

INSTITUTO TECNOLOGICO DE COSTA RICA

ADMINISTRACION DE TECNOLOGIAS DE INFORMACION

ESCUELA DE COMPUTACION

LENGUAJES DE PROGRAMACIÓN

I TAREA PROGRAMADA

“MENSAJERIA INSTANTANEA – LENGUAJE C”

ESTUDIANTES:

LESLIE BECERRA

BAYRON PORTUGUEZ

BRAULIO ALPIZAR

GRUPO 1

FECHA ENTREGA:

JUEVES 13 DE SETIEMBRE DE 2012

Tabla de Contenidos

Descripción del problema.....3

Diseño del programa.....4

Librerías usadas.....5

Análisis de resultados.....6

Manual de usuario.....7

Conclusiones personales.....11

DOCUMENTACION EXTERNA

Descripción del problema

Este proyecto programado consistio en el desarrollo de un programa de mensajería instantanea haciendo uso del paradigma imperativo. Para ello este proyecto fue programado en lenguaje C el cual fue creado en 1972 por Dennis M. Ritchie.

Para la elaboración de este programa será necesaria la implementación de sockets los cuales son mecanismos para transportar datos mediante las comunicaciones orientadas a conexión. Para ello será de necesario realizar una investigación así como repetidas pruebas para alcanzar el dominio suficiente para lograr la correcta impletación de los mismos pues son la parte fundamental del sistema de mensajería.

Por otro lado se deberá investigar sobre la función `fork()` la cual bifurca procesos dentro de un programa. Esta función será necesaria pues se espera que el programa trabaje bajo un archivo de ejecución único, es decir que no se requiera de archivos distintos para que se conecten servidores o clientes, si no que opere en uno mismo según la necesidad del proceso.

Este programa se desarrollará para el Sistema Operativos Linux por lo que tambien será necesario investigar sobre sus particularidades en el uso de las funciones antes descritas y para el desarrollo general del programa.

Diseño del programa

En el main se definen tres arreglos que contienen los dos puertos y una IP que introduce en usuario para poder realizar la conexión estos datos se pasan como parámetros al método cliente y al método servidor, después este se inicia para esperar a un cliente, el método cliente verifica esta verificando que exista un servidor al cual conectarse, una vez lo encuentre realiza la conexión entre ambas computadoras. Esto se puede realizar gracias al “fork” el cual bifurca los procesos y permite ejecutar servidor y cliente en la misma computadora, de esta forma los métodos del servidor solo reciben información y los métodos del cliente solo se encargan de escribir.

ESTADISTICA DE TIEMPOS

	Horas
Analisis de Requerimientos	8
Diseño de la Aplicacion	15
Investigacion de Funciones	30
Programacion	10
Documentacion Interna	2
Pruebas	3
Elaboracion de documentos	2
TOTAL	60

Librerías usadas

`#include <sys/socket.h>`

Librería para el manejo de la declaración de constantes tipos y funciones sobre sockets

`#include <sys/types.h>`

`#include <netinet/in.h>`

`#include <arpa/inet.h>`

`#include <netdb.h>`

`#include <stdio.h>`

Standard input-output header proporciona el núcleo de las capacidades de entrada/salida mediante las definiciones de macros, las constantes, las declaraciones de funciones y la definición de tipos usados por varias operaciones estándar de entrada y salida.

`#include <string.h>`

Para manipulación de cadenas de caracteres.

`#include <stdlib.h>`

Para realizar ciertas operaciones como conversión de tipos, generación de números pseudo-aleatorios, gestión de memoria dinámica, control de procesos, funciones de entorno, de señalización (??), de ordenación y búsqueda.

`#include <unistd.h>`

Es la librería que proporciona acceso al sistema operativo POSIX API. Se define por el estándar POSIX.1, la base de la Single Unix Specification ', y por lo tanto deben estar disponibles en todo conforme (o casi-conforme) del sistema operativo (todas las versiones oficiales de Unix, incluyendo Mac OS X, Linux, etc).

