Universidad Simón Bolívar
Departamento de Computación y TI
Redes de Computadoras I (CI-4835)

MINIPROYECTO N° 1

OBJETIVOS ESPECÍFICOS:

Al finalizar esta actividad el estudiante deberá estar en capacidad de:

- Comprender el uso y la programación de la Interfaz de Aplicaciones (API) Sockets de Berkeley.
- Entender el uso y programación de las llamadas a Procedimientos Remotos (RPC).

Descripción

El Sistema Ferroviario Nacional de Venezuela, desea implementar un proyecto de prueba para algunos vagones de trenes que lleven solamente personas sentadas a fin de mejorar la calidad de vida de venezolanos con problemas de discapacidad, mujeres embarazadas y personas de la tercera edad. Para ello se desea construir un sistema de reservación y compra de boletos que sea implementado a través del uso de la Interfaz de Aplicaciones Sockets de Berkeley bajo una arquitectura Cliente/Servidor. Con el fin de resolver el planteamineto se le ha pedido a usted que diseñe y programe un módulo simple que contemple los siguientes aspectos:

Parte I.

1.- El módulo que usted debe construir procesará únicamente las reservaciones de un vagón por tren. Para ello se quiere que construya un mecanismo de software que mantenga una matriz de dimensión 10 x 4 para representar los asientos de un vagón del tren. Los diferentes asientos se identificarán por la fila y la columna en la que se encuentran. Además, se requiere construir procesos clientes que reserven boletos especificando el asiento concreto, identificado por un número de fila y un número de columna. Cuando el cliente intente realizar una reserva del boleto, el servidor le deberá responder una de las siguientes opciones:

Reservado. En este caso, el cliente al recibir la respuesta muestra un mensaje por salida estándar indicando que la reserva se ha realizado correctamente y finaliza su ejecución.

Ocupado y una lista de los asientos que se encuentran libres actualmente. En este caso, el cliente al recibir la respuesta solicita otro asiento hasta que logre reservar uno que esté libre.

Vagón completo. En este caso, el cliente, al recibir la respuesta, la muestra por salida estándar y finaliza su ejecucición.

El cliente hace la petición hasta 3 veces, si el servidor no responde envía un mensaje indicando que el tiempo de repuesta se agotó.

Se deben hacer las validaciones necesarias, tales como que un puesto no esté fuera del rango, así como también que se respete el orden de la secuencia de opciones y cualquier otra que sea considerada.

La sintaxis del servidor es:

reserva_bol_ser -f <filas> -c <col> [-p puerto]

La sintaxis del clente deberá ser:

reserva_bol_cli <ip-servidor> -p <puerto servicio> -f <fila> -c <col>

Implemente un sistema cliente servidor, que provea atención a un número, concurrente o paralelo, suficiente de clientes que simulen el proceso de reserva. En una invocación solamente se puede reservar un boleto.

2.- Para la definición del protocolo se deben tener en cuenta los siguientes aspectos, describiendo en forma clara y sencilla cada uno de ellos:

1. Servicio que proporciona el protocolo
2. Suposiciones sobre el entorno donde se ejecuta el protocolo
3. Vocabulario de los mensajes utilizados en el protocolo
4. Formato de los mensajes del vocabulario del protocolo
5. Reglas de procedimiento que controlan la consistencia del intercambio de mensajes. Se recomienda usar diagramas temporal y/o máquinas de estados finitos.

Esta sección debe imprimirse y ser dejada en los casilleros de cada profesor ubicados en el departamento de computación y TI de 8:30 a.m. a 3:45 p.m.

3.- La entrega debe hacerse de manera electrónica a través del Aula Virtual, antes de la fecha y hora indicada: lunes 18/05/15 a las 10 p.m. El acceso para subir los mini proyectos será cerrado luego de la hora establecida. Los grupos son de 2 personas.

- •La entrega es un archivo comprimido con nombre Apellido1_Apellido2_X.tar.gz (ver http://ldc.usb.ve/~figueira/Cursos/ci3825/taller/material/TGZ.html de cómo hacer un archivo tar.gz), donde X será reemplazado por los términos Sockets o RPC y Apellido1 y Apellido2 corresponden a los apellidos de los integrantes del grupo (por ejemplo, Mora_Azuaje_Sockets.tar.gz) con los siguientes archivos:
  - •Archivos fuente. Por ejemplo, si se trata de un proyecto en lenguaje C, debe incluir todos los ".c" y ".h.
  - •Makefile
  - •Un archivo LEEME.txt, que explique el contenido del "tar.gz":
    - •nombres y apellidos de los integrantes del grupo
    - •números de carnet
    - •qué archivos lo componen y qué tiene cada archivo
    - •qué hace el programa,

•cómo se ejecuta,
•qué condiciones particulares tiene. Por ejemplo, limitaciones ("la entrada XX tiene maximo 50 caracteres" o "el máximo número de registros YY es 20") o errores.
• Tenga cuidado de no incluir ejecutables, archivos objeto (.o) o cualquier otro archivo generado automáticamente a partir de los fuentes.

Parte II.

Realice los cambios pertinentes para implementar la solución del mismo problema a través de llamadas a procedimientos remotos RPC. Debe entregar también la documentación con los cambios que correspondan y el archivo comprimido.

GDRC/Abril-Julio 2015