

Arquitectura de Redes y Servicios

Práctica tema 5: Cliente-servidor no conectado e iterativo

Dr. Diego R. Llanos
Departamento de Informática
Universidad de Valladolid
Versión 2.4

21 de octubre de 2019

1. Introducción

El servicio DAYTIME permite consultar de forma remota la fecha y hora del día, en un formato comprensible al ser humano. El protocolo es muy simple: cuando el servidor recibe una conexión, envía una cadena con la fecha y la hora del día, como:

Mon Sep 28 18:28:20 CET 2019

El servicio DAYTIME 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 DAYTIME, 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 DAYTIME UDP

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

```
./daytime-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 DAYTIME. 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 y al servidor correspondiente. En respuesta, el servidor enviará un datagrama con una cadena que incluirá el nombre del servidor, la fecha y la hora. 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 DAYTIME UDP iterativo

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

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

Si no se suministra número de puerto, el servidor recuperará el puerto asociado al servicio DAYTIME. 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 (utilizando la función `gethostname(2)`).
- La fecha y la hora. Para obtener la fecha y la hora, la forma más sencilla es utilizar la función `system(3)`, que permite ejecutar un comando del `shell`, para ejecutar el comando `date` redirigiendo el resultado a un fichero, y luego leyendo dicho fichero. Un ejemplo de uso es el siguiente:

```
#define SIZEBUFF 30
char buff[SIZEBUFF];
FILE *fich;
...
system("date > /tmp/tt.txt");
fich = fopen("/tmp/tt.txt", "r");
if (fgets(buff, SIZEBUFF, fich) == NULL) {
    printf("Error en system(), en fopen(), o en fgets()\n");
    exit(EXIT_FAILURE);
}
printf("La fecha y la hora son %s\n", buff);
```

Para poder enviar el resultado al cliente, habrá que concatenar las dos cadenas (la que contiene el nombre del host y la que contiene la fecha y hora) haciendo uso de la función `strcat(3)`.

La cadena que deberá enviar el servidor deberá tener **exactamente** el siguiente aspecto:

```
lab05v00: Mon Sep 27 08:58:30 CET 2019
```

Una vez enviada la respuesta, el servidor quedará a la espera del siguiente datagrama.

Nota: 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: 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 3 de noviembre de 2019 a las 23:55**.
7. En la sesión del **lunes 4 de noviembre de 2019** se evaluará el trabajo de seis alumnos de manera *pseudoaleatoria*, 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 figurarán como suspensos en la convocatoria ordinaria, debiendo presentar todas las prácticas por e-mail al profesor para poder presentarse a la convocatoria extraordinaria. Todas las prácticas se corregirán sobre 7.
10. No se admitirán entregas fuera de plazo.