VIRTUALIZACIÓN Y SISTEMAS OPERATIVOS AVANZADOS

TRABAJO PRÁCTICO 2. SOCKETS Y RPC

OBJETIVO

- Implementar aplicaciones cliente y servidor para intercambiar información entre dos contenedores, que haga uso tanto de Sockets como de Remote Procedure Calls (RPC).

FORMATO DE ENTREGA

Entregar un archivo comprimido .tar.gz o .zip que incluya:

Documentación:

- Nombre, apellido y correo electrónico de cada integrante.
- Las instrucciones para compilar y ejecutar las aplicaciones. De ser necesario, incluir el código fuente con las aclaraciones que se consideren necesarias.

Código fuente:

- Entregar dentro de una carpeta src todo el código fuente desarrollado.

Archivos adicionales:

- Incluir el archivo con usuarios y contraseñas para poder probar los programas generados. Se explica mejor en la sección de la consigna.
- Incluir el archivo con el/los tokens que son válidos (si se decide hacer así) o incluir en la documentación una explicación clara del funcionamiento pretendido. Esto también se detalla mejor en la sección de consigna.

La entrega se realiza por e-mail a: dharispe@frsf.utn.edu.ar

Fecha de entrega: 15 de junio de 2023

CONSIGNA

Iniciar cada uno de los contenedores instalados y configurados en la guía para el TP2. Luego, utilizarlos para realizar las actividades que se mencionan a continuación:

- Implementar una aplicación servidor que atienda conexiones por socket (denominado a partir de ahora serverAuth). Esta aplicación validará un usuario y contraseña, que se reciban desde un cliente, comparando con la información que se encuentra en un archivo (se puede ubicar junto al ejecutable del programa servidor). Las credenciales pueden estar almacenadas en texto plano.

- Cuando la aplicación serverAuth corrobore la correcta combinación de usuario y contraseña, deberá proporcionar al cliente un código especial (similar a un token), que será usado luego.
- Implementar una aplicación servidor que proporcione un procedimiento remoto para utilizar RPC (denominado a partir de ahora serverInteract). El procedimiento permitirá la edición del contenido de archivos, que se ubiquen junto al ejecutable de la aplicación. La edición puede estar limitada a un archivo especifico en el cual, por ejemplo, se vayan agregando entradas de datos (como registros en nuevas líneas). Este procedimiento remoto solo se ejecutará satisfactoriamente si se recibe un código de uso valido (token recibido en la interacción con el serverAuth). El token puede obtenerse desde un archivo.
- Implementar una sola aplicación cliente que sea capaz de interactuar tanto con el servidor de autentificación (serverAuth) como con el servidor de interacción (serverInteract). El comportamiento que se llevará a cabo puede depender de la cantidad o tipo de parámetros recibidos por la aplicación cliente. Otra opción puede ser tener un menú al momento de ejecutar.

Consideraciones:

- La interacción con archivos no necesita ser demasiado compleja. La idea es que exista la posibilidad de alterar archivos locales con información que llega como parámetro al momento de llamar el procedimiento remoto.
- El token se puede tratar de un número o código que se guarda en un archivo. No es necesario que varie en el tiempo o con los usos.
- Ambos programas servidores pueden estar corriendo en simultaneo y esperando conexiones o llamadas remotas. Otra alternativa podría ser que serverAuth de inicio a serverInteract al tener una autentificación correcta.
- Los programas servidores deberán dar como respuesta algún indicador (puede ser un numero o texto) de como resulto cada una de las operaciones.

EJEMPLO DE USO DE CLIENTE (si comportamiento depende de parámetros)

Autentificación

clientSRPC <ip contenedor servidor> -u <usuario> -p <contraseña>

Interacción

clientSRPC <ip contenedor servidor> -t <token> -o <opción> -c <contenido>

token: Valor recibido en la interacción de autentificación previa.

opción: Si del lado de serverAuth se soportan más de una acción con los archivos, este parámetro podría indicar que es lo que se quiere hacer.

contenido: Si la acción corresponde a agregar contenido a un archivo, el contenido a agregar se especifica en ese lugar.

INFORMACIÓN DE UTILIDAD PARA RECORDAR

- Para especificar datos de tipo string en los archivos .x que se usan con rpcgen, se puede usar el dato aislado o dentro de un struct como se muestra a continuación:

```
/*  EJEMPLO example.x */

struct data{  string a<10>;  };

program EX_PROG{

  version EX_VERS{

     int ex(data)=1;    }=1;

}=0x4562877;
```

- Para acceder a, leer de, escribir en y desplazarse dentro de un archivo en C, usar las funciones fopen, fscanf, fprintf y fseek respectivamente.
- Con la función fscanf se puede leer hasta encontrar un carácter determinado. Por ejemplo, si se quiere leer hasta encontrar ":" (dos puntos). Se debe escribir:

```
int readCount = fscanf(fptr, "%[^:]", buffer);
        readCount: cantidad de caracteres leídos.
        fptr: handle para el archivo, que se obtuvo con fopen.
        buffer: variable donde se almacena lo que se leyó del archivo.
```

- Para ejecutar un programa en segundo plano desde la terminal, agregar un espacio y & al final. Por ejemplo, para iniciar un programa llamado server, ejecutar: ./server &
  Esto liberará la terminal luego de ejecutar el programa, para poder seguir ejecutando comandos.
- Con el comando kill y la opción -9 se puede matar procesos (que podrían estar en segundo plano) si se especifica el PID del proceso. Esto se puede obtener ejecutando el comando "ps".

  Ejemplo: kill -9 1234