

1. Semáforos: construir la implantación de la clase. Completar la clase Semaforo para poder sincronizar procesos de una misma computadora.

- Se utilizó la interfaz Semaforo.h (provista por el profesor) para generar la estructura de la clase Semaforo. Dentro del constructor se define el ID del semáforo creado, utilizando el comando semget. En este, se usa el carné como key, se establece que será 1 semáforo creado y se define que se quiere crear el mismo con IPC_CREAT. Además, se adhiere una verificación para asegurar si el semáforo fue creado correctamente, en caso contrario, se avisará de un error (esto con perror). Por último se utiliza el comando semctl para asignarle un valor al semáforo creado (aquí entra en juego el valor adquirido por la unión) por medio de SETVAL. Al inicio se realizaron las declaraciones de union y struct dentro de los métodos del programa, sin embargo, luego se mejoró al comprender que ambas son parte de un todo (se utilizan en diferentes métodos en toda la clase). El struct se declaró para luego otorgarle las indicaciones necesarias según fuera el caso en los métodos clave del Semáforo: Wait (P) y Signal (V). Esta se necesitaba para poder implementar la función semop. La diferencia entre ambos es que al asignarle un valor de operador al Wait, este debe ser -1, para así decrementar en uno el valor del semáforo (si este es menor o igual a 0 el proceso espera). Por otro lado, en el método Signal se establece el operador con un +1, para incrementar el valor del semáforo y así "despertar" al primer proceso que exista en la cola. Al declarar el destructor se utiliza semctl nuevamente, esta vez con la instrucción IPCRMID para eliminar el semáforo creado (según su id).

- Compilación y resultados de ejecución

```
Actividades Terminal Jue 21:15
jennifer@jennifer-VirtualBox: ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Labora...
Archivo Editar Ver Buscar Terminal Ayuda
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana1/Semaforo$ g++ -Wall main.cpp Semaforo.cpp -o ejecutable
main.cpp: In function 'int main()':
main.cpp:11:12: warning: unused variable 'texto' [-Wunused-variable]
    char * texto = NULL;
           ^~~~~~
Semaforo.cpp: In constructor 'Semaforo::Semaforo(int)':
Semaforo.cpp:32:9: warning: unused variable 'aux' [-Wunused-variable]
    int aux = semctl(id, 0, SETVAL, u);
        ^~~~
Semaforo.cpp: In member function 'int Semaforo::Wait()':
Semaforo.cpp:46:9: warning: unused variable 'aux' [-Wunused-variable]
    int aux = semop(id, &b, 1);
        ^~~~
Semaforo.cpp:47:1: warning: no return statement in function returning non-void
[-Wreturn-type]
}
^
Semaforo.cpp: In member function 'int Semaforo::Signal()':
Semaforo.cpp:55:9: warning: unused variable 'aux' [-Wunused-variable]
    int aux = semop(id, &b, 1);
        ^~~~
Semaforo.cpp:56:1: warning: no return statement in function returning non-void
[-Wreturn-type]
}
^
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana1/Semaforo$
```

(a) Compilar en terminal

```
Actividades Terminal Jue 21:18
jennifer@jennifer-VirtualBox: ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Labora...
Archivo Editar Ver Buscar Terminal Ayuda
    int aux = semop(id, &b, 1);
        ^~~~
Semaforo.cpp:47:1: warning: no return statement in function returning non-void
[-Wreturn-type]
}
^
Semaforo.cpp: In member function 'int Semaforo::Signal()':
Semaforo.cpp:55:9: warning: unused variable 'aux' [-Wunused-variable]
    int aux = semop(id, &b, 1);
        ^~~~
Semaforo.cpp:56:1: warning: no return statement in function returning non-void
[-Wreturn-type]
}
^
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana1/Semaforo$ ./ejecutable
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 para activar el semaforo 7
Esperando para activar el semaforo 8
Esperando para activar el semaforo 9
Esperando que el semaforo se active ...
^
```

(b) Esperando lectura

```
Actividades Terminal Jue 21:18
jennifer@jennifer-VirtualBox: ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Labora...
Archivo Editar Ver Buscar Terminal Ayuda
[-Wreturn-type]
}
^
Semaforo.cpp: In member function 'int Semaforo::Signal()':
Semaforo.cpp:55:9: warning: unused variable 'aux' [-Wunused-variable]
    int aux = semop(id, &b, 1);
        ^~~~
Semaforo.cpp:56:1: warning: no return statement in function returning non-void
[-Wreturn-type]
}
^
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana1/Semaforo$ ./ejecutable
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 para activar el semaforo 7
Esperando para activar el semaforo 8
Esperando para activar el semaforo 9
Esperando que el semaforo se active ...
2
Semaforo activado
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana1/Semaforo$
```

(c) Fin de ejecución

Figure 1: Demostración de compilación y ejecución.

2. Correr el ejemplo provisto "PiPorSeries.c", anotar los resultados obtenidos explicando por qué no funciona.

- [-Explicación de cómo se dio solución al problema. Puede ser en escrito en prosa o los pasos que se siguieron para llegar a la solución-]
- [-Demostración del cumplimiento del punto problema captura de pantalla (compilación y resultados de ejecución)]

: