

Arquitectura de Redes y Servicios

Práctica tema 5: Cliente-servidor no conectado e iterativo

Versión 2

Dr. Jorge Fernández, Dr. Diego R. Llanos
Departamento de Informática
Universidad de Valladolid

18 de octubre de 2018

Índice

1. Introducción	1
2. Desarrollo de un cliente QOTD UDP	1
3. Desarrollo de un servidor QOTD UDP iterativo	2
4. Condiciones de entrega y evaluación	3

1. Introducción

El servicio QOTD (Quote of the Day) es una herramienta habitual de depuración de errores y medida de rendimiento en redes. El protocolo es muy simple: cuando el servidor recibe una solicitud de conexión TCP o un datagrama UDP dirigido a ese puerto, éste responde enviando una cadena de texto que puede estar formada por **una o varias líneas**, debe componerse de **caracteres imprimibles ASCII**, **espacios en blanco** (), **retornos de carro** (\r) y/o **saltos de línea** (\n), y no puede superar los **512 caracteres**. El servicio Quote of the Day es un servicio estándar, y como tal está asociado a un puerto bien conocido.

La práctica consiste en desarrollar un cliente y un servidor iterativo del protocolo QOTD, utilizando sockets UDP. La práctica deberá desarrollarse en lenguaje C en las máquinas virtuales instaladas en la práctica anterior.

Es imprescindible que el alumno comprenda perfectamente el significado de cada línea de código desarrollada. Los únicos recursos necesarios para realizar la práctica son (a) los apuntes desarrollados por el profesor y (b) las páginas **man**.

Esta práctica está dimensionada para su realización en seis horas lectivas. Deberá desarrollarse fundamentalmente en clase.

2. Desarrollo de un cliente QOTD UDP

La primera tarea a realizar es el desarrollo de un cliente QOTD utilizando el protocolo UDP, denominado `qotd-udp-client-Apell1-Apell2`. El cliente se invocará con el siguiente formato:

```
./qotd-udp-client-Apell1-Apell2 direccion.IP.servidor [-p puerto-servidor]
```

El cliente utilizará la función `inet_aton(3)` para transformar la dirección IP recibida como parámetro a una dirección formada por 32 bits en *network byte order*.

Si el usuario no especifica el número de puerto en el que el servidor está escuchando, el cliente recuperará con `getservbyname()` el número de puerto asociado al servicio QOTD. En caso contrario, el cliente utilizará como puerto del servidor el número de puerto suministrado. Para ello, el cliente deberá transformar la cadena con el número de puerto a un entero de 16 bits en *network byte order*. Esto puede conseguirse utilizando la función `sscanf(cadena, "%d", &numero)`, que se encarga de transformar la cadena en un número, y luego utilizar la función `htons(3)`, que recibe un número `short int` de 16 bits y lo convierte a *network byte order*.

El cliente enviará un datagrama UDP con una cadena arbitraria al puerto QOTD del servidor correspondiente. En respuesta, el servidor enviará un datagrama con una cadena formada por el nombre del servidor y seguida del Quote of the Day. El cliente deberá entonces imprimir por pantalla la cadena recibida y terminará.

Para probar el funcionamiento del cliente desarrollado, podrá intentarse la conexión con el servidor desarrollado por el profesor. La dirección IP del servidor del profesor es 10.0.25.250.

3. Desarrollo de un servidor QOTD UDP iterativo

El siguiente paso consiste en el desarrollo del servidor QOTD UDP. Este servidor tendrá un nombre de la forma `qotd-udp-server-Apell1-Apell2`. El formato de invocación del servidor será

```
./qotd-udp-server-Apell1-Apell2 [-p puerto-servidor]
```

Si no se suministra número de puerto, el servidor recuperará el puerto asociado al servicio QOTD. El servidor colocará un socket UDP a la escucha en el puerto correspondiente, y entrará en un bucle infinito. Cada vez que reciba un datagrama, e independientemente de su contenido, el servidor generará una cadena formada por el nombre del servidor y la cadena de texto a enviar.