`#include <errno.h>`

Para analizar los códigos de error devueltos por las funciones de biblioteca. Un valor es almacenado en `errno` por determinadas funciones de biblioteca cuando detectan errores. Al inicio del programa, el valor almacenado es cero. Las funciones de la librería sólo almacena valores superiores a cero. Cualquier función de biblioteca puede alterar el valor almacenado antes de vuelta, o no detectar errores.

`#include <wait.h>`

Para definición de constantes simbólicas de espera.

Análisis de resultados

A continuación se presentará una tabla con el análisis de resultados de las principales funcionalidades dentro del programa desarrollado (MI). Por cada rubro se asignó un puntaje de 1 a 5 según el nivel de éxito y satisfacción que el grupo considera alcanzado.

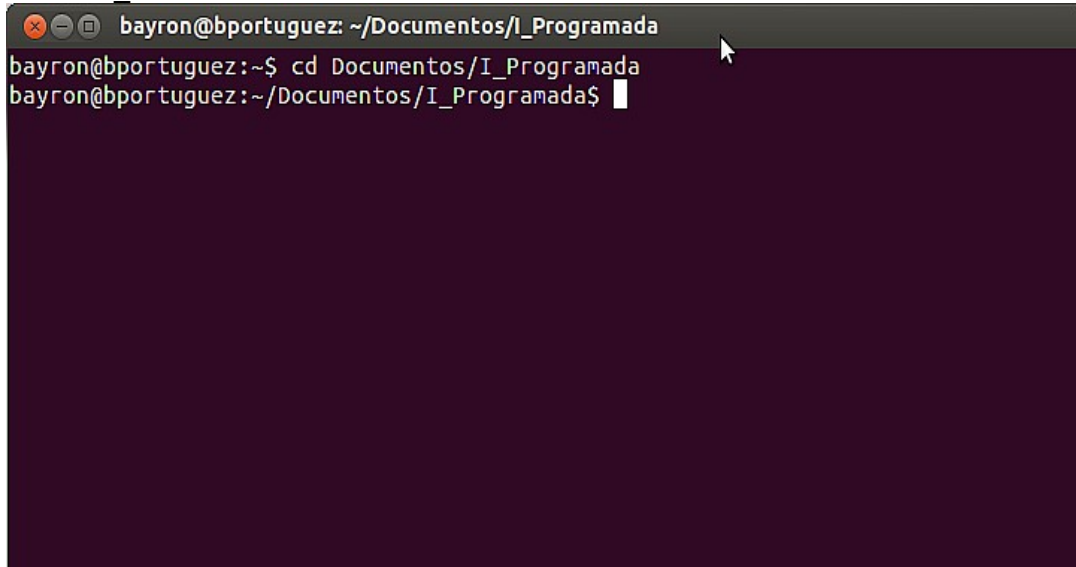
Función o segmento del programa	Se logró implementar	No se logró implementar
Implementación del programa Servidor (sockets)	4	
Implementación del programa Cliente (sockets)	5	
Comunicación entre Servidor y Cliente mediante puertos y IP entre diferentes maquinas por conexión de red cableada	5	
Bifurcación del programa Servidor	5	
Bifurcación del programa Cliente	5	
Manejo y utilización de colores en la terminal de linux	5	
Utilización de un medio de fácil comprensión para las comunicaciones	5	
Implementación de interfaz grafica		1
Utilización del espacio Github para el trabajo del código	3	

El uso de los sockets se logró con éxito sin embargo uno de los puertos del servidor no se logró cerrar, se hicieron varios intentos y pruebas pero por cuestiones de tiempo y de como se trabajaron los algoritmos no se pudo solucionar. Pensamos que esto podría solucionarse con una reacomodo de las validaciones general y corrección de las variables, además separaríamos de una forma diferente los procesos de forma que permita una mayor interacción con los procesos.

