

Proyecto # 1

Servidor de Streaming

Resumen

El uso de los principales servicios que ofrecen los Sistemas Operativos son de vital importancia para el desarrollo de prácticamente todas las aplicaciones disponibles hoy en día mediante el acceso a redes como *Internet*.

Este trabajo busca generar una experiencia básica en el uso de servidores de software y el acceso a servicios de datos que éstos pueden ofrecer a múltiples clientes de forma concurrente.

1. Descripción

Este proyecto consiste en la programación de un servidor para la distribución en línea o “streaming” de archivos multimedia (audio, vídeo, imágenes, pdf, texto, etc) hacia múltiples clientes conectados de forma concurrente al sistema.

Múltiples usuarios clientes remotos, en la misma red o VPN, podrán reproducir el contenido de estos archivos mediante streaming.

2. Servidor

El servidor deberá estar programado completamente sobre un sistema operativo Linux y en el lenguaje C, usando el compilador *gcc* únicamente. Deberá usar sockets para la comunicación de los clientes remotos hacia un puerto y una IP conocidos el cual deberá establecer o bien configurar una resolución de nombres vía DNS o similar.

El servidor deberá atender las solicitudes concurrentemente. Al recibir una conexión deberá crear un proceso liviano tipo *thread* para que atienda a dicho cliente. Para el manejo de las solicitudes el servidor tendrá control de usuarios, es decir cada solicitud estará asociada a una cuenta de usuario. Deberá facilitar algún mecanismo para el registro, autenticación e inicio de sesión de cada usuario desde múltiples clientes (terminales) ante el servidor.

Cada *thread* deberá enviar el contenido del archivo al cliente, en paquetes de un tamaño que deberá definir. El cliente recibirá los paquetes para ir reproduciendo los datos conforme los recibe. Debe definir un buffer de tamaño delimitado para almacenamiento temporal de información que se vaya a reproducir. Debe documentar y explicar el uso de

este buffer y todas las estructuras de datos requeridas de forma gráfica (con diagramas) y el código correspondiente en su documentación.

El programa servidor deberá informar mediante una bitácora en pantalla las acciones que ocurren en el sistema tales como: un usuario conectado, un cliente que reproduce un archivo, identificando dicho cliente con su nombre de inicio de sesión y el archivo que solicita, además de la cantidad de bytes transmitidos y la velocidad de envío. Dicha información también debe quedar almacenada en un archivo de texto tipo Log en el servidor por cada sesión del servidor, los logs anteriores no deben ser borrados, use la fecha y hora como parte del nombre de los archivos para llevar un control histórico de los mismos. Para cada archivo transmitido, el servidor deberá incluir en la bitácora la firma MD5sum del mismo, para comprobación posterior de seguridad de datos transmitidos.

El servidor leerá los archivos multimedia desde una carpeta en la máquina donde ejecuta y debe ser fácil poder agregar y registrar nuevos archivos en el sistema. Es decir, los elementos multimedia disponible no deberían estar predefinidos en el código. Al iniciar una nueva sesión del servidor, la lista actualizada de los archivos debe mostrarse al usuario.

Si el servidor no está en ejecución o al cerrarse, se deberán desconectar los clientes, indicándoles el cierre del servicio.

Deberá definir y documentar el protocolo y mensajes que vaya a utilizar entre el Servidor y los Clientes para la comunicación y sincronización de actividades.

Deberá tener en cuenta el manejo de concurrencia y acceso a recursos compartidos por parte de los clientes, 1 o N clientes deben poder tener acceso al mismo archivo multimedia para ser transmitido y reproducido en cualquier momento.

Haga uso de la biblioteca Pthread, semáforos, y sockets de forma apropiada. Documente las formas en que las emplea.

La máquina que usará para ejecutar el programa Servidor debe correr completamente y únicamente desde los servidores de la Escuela de Computación, sobre la red de la Escuela, en computadoras Linux provistas para este fin. Los accesos a estos equipos será remoto, usando el servicio gratuito VPN de la Escuela y mediante protocolos SSH, HTTP u otros que requiera y vía sockets para la comunicación entre el Servidor y sus clientes, que podrán estar en máquinas remotas (lógica y físicamente) fuera de la Escuela.

Todos los parámetros de IPs, números de puertos a usar, etc. deben ser parametrizables al momento de ejecución y no estar pre-establecidos en el código o que sea necesario re-compile el código para hacer cambios.

3. Clientes

Va a requerir programar 2 tipos de clientes: uno en modo texto (terminal) y otro que permita reproducir videos e imágenes principalmente y adicionalmente otros tipos como audio, pdfs, textos, etc.