Para generar el Quote of the Day, utilizaremos la salida generada por el juego `fortune`, incluido en el paquete `bsd-games` de Slackware. Al invocarlo sin parámetros, `fortune` devuelve un Quote. Este comando no fue instalado en la práctica 1, así que lo primero es instalarlo. Los comandos para instalar dicho paquete en la máquina virtual son:

```
mount /dev/cdrom /mnt/dvd # Montar DVD de instalación de Slackware
cd /mnt/dvd/slackware64/y # Ir al directorio que contiene bsd-games
installpkg bsd-games*txz # Instalar el paquete bsd-games
```

Para obtener la cadena de caracteres que contenga el aforismo generado con `fortune`, la forma más sencilla es utilizar la función `system(3)`, que permite ejecutar un comando del `shell`, redirigiendo el resultado a un fichero y luego leyendo dicho fichero. Un ejemplo de uso para obtener los **100 primeros caracteres** de una cita sería el siguiente:

```
#define MAXLENGTH 100
static char buffQuote[MAXLENGTH];
system("/usr/games/fortune -s > /tmp/tt.txt");
FILE *fich = fopen("/tmp/tt.txt", "r");
int nc = 0;
do {
    buffQuote[nc++] = fgetc(fich);
} while(nc < MAXLENGTH-1);
fclose(fich);
```

Para poder enviar el resultado al cliente, habrá que concatenar las dos cadenas (la que contiene el encabezado con el nombre del servidor y la que contiene la cita) haciendo uso de la función

`strcat(3)` o alguna de sus variantes. La cadena que deberá enviar el servidor deberá tener **exactamente** el siguiente formato (nótese el salto de línea entre el encabezado y la cita):

```
Quote Of The Day from vm25XX:
<cita>
```

Por ejemplo:

```
Quote of the day from vm2501:
NOBODY EXPECTS THE SPANISH INQUISITION
```

Algunos de los aforismos generados por **fortune** son especialmente largos, por lo que el alumno deberá utilizar la opción **-s** para solicitar a **fortune** aforismos cortos. Finalmente, una vez enviada una respuesta conforme con todos los requisitos establecidos, el servidor quedará a la espera del siguiente datagrama.

Nota 1: El comando **netstat -tulpn** muestra información sobre los puertos que están siendo utilizados en la máquina local. Para obtener toda la información es necesario ejecutar este comando como **root**.

Nota 2: Tanto el cliente como el servidor deberán compilarse con la opción **-Wall**, que muestra por pantalla todas las advertencias del compilador. Ambos programas deberán compilar sin ningún error o advertencia. Se penalizarán las advertencias no resueltas.

4. Condiciones de entrega y evaluación

1. Esta práctica deberá realizarse en la máquina virtual instalada a principios de curso.
2. Los dos ficheros de código fuente (correspondientes al cliente y al servidor) deberán comenzar con un comentario indicando el nombre de su autor, con el siguiente formato:

```
// Practica tema 5, Apellido1 Apellido2 Nombre
```

3. Tanto el cliente como el servidor deberán estar adecuadamente comentados. De lo contrario, se restarán dos puntos de la nota final.
4. Tanto el cliente como el servidor deberán compilar sin advertencias (opción **-Wall** del compilador **gcc**). De lo contrario, se restarán tres puntos de la nota final.
5. Cuando esté finalizada, se creará un fichero comprimido en formato **ZIP** que contenga exclusivamente los dos ficheros fuente (sin directorios) y se subirá al entorno virtual. Un fallo en las condiciones de entrega supondrá otro punto menos en la nota final.
6. El plazo de entrega finalizará el **domingo 28 de octubre de 2018 a las 23:55**.
7. En la sesión del **lunes 29 de octubre de 2018** se describirá la siguiente práctica, por lo que la asistencia a esa sesión es obligatoria para todos los alumnos.
8. Esta práctica supondrá un 20 % de la calificación de prácticas de la asignatura.
9. Se utilizará un sistema automático de detección de copias. En caso de copia, todos los alumnos involucrados deberán presentar todas las prácticas, incluida la práctica copiada, y se corregirán sobre 7, no sobre 10. En el caso de que la calificación media ponderada de todas las prácticas sea menor que cinco, dichos alumnos habrán suspendido la convocatoria ordinaria de la asignatura.
10. No se admitirán entregas fuera de plazo.