



PRÁCTICA DE NIVEL DE TRANSPORTE (SOCKETS)

(CC) Moreno, A. M. & Bravo, S. (Redes I, 2023)

Tabla de contenido



- Presentación
 - ▣ Objetivos y Entorno de desarrollo
 - ▣ Servicio “Patata caliente”
 - Especificaciones del protocolo
 - Mensajes
 - Ejemplo de dialogo
 - Requisitos

Objetivos y Entorno de desarrollo



- El objetivo de esta práctica es implementar una aplicación en red como
 - ▣ usuario del nivel de transporte y
 - ▣ según la arquitectura o modelo de programación cliente-servidor
- Entorno de desarrollo
 - ▣ Estación de trabajo con S.O. Debian GNU/Linux 9 (*stretch*) (nopal.usal.es)
 - ▣ Sockets de Berkeley
 - ▣ Lenguaje de programación C

Especificaciones del protocolo



- El servicio que vamos a implementar se denomina “Patata Caliente” (PC).
- Consiste en acertar la respuesta numérica a una pregunta en un determinado número de intentos. En cada intento se contestará si la respuesta correcta es mayor o menor que la indicada por el jugador o éste ha acertado.

Mensajes PC (I)

5

- PC emplea dos tipos de mensajes:
 - ▣ Peticiones de los clientes a los servidores
 - ▣ Respuestas de los servidores a los clientes
 - ▣ Las líneas de los mensajes PT
 - Son siempre líneas de caracteres terminadas con los caracteres CR-LF (retorno de carro ‘\r’ (ASCII 13 (0x0D)), - avance de línea ‘\n’ (ASCII 10 (0x0A)))
 - La longitud máxima de las líneas no debe exceder 516 bytes, contando todos los caracteres (incluido el CR-LF final)

Mensajes PC (II)

6

□ **Peticiones del cliente al servidor**

- La orden HOLA es la primera en enviarse
 - Se forma: HOLA[CR-LF]
- La orden RESPUESTA envía una respuesta
 - Se forma: RESPUESTA <numero>[CR-LF]
 - Donde *número* es la respuesta numérica a la pregunta
- La orden + solicita otra pregunta después de un acierto o si se agotan los reintentos
 - Se forma: +[CR-LF]
- Se pueden seguir jugando hasta que se finalice con la orden ADIOS[CR-LF]

Mensajes PC (III)

□ Respuestas del servidor

Núm.	Cadena	Descripción
220	Servicio preparado	Respuesta cuando el cliente realiza la conexión
250	<pregunta>#<intentos>	Respuesta a las ordenes HOLA y + con la pregunta y número de intentos
354	MAYOR/MENOR#<intentos>	Respuesta al envío de la orden RESPUESTA indicando si la respuesta correcta es mayor o menor y el número de reintentos que quedan
350	ACIERTO	Respuesta al envío de la orden RESPUESTA cuando hay acierto
375	FALLO	Respuesta al envío de la orden RESPUESTA cuando se ha superado el número de reintentos
221	Cerrando el servicio	Respuesta a la orden ADIOS
500	Error de sintaxis	Respuesta a errores de sintaxis en cualquier orden o secuencias incorrectas (Ej. Envío de RESPUESTA sin haber saludado (sin orden HOLA previa))

Mensajes PC (IV)

8

Ejemplo de dialogo (S: Servidor, C: Cliente)

- ▣ S:220 Servicio preparado
- ▣ C: HELO
- ▣ S: 500 Error de sintaxis
- ▣ C: HOLA
- ▣ S: 250 ¿A cómo está hoy la prima de riesgo?#4
- ▣ C: no lo se
- ▣ S: 500 Error de sintaxis
- ▣ C: RESPUESTA 67
- ▣ S: 354 MAYOR#3
- ▣ C: RESPUESTA 110
- ▣ S: 350 ACIERTO
- ▣ C: +
- ▣ S: 250 ¿Cuántos estados tiene EEUU?#4
- ▣ C: RESPUESTA 51
- ▣ S: 350 ACIERTO
- ▣ C: ADIOS
- ▣ S: 221 Cerrando el servicio

NOTA: Las líneas de los mensajes PC son siempre líneas de caracteres terminadas con los caracteres CR-LF (retorno de carro '\r' (ASCII 13 (0x0D)), - avance de línea '\n' (ASCII 10 (0x0A)))

Requisitos (I)

□ Programa Servidor

- Aceptará peticiones de sus clientes tanto en TCP como en UDP
 - Registrará todas las peticiones en un fichero de "log" llamado peticiones.log en el que anotará:
 - Fecha y hora del evento
 - Descripción del evento:
 - Comunicación realizada: nombre del host, dirección IP, protocolo de transporte, nº de puerto efímero del cliente
 - Una línea por cada uno de los mensajes intercambiado. Indicando en las órdenes recibidas
 - Orden recibida: nombre del host, dirección IP, protocolo de transporte, el puerto del cliente y la orden.
 - Para las respuestas
 - Respuesta enviada: nombre del host, dirección IP, protocolo de transporte, puerto del cliente y la respuesta mandada.
 - Comunicación finalizada: nombre del host, dirección IP, protocolo de transporte, nº de puerto efímero del cliente
 - Se ejecutará como un "daemon"

□ Programa Cliente

- Se comunicará con el servidor bien con TCP o con UDP
- Leerá por parámetros el nombre o IP del servidor y el protocolo de transporte TCP o UDP de la siguiente forma:
 - cliente nombre_o_IP_del_servidor TCP
- Realizará peticiones al servidor como se ha indicado anteriormente
- Realizará las acciones oportunas para su correcta finalización

Requisitos (II): pruebas

10

- Durante la fase de pruebas el cliente podrá ejecutarse como se muestra en el ejemplo de diálogo anterior, pero en la versión para entregar el cliente
 - ▣ Leerá de un fichero las órdenes que ha de ejecutar. El nombre del fichero lo recibirá como parámetro
 - ▣ Escribirá las respuestas obtenidas del servidor y los mensajes de error y/o depuración en un fichero con nombre el número de puerto efímero del cliente y extensión .txt

Requisitos (III): versión entregable

11

- Para verificar que esta práctica funciona correctamente y permite operar con varios clientes, se utilizará el *script* `lanzaServidor.sh` que ha de adjuntarse obligatoriamente en el fichero de entrega de esta práctica junto a los ficheros de ordenes en él utilizados.
- El contenido de `lanzaServidor.sh` es el siguiente:

```
# lanzaServidor.sh
# Lanza el servidor que es un daemon y varios clientes
# las ordenes están en un fichero que se pasa como tercer parámetro
./servidor
./cliente nogal TCP ordenes.txt &
./cliente nogal TCP ordenes1.txt &
./cliente nogal TCP ordenes2.txt &
./cliente nogal UDP ordenes.txt &
./cliente nogal UDP ordenes1.txt &
./cliente nogal UDP ordenes2.txt &
```

Requisitos (IV): documentación

12

- Entregar un informe en formato PDF que contenga:
 - ▣ Detalles relevantes del desarrollo de la práctica
 - ▣ Documentación de las pruebas de funcionamiento realizadas



PRÁCTICA DE NIVEL DE TRANSPORTE (SOCKETS)

(CC) Moreno, A. M. & Bravo, S. (Redes I, 2023)