Manual de usuario

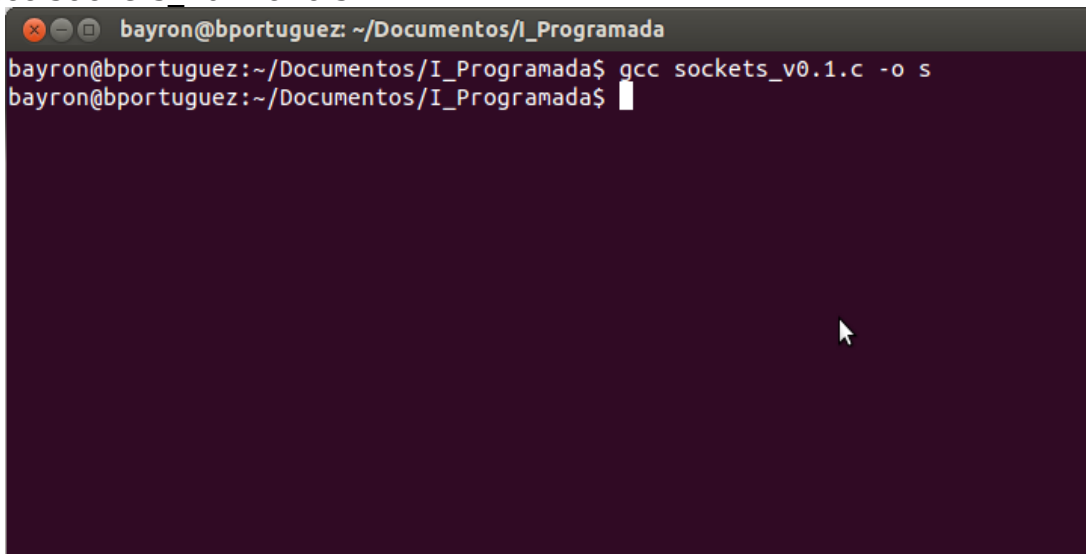
A continuación se explicara sobre el uso del programa MI.

1. Primero deben abrirse en linux 2 terminales y localizar el archivo "sockets_v0.1.c"

A terminal window with a dark purple background. The title bar shows the window name 'bayron@bportuguez: ~/Documentos/I_Programada'. The terminal content shows the user navigating to the directory 'Documentos/I_Programada' using the 'cd' command. The prompt changes from '~\$' to '~/Documentos/I_Programada\$'.

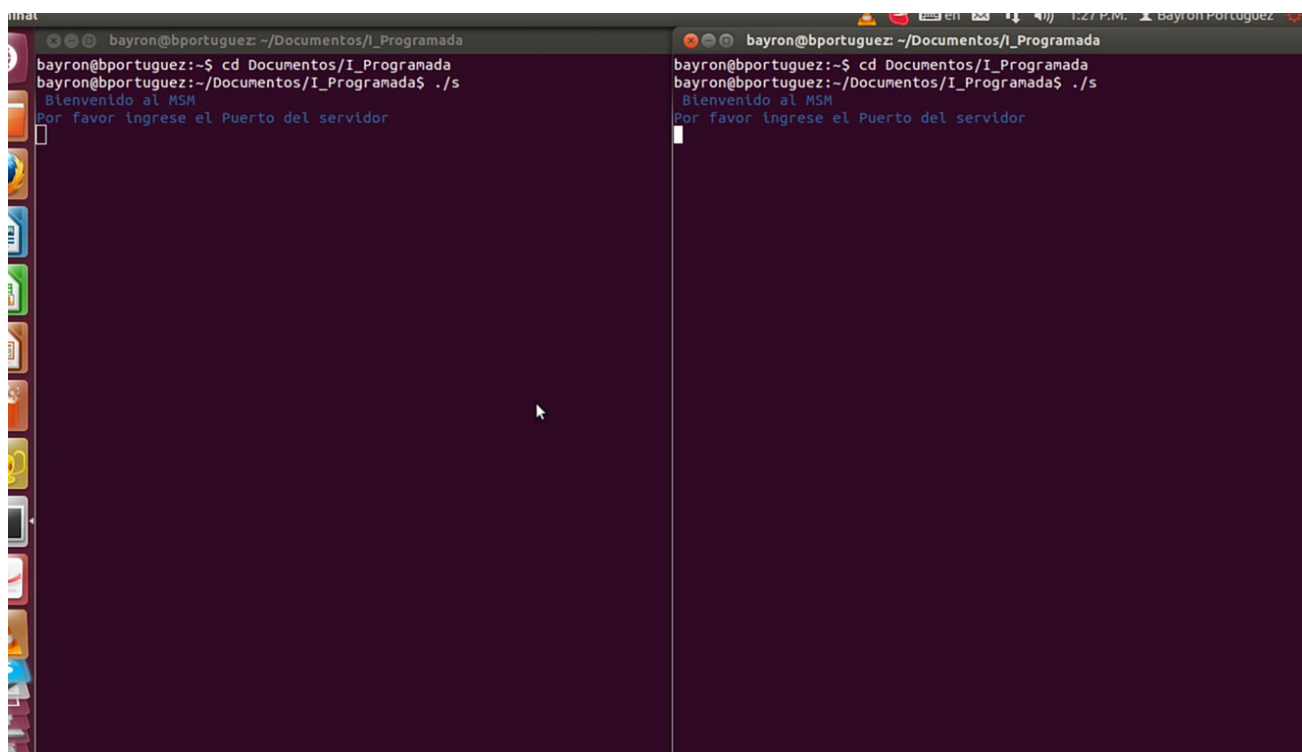
```
bayron@bportuguez: ~/Documentos/I_Programada
bayron@bportuguez:~$ cd Documentos/I_Programada
bayron@bportuguez:~/Documentos/I_Programada$
```

