



INSTITUTO TECNOLÓGICO DE COSTA RICA
INGENIERÍA EN COMPUTACIÓN
PRINCIPIOS DE SISTEMAS OPERATIVOS

Documentación : Rust the Future Machine

Andres Ramirez Ortega - 2018172107
Joel Vega Godínez - 2018163840

Profesor:
Kevin Moraga García

**Sede Inter-Universitaria Alajuela
27 de Abril 2021**

Contents

1	Introducción	4
2	Ambiente de desarrollo	5
2.1	Joel Vega Godínez	5
2.1.1	C	5
2.1.2	Navegador	5
2.1.3	Sistema Operativo	5
2.1.4	Especificaciones de Computadora	5
2.1.5	Bash	5
2.1.6	Editor de texto	5
2.1.7	Gcc	5
2.1.8	Rustc	5
2.1.9	Herramienta de diagramas	5
2.2	Andrés Ramírez Ortega	6
2.2.1	C	6
2.2.2	Navegador	6
2.2.3	Sistema Operativo	6
2.2.4	Especificaciones de Computadora	6
2.2.5	Bash	6
2.2.6	Editor de texto	6
2.2.7	Gcc	6
2.2.8	Rustc	6
2.2.9	Herramienta de diagramas	6
3	Estructuras de datos usadas y funciones	7
4	Instrucciones para ejecutar el programa	8
5	Actividades realizadas por estudiante	9
5.1	Actividades Joel Vega	9
5.2	Actividades Andrés Ramírez	9
6	Autoevaluación	10
6.1	Joel Vega	10
6.2	Andrés Ramírez	10

7	Lecciones Aprendidas	11
7.1	Joel Vega	11
7.2	Andrés Ramírez	11

1 Introducción

El intercambio de datos entre los diferentes servidores y computadoras es muy importante. Los beneficios de contar con un servidor FTP son muchos, desde la comodidad y la facilidad hasta los costos de implementación, la velocidad y la flexibilidad en los diferentes procesos que se realizan.

Muchas veces, estos servidores se sobre utilizan debido a la gran cantidad de peticiones que se hacen y a la alta demanda. Esto se demuestra en los casos de eventos masivos y ventas de boletos de conciertos y películas muy famosas. Para evitar estos colapsos y caídas de servidores existen técnicas que pueden ayudar a mejorar la calidad y el funcionamiento. El pre-thread consiste la creación con anterioridad de hilos, los cuales atienden las solicitudes. El pre-forked es la creación de bifurcaciones que manejan las diferentes solicitudes, la cual se puede utilizar cuando se tienen bibliotecas que no son seguras para subprocesos, lo cual permite que, los problemas dentro de una solicitud, afecten solamente el proceso en el que se encuentran y no todo el servidor. Estas dos técnicas ayudan a administrar los recursos de un servidor y estas son las que serán implementadas en esta tarea.

2 Ambiente de desarrollo

2.1 Joel Vega Godínez

2.1.1 C

Ubuntu GLIBC 2.31-0ubuntu9.1

2.1.2 Navegador

Mozilla Firefox 82.0 (64 bits)

2.1.3 Sistema Operativo

Linux Lite 5.2

2.1.4 Especificaciones de Computadora

Notebook HP

producto: HP Notebook (X6Y01LAABM)

descripción: CPU AMD E2-7110 APU with AMD Radeon R2 Graphics

descripción: BIOS vendor: Insyde version: F.26

descripción: System Memory slot: System board or motherboard size: 4GiB

2.1.5 Bash

GNU bash, version 5.0.17(1)-release (x86_64 - pc - linux - gnu)

2.1.6 Editor de texto

gedit - Version 3.36.2

2.1.7 Gcc

gcc (Ubuntu 9.3.0-17ubuntu1 20.04) 9.3.0

2.1.8 Rustc

rustc 1.51.0 (2fd73fabe 2021-03-23)

2.1.9 Herramienta de diagramas

<http://www.draw.io/>

2.2 Andrés Ramírez Ortega

2.2.1 C

Ubuntu GLIBC 2.31-0ubuntu9.1

2.2.2 Navegador

Google Chrome (64 bits)

