

Tecnológico de Costa Rica

Escuela de Computación

Curso: Principios de Sistemas Operativos

Tercer Proyecto

Un problema que se presenta con el servicio FTP (file transfer protocol), utilizado tradicionalmente para la transferencia de archivos entre máquinas, es la tarea a veces complicada de instalar un programa servidor en la máquina a la que se desean transferir/recuperar archivos (modelo cliente-servidor), ya que se requieren privilegios de administrador.

Un enfoque más sencillo para implementar un servicio FTP sería simplemente que dos programas clientes se comuniquen entre sí y transfieran archivos entre ellos sin necesidad de instalar un servidor (modelo peer-to-peer).

En este proyecto se pretende desarrollar un cliente FTP que funcione de ese modo, es decir, que permita transferir y recibir archivos entre programas clientes que se encuentren instalados en máquinas remotas.

Autenticación

El programa a desarrollar, que llamaremos *bftp* (basic ftp), realizará la autenticación de usuarios al inicio de la ejecución del programa:

```
bftp://<usuario>:<contraseña>@<dirección-ip>/<url-ruta>
```

Donde *usuario* es el nombre de usuario, *dirección-ip* es el IP de la máquina remota, *contraseña* es la contraseña de acceso, y *url-ruta* es el directorio donde se iniciará la sesión.

Cada máquina en donde se ejecute el programa contendrá un archivo llamado *.passwords* que almacenará el nombre de usuario y su contraseña. De este modo al iniciar una conexión se debe primero verificar las credenciales del usuario antes de permitirle ejecutar comandos.

Optativo: No es conveniente que los password queden expuestos en dicho archivo, por lo que se deben codificar utilizando una librería tipo “BASE64” (10 % adicional)

Comandos ftp

El programa *bftp* reconocerá un subconjunto de instrucciones de FTP que permitirán la conexión y transferencia de archivos:

open <dirección-ip>: establece una conexión remota
close: cierra la conexión actual
quit: termina el programa
cd <directorio>: cambia de directorio remoto
get <archivo>: recupera un archivo remoto
lcd <directorio>: cambia de directorio local
ls: lista los archivos del directorio remoto
put <archivo>: envía un archivo a la máquina remota
pwd: muestra el directorio activo remoto

La transferencia de archivos realizará una copia del archivo en cuestión desde la máquina remota hasta la máquina local (o viceversa) almacenando éste con el mismo nombre que se encontraba en la otra máquina.

Comandos adicionales (optativo - 25 %)

El programa también podrá ejecutar un subconjunto adicional de instrucciones:

delete <archivo>: borra el archivo en la máquina remota
mdelete <patrón>: borra los archivos que cumplen el patrón
mget <patrón>: recupera los archivos que cumplen el patrón
mput <patrón>: envía los archivos que cumplen el patrón
rename <archivo> <nombre>: cambia el nombre del archivo
rmdir <directorio>: borra el directorio remoto
mkdir <directorio>: crea un directorio en la máquina remota

El *patrón* consiste en una hilera de texto junto con el carácter comodín *, por ejemplo:

```
tareas*.txt
documentos.*
*.txt
```

Modo de operación

El programa *bftp* se debe ejecutar en al menos dos máquinas al mismo tiempo, y quedarán esperando conexiones en el puerto 8889. Posteriormente, permitirá que el usuario ingrese comandos en la terminal y los ejecutará. Note que un usuario solo puede establecer una conexión con una máquina a un tiempo, pero múltiples máquinas remotas pueden establecer conexión con la máquina de dicho usuario al mismo tiempo.

Internamente cada programa se deberá comportar con un servidor y un cliente al mismo tiempo. Por eso se debe utilizar un mecanismo (tareas) que le permita al programa estar haciendo varias cosas al mismo tiempo: recibir comandos del usuario, atender todas las conexiones entrantes y ejecutar sus comandos.

El programa debe utilizar tanto *sockets* clientes como un *socket* servidor para realizar las conexiones adecuadamente.

Documentación

Se deberá generar una documentación formal, en formato pdf, en donde se describan las diferentes etapas del desarrollo del proyecto, las decisiones de diseño que se tomaron, los mecanismos de programación utilizados, y los resultados de las diferentes pruebas al programa.

Dicha documentación deberá incluir al menos las siguientes secciones:

- Introducción
- Descripción del problema (este enunciado)
- Definición de estructuras de datos
- Descripción detallada y explicación de los componentes principales del servidor/cliente:
 - Manejo de sockets
 - Manejo de solicitudes
 - Rutina de transferencia
 - Manejo de autenticación
 - Manejo de hilos
- Descripción del protocolos y formatos
- Análisis de resultados de pruebas
- Conclusiones sobre rendimiento y correctitud

Forma de evaluación

- Programa *bftp* (80 %)
 - Operaciones de transferencia: put, get (20%)
 - Operaciones sobre directorios: cd, pwd, lcd, ls (20%)
 - Procedimiento de autenticación (20%)
 - Manejo de múltiples conexiones simultáneas (20%)
- Documentación (20%)
 - Descripción de estructuras, componentes, etc. (10%)
 - Análisis de resultados y conclusiones (10%)

Consideraciones generales

- El proyecto debe ser ejecutado en grupos de a lo más dos personas. Se puede desarrollar en forma individual pero bajo ninguna circunstancia se aceptarán grupos de tres o más personas.
- El proyecto debe ser realizado en lenguaje C y utilizando la librería *pthread*.

- No se permite la copia de código entre grupos de estudiantes, tampoco es permitido utilizar clases o librerías adicionales (desarrolladas por terceros) para simplificar el manejo de socket, autenticación, o copia de archivos. Sí es permitido utilizar una librería externa para encriptar los passwords.
- Se deberá crear un archivo zip que incluya: el código fuente de todos los programas desarrollados, binarios generados, y documentación en formato pdf.
- Puede buscar en Internet mayor información sobre el uso de las diferentes opciones del comando *ftp*.