Práctica 1. Introducción a la programación de aplicaciones con las primitivas Socket en C

Objetivos

- Familiarización con las primitivas socket.
- Creación de aplicaciones cliente sencillas basadas en sockets.
- Creación de aplicaciones servidor sencillas basadas en sockets.
- Familiarización con el entorno de desarrollo de aplicaciones usado en el curso.
- Familiarización con los recursos de referencia de Internet.
- Familiarización con la sintaxis ABNF.

Programación

- Dos sesiones de 2 horas en el laboratorio, distribuidas de la siguiente manera:
    - Primera sesión: Análisis y modificación de un cliente y un servidor TCP sencillos.
    - Segunda sesión: Añadir nueva funcionalidad a un cliente y un servidor TCP sencillos.

Recursos

- Ordenador PC.
- Conexión a Internet
- Microsoft Visual Studio 12
- Programa de control de versiones GitHub.

Descripción

En esta práctica se revisarán las primitivas socket para la programación de clientes y servidores sobre TCP. Y se estructurará de la siguiente manera:

- Introducción a las funciones socket y sus formatos.
- Manejo del entorno de desarrollo a usar durante el curso.
- Diseño general de clientes TCP usando primitivas socket.

Primera sesión -

Parte 1ª. Análisis del código (1,0 hora)

1. Crear dos proyectos en Visual Studio de Aplicación de Consola Win32. En el primero se incluirá el fichero `cliente.c`, y en el segundo el fichero `servidor.c` y el fichero de cabecera `protocol.h`.
2. Compilar y ejecutar.
3. Estudiar el funcionamiento y la estructura de ambos códigos:

a. Enumerar cada una de las primitivas sockets utilizadas.
b. Enumerar y comentar qué función tienen cada uno de los comandos de aplicación.

Parte 2º. Mejoras al código (1,0 horas)

Añada todo lo necesario a ambos programas para soportar adecuadamente los errores de las primitivas de transporte y así conseguir que ambos programas evolucionen de manera predecible.

Opcional: Detecte los posibles fallos en el soporte al protocolo de aplicación y describa en el código las causas. Implemente una solución para los mismos. (+10%).

Segunda sesión

Parte 3ª. Adición de nueva funcionalidad (2 horas)

1. Modificar el cliente para que solicite al usuario, una vez autenticado, dos números enteros de cuatro dígitos como máximo, precedidos el comando **<SUM>**[1]**.** El formato deberá ser exactamente el siguiente:

   SUM SP NUM1 SP NUM2 CRLF

   Donde:

   SUM = "sum"
   NUM1 =1*4DIGIT
   NUM2=1*4DIGIT

2. Modifica el servidor para que en el caso de recibir el comando **<SUM>** seguido de un espacio, un número entero, un espacio y otro número entero, el servidor responda al cliente con un mensaje **<OK>** seguido de un espacio y el resultado de la suma de ambos números enteros. El formato será el siguiente:

   OK SP SUMA CRLF

   Donde:

   OK = "ok"
   SUMA=1*5DIGIT

   Si el formato del comando **<SUM>** no es el correcto, se deberá devolver **<error>.**

   ERROR = "error"

   Ayuda: *usar la función sscanf_s para la lectura de los dos sumandos de la cadena recibida del cliente.*
3. El cliente deberá mostrar lo que le devuelva el servidor en cualquier caso.

[1] Recuerde que se usa la notación ABNF.

Material a entregar

Ficheros de código fuente: Todos aquellos ficheros necesarios para la adecuada compilación y construcción del código, nunca ficheros ejecutables o de código objeto.

El código entregado deberá cumplir con los objetivos concretos descritos en la práctica, tener comentarios que ayuden a su seguimiento y corrección y no tener errores de sintaxis, ni de ejecución.

Todo el código deberá estar accesible al profesor a través de la **herramienta de control de versiones usada**, así como en la plataforma de docencia.

Evaluación

La evaluación de esta práctica, una vez obtenido el visto bueno del profesor, se realizará a través de dos instrumentos:

- La corrección del material entregado. 50% de la calificación.
- La realización de un test sobre la práctica. %50% de la calificación.

Téngase en cuenta que el plagio conllevará la calificación de 0 en la práctica.

Bibliografía

1. Donahoo, M. J. "**TCP/IP sockets in C**" 2ª Edición, Morgan Kaufmann.
2. Crocker, D., Ed., and P. Overell, **"Augmented BNF for Syntax Specifications: ABNF"**, STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.