

I. Memoria Compartida

1. pruebaShM.cc

```
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas cor
2/Memoria Compartida$ ./a.out
Escriba un numero y para continuar ...
1
Padre-> valor de la variable compartida es Area de memoria compartida
Hijo-> valor de la variable compartida es Area de memoria compartida
Fin del programa ...
```

El programa utiliza la clase Semaforo para funcionar correctamente.

2. PiPorSeries.c con Memoria Compartida

```
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas
2/Memoria Compartida$ g++ PiPorSeries.c
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas
2/Memoria Compartida$ ./a.out
Creating process 11814: starting value 0, finish at 100000
Creating process 11815: starting value 100000, finish at 200000
Creating process 11816: starting value 200000, finish at 300000
Creating process 11817: starting value 300000, finish at 400000
Creating process 11818: starting value 400000, finish at 500000
Creating process 11819: starting value 500000, finish at 600000
Creating process 11820: starting value 600000, finish at 700000
Creating process 11821: starting value 700000, finish at 800000
Creating process 11822: starting value 800000, finish at 900000
Creating process 11823: starting value 900000, finish at 1000000
Valor calculado de Pi es 3.141592 con 1000000 terminos
```

3. sumaUno.cc con Memoria Compartida

```
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas cortas/Semana 2/
Serial version: Valor acumulado por 100 procesos es 100000 en 0.511000 ms
Fork version: Valor acumulado por 100 procesos es 100000 en 75.459000 ms
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas cortas/Semana 2/
```

El programa utiliza también la clase Semaforo para funcionar correctamente.

Instrucciones: compilar los programas utilizando el comando `make` y correrlos con los siguientes comandos: `./pruebashm` `./pishm` `./sumashm`.

II. Buzones

1. Funcionamiento enviarSinClases.c y recibirSinClases.c

```
rigovil@rodrigo:~/Escritorio/UCR/2020
2/Buzones$ ./enviarSin.out
Label: a, status 0
Label: b, status 0
Label: c, status 0
Label: d, status 0
Label: e, status 0
Label: li, status 0
rigovil@rodrigo:~/Escritorio/UCR/2020
2/Buzones$ ./recibirSin.out
Label: a, times 0
rigovil@rodrigo:~/Escritorio/UCR/2020
```

Los archivos no se encuentran en la carpeta de entrega pues solo se pide mostrar su funcionalidad.

2. Funcionamiento enviarConClases.cc y recibirConClases.cc

```
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semest
2/Buzones/enviar & recibir con clases$ make
g++ Buzon.cc enviarConClases.cc -o enviar
g++ Buzon.cc recibirConClases.cc -o recibir
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semest
2/Buzones/enviar & recibir con clases$ ./enviar
Label: a, status 32764
Label: b, status 32764
Label: c, status 32764
Label: d, status 32764
Label: e, status 32764
Label: li, status 32764
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semest
2/Buzones/enviar & recibir con clases$ ./recibir
Label: a, times 0
Label: b, times 1
Label: c, times 2
Label: d, times 3
Label: e, times 4
Label: li, times 5
Buzon::Recibir: No message of desired type
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semest
2/Buzones/enviar & recibir con clases$
```

Instrucciones: ir a la carpeta correspondiente y compilar ambos programas utilizando el comando `make` y correr primero el programa de enviar con el comando `./enviar` y luego correr el programa de recibir con el comando `./recibir`.

3. PiPorSeries.c con Buzones

```
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tarea
2/Buzones$ gcc PiPorSeries.c
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tarea
2/Buzones$ ./a.out
Creating process 65696: starting value 0, finish at 100000
Creating process 65697: starting value 100000, finish at 200000
Creating process 65698: starting value 200000, finish at 300000
Creating process 65699: starting value 300000, finish at 400000
Creating process 65700: starting value 400000, finish at 500000
Creating process 65701: starting value 500000, finish at 600000
Creating process 65702: starting value 600000, finish at 700000
Creating process 65703: starting value 700000, finish at 800000
Creating process 65704: starting value 800000, finish at 900000
Creating process 65705: starting value 900000, finish at 1000000
Valor calculado de Pi es 3.141591 con 1000000 terminos
rigovil@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tarea
```

4. sumaUno.c con Buzones

```
rigovill@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas cortas/tarea 2
/Buzones$ gcc sumaUno.c
rigovill@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas cortas/tarea 2
/Buzones$ ./a.out
Serial version:      Valor acumulado por 100 procesos es  100000  en 0.501000 ms
Fork   version:      Valor acumulado por 100 procesos es  100000  en 76.133000 ms
rigovill@rodrigo:~/Escritorio/UCR/2020 - II Semestre/CI-0122/tareas cortas/tarea 2
```

Instrucciones: compilar ambos programas con el comando `make` y correrlos con los comandos `./piBuzones` `./sumaBuzones`.