I.T.C.R.

Departamento de Computación
Prof: Ing. Erika Marín Schumann.

Sistemas Operativos

Servidor Imágenes
I Semestre del 2025

Objetivos del proyecto

- Aprender a desarrollar un esquema de cliente-servidor.
- Conocer la comunicación por medio de sockets.
- Desarrollar un programa que maneje procesos.
- Desarrollar un programa donde se utilicen threads.
- Desarrollar un programa en C para ser corrido en ambiente Linux.

Definición general

Este primer proyecto tiene como finalidad desarrollar diferentes esquemas de servidores. Los servidores a desarrollar responderán a peticiones de archivos que realizaran los clientes. Para este proyecto el cliente podrá ser cualquier Browser o el cliente que se desarrollará.

Obviamente, el estudiante debe investigar sobre el protocolo de comunicación que tienen los browsers y la manera en que estos hacen las peticiones, para que los servidores que desarrollen puedan responder satisfactoriamente a las peticiones hechas por los clientes.

El cliente que desarrollarán utilizará el mismo protocolo del Browser. La diferencia entre ambos clientes es que el Browser al hacer la petición al servidor muestra el archivo recibido en cambio el cliente salva el archivo en una carpeta del disco de la máquina cliente.

Con el proyecto, el estudiante deberá desarrollar diferentes tipos de servidores que satisfagan de manera distinta las solicitudes de los clientes. Con estos diferentes modelos, se adquirirá conocimientos en lo relacionado a procesos e hilos.

Como se mencionó una forma de realizar las solicitudes es por medio del browser. Por lo que en la dirección URL se digitará algo como: http://<ipDelServidor>:<puerto>/<nombredelarchivo>. Lo mismo con el cliente no browser.

En el mismo lugar donde se colocan los servidores, habrá una carpeta donde se tengan todos los archivos que se le puedan enviar al cliente.

Es claro, que el cliente podría solicitar un archivo que el servidor (cualquiera de los 3) no conozca, por lo que se debe enviar un mensaje de error para que el browser lo despliegue y el usuario sepa porque razón no se le despliega el archivo solicitado.

Las solicitudes de uno o más clientes serán atendidas dependiendo del esquema de cada uno de los servidores. Basta con la cola de atenciones que tienen los sockets, sin embargo, si el grupo así lo desea puede implementar una cola de solicitudes.

Los servidores deben ser lo más estables, esto significa que no debería caerse por ninguna razón y deben saber responder a solicitudes que ocurran “en el mismo momento” y debe poder manipular archivos grandes. Para esto, en las revisiones el grupo debe contar con un archivo bastante grande para poder hacer las pruebas (Cada equipo debe asegurarse que se cuente con este archivo en el día de la revisión).

Se debe tener cuidado, dependiendo del esquema o servidor de no dejar procesos sin matar o sockets abiertos.

Igualmente se debe evitar que el cliente o el servidor entren en un ciclo infinito de espera, aún cuando ya se haya concluido con la transferencia del archivo.

Descripción Detallada

Los tipos de servidores a implementar son los siguientes:

- FIFO: Este es el más simple. La idea es que el servidor esté oyendo en uno de los puertos y vaya atendiendo las peticiones conforme van llegando. Existe un solo proceso secuencial, ninguna acción se ejecuta en paralelo. De manera que el servidor irá atendiendo las peticiones una por una.
- FORK: Este servidor recibirá las solicitudes y por cada nueva solicitud creará un nuevo proceso que la atienda. Esta claro que el servidor solo atiende solicitudes y los que se encargan de responder y de mantener la comunicación con los clientes son los nuevos procesos creados. Los procesos hijos deben morir cada vez que haya terminado la transferencia del archivo, al igual que la comunicación o socket debe ser cerrado. El proceso padre no debe morir, ni es necesario que lleve control de sus hijos.

- THREAD: Este servidor recibirá las solicitudes y por cada nueva solicitud creará un nuevo hilo que la atienda. Esta claro que el servidor solo atiende solicitudes y los que se encargan de responder y de mantener la comunicación con los clientes son los hilos creados. Los hilos deben morir cada vez que haya terminado la transferencia del archivo, al igual que la comunicación o socket debe ser cerrado. El proceso padre no debe morir, ni es necesario que lleve control de sus hilos. Sin embargo, el proceso padre debe seguir recibiendo solicitudes, no debe quedarse esperando a que ningún hilo termine.

Se puede solicitar cualquier tipo de archivo, sin embargo, nos concentraremos en imágenes, archivos de texto o html.

- Cliente: el cliente debe permitir que el usuario introduzca un nombre de un archivo y después solicitarlo al servidor. Con la información que obtiene del servidor debe salvar en el disco de la computadora donde corre el cliente el archivo que solicitó. Es claro que se pueden levantar varias instancias del cliente. Pero, además, el usuario puede introducir el nombre de varios archivos separados por comas. Si este es el caso la aplicación debe generar un hilo por cada archivo. Y será el hilo el que se debe comunicar con el servidor por medio del Socket.

Documentación

Se espera que sea un documento donde especifique lo siguiente:

a. Portada, índice, introducción
b. Estrategia de Solución
c. Análisis de Resultados: Deberá elaborar un listado de todas y cada una de las actividades y tareas que deben cubrirse a nivel funcional, para cada una de ellas debe aportar el porcentaje de realización y en caso de no ser el 100% debe justificarse.
d. Casos de pruebas: se espera que definan claramente cada prueba, cuáles son los resultados esperados y cuáles fueron los resultados obtenidos. No es necesario que sean grandes, pero deben evaluar la funcionalidad completa del programa.
e. Comparación: Investigación comparativa entre los hilos de Java y Phtreads. La comparación debe incluir usabilidad, detalles técnicos y rendimiento.
f. Evaluación: explicar sus experiencias al implementar cada uno de los servidores y determinar cuál considera que atiende mejor las solicitudes y que da mejor rendimiento.
g. Manual de usuario: especificar como compilar y correr su tarea.
h. Bitácora de trabajo durante las tres semanas de trabajo, incluyendo verificaciones realizadas (si existieran) de consultas realizadas con el profesor o asistente.
i. Bibliografía y fuentes digitales utilizadas

Aspectos Administrativos

- El desarrollo de este programa debe de realizarse en grupos de no más de tres personas.
- El trabajo se debe de entregar antes del 13 de abril de 2025 a medianoche, por medio del TecDigital