

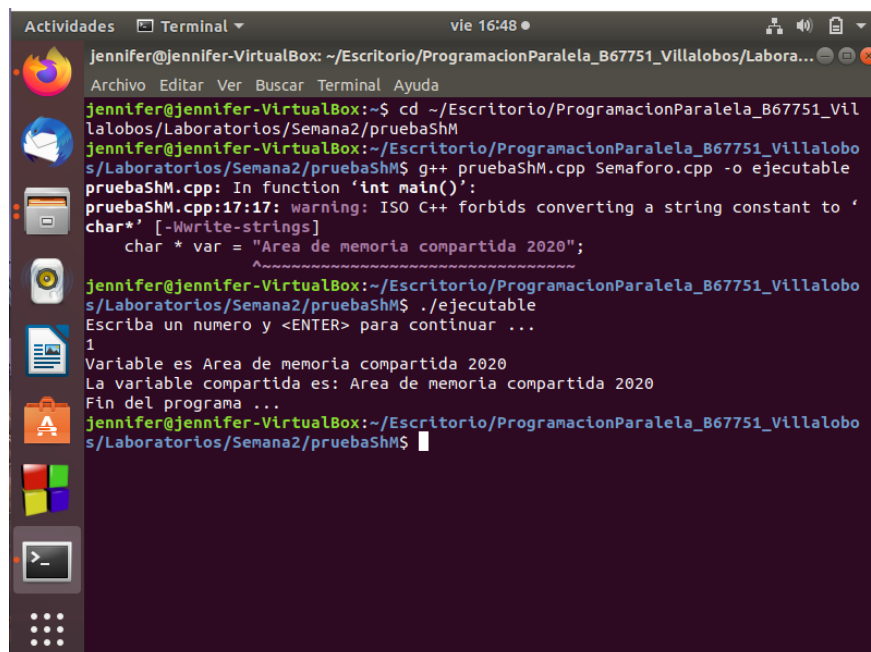
1. Utilizar el "man" para comprender que hacen los llamados al sistema a utilizar: shmget, shmat, shmdt, shmctl.

Utilizando el manual de Linux, se llega a la siguiente conclusión:

- shmget (llave, tamaño, bandera). Este comando retorna el identificador de un nuevo segmento de memoria compartida. Si el proceso tuvo éxito, retorna el identificador, de lo contrario, devolverá un -1.
- shmat(id, dirección, bandera). Esta herramienta se encarga de adjuntar el segmento de memoria compartida a la dirección especificada.
- shmdt(dirección). Se ocupa se separar (o dividir) el segmento de memoria compartida localizado en el parámetro especificado, del espacio localizado en la dirección del proceso invocador del comando.
- shmctl(id, comando, buf). Realiza el comando indicado como parámetro sobre el segmento de memoria compartida con el id especificado.

2. Ejecutar el programa "pruebaShM.c".

El programa pruebaShM necesitaba de las clases Semaforo.h y Semaforo.cpp para lograr su funcionamiento. Se agregaron ambos a un proyecto, junto con el código de la prueba provista por el profesor y se realizó su ejecución.