2.2.3 Sistema Operativo

Ubuntu(64-bit)

2.2.4 Especificaciones de Computadora

HP Laptop 15-gw0xxx

Procesador AMD Ryzen 3 3250U with Radeon Graphics 2.60 GHz

RAM instalada 12,0 GB (9,94 GB utilizable)

Tipo de sistema Sistema operativo de 64 bits, procesador x64

2.2.5 Bash

GNU bash, version 5.0.17(1)-release (x86_64 - pc - linux - gnu)

2.2.6 Editor de texto

gedit - Version 3.36.2

2.2.7 Gcc

gcc (Ubuntu 9.3.0-17ubuntu1 20.04) 9.3.0

2.2.8 Rustc

rustc 1.51.0 (2fd73fabe 2021-03-23)

2.2.9 Herramienta de diagramas

<http://www.draw.io/>

3 Estructuras de datos usadas y funciones

- Función CONNECT: Esta función esta presente en ambos clientes, es la encargada de conectar al servidor en base al url y el puerto que tomó de la entrada de usuario
- Función attend client: Presente en ambos tipos de servidores, se encarga de crear un hilo (si es prethread) o un proceso (si es preforked) para atender a un client.
- Función cd: El FTPServer implementa el comando cd para buscar en un directorio.
- Función get: El FTPServer implementa el comando get para obtener un archivo.
- Función put: El FTPServer implementa el comando put para poner un archivo.
- Función quit: El FTPServer implementa el comando quit para salir del FTPServer.
- Función receive clients: Presente en ambos servidores, se encarga de manejar los procesos o hilos (dependiendo si es prethread o preforked) para que acepten a los clientes
- STRUCT command: Este struct contiene un id, una dirección y un nombre de archivo, con estos datos se podía comunicar al servidor cual comando esta solicitando y con que datos se va a realizar desde la entrada del usuario.
- STRUCT packet: Este struct contiene entre sus datos una bandera, el id del comando, el largo de los datos, y los datos. Con esta información tanto el cliente como el servidor, se lograban comunicar datos de entrada y salida.
- Struct sockaddr_in: Proveniente de socket.h, este facilitaba la creación de la dirección y el puerto, a donde el socket se va a conectar.

4 Instrucciones para ejecutar el programa

Las instrucciones para ejecutar son las siguientes:

- Se tiene un makefile, con el cual inicializamos.
- Luego debemos dirigirnos a la carpeta bin.
- Luego el cliente debe dirigirse a la carpeta FTPClient.
- Luego se ejecuta `ftpclient -h ¡host-a-conectar! ¡lista-de-comandos-a-ejecutar!`
- Para el server prethread se debe dirigir a la carpeta bin y luego a la carpeta prethread-FTPserver.
- Luego se ejecuta `prethread-FTPserver -n ¡cantidad-hilos! -w ¡ftp-root! -p ¡port!`
- Para el server preforked se debe dirigir a la carpeta bin y luego a la carpeta preforked-FTPserver.
- Luego se ejecuta `preforked-FTPserver -n ¡cantidad-hilos! -w ¡ftp-root! -p ¡port!`

5 Actividades realizadas por estudiante

5.1 Actividades Joel Vega

Lista de Actividades		
Actividad	Día	Cantidad Horas
Reunión inicial Kick-Off	8/4/2020	1 hrs
Investigar sobre pre-forked	10/4/2020	2 hrs
Desarrollar el Kick-Off	11/4/2020	3 hrs
Crear los diagramas UML	11/4/2020	3 hrs
Entrega del Kick-Off	12/4/2020	1 hrs
Códicar y probar ambiente	17/4/2020	2 hrs
Programar Clientes	19/4/2020	4 hrs
Investigar sobre los sockets	21/4/2020	2 hrs
Investigar sobre los Hilos	21/4/2020	4 hrs
Reunión para corregir errores	22/4/2020	2 hrs
Documentar el código	24/4/2020	2 hrs
Pruebas sobre el código	24/4/2020	2 hrs
Reunión para la documentación	26/4/2020	1 hrs
Terminar documentación	27/4/2020	4 hrs
Entregar Tarea	27/4/2020	1 hrs

