

CI-0117 Laboratorio/Tarea Programación Paralela y Concurrente

Dayana Marín Mayorga | B64096

Completar una clase Socket para poder intercambiar mensajes entre procesos que no comparten memoria.

Solución:

Se crea dos constructores uno que recibe un entero por parámetro “un identificador”y un constructor vacío.

```
Socket::Socket(int ID){
    this->idSocket = socket(AF_INET, SOCK_STREAM, 0);
    if(-1 == idSocket){
        perror("Socket::Socket");
    }
    SetIDSocket(ID);
}

Socket::Socket(){
    this->idSocket= socket(AF_INET, SOCK_STREAM, 0);
    if(-1 == idSocket){
        perror("Socket::Socket()");
    }
}
// Destrucción
```

Se crea el método Listen, va a recibir conexiones por medio del método Accept().

```
/*
 * Indica al sistema operativo que el socket va a actuar de manera pasiva
 * Utilizara conexiones establecidas por medio de Accept
 */
int Socket::Listen( int queue ) {
    int pas = listen(this->idSocket, queue);
    if ( -1 == pas ) {
        perror( "Socket::Listen" );
        exit( 1 );
    }
    return pas;
}
// ↩
```

Se crea el método Bind, el parámetro determina el número de puerto a utilizar.

```
/*
 * Asocia al socket con el puerto indicado como parametro
 */
int Socket::Bind ( int port ) {
    struct sockaddr_in dir;

    dir.sin_family = AF_INET;
    dir.sin_addr.s_addr = htonl( INADDR_ANY );
    dir.sin_port = htons( port );

    int ID = bind( this->idSocket, (const sockaddr *) & dir, sizeof(dir));

    if ( -1 == ID) {
        perror( "Socket::Bind" );
        exit( 1 );
    }

    return ID;
}
/*
```

Se crea el método Socket * Accept para establecer conexiones con el servidor.

```
/*
 * Acepta conexiones desde los clientes
 * Devuelve una nueva instancia de la clase Socket para manejar la conexion de
 * un cliente
 */
Socket * Socket::Accept()
{
    struct sockaddr_in dir;
    socklen_t dir_len = sizeof(dir);
    int nuevoId = accept(this->idSocket, (struct sockaddr*) &dir, &dir_len);

    if(nuevoId == -1)
    {
        perror("Socket::Accept");
        exit(1);
    }
    Socket * nuevoSocket = new Socket(nuevoId);
    return nuevoSocket;
}
/*
```

Se crea el método ShutDown el cual cierra parcialmente el socket de acuerdo con el parámetro.

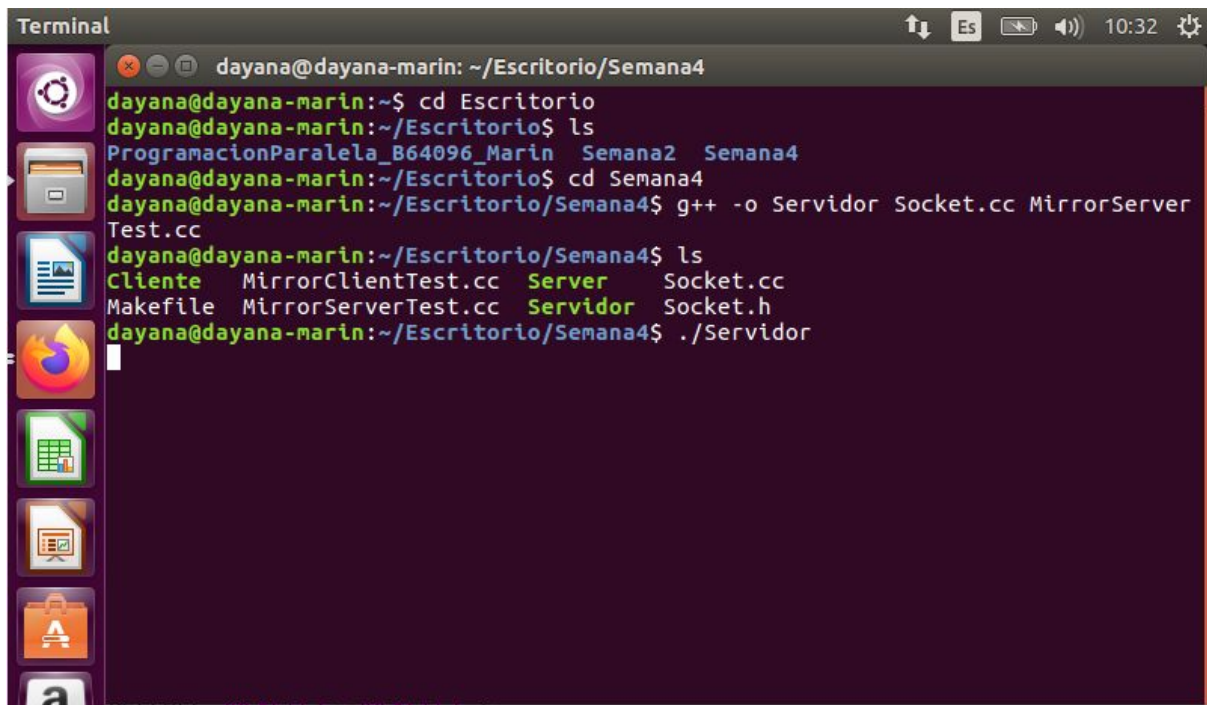
```
/*
 * Cierra parcialmente la conectividad de un socket, puede ser por escrituras o
 lecturas
 * El parametro "mode" indica el tipo de cierre que se quiere efectuar
 */
int Socket::Shutdown( int mode ) {
int apagado = shutdown(this->idSocket,mode);
if( apagado == -1){
perror( "Socket::Shutdown" );
exit( 1 );
}
return apagado;
}
```