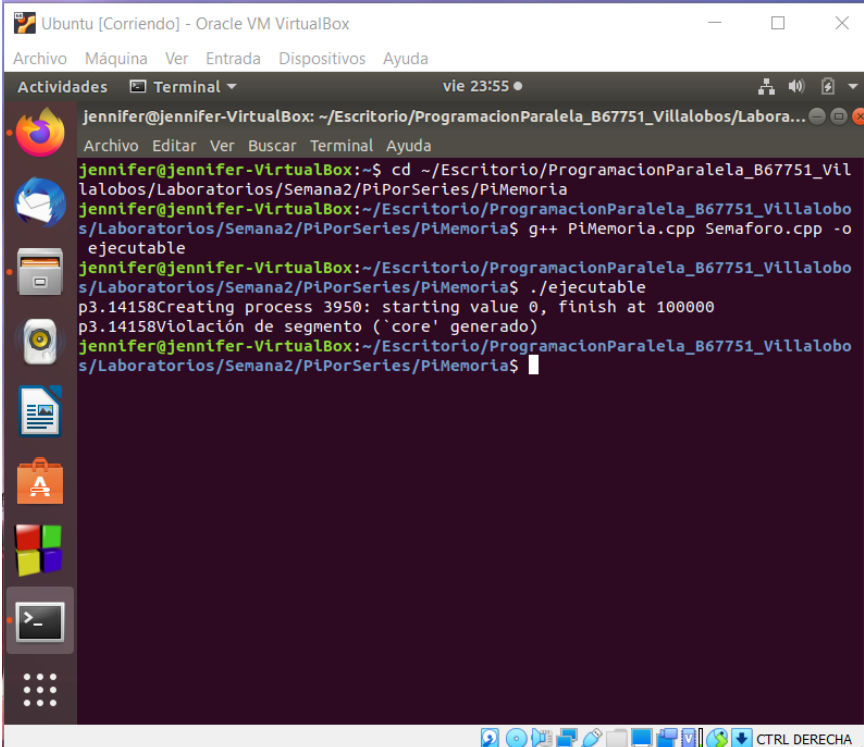


```
Actividades Terminal vie 16:48
Jennifer@jennifer-VirtualBox: ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Labora...
Archivo Editar Ver Buscar Terminal Ayuda
Jennifer@jennifer-VirtualBox:~$ cd ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/pruebaShM
Jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/pruebaShM$ g++ pruebaShM.cpp Semaforo.cpp -o ejecutable
pruebaShM.cpp: In function 'int main()':
pruebaShM.cpp:17:17: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
    char * var = "Area de memoria compartida 2020";
                   ^
Jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/pruebaShM$ ./ejecutable
Escriba un numero y <ENTER> para continuar ...
1
Variable es Area de memoria compartida 2020
La variable compartida es: Area de memoria compartida 2020
Fin del programa ...
Jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/pruebaShM$
```

(a) Ejecución del programa "pruebaShM".

3. Crear una versión de "PiPorSeries.c" que utilice un segmento de memoria compartida para lograr hacer que el programa realice los cálculos correctamente.

El programa se basó en el ejemplo provisto por el profesor (pruebaShM). Se implementó la creación de una segmentación de memoria de una variable (acum) la cual se encarga de acumular el cálculo de Pi provisto por los distintos procesos hijos (del fork). La acumulación, luego de la destrucción del proceso hijo se mantiene, también se encontró la manera de pasar los datos al vector central (casiPi) para ordenar los datos por proceso. Sin embargo, al intentar buscar una forma de comunicar que el proceso hijo había culminado su función, no se encontró una solución. No se logró comunicar que el hijo había cumplido su trabajo y por ende, se podían utilizar los comandos para transferir los resultados al vector. El error encontrado es un segmentation fault.



```
Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal vie 23:55
jennifer@jennifer-VirtualBox: ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Labora...
Archivo Editar Ver Buscar Terminal Ayuda
jennifer@jennifer-VirtualBox:~$ cd ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/PiPorSeries/PiMemoria
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/PiPorSeries/PiMemoria$ g++ PiMemoria.cpp Senaforo.cpp -o ejecutable
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/PiPorSeries/PiMemoria$ ./ejecutable
p3.14158Creating process 3950: starting value 0, finish at 100000
p3.14158Violación de segmento ('core' generado)
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobos/Laboratorios/Semana2/PiPorSeries/PiMemoria$
```

(b) Compilación y ejecución del PiPorSeries de memoria compartida

4. Buzones: construir la implantación de la clase.

Se utilizó el header provisto en el repositorio. La implementación de la clase se dio tomando como base los ejemplos del profesor (en las clases de Enviar y Recibir SinClases). El constructor posee su inicialización con el comando msgget, utilizando el carné como llave. El destructor posee el comando msgctl con su respectiva identificación para lograr la destrucción. Al implementar el método de enviar se utilizó la estructura msgbuf con solo 2 valores: el tipo de mensaje y el mensaje en sí. Se utilizó el comando strcpy para generar una copia del mensaje entrante y depositarlo en el mtext de la estructura, tanto el tipo como el mensaje se enviaron por el comando msgsnd. El último método, Recibir, también poseía su estructura msgbuf y recibía el mensaje y el tipo del mismo como parámetro. Este, sin embargo, presentaba un error al poder recibir los mensajes (sin embargo, los mensajes enviados con EnviarConClases sí eran recibidos por el RecibirSinClases).

```

7  #include <sys/msg.h>
8
9  Buzon::Buzon()
10 {
11     this->id = msgget(0xB67751, IPC_CREAT | 0600); //Identificador del buzón
12     //Para saber si falla
13     if (-1 == id)
14     {
15         perror("Buzon::Buzon");
16         exit(1);
17     }
18 }
19
20 Buzon::~Buzon()
21 {
22     int status;
23     status = msgctl(id, IPC_RMID, NULL);
24     //Para saber si falla
25     if (status == -1)
26     {
27         perror("Buzon::~Buzon");
28         exit(1);
29     }
30 }
31
32

