

Pràctica 6: Programació de clients amb TCP i UDP.

Autor: Enrique Bonet

Objectiu i descripció general.

L'objectiu d'esta pràctica és la programació de dos clients, un utilitzant el protocol UDP i un altre amb el protocol TCP. El client (UDP o TCP) consultarà el servici *daytime* d'un ordinador remot i rebrà com a resposta la informació sobre el dia i l'hora de l'ordinador remot, havent de mostrar esta informació per pantalla.

Daytime és un servici dels ordinadors que es troba instal·lat en el port 13 dels protocols TCP i UDP de la capa de transport.

Més concretament, el servici *daytime* amb UDP espera la recepció d'un datagrama UDP buit, açò és, sense cap tipus de dada, i respon a este paquet amb una cadena de text que conté informació sobre el dia i l'hora del sistema remot consultat.

Pel seu costat, el servici *daytime* amb TCP espera la recepció d'una petició de connexió TCP al port 13, i sense necessitat d'esperar la recepció de cap dada envia una cadena de text amb idèntica informació que per a UDP, tancant a continuació de forma automàtica la connexió establida.

Un exemple del format de la cadena rebuda és:

Fri Feb 28 12:14:18 2003

El funcionament del servici *daytime* amb UDP és el següent:

1. El client envia un datagrama buit al servidor.
2. El servidor rep el datagrama buit, a partir del qual obté l'adreça IP i el port UDP del client que sol·licita la informació.
3. El servidor realitza una consulta per mitjà de la funció *time* de C/C++ i converteix la dita consulta en una cadena de text, amb el format especificat amb anterioritat, per mitjà d'una crida a la funció *asctime* de C/C++.
4. El servidor crea un datagrama amb la informació obtinguda en el punt 2, datagrama que té com a única dada la cadena de text tornada per *asctime*.
5. El datagrama arriba al client, el qual pot llegir-lo i extraure la informació del dia i hora del servidor.

Com tot servici UDP, este és no fiable, és a dir, tant el datagrama de petició del servici del client al servidor com la resposta del servidor al client poden perdre's i el client podria no rebre, en alguns casos, cap resposta. A més, el servici no té perquè estar

implementat en tots els ordinadors, per la qual cosa, també pot no obtindre's cap resposta d'aquells ordinadors que no tinguen implementat el servici.

Per la seua banda, el funcionament amb TCP és el següent:

1. El client sol·licita una connexió al servidor.
2. El servidor accepta la connexió.
3. El servidor realitza una consulta per mitjà de la funció *time* de C/C++ i converteix la dita consulta en una cadena de text, amb el format especificat amb anterioritat, per mitjà d'una crida a la funció *asctime* de C/C++.
4. El servidor envia la cadena de text tornada per *asctime* i tanca la connexió.
5. El client rep la cadena de text i extrau la informació del dia i hora del servidor.

Tot servici TCP és fiable, és a dir, les dades sempre arriben de l'emissor al receptor o bé es produeix un missatge d'error, amb la qual cosa no cal tindre en compte la possibilitat que les dades no arriben al seu destí. A més, si el servici no està disponible en un ordinador, l'intent de connexió al port de *daytime* del dit ordinador produirà un error.

Desenrotllament de la pràctica.

Desenrotllar un programa en C/C++ que implemente dos clients de *daytime*, un utilitzant el protocol UDP i un altre utilitzant TCP. Els clients a desenrotllar, de noms, per exemple, *horaUDP.c* i *horaTCP.c*, hauran de respondre a les següents línies de comandaments:

```
horaUDP <adreça IP del servidor> [timeout]  
horaTCP <adreça IP del servidor>
```

On <> indica un paràmetre de pas obligatori i [] un paràmetre de pas opcional.

El paràmetre opcional *timeout* del client UDP servirà per a poder incrementar els segons del *timeout* per defecte que tindrà el programa. Dit *timeout* per defecte haurà de ser, si no s'especifica, d'1 segon.

