  
    print("Pero veo");  
    print("toda esa gente caminar  
con mala cara,");  
    print("pibes descalzos bajo un  
cartel de las nike más caras.");  
    print("Piden una ayuda, pero a  
todos les resbala.");  
    print("La seguridad de su país,  
solo regala balas.");  
    print("Diganme que no es en  
serio,");  
    print("que me muestran tantos  
fines, sin proporcionar los  
medios.");  
    print("Diganme que no es en  
serio,");  
    print("que después de esto, no
```

```
thread Batería:{  
    while(true){  
        print("Turummm")  
    }  
}
```

```
thread Campana:{  
    while(true){  
        print("Tilimmm");  
    }  
}
```

```
thread Instrumental:{  
    while(true){  
        print("Burum psss psss")  
    }  
}
```

```
thread Bajo{  
    while(true){  
        print("Booom boom");  
    }  
}
```

¹ Fragmento del tema [Pantano](#)

² Fragmento del tema [Banzai](#)



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



```
quiera empezar un incendio");  
}
```

Se pide utilizar semáforos para garantizar que:

- Primero suene la instrumental una vez, antes que Vos empiece a rapear.
- Cada vez que una rima termina con **o**, suena la batería una vez.
- Cada vez que una rima termina con **a**, suena la campana una vez.
- Cuando la rima termine en consonante, suene el bajo.
- Suene la instrumental cuando Vos termine de rapear.

❑ F:

Dado este thread y los instrumentos/sonidos del ejercicio **2-E**, utilizar semáforos para garantizar que antes y después de cada verso de esta rima, suene una única vez un sonido/instrumento.³

```
thread WOS:{  
    print("Y no hables de meritocracia,");  
    print("Me da gracia, no me jodan");  
    print("Que sin oportunidades,");  
    print("esa mie*** no funciona");  
}
```

Bonus: Realizar los cambios en el código, necesario para que al finalizar Vos de rapear ningún instrumento quede bloqueado, pero aún así ninguno suene. ¿Necesitamos semáforos para esto?

❑ G:

Considerando este thread, similar al del ejercicio anterior:

global Semaphore puedeCantar = **new** Semaphore(0);

```
thread WOS:{  
    puedeCantar.acquire();  
    print("Y no hables de meritocracia,");  
    print("Me da gracia, no me jodan");  
    print("Que sin oportunidades,");  
    print("esa mie*** no funciona");  
    puedeCantar.release();  
}
```

¿Qué problema/s encuentran? Investiguen como se llama ese problema e intenten explicar en que consiste. ¿Como lo solucionaría? De dos posibles soluciones, una utilizando semáforos y otra no.

❑ H:

global int n = 0;

³ Fragmento de la canción [Patada de canguro](#)



Lab. Programación Concurrente

7° Programación - 2022

Valenzuela/Varela



```
thread Invert:{  
    print("La réplica no la entiendo,")  
    print("¿cuántas gotas tienen que caer para  
saber que está lloviendo?")  
}
```

```
thread Reloj:{  
    while(true){  
        print(x);  
    }  
}
```

Garantizar que:

- Dar una solución utilizando semáforos para que Invert empiece a cantar únicamente cuando el reloj haya impreso de 0 a 10.
- Proponer los cambios necesarios para que el reloj no quede bloqueado una vez y aun así cumpla lo pedido en el ítem anterior.