5.2 Actividades Andrés Ramírez

Lista de Actividades		
Actividad	Día	Cantidad Horas
Reunión inicial Kick-Off	8/4/2020	1 hrs
Investigar sobre pre-thread	9/4/2020	2 hrs
Investigar sobre otros temas importantes	10/4/2020	2 hrs
Desarrollar el Kick-Off	11/4/2020	3 hrs
Entrega del Kick-Off	12/4/2020	1 hrs
Códicar y probar ambiente	17/4/2020	2 hrs
Investigar aspectos sobre los lenguajes a utilizar	15/4/2020	4 hrs
Programar Clientes	19/4/2020	4 hrs
Investigar sobre los sockets	21/4/2020	2 hrs
Investigar sobre los Hilos	21/4/2020	4 hrs
Reunión para corregir errores	22/4/2020	2 hrs
Arreglar errores	25/4/2020	3 hrs
Reunión para la documentación	26/4/2020	1 hrs
Terminar documentación	27/4/2020	4 hrs
Entregar Tarea	27/4/2020	1 hrs

6 Autoevaluación

Se logro realizar el prethread y el preforked, pero no se logro la parte del cliente en Rust y el StressCMD, además se tuvieron problemas con la documentación del lenguaje rust ya que se nos hizo muy difícil trabajar en este lenguaje. La función connect no se implementó ya que se conecta automáticamente al servidor, por lo cual no entendimos la función. Los commits se pueden observar en este repositorio: <https://github.com/JoelVega10/FTP-Server-Sistemas-Operativos>

6.1 Joel Vega

Autoevaluación	
Aprendizaje	Puntaje
Aprendizaje de pthreads	4
Aprendizaje de forks	4
Aprendizaje de comunicacion entre procesos	5
Aprendizaje de sockets	5

6.2 Andrés Ramírez

Autoevaluación	
Aprendizaje	Puntaje
Aprendizaje de pthreads	3
Aprendizaje de forks	3
Aprendizaje de comunicacion entre procesos	5
Aprendizaje de sockets	5

7 Lecciones Aprendidas

7.1 Joel Vega

En el presente proyecto me fue posible lograr reforzar los siguientes aspectos:

- La creación de sockets y su funcionamiento en el lenguaje C.
- El funcionamiento de los comandos del servidor FTP y su interacción con el cliente.
- La librería pthread y como se manejan los hilos con esta, me ayudo a entender como trabajan los hilos.
- La creación de diagramas para un trabajo con este es de suma importancia ya que nos recuerda siempre cual camino seguir.
- El manejo de solicitudes de los clientes hacia el servidor, y la respuesta que da el servidor al estar los espacios disponibles completos.

7.2 Andrés Ramírez

- La función de un web server, los clientes y la importancia de que un web server funcione correctamente.
- Pensar e investigar antes de codificar, buscar información clara que ayude a entender mejor el problema y sirva como una guía.
- El uso de hilos queda más claro, mediante la investigación, las lecturas de la clase y la explicación del profesor queda todo más claro.
- La importancia de las conexiones entre clientes y un servidor.
- La ventaja de hacer un Kick-Off para no dejar todo hasta el final y tener una idea de los pasos a seguir para el proyecto.

References

- [1] Kevin Moraga.(2021).Documentación Tarea 3
- [2] How to Set up FTP Server - Windows FTP— Serv-U. (s. f.). FTP.
<https://www.serv-u.com/ftp-server-windows/server-setup>
- [3] GeeksforGeeks. (2018, 10 octubre). Multithreading in C.
<https://www.geeksforgeeks.org/multithreading-c-2/>
- [4] Ippolito, G. (s. f.). Linux Tutorial: POSIX Threads. Posix.
<https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>
- [5] UNIX Network Programming: The sockets networking API. (s. f.). Google Books.