2. Luego se debe proceder a compilar el programa de la siguiente forma:
gcc sockets_v0.1.c -o s

A terminal window with a dark purple background. The title bar shows the window name 'bayron@bportuguez: ~/Documentos/I_Programada'. The terminal content shows the user compiling the file 'sockets_v0.1.c' into an executable named 's' using the 'gcc' command. The prompt remains the same as in the previous terminal.

```
bayron@bportuguez:~/Documentos/I_Programada$ gcc sockets_v0.1.c -o s
bayron@bportuguez:~/Documentos/I_Programada$
```

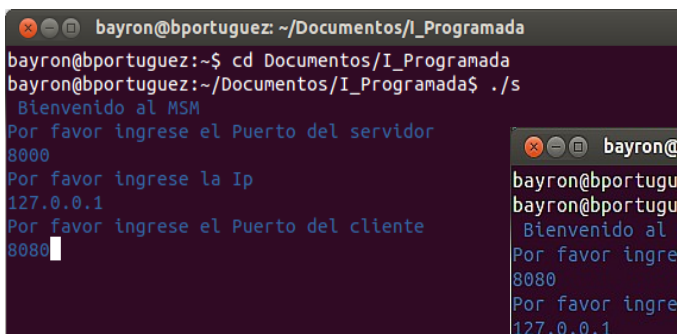
3. Despues procedemos a ejecutar el archivo en ambas consolas de la siguiente forma:
./s



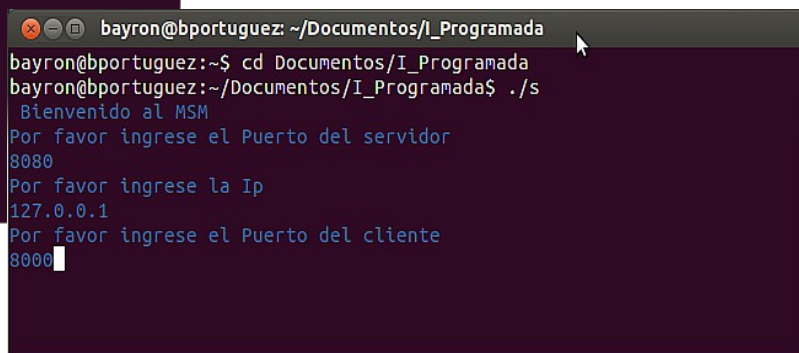
4. Procedemos a ingresar los datos en una terminal
 - Primero el puerto del servidor
 - Segundo la IP de la maquina a que queremos llegar
 - Tercero nuestro puerto
 - Y damos Enter

