

CI-0117 Laboratorio/Tarea Programada/Programación Paralela y Concurrente/Dayana Marín Mayorga | B64096

Ejercicio a resolver:

Los ejemplos "ipv4-test.cc" y "ipv6-test.cc" deben funcionar correctamente.

Se crea la clase Socket.cc en la cual se implementa los siguientes métodos:

Primera se crea el constructor. El primer parámetro nos indica el tipo socket y el segundo indica si es un socket tipo IPv4 o IPv6.

Se establece el dominio de los sockets en este constructor.

```
Socket::Socket( char tipo, bool ipv6 ){
    if (ipv6){
        this->idSocket = socket(AF_INET, SOCK_STREAM, 0);
    }else{
        switch(tipo){
            case 'd':
            case 'D':
                this->idSocket = socket(AF_INET, SOCK_DGRAM, 0);
                break;
            default:
                this->idSocket = socket(AF_INET, SOCK_STREAM, 0);
                break;
        }
    }
    if(-1 == this->idSocket){
        perror("Socket::Socket");
        exit(1);
    }
}
```

- Después se crea el destructor utilizando el método close().

```
//Destructor
Socket::~Socket(){
    close();
}

void Socket::Close(){
    int retorno;
    retorno = close(this->idSocket);
    if(-1 == retorno){
        perror("Socket::Close()");
        exit(1);
    }
}
```

- Se crea el m3todo Connect().

```
int Socket::Connect( char * hostip, int port ) {  
    struct sockaddr_in direction;  
    direction.sin_family = AF_INET;  
    direction.sin_port = htons(port);  
    inet_pton(AF_INET, hostip, & direction.sin_addr);  
    int retorno = connect(this->idSocket,(struct sockaddr*)& direction,sizeof  
    (direction));  
    return retorno;  
}
```

Se crean los m3todos Read () y Write()

```
int Socket::Read( char *text, int len ) {  
    int retorno;  
    retorno = read(this->idSocket,(void*)text,len);  
    if (-1 == retorno){  
        perror("Socket::Read()");  
    }  
    return retorno;  
}  
  
int Socket::Write( char *text, int len) {  
    int retorno;  
    retorno = write(this->idSocket,(void*)text,len);  
    if (-1 == retorno){  
        perror("Socket::Write");  
    }  
    return retorno;  
}  
  
void Socket::SetIDSocket(int id){  
    idSocket = id;  
}
```

-Al compilar ipv4-test.cc se genera lo esperado.

```
dayana@dayana-marin:~/Escritorio/Semana3$ ls
ipv4-test.cc  ipv6-test.cc  lab3  Makefile  Socket.cc  Socket.h
dayana@dayana-marin:~/Escritorio/Semana3$ ./lab3
HTTP/1.1 200 OK
Date: Fri, 01 May 2020 20:50:54 GMT
Server: Apache
Last-Modified: Thu, 23 Mar 2017 23:50:36 GMT
ETag: "4b-54b6e8b1d6b00"
Accept-Ranges: bytes
Content-Length: 75
Content-Type: text/html; charset=UTF-8

<HTML>
<CENTER>
<IMG SRC=/RioCeleste-VolcanTenorio.jpg>
</CENTER>
</HTML>
```

-Para este laboratorio también se esperaba que al compilar el archivo ipv6-test.cc lo cual se hizo pero dio el siguiente resultado:

```
dayana@dayana-marin:~/Escritorio/ProgramacionParalela_B64096_Marin/Laboratorios/
Semana3.B64096$ ls
a.out  ipv6-test.cc  lab3  Makefile  Socket.cc
ejeipv6  lab  laipv6  server.c  Socket.h
dayana@dayana-marin:~/Escritorio/ProgramacionParalela_B64096_Marin/Laboratorios/
Semana3.B64096$ g++ -o UltimaPrueba Socket.cc ipv6-test.cc
dayana@dayana-marin:~/Escritorio/ProgramacionParalela_B64096_Marin/Laboratorios/
Semana3.B64096$ ls
a.out  ipv6-test.cc  lab3  Makefile  Socket.cc  UltimaPrueba
ejeipv6  lab  laipv6  server.c  Socket.h
dayana@dayana-marin:~/Escritorio/ProgramacionParalela_B64096_Marin/Laboratorios/
Semana3.B64096$ ./UltimaPrueba
H
dayana@dayana-marin:~/Escritorio/ProgramacionParalela_B64096_Marin/Laboratorios/
Semana3.B64096$
```

Metodo connect() para ipv6:

```
int Socket::Connect( char *host, char *service ) {
    int retorno;
    struct addrinfo hints, *resultado, *rp;

    hints.ai_family = AF_UNSPEC;
    hints.ai_socktype = SOCK_STREAM;
    hints.ai_flags = 0;
    hints.ai_protocol = 0;

    retorno = getaddrinfo( host, service, &hints, &resultado);

    for ( rp = resultado; rp; rp = rp->ai_next ) {
        retorno = connect( idSocket, rp->ai_addr, rp->ai_addrlen );
        if ( -1 == retorno )
            break;
    }

    freeaddrinfo( resultado );

    return retorno;
}
```