

Sistemas Operativos y Redes (E0224)

Año 2021

Trabajo Práctico N°5

Grupo N°4:

Ignacio Hamann - 68410/3

Juan Pablo Elisei - 68380/5

Tomás Tavella - 68371/4

Resumen

Realizacion del entregable 6.



Facultad de Ingeniería
Universidad Nacional de La Plata

Índice

| | |
|--------------------------------|---|
| 1. Enunciado | 2 |
| 2. Interpretación del problema | 2 |
| 3. Resolución | 3 |
| 4. Conclusiones | 3 |

1. Enunciado

Queremos un programa *servidor concurrente* que permita a un cliente que se conecta utilizando el protocolo TCP enviar un archivo para que el servidor lo almacene.

- Cuando un cliente se conecta, el servidor enviará el mensaje: “listo” y esperará recibir la palabra “archivo”.
- A continuación el servidor espera un espacio y luego el nombre con que se quiere guardar el archivo (solo letras, números y el carácter “.”) finalizado con un espacio.
- Luego se quedará esperando un número codificado en *ascii* que indica el tamaño en bytes del archivo, también finalizado con un espacio.
- A continuación comenzará la recepción de los datos, que serán almacenados en un archivo cuyo nombre es el recibido en primer término, hasta completar la cantidad de bytes correspondiente.
- El servidor debe permitir la recepción de archivos binarios.
- Una vez recibida la totalidad de los datos, el servidor contestará con el siguiente mensaje:
 - “Archivo xx completo, tamaño declarado yy bytes, tamaño real zz bytes.”
 - xx: nombre del archivo.
 - yy: tamaño enviado en el mensaje del cliente.
 - zz: total de bytes recibidos por el servidor.
- Luego el servidor cerrará la conexión con el cliente.
- El servidor debe llevar un archivo de registro de todas las conexiones entrantes, con fecha y hora de inicio y de finalización, tamaño del archivo recibido, cantidad de bytes enviados y cantidad de bytes recibidos en formato *csv*.
- Defina explícitamente cómo será el manejo de errores.
- Para probar el programa puede utilizar el cliente TCP que se vio como ejemplo, o el programa *nc*.
- Recuerde verificar que no queden procesos *zombie* cuando finalizan los hijos generados.

2. Interpretación del problema

Se debe crear el programa de un servidor concurrente que utilice el protocolo TCP, el cual aceptará conexiones de clientes que envían archivos para almacenar en el servidor. El procedimiento a seguir para la esta transferencia es el siguiente:

1. Una vez que un cliente establezca la conexión con el servidor, este ultimo le enviará el mensaje “listo” al cliente, quedando en espera del mensaje de confirmación “archivo” por parte del cliente.
2. Luego el servidor debe recibir un caracter *espacio*, seguido del nombre para almacenar el archivo (solo caracteres alfanuméricos y un punto “.”) y un caracter *espacio* al final.
3. Una vez recibido un nombre valido, el servidor espera la recepción de un numero codificado en *ascii* que indique el tamaño del archivo en bytes, seguido de un caracter *espacio*.
4. Ahora, el servidor comienza la recepción de datos, los que almacena en un archivo binario con el nombre obtenido en el paso 2, hasta llegar a la cantidad de bytes obtenidos en el paso 3.
5. Como forma de confirmación, una vez terminada la recepción el servidor le enviará al cliente el siguiente mensaje:
“Archivo «*Nombre de archivo*», tamaño declarado «*Cantidad de bytes declarada*» bytes, tamaño real «*Cantidad de bytes recibidos*» bytes.”.

6. Finalmente, el servidor cierra la conexión con el cliente.

Adicionalmente, el servidor debe mantener un archivo en formato `csv` que contenga registro de:

- Las conexiones entrantes.
- Fecha y hora de comienzo de la conexiones.
- Fecha y hora de finalización de la conexiones.
- Tamaño de los archivos recibidos.
- Cantidad de bytes enviados.
- Cantidad de bytes recibidos.

Para el manejo de errores se presentan varios casos de errores distintos:

- Error en el nombre de archivo y en el mensaje de confirmación “archivo”: En estos casos, como no se llega a abrir el archivo, simplemente se envía un mensaje de error al cliente y se cierra la conexión.
- Si el archivo enviado es de tamaño superior al declarado, el archivo se trunca al llegar a la cantidad declarada de bytes y se cierra con un código de error, se envía un mensaje de error al cliente y se cierra la conexión.
- Si el archivo enviado es de tamaño inferior al declarado, debe existir un tiempo de *timeout*. Una vez excedido este tiempo, si se recibió una cantidad de bytes menor a la declarada, se cierra el archivo con un código de error, se envía un mensaje de error al cliente y se cierra la conexión.
- Si se cierra la conexión con el cliente prematuramente, se cierra el archivo con un código de error.

Adicionalmente, en todos estos casos el código de error correspondiente se almacena en el registro que lleva el servidor.

3. Resolución

Placeholder text.

4. Conclusiones

Placeholder text.