** Lo mismo debe ejecutarse en la terminal 2

Terminal 1



Terminal 2



5. Posterior a esto estamos listos para comunicarnos las pantallas mostrarán lo siguiente:

```
bayron@bportuguez: ~/Documentos/l_Programada
Por favor ingrese el Puerto del servidor
8000
Por favor ingrese la Ip
127.0.0.1
Por favor ingrese el Puerto del cliente
8080

Esperando la conexion en el puerto 8000

Se establecio la conexion (127.0.0.1 , 43108)

YO (Adios para cerrar el MSN) :
```

Esto indica que usted ya puede enviar mensajes a otro usuario del programa.

6. Para enviar un mensaje solo escriba lo que desea comunicar y presione Enter

Terminal 1

```
bayron@bportuguez: ~/Documentos/l_Programada
8000
Por favor ingrese la Ip
127.0.0.1
Por favor ingrese el Puerto del cliente
8080

Esperando la conexion en el puerto 8000

Se establecio la conexion (127.0.0.1 , 43108)

YO (Adios para cerrar el MSN) :Hola
YO (Adios para cerrar el MSN) :
```

Terminal 2

```
bayron@bportuguez: ~/Documentos/l_Programada
Por favor ingrese la Ip
127.0.0.1
Por favor ingrese el Puerto del cliente
8080

Esperando la conexion en el puerto 8080

YO (Adios para cerrar el MSN) :
Se establecio la conexion (127.0.0.1 , 54322)

Otro Usuario = Hola
```

Terminal 1

```
bayron@bportuguez: ~/Documentos/I_Programada
127.0.0.1
Por favor ingrese el Puerto del cliente
8080

Esperando la conexion en el puerto 8080

Se establecio la conexion (127.0.0.1 , 43108)

Y0 (Adios para cerrar el MSN) :Hola

Y0 (Adios para cerrar el MSN) :
Otro Usuario = Hola, ¿como estas?
```

Terminal 2

```
bayron@bportuguez: ~/Documentos/I_Programada

Esperando la conexion en el puerto 8080

Y0 (Adios para cerrar el MSN) :
Se establecio la conexion (127.0.0.1 , 54322)

Otro Usuario = Hola

Y0 (Adios para cerrar el MSN) :Hola, ¿como estas?

Y0 (Adios para cerrar el MSN) :
```

7. Para cerrar la comunicación digite “Adios”

```
bayron@bportuguez: ~/Documentos/I_Programada

Esperando la conexion en el puerto 8080

Se establecio la conexion (127.0.0.1 , 43108)

Y0 (Adios para cerrar el MSN) :Hola

Y0 (Adios para cerrar el MSN) :
Otro Usuario = Hola, ¿como estas?

Y0 (Adios para cerrar el MSN) :Adios

Y0 (Adios para cerrar el MSN) :
```

```
bayron@bportuguez: ~/Documentos/I_Programada

Esperando la conexion en el puerto 8080

Y0 (Adios para cerrar el MSN) :
Se establecio la conexion (127.0.0.1 , 54322)

Otro Usuario = Hola

Y0 (Adios para cerrar el MSN) :Hola, ¿como estas?

Y0 (Adios para cerrar el MSN) :bayron@bportuguez:~/Documentos/I_Programada$
```

Con esto el otro usuario ya no podrá enviarle mas mensajes.
Posteriormente puede cerrar las terminales.

Conclusiones personales

Este trabajo nos permitió adentrarnos un poco más en la programación en lenguaje C. Mesclando lo aprendido en clase con la intensa investigación realizada por el equipo de trabajo para la elaboración del código. Aunque faltó afinar algunos detalles el equipo está bastante satisfecho con la realización de este proyecto. Con la elaboración de este proyecto aprendimos la importancia del uso de punteros así como el uso de las distintas librerías que C ofrece.

Además nos permitió desarrollar habilidades como trabajo en equipo y liderazgo las cuales son pilares importantes dentro de la carrera.