Redes de Computadores – Laboratorio evaluado 1: Transferencia de archivos por TCP

13 de abril de 2022

Algo que constantemente tenemos que hacer en nuestro día a día trabajando con computadores es transferir archivos de una parte a otra, ya sea desde entre carpetas, de un disco duro a un *pendrive*, o inclusive por medio de la red, ya sea local o a través de internet. Considerando lo fundamental que es esto, en esta tarea vamos a enfocarnos en implementar nuestro propio programa de transferencia de archivos, donde ustedes tendrán que crear un programa que permita traspasar archivos entre computadores por medio de la red.

Para hacer esto, tendrán que implementar este programa usando *sockets* TCP, pudiendo usar el lenguaje de su preferencia, mientras trabajen directamente con bibliotecas de *sockets* (a bajo nivel) y no ocupen bibliotecas que ya implementen protocolos de transferencia de archivos.

1. Requerimientos

Los requerimientos a cumplir son los siguientes:

- El programa puede realizarse con cualquier lenguaje de programación.
- El programa debe trabajar con conexiones TCP de forma manual, usando la API del sistema operativo para *sockets* o una biblioteca de terceros.
- El programa debe poder ejecutarse en modo receptor o transmisor (pueden ser ejecutables diferentes o uno solo, indicando cuál modo usar mediante argumentos).
  - El receptor corresponde al programa que va a estar a la escucha de conexiones, esperando recibir algún archivo y guardándolo en disco, mientras que el transmisor corresponde al programa que desea enviar un archivo en disco a un receptor.
- El protocolo debe permitir la transferencia de archivos (binarios o de texto plano) de cualquier tamaño, así como también indicar el nombre y tamaño del archivo a transferir, para que pueda ser llamado de la misma forma en su destino.
- Ambas partes deben indicar el progreso de la transferencia a medida que el archivo se envía.
- El protocolo debe permitir la transferencia cifrada de archivos mediante algún protocolo de cifrado simétrico (por contraseña o archivo de llave).

2. Evaluación

La fecha de entrega para este laboratorio es el 11 de mayo a las 23:59. La no entrega de la actividad sin justificación equivale a NCR. Si, por alguna razón, no pudieran entregar a tiempo su tarea, les ruego que me lo comuniquen con tiempo.

El laboratorio se evaluará individualmente o de a pares bajo los siguientes requerimientos, mediante una escala de 6 puntos:

- **Funcionalidad** (2 puntos): El programa debe permitir transferir archivos de forma exitosa y confiable, señalizando errores en caso de haberlos.
  - **Pista**: Una buena forma de comprobar que un archivo se transfirió bien es comparando el *hash* en la copia enviada y la recibida.
- **Transferencias de tamaño arbitrario** (1 punto): El programa debe poder transmitir archivos de cualquier tamaño.
- **Información sobre la transferencia** (1 punto): Tanto receptor como transmisor deben poder indicar el progreso de la transferencia. Además, el receptor debe poder indicar por pantalla el nombre y tamaño del archivo que está recibiendo.
- **Cifrado** (1 punto): La transferencia debe poder realizarse mediante algún tipo de cifrado simétrico (como Blowfish o AES). Pueden ocupar bibliotecas de terceros para implementar esta parte.
- **Demostración** (1 punto): El uso del programa debe ser demostrado, mostrando que es posible la transferencia de archivos, tanto de forma cifrada como sin cifrar.

De forma adicional, cada uno de estos puntos extra otorgarán 0.5 puntos adicionales sobre la calificación anterior, hasta completar 6 puntos:

- **Limpieza de código**: El código del programa tiene una sintaxis y estructura clara, consistente y limpia.
  - **Pista**: Pueden utilizar programas de formato automático como `clang-format` (C/C++), `autopep8` (Python) o `rustfmt` (Rust), dependiendo del lenguaje que quieran utilizar. Esta clase de utilidades pueden ser integradas en editores de código como Visual Studio Code, (Neo)vim o Emacs.
- **Compilación automatizada**: El proyecto utiliza un *toolchain* de compilación, que se encarga de ejecutar los pasos de compilación de forma automática, sin tener que llamar manualmente al compilador para generar los ejecutables.
  - **Pista**: Dependiendo del lenguaje, pueden utilizar herramientas como CMake, `autotools`[1] o Cargo.
- **Cifrado asimétrico**: El programa permite utilizar algún algoritmo de cifrado asimétrico para transferir archivos, como RSA.

El código debe ser entregado en Canvas como un archivo comprimido (de cualquier tipo) o un *link* a un repositorio público de código, que contenga un `README` con instrucciones de cómo compilar y ejecutar el proyecto. También se requerirá un video corto de demostración (no mayor a 5 minutos, idealmente) mostrando el funcionamiento del programa, para el cual deben entregar un *link* (consejo: para hacer más fácil la grabación, pueden ejecutar cliente y servidor en `localhost`; es igual de válido).

[1] Sólo se los recomiendo si no tienen amor propio y desprecian su salud mental.