Para iniciar sesión se solicitará un nombre de usuario al ingresar. Cuando un usuario se conecta al sistema de Streaming podrá consultar al Servidor los archivos multimedia disponibles para ser reproducidos, seleccionar cualquiera para que el mismo se empiece a reproducir vía streaming. El usuario podría pausar o detener la reproducción. Cuando una reproducción se detiene o cancela, el usuario podrá seleccionar un nuevo archivo. El usuario podrá saltar hacia atrás o hacia adelante al siguiente archivo por reproducir en la lista de archivos. No es necesario que el usuario puede devolverse en la reproducción

o adelantar el audio o video por ejemplo, sólo pausar y detener. Se puede configurar un modo aleatorio de reproducción, para que los archivos se reproduzcan indefinidamente hasta que el usuario lo pause o detenga.

En caso que el usuario inicie una nueva sesión desde otro cliente (terminal, navegador, app, etc), el streaming debe continuar de forma automática a la nueva conexión, trasladándose la reproducción a la nueva sesión del usuario. Tome en cuenta que esta nueva sesión podría ser en una terminal textual en máquina o dispositivo. En el caso de audio, la reproducción deberá continuar en cualquiera de los 2 tipos de clientes a realizar. En el caso de imágenes, PDFs o Videos, el cliente textual, deberá entonces descargar el archivo, para que pueda ser visualizado de forma local. En el caso del segundo cliente, el video debe continuar por donde iba la reproducción, los otros tipos de archivos se mostrarán en el nuevo cliente conectado.

3.1. Cliente para modo texto

El cliente en modo texto, deberá poder solicitar cualquier tipo de archivo. Acá, los archivos multimedia serán descargados para reproducción local, sin embargo los archivos de audio si podrán reproducirse, pausarse y detenerse. Los archivos de texto, pueden ser mostrados en pantalla o guardados en archivo, según escoja el usuario. Los demás archivos multimedia serán guardados localmente. Para cada archivo recibido, el cliente hará una verificación mediante MD5sum mostrando el comando ejecutado y la firma en pantalla generada al recibir cualquier archivo. Como se indicó antes, si el usuario se conecta desde otro cliente, la reproducción de audio debe continuar en el punto donde estaba en la nueva sesión del cliente y cerrar la sesión anterior. Siempre que se cierre una sesión de usuario, por salir del programa o por haber iniciado sesión en un cliente nuevo, deberán mostrarse las estadísticas locales de esa sesión. No es necesario que el cliente migre las estadísticas. Se considerará como 10 % extra el traslado de estadísticas entre sesiones, en cuyo caso deberá indicar la cantidad de sesiones y las IPs y puertos desde donde se realizaron esas conexiones.

3.2. Cliente para modo gráfico

Este cliente puede ser desarrollado en cualquier lenguaje de programación que considere pertinente, y en la plataforma que prefiera: desktop, móvil, web, etc.

La creatividad de la interfaz y facilidad de uso del programa cliente será considerado en la calificación de este cliente.

4. Generalidades

- Este trabajo se debe realizar en parejas de máximo DOS personas o individual.
- El cliente gráfico no deberá almacenar ninguna copia de los archivos multimedia, ni grandes bloques de almacenamiento temporal en memoria, etc.
- En general el sistema deberá soportar la reproducción continua de archivos extensos, es decir de duración larga, muchos minutos o varios gigas de tamaño.

- Podrá emplear más de un socket por cliente si así lo requiere. Podrá hacer uso de pipes o memoria compartida entre los procesos hilos y el padre si requiere información adicional o de control. Podrá usar semáforos según lo requiera para el control de las operaciones.
- Cualquier código de libre acceso y abierto, de bibliotecas o no que se vaya a utilizar debe estar debidamente documentado y justificado su uso. Caso contrario sería considerado como plagio. No puede usar bibliotecas que implementen la funcionalidad principal del Proyecto.
- Podrá hacer uso de bibliotecas open-source para la reproducción del audio y vídeo cuando lo requiera. Deberá documentar en su bibliografía dichas fuentes, así como hacer las referencias en su documentación y dar los créditos correspondientes en el código.
- Debe ser posible tener múltiples clientes conectados de forma concurrente al servidor, reproduciendo streaming.
- Tanto el servidor como el cliente deben mostrar estadísticas de uso: tiempo de ejecución, cantidad de memoria principal usada, cantidad de bytes enviados y recibidos, cantidad de conexiones hechas, cantidad de clientes diferentes conectados, cantidad de archivos transmitidos, velocidad de transmisión, etc.
- Use la siguiente invitación Git para crear su repositorio de trabajo con su pareja de trabajo: <https://classroom.github.com/g/zCAvNqS2>
- Para facilitar el trabajo, configure un archivo Makefile, para la construcción de su proyecto, servidor y clientes. Documente su Makefile y el uso del mismo.
- Al finalizar el proyecto, para su entrega, deberá preparar una carpeta con el nombre de ambos participantes, donde deberá incluir sus archivos fuentes .c y archivos fuente requeridos. No incluya acá archivos multimedia o similares. Tampoco incluya acá archivos de bibliotecas de terceros, pues esto ya estará referenciado en su documentación y comentarios de uso en su código. No incluya archivos ejecutables. Prepare un archivo comprimido .tgz de esta carpeta. NO incluya acá el archivo PDF de su documentación.