nuevo método Write con solo un parámetro.

```
/*
 * Escribe la tira de caracteres a traves de socket
 */
int Socket::Write( char *text ) {
int escr;
escr = write(this->idSocket,(void*)text,strlen(text));
if ( -1 == escr){
perror("Socket::Write");
}
return escr;
}
```

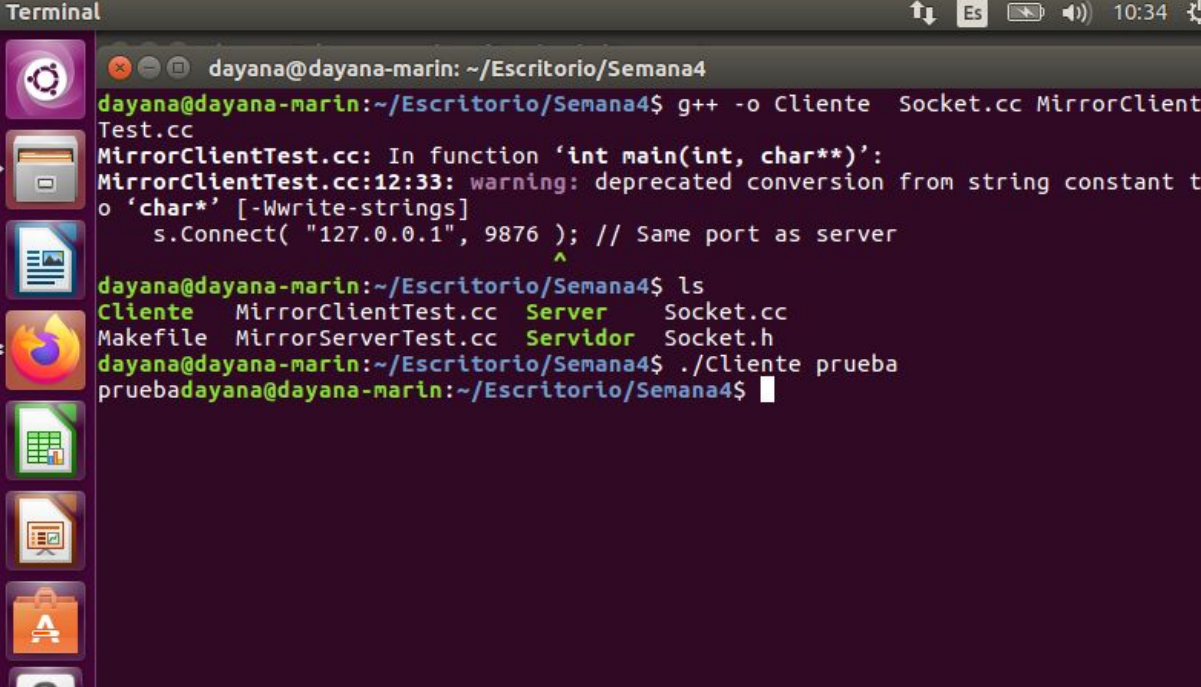
Las clases MirrorServerTest.cc y MirrorClientTest.cc compilan conforme a lo esperado:

Compilación MirrorServerTest.cc:

A terminal window titled "Terminal" with a dark background and light text. The window shows a series of commands and their outputs. The user is in the directory ~/Escritorio/Semana4. The commands executed are: 1. cd Escritorio, 2. ls (showing ProgramacionParalela_B64096_Marin, Semana2, and Semana4), 3. cd Semana4, 4. g++ -o Servidor Socket.cc MirrorServerTest.cc, 5. ls (showing Cliente, MirrorClientTest.cc, Server, Socket.cc, Makefile, MirrorServerTest.cc, Servidor, and Socket.h), and 6. ./Servidor. The terminal output shows the successful compilation of the program into an executable named Servidor.

```
Terminal
dayana@dayana-marin: ~/Escritorio/Semana4
dayana@dayana-marin:~$ cd Escritorio
dayana@dayana-marin:~/Escritorio$ ls
ProgramacionParalela_B64096_Marin  Semana2  Semana4
dayana@dayana-marin:~/Escritorio$ cd Semana4
dayana@dayana-marin:~/Escritorio/Semana4$ g++ -o Servidor Socket.cc MirrorServer
Test.cc
dayana@dayana-marin:~/Escritorio/Semana4$ ls
Cliente  MirrorClientTest.cc  Server  Socket.cc
Makefile MirrorServerTest.cc  Servidor Socket.h
dayana@dayana-marin:~/Escritorio/Semana4$ ./Servidor
```

Compilación MirrorClientTest.cc:



The image shows a terminal window titled "Terminal" with a dark background. The user is logged in as "dayana" on a machine named "dayana-marin". The current directory is "~/Escritorio/Semana4". The terminal shows the following commands and output:

```
dayana@dayana-marin: ~/Escritorio/Semana4
dayana@dayana-marin:~/Escritorio/Semana4$ g++ -o Cliente Socket.cc MirrorClientTest.cc
MirrorClientTest.cc: In function 'int main(int, char**)':
MirrorClientTest.cc:12:33: warning: deprecated conversion from string constant to 'char*' [-Wwrite-strings]
    s.Connect( "127.0.0.1", 9876 ); // Same port as server
                        ^
dayana@dayana-marin:~/Escritorio/Semana4$ ls
Cliente  MirrorClientTest.cc  Server  Socket.cc
Makefile MirrorServerTest.cc  Servidor Socket.h
dayana@dayana-marin:~/Escritorio/Semana4$ ./Cliente prueba
pruebadayana@dayana-marin:~/Escritorio/Semana4$
```

The terminal window has a sidebar on the left with icons for various applications: a terminal icon, a file manager icon, a web browser icon, a spreadsheet icon, a presentation icon, and a folder icon. The top of the window shows the title "Terminal" and some system icons on the right, including a volume icon and a clock showing "10:34".


```
dayana@dayana-marin: ~/Escritorio/Semana4
dayana@dayana-marin:~$ cd Escritorio
dayana@dayana-marin:~/Escritorio$ cd Semana4
dayana@dayana-marin:~/Escritorio/Semana4$ ls
Makefile  MirrorClientTest.cc  MirrorServerTest.cc  Socket.cc  Socket.h
dayana@dayana-marin:~/Escritorio/Semana4$ g++ -o Server Socket.cc MirrorServerTest.cc
dayana@dayana-marin:~/Escritorio/Semana4$ ls
Makefile  MirrorClientTest.cc  MirrorServerTest.cc  Server  Socket.cc  Socket.h
dayana@dayana-marin:~/Escritorio/Semana4$ ./Server

dayana@dayana-marin: ~/Escritorio/Semana4
dayana@dayana-marin:~$ cd Escritorio
dayana@dayana-marin:~/Escritorio$ cd Semana4
dayana@dayana-marin:~/Escritorio/Semana4$ ls
Makefile  MirrorClientTest.cc  MirrorServerTest.cc  Server  Socket.cc  Socket.h
dayana@dayana-marin:~/Escritorio/Semana4$ g++ -o Cliente Socket.cc MirrorClientTest.cc
MirrorClientTest.cc: In function 'int main(int, char**)':
MirrorClientTest.cc:12:33: warning: deprecated conversion from string constant to 'char*' [-Wwrite-strings]
    s.Connect( "127.0.0.1", 9876 ); // Same port as server
                        ^
dayana@dayana-marin:~/Escritorio/Semana4$ ls
Cliente  MirrorClientTest.cc  Server  Socket.h
Makefile  MirrorServerTest.cc  Socket.cc
dayana@dayana-marin:~/Escritorio/Semana4$ ./Cliente Prueba
Pruebadayana@dayana-marin:~/Escritorio/Semana4$
```

PD: Para esta documentación no se incluye los métodos del laboratorio 3.