

1. Semáforo

- Con base en el header del semáforo ya proporcionado, se crearon los cuatro métodos necesarios para que el programa corriera. En el constructor, se utilizó una unión que es utilizada como argumento al momento de crear el semáforo. Se utilizó el llamado al sistema **semget** para crear el identificador de semáforo, con la llave correspondiente a mi carné, dándole el valor de 1 correspondiente a que solo habrá un semáforo y la bandera de los permisos. Posteriormente, con **semctl** se le asigna el valor inicial al semáforo. Lo mismo con el método destructor, solo que la operación a realizar, indicada por la bandera, es la que elimina el semáforo. En los métodos Signal y Wait, se utiliza una estructura con valores "quemados", pues son los necesarios para que el programa funcione en este caso particular, donde cada valor corresponde respectivamente al número de semáforo a operar, la operación a realizar y la bandera. La única diferencia es en la operación a realizar, donde se le asigna un valor negativo o positivo, dependiendo del método, para que el valor del semáforo aumente o disminuya.

- Demostración:

```
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117/Practicas/Semana 1/semaforo$ make
g++ -c -g pruebaSem.cc
g++ -c -g Semaphore.cc
Semaphore.cc: In member function 'int Semaphore::Signal()':
Semaphore.cc:25:1: warning: no return statement in function returning non-void [-Wreturn-type]
 25 | }
    | ^
Semaphore.cc: In member function 'int Semaphore::Wait()':
Semaphore.cc:31:1: warning: no return statement in function returning non-void [-Wreturn-type]
 31 | }
    | ^
g++ -g pruebaSem.o Semaphore.o -o pruebaSem
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117/Practicas/Semana 1/semaforo$ ./pruebaSem
Esperando para activar el semaforo 0
Esperando para activar el semaforo 1
Esperando para activar el semaforo 2
Esperando para activar el semaforo 3
Esperando para activar el semaforo 4
Esperando para activar el semaforo 5
Esperando para activar el semaforo 6
Esperando que el semaforo se active ...
Esperando para activar el semaforo 7
Esperando para activar el semaforo 8
Semaforo activado
Esperando para activar el semaforo 9
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117/Practicas/Semana 1/semaforo$
```

2. PiPorSeries

- El problema no está en la implementación de la lógica para calcular pi, sino que, al hacer uso de creación de procesos mediante **fork()**, lo que sucede es que todos los procesos utilizan espacios de memoria distintos entre sí, por lo que el vector, utilizado para calcular una aproximación de pi, nunca es el mismo en cualquiera de los procesos pues no se está utilizando uno que esté en un espacio de memoria compartido por todos los procesos involucrados.

- Demostración:

```
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117/Practicas/Semana 1$ ./a.out
Creating process 5919: starting value 0, finish at 100000
Creating process 5920: starting value 100000, finish at 200000
Creating process 5921: starting value 200000, finish at 300000
Creating process 5922: starting value 300000, finish at 400000
Creating process 5923: starting value 400000, finish at 500000
Creating process 5924: starting value 500000, finish at 600000
Creating process 5925: starting value 600000, finish at 700000
Creating process 5926: starting value 700000, finish at 800000
Creating process 5927: starting value 800000, finish at 900000
Creating process 5928: starting value 900000, finish at 1000000
Valor calculado de Pi es  6.95275e-310  con 1000000 terminos
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117/Practicas/Semana 1$
```

3. Makefiles

- Con la ayuda de makefiles proporcionados por profesores en cursos pasados, se logra realizar el necesario para compilar la solución al problema del semáforo. La parte en verde representa el archivo que se va a crear a partir de los archivos que están después de los dos puntos, junto con el comando necesario para crearlo a partir de la terminal.

- Demostración:

```
home > rigovil > Escritorio > UCR > I Semestre 2020 > CI-0117 > Practicas > Semana 1 > semaforo > Makefile
1  pruebaSem: pruebaSem.o Semaphore.o
2      g++ -g pruebaSem.o Semaphore.o -o pruebaSem
3
4  pruebaSem.o: pruebaSem.cc
5      g++ -c -g pruebaSem.cc
6
7  Semaphore.o: Semaphore.cc Semaphore.h
8      g++ -c -g Semaphore.cc
```

```
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117/Practicas/Semana 1/semaforo$ make
g++ -c -g pruebaSem.cc
g++ -c -g Semaphore.cc
Semaphore.cc: In member function 'int Semaphore::Signal()':
Semaphore.cc:25:1: warning: no return statement in function returning non-void [-Wreturn-type]
   25 | }
      | ^
Semaphore.cc: In member function 'int Semaphore::Wait()':
Semaphore.cc:31:1: warning: no return statement in function returning non-void [-Wreturn-type]
   31 | }
      | ^
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117/Practicas/Semana 1/semaforo$
```