5. Acceso VPN y Máquinas Linux

Para el acceso VPN a la Escuela deberá hacer uso de su propia cuenta de acceso de la Escuela en el dominio @ic-itcr.ac.cr para conectarse a los equipos clientes en la Escuela y acceder el servidor de forma remota. Tenga en cuenta que las cuentas y la información en éstas está disponible de forma centralizada en un servidor LDAP de la Escuela, y que al conectarnos de forma local o remota desde cualquier equipo siempre tendrá acceso a la información en su cuenta (como una nube) en los servidores de la Escuela. Sin embargo las 3 máquinas provistas no están conectadas a los servicios de usuarios LDAP, son máquinas independientes en ese sentido. La administración de las cuentas de usuario es local, por lo que, para poder trabajar su proyecto, cada uno deberá contar con una cuenta de usuario local usando su mismo nombre de usuario de la red de la Escuela.

Para esto deberá facilitar su nombre de usuario en la red de la Escuela y nombre completo, para poder crearle una cuenta de usuario en dichas máquinas con una contraseña genérica. Por seguridad, deberá cambiar su contraseña al ingresar por primera vez en cada máquina.

En caso de requerir herramientas de software especiales que requieran ser instaladas en las máquinas brindadas para programar, deberá comunicarlo con tiempo al profesor vía Telegram o Correo Electrónico, para coordinar la instalación.

Es posible tener acceso remoto a los servicios en red de la Escuela vía VPN y vía SSH, consulte la información disponible en el material digital del curso en la carpeta *AccesoVPN*, descargando el archivo *VPN.zip*. En el caso de máquinas Linux o Mac, deberá editar el archivo de configuración *client.ovpn* y ajustar las rutas de los directorios donde se ubicarán los archivos *ca.crt* y *ta.key* requeridos para la conexión.

Las máquinas a usar son:

- esteban-1 : 172.19.127.76
- esteban-2 : 172.19.127.71
- esteban-3 : 172.19.127.72

6. Documentación

Realice un documento formal en L^AT_EX, y genere un PDF para presentar los resultados, este debe incluir:

- Portada con un abstract en inglés, ambos en 1 sola página
- Introducción, Marco Teórico de Todas las Herramientas empleadas como parte de su investigación para su trabajo
- Desarrollo, detallando el diseño de su solución, estructuras de datos empleadas, protocolos usados y el diseñado para su proceso de comunicación. Deberá documentar de forma gráfica (con diagramas) y en código estas estructuras y protocolo creado y sus usos.
- Pruebas de la ejecución del Proyecto con diversos clientes, cambios de archivos multimedia disponibles, etc.
- Conclusiones, comentarios y observaciones
- Bibliografía en formato IEEE. Referencias bibliográficas empleadas, la bibliografía correctamente empleada vale 10 puntos del trabajo. Se deben usar y anotar correctamente las fuentes bibliográficas en el texto del documento y los sitios de internet consultados para la actividad (referencias).
- Presente la información de forma esquematizada y sintetizada para mostrar la información importante y relevante de manera concisa.
- Trabaje bien la compresión de las imágenes en el documento a presentar para reducir el tamaño del archivo a enviar, recorte las imágenes para mostrar solo los datos importantes y que sean legibles, no es necesario mostrar todo el escritorio en las capturas.

7. Indicaciones de Entrega

Generar una versión del reporte en PDF y súbalo como respuesta a la Tarea asignada en Google Classroom, junto con un archivo .tgz con el código de su proyecto, como se indicó previamente, deberá entregar 2 archivos.

- Identifique los archivos a entregar correctamente con el formato:

[S0]_Proy1_Stream_NombreApellido1_NombreApellido2

- Un miembro deberá subir el archivo PDF al Classroom y el otro miembro deberá subir el archivo .tgz. Al subir sus archivos, incluya como comentario en Classroom la firma MD5sum del mismo.
- Las copias o intento de copia se penalizaran con 0 y un reporte al expediente y a la Escuela.
- Se calificará el uso correcto de fuentes bibliográficas y el uso de referencias a las mismas en el texto del documento.
- Las entregas tardías se contabilizan con 15 puntos menos por cada día de atraso.
- Las instrucciones que no se cumplan se penalizarán con 5 puntos menos cada falta.
- Fecha máxima de entrega: Viernes 3 de Julio del 2020, antes de las 11:59pm.

8. Revisión

- La revisión será de forma remota, con un demo en vivo, donde se proveerán diversos archivos de varios formatos y tamaños para ser transmitidos con sus clientes y servidor. Adicionalmente, deberá preparar un escenario de pruebas para poder evaluar el desempeño del servidor con al menos 3 clientes remotos diferentes para pruebas simultáneas. El profesor proveerá archivos y clientes adicionales para las pruebas.
- Posteriormente se definirán horarios de revisión a convenir.

Cualquier consulta puede ser enviada al correo earias@ic-itcr.ac.cr, durante las clases o Telegram [@estebanarias](https://www.instagram.com/estebanarias)

Prof. Esteban Arias M.