

Tarea Programada 1

Lenguajes de Programación



Andrei Fuentes Leiva

Kevin Piedra

Daniel Rodríguez

2016136187

Gabriel Omar Piedra Camacho

2016078985

23 de Marzo del 2019

<b>Introducción</b>	<b>3</b>
<b>Diseño</b>	<b>4</b>

# Introducción

Se desarrollara un sistema chat que utilizan sockets para lo coneccion entre el cliente y el servidor, el servidor podrá aceptar varias conecciones de clientes al mismo tiempo además de esto los clientes pueden enviar y recibir mensajes al mismo tiempo, para esto se usarán forks(), además de esto se mantendrá control de los clientes conectados, una vez conectados un cliente decidirá a quién quiere enviar un mensaje y puede escribirlo, el server se encargará de enviar el mensaje al usuario correcto.

La función fork() una vez después de llamada crea un nuevo proceso que correrá de donde se llamó el fork() en adelante, este nuevo proceso es conocido como el proceso hijo, el padre también ejecutará el código que sigue después de la llamada del fork(), para poder identificar cual es el proceso hijo nadamas se debe ver el valor del return del fork() si es 0 significa que es el hijo.

# Diseño

El sistema será diseñado para el sistema operativo Linux en el lenguaje de programación C, el sistema consistirá de dos programas independientes el programa del cliente y el programa del server, el server se abrirá en un puerto especificado y los clientes especificara a qué puerto se quieren conectar a través de un archivo aparte.

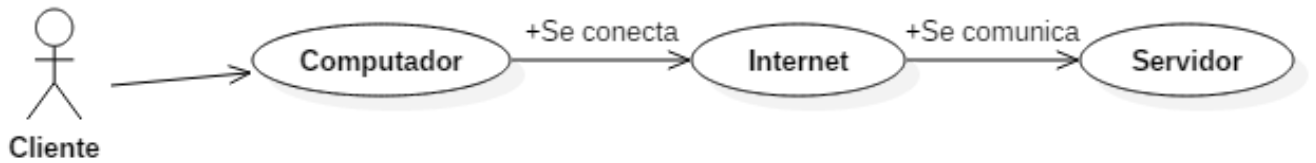
El cliente usará la función `fork()` para poder enviar mensajes y recibirlos al mismo tiempo, al usar `fork()` uno de los procesos se enfocara en escuchar cualquier mensajes que pueda venir del server mientras que el otro procesos escuchara el input del usuario para poder enviar un mensaje al server.

El cliente tendrá un archivo aparte del programa donde especificara en qué puerto se quiere conectar, una vez que se inicia el programa del cliente se leerá este archivo se verá el puerto que desea usar el usuario y se intentara crear una coneccion en ese puerto con el servidor.

Los mensajes que se reciben de otros usuarios serán formateados de manera diferente, cuando se recibe un mensaje de otro usuario este se colocara a la derecha de la terminal y además se mostrarán en color verde para diferenciarlos de los mensajes que se escriben.

El servidor mantendrá control de todos los usuarios que se conectan a este mismo, el servidor se asegurara que no pueden haber dos usuarios con el mismo nombre en el servidor al mismo tiempo, y se asegurará que una vez que algún cliente salga del sistema ese nombre de usuario se libere y pueda ser usado por otros nuevos usuarios.

# Diagrama de Arquitectura



## Referencias

A. M. (2016, September 17). Adding Color to Your Output From C. Tomado de:  
<http://web.theurbanpenguin.com/adding-color-to-your-output-from-c/>

Fork(2). (2015, September 15). Tomado de:  
<http://man7.org/linux/man-pages/man2/fork.2.html>

Robert I. Pitts. (s.f). Intro to File Input/Output in C. Tomado de:  
<https://www.cs.bu.edu/teaching/c/file-io/intro/>

Programming simplified. (s.f.). C read file program. Tomado de:  
<https://www.programmingsimplified.com/c-program-read-file>