

Protocolos de Comunicación

Trabajo práctico especial – 2024 1C

Objetivo

El objetivo del presente trabajo es comprender el desarrollo de protocolos de aplicación para TCP/IP, orientados y no orientados a conexión.

El mismo deberá ser realizado en grupos de hasta 3 alumnos

Especificaciones

Desarrollar programas cliente/servidor que permitan compartir archivos en forma binaria a través de Internet. Para ello el grupo deberá desarrollar protocolos de aplicación originales e implementar aplicaciones cliente y servidor -usando sockets TCP/IP- basadas en las especificaciones del mismo.

Cada usuario tendrá una serie de archivos que desea compartir. Todos los archivos tendrán que estar en una misma carpeta de su disco local, que llamaremos *repositorio*.

A fin de compartir los archivos, deberá existir un servidor que tenga información sobre los usuarios y qué archivos está compartiendo. Este servidor será llamado *tracker*. El intercambio de archivos se realizará únicamente entre usuarios registrados.

Cuando una persona desea descargar un archivo, la aplicación le solicitará al tracker una lista de los usuarios que posean ese archivo en forma completa. Una vez obtenida la lista, podrá comenzar a pedir a algunos o todos esos usuarios la descarga del archivo, que será copiado en el repositorio.

De esto se desprende que hay dos protocolos a desarrollar: uno para la comunicación con el *tracker* (aplicación A) y otro para la interconexión con los usuarios (aplicación B).

Aplicación A

La aplicación A deberá contemplar como mínimo los siguientes puntos:

- El protocolo de transporte a utilizar entre el *tracker* y sus clientes será UDP
- El servidor mantendrá una lista actualizada de los archivos compartidos. Por cada uno de ellos mantendrá la siguiente información:
 - Nombre del archivo
 - Tamaño en bytes del archivo
 - MD5 (identifica en forma unívoca al archivo). Tener en cuenta que el nombre de un archivo puede variar de un host a otro, es el MD5 el que lo identifica.
 - Lista de usuarios que lo tienen en forma completa (nombre de cada uno). Estos usuarios serán llamados *seeders*.
 - Lista de los usuarios que lo están bajando, que serán llamados *leechers*.¹
- Sólo podrán conectarse usuarios que estén previamente registrados, por lo que el tracker

¹ Sólo para los que entreguen en segunda fecha, opcional para los que entreguen en primera fecha

- tendrá que mantener una base o archivo simple con el nombre y clave de cada usuario.
- Por cada usuario deberá mantener actualizado su número de IP y el o los puertos que utiliza para su aplicación servidor (el servidor de la aplicación B). ²
- Cada *seeder* deberá enviar información actualizada sobre los puertos en los cuales escucha.
- Cuando un *seeder* detecte cambios en el repositorio, se lo informará al *tracker*.
- Cuando un usuario nuevo se registre deberá enviar su nombre, clave e IP actual. El servidor le informará en caso de que el usuario ya exista. Si no existe, le enviará un paquete pidiendo que confirme su registración (para evitar que se envíen pedidos indiscriminadamente).
- El nombre y la clave del usuario podrán ser enviados como texto plano.

Aplicación B

Una vez que un usuario obtuvo del *tracker* la lista de *seeders*, podrá solicitarle a los mismos que le envíen porciones de ese archivo, hasta completar la transferencia total.

- El protocolo de transporte a utilizar entre los *seeders* y *leechers* será TCP
- La aplicación debe transferir los archivos en forma confiable entre los *seeders* y *leechers*.
- Deberá solicitar distintas porciones del archivo en forma concurrente a distintos *seeders*.
- Tendrá que instrumentar los mecanismos y controles necesarios a fin de lograr obtener el archivo completo en el menor tiempo posible.
- Los archivos se transferirán sin encriptar.
- Un *seeder* deberá atender en forma concurrente a varios *leechers*. La aplicación no debe imponer a priori un límite de comunicaciones concurrentes.
- El *seeder* deberá almacenar en un archivo de logs tanto las conexiones que se realizaron como los bytes transferidos a cada usuario. El formato del log deberá ser estándar, para poder ser leído por herramientas analizadoras de log. Sugerencia: investigar syslog.³
- Podrá transferir todo tipo de archivos y de cualquier tamaño.
- Sólo se le transferirá información a *leechers* registrados, verificando contra el *tracker* si el usuario e IP son correctos. No se transfieren claves entre *seeders* y *leechers*.

A su vez podrán contemplar los siguientes requerimientos opcionales

- Solicitar al gateway la "apertura" de puerto/s públicos (DNAT)
-

Interacción con el usuario.