```

Unix (LF) UTF-8 Line 62, Column 23 Insert Read/Wri... default CTRL DERECHA

(c) Declaración del constructor y destructor.

```

31 }
32
33 int Buzon::Enviar(const char * mensaje, int veces, long tipo)
34 {
35     int status;
36     struct msgbuf
37     {
38         long int mtype; // Tipo de mensaje, debe ser >0
39         char mtext[80];
40     } mens;
41     mens.mtype = tipo;
42     strncpy(mens.mtext, mensaje, strlen(mensaje));
43     status = msgsnd(id, & mens, sizeof(mens.mtext), IPC_NOWAIT); //La bandera
44     if (status == -1)
45     {
46         perror("Buzon::Enviar");
47         exit(1);
48     }
49     return status;
50 }
51
52 int Buzon::Recibir(char * mensaje, int * veces, long tipo)
53 {
54     int status;
55     struct msgbuf
56

```

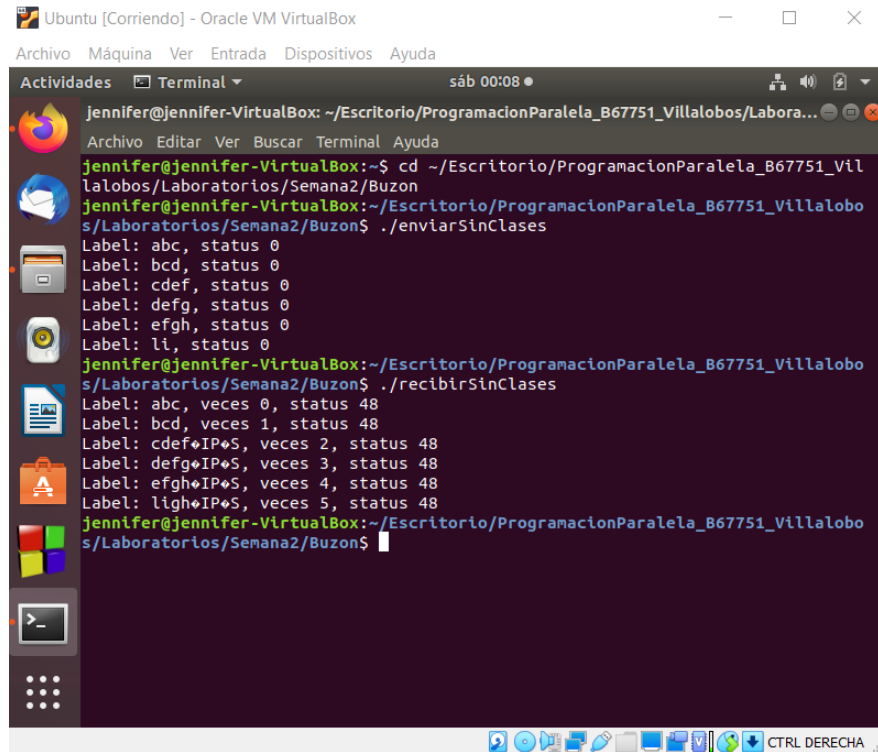
Unix (LF) UTF-8 Line 62, Column 23 Insert Read/Wri... default CTRL DERECHA

(d) Método Enviar.

```
49     }
50     return status;
51 }
52
53 int Buzon::Recibir(char * mensaje, int * veces, long tipo)
54 {
55     int status;
56     struct msgbuf
57     {
58         long int mtype;    // Tipo de mensaje, debe ser >0
59         char mtext[80];
60     } mens;
61
62     mens.mtype = tipo;
63
64     status = msggrcv(id, & mens, sizeof(mens.mtext), tipo, IPC_NOWAIT); //La
65     if (status == -1)
66     {
67         perror("Buzon::Recibir");
68         exit(1);
69     }
70     return status;
71 }
72
73 //int Buzon::Enviar(void * mensaje, int len, int veces, long tipo)
74 //{
```

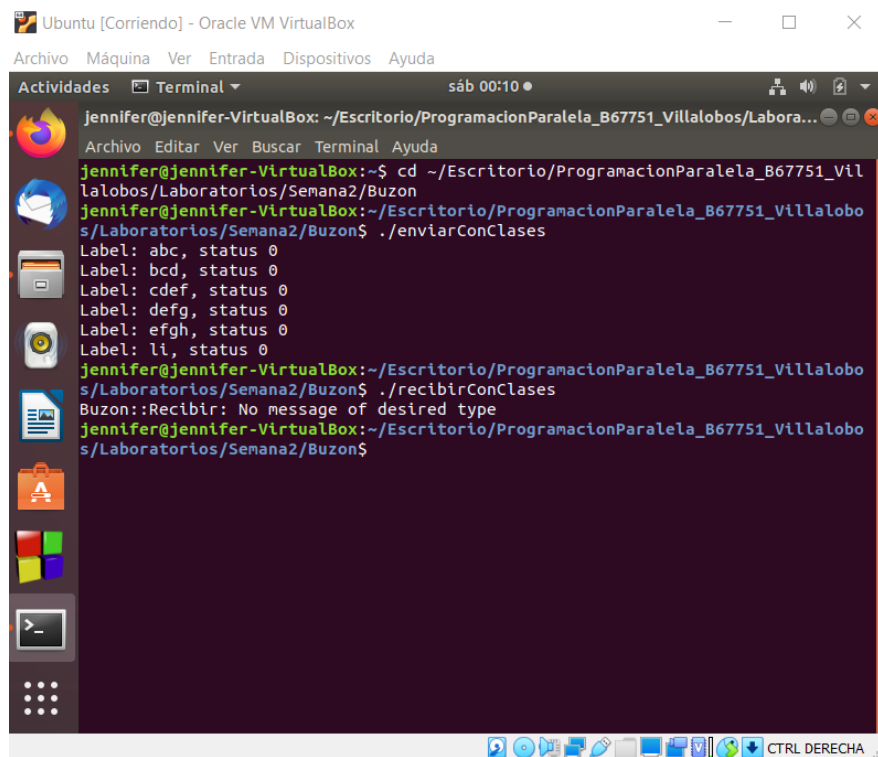
(e) Método Recibir.

- Resultados de ejecución



```
jennifer@jennifer-VirtualBox: ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Labora...
Archivo Editar Ver Buscar Terminal Ayuda
jennifer@jennifer-VirtualBox:~$ cd ~/Escritorio/ProgramacionParalela_B67751_Vil
lalobos/Laboratorios/Semana2/Buzon
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobo
s/Laboratorios/Semana2/Buzon$ ./enviarSinClases
Label: abc, status 0
Label: bcd, status 0
Label: cdef, status 0
Label: defg, status 0
Label: efgh, status 0
Label: li, status 0
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobo
s/Laboratorios/Semana2/Buzon$ ./recibirSinClases
Label: abc, veces 0, status 48
Label: bcd, veces 1, status 48
Label: cdef+IP+S, veces 2, status 48
Label: defg+IP+S, veces 3, status 48
Label: efgh+IP+S, veces 4, status 48
Label: ligh+IP+S, veces 5, status 48
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobo
s/Laboratorios/Semana2/Buzon$
```

(f) Ejecución de enviar y recibir SinClases.



```
jennifer@jennifer-VirtualBox: ~/Escritorio/ProgramacionParalela_B67751_Villalobos/Labora...
Archivo Editar Ver Buscar Terminal Ayuda
jennifer@jennifer-VirtualBox:~$ cd ~/Escritorio/ProgramacionParalela_B67751_Vil
lalobos/Laboratorios/Semana2/Buzon
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobo
s/Laboratorios/Semana2/Buzon$ ./enviarConClases
Label: abc, status 0
Label: bcd, status 0
Label: cdef, status 0
Label: defg, status 0
Label: efgh, status 0
Label: li, status 0
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobo
s/Laboratorios/Semana2/Buzon$ ./recibirConClases
Buzon::Recibir: No message of desired type
jennifer@jennifer-VirtualBox:~/Escritorio/ProgramacionParalela_B67751_Villalobo
s/Laboratorios/Semana2/Buzon$
```

(g) Ejecución de enviar y recibir ConClases.

Figure 1: Demostración de ejecución.

⋮