Se deberán contemplar los siguientes puntos:

- Al iniciar un servicio, se le podrán pasar por línea de comandos los siguientes parámetros.
En caso de no pasarle parámetros, tomará los datos por defecto
Puertos en los que escucha. (por defecto sólo en el puerto 15555)
IPs en los cuales escucha (por defecto en todos los IPs del host). (por defecto en todos)
Cantidad máxima de conexiones simultáneas a aceptar. (por defecto no hay máximo)
- Se debe instrumentar un mecanismo para que el usuario pueda ver, tanto para el *seeder* como para el *leecher*, las conexiones que tiene abiertas y sobre qué archivo está operando cada una.

² Para los que entreguen en primera fecha, habrá un único puerto por cada *seeder*, para los que entreguen en segunda fecha deberán contemplar la posibilidad de varios puertos

³ Sólo para los que entreguen en segunda fecha

- El usuario debe poder realizar como mínimo las siguientes operaciones
 - Indicar su usuario y clave (si el usuario no existe, poder crearlo)
 - Pedirle al tracker la lista de archivos
 - Cuantos seeders hay por archivo
 - Cambiar su clave
 - Bajar archivos
 - Opcionalmente** se pueden agregar las siguientes operaciones
 - Hacer una pausa en el download
 - Resumir un download pausado

Dudas sobre el TPE

Es normal que surjan dudas acerca de cómo interpretar ciertos casos o si se deben contemplar ciertos errores o no. O que una consigna genere más de una posible solución, por lo que es importante que analicen bien el enunciado, y ante cualquier duda pregunten. Las dudas deben volcarse en Github: <https://github.com/marcelogarberoglio/PDC-ITBA/issues>

En caso de realizar alguna aclaración o consideración sobre el enunciado, la misma deberá ser tomada en cuenta por todos los grupos, no solo para el grupo que haya hecho la pregunta.

Conformación de grupos

Los grupos (de hasta 3 integrantes) deberán ser informados también en <https://github.com/marcelogarberoglio/PDC-ITBA/issues>.

Los grupos que deseen rendir en primera fecha deberán informar la conformación del mismo antes del 3/7 a las 23:59

Los que rindan en segunda fecha deberán hacerlo antes del 9/7 a las 23:59

En todos los casos deberán estar inscriptos en SGA

Uso de Git

Es obligatorio el uso de un repositorio Git para la resolución de este final. Deberán crear un repositorio en GitHub donde todos los integrantes del grupo colaboren con las modificaciones del código provisto. **No se aceptarán entregas que utilicen un repositorio git con un único commit que consista en la totalidad del código a entregar.**

Los distintos *commits* deben permitir ver la evolución del trabajo, tanto grupal como **individual**, por lo que el nombre de cada usuario deberá tener relación con el nombre de cada miembro del grupo.

Muy importante: los repositorios creados deben ser privados.

Material a entregar

Cada grupo deberá subir al Campus ITBA **únicamente un archivo compactado**.

El nombre del archivo compactado **tiene** que ser **TPE_XX** seguido de la extensión (rar, zip, etc.) donde **XX** es el número del grupo, por ejemplo TPE_02.rar, TPE_10.zip

El archivo debe contener como mínimo los siguientes archivos:

- Programas fuente para cliente y servidor, desarrollados en C y para Linux. Para cada función se debe detallar si:
 - Fue desarrollada en su totalidad por el grupo.
 - Fue extraída de un libro o sitio, indicando la fuente.

- Fue desarrollada en base a un libro o sitio, indicando la fuente y cuáles fueron las modificaciones realizadas a la versión original.
- Carpeta con archivos de prueba
- Archivo makefile
- Programas o scripts de testeo
- "Man page" de la aplicación.
- Un informe que contenga, en este orden
 - Carátula
 - Índice
 - Descripción detallada de los protocolos desarrollados (formato de los paquetes, casos de error, etc.)
 - Problemas encontrados durante el diseño y la implementación
 - Limitaciones de la aplicación
 - Posibles extensiones
 - Conclusiones
- El directorio oculto .git/ donde se detallan todas las modificaciones realizadas

No se aceptarán trabajos en forma incompleta.

En caso de detectar plagio el trabajo estará desaprobado y sin posibilidad de recuperación

Fecha de entrega:

- Para los que rindan en primera fecha: 06/07/2024 antes de las 23:59 hs
- Para los que rindan en segunda fecha: 11/07/2024 antes de las 23:59 hs

Una vez realizadas las pruebas mínimas por parte de la Cátedra se programará un coloquio (presencial u online) con cada grupo para realizar la defensa del mismo y recibir una devolución. También se podrá pedir una demo para algún caso particular.

Aquellos que entreguen en primera fecha y no aprueben se los considerará como ausente y podrán entregar en segunda fecha (con los requisitos para la segunda fecha)