L'anàlisi bàsica de desenrotllament del programa amb el protocol UDP és:

- 1 Creació d'un *socket* UDP.
- 2 Creació d'un datagrama UDP buit i enviament de dit datagrama al servidor.
- 3 Espera d'un temps donat pel valor de *timeout* (o el *timeout* per defecte si no s'ha especificat) a la recepció del datagrama de resposta.
- 4 Si el susdit datagrama és rebut, es col·loca en la pantalla la resposta obtinguda, sinó es col·loca un missatge indicant que el *timeout* ha vençut.
- 5 Tancament del *socket* UDP creat.

L'anàlisi bàsica de desenrotllament del programa amb el protocol TCP és:

- 1 Creació d'un *socket* TCP.
- 2 Connexió del *socket* creat al port *daytime* del servidor.
- 3 Si la connexió ha sigut realitzada, es col·loca en pantalla la resposta obtinguda, sinó es col·loca un missatge que indique que el servei no esta disponible.
- 4 Tancament del *socket* TCP creat.

Hi ha exemples compilats dels programes en la pàgina web de l'assignatura. Els programes tenen el nom *horaUDP* i *horaTCP*.

Nota: Pel fet que els ordinadors de pràctiques tenen configurat el tallafocs de Linux en un format molt restrictiu, és possible que els programes, especialment el de protocol UDP, no funcionen al no rebre mai la resposta. Per a desactivar el tallafocs i que funcionen correctament, podeu executar el comandament "*service iptables stop*". Per a provar el funcionament de la pràctica l'ordinador lab3inf005 té habilitat el servei de *daytime* amb TCP i UDP.

Práctica 6: Desarrollo de clientes bajo TCP y UDP.

Autor: Enrique Bonet

Objetivo y descripción general.

El objetivo de esta práctica es el desarrollo de dos clientes, uno utilizando el protocolo UDP y otro el protocolo TCP. El cliente (UDP o TCP) consultará el servicio daytime de un ordenador remoto y recibirá como respuesta la información sobre el día y la hora del ordenador remoto, debiendo mostrar esta información por pantalla.

El servicio daytime es un servicio existente en los ordenadores que se encuentra instalado en el puerto 13 de los protocolos TCP y UDP de la capa de transporte.

En concreto, el servicio daytime bajo UDP espera la recepción de un datagrama UDP vacío, esto es, sin ningún tipo de dato, y responde al mismo con una cadena de texto que contiene información sobre el día y la hora del sistema remoto consultado.

Por su lado, el servicio daytime bajo TCP espera la recepción de una petición de conexión TCP al puerto 13, y sin necesidad de esperar la recepción de ningún dato envía una cadena de texto con idéntica información que bajo UDP, cerrando a continuación de forma automática la conexión establecida.

Un ejemplo del formato de la cadena recibida es:

Fri Feb 28 12:14:18 2003

El funcionamiento del servicio daytime bajo UDP es el siguiente:

1. El cliente envía un datagrama vacío al servidor.
2. El servidor recibe el datagrama vacío, a partir del cual obtiene la dirección IP y el puerto UDP del cliente que solicita la información.
3. El servidor realiza una consulta mediante la función `time` de C/C++ y convierte dicha consulta en una cadena de texto, con el formato especificado con anterioridad, mediante una llamada a la función `asctime` de C/C++.
4. El servidor crea un datagrama con la información obtenida en el punto 2, datagrama que tiene como único dato la cadena de texto devuelta por `asctime`.
5. El datagrama llega al cliente, el cual puede leerlo y extraer la información del día y hora del servidor.

Como todo servicio UDP, este es no fiable, por lo cual, tanto el datagrama de petición del servicio del cliente al servidor, como la respuesta del servidor al cliente,

pueden perderse, por lo que el cliente puede no recibir en algunos casos respuesta. Además, el servicio no tiene porque estar implementado en todos los ordenadores, por lo cual, también puede no obtenerse respuesta de aquellos ordenadores que no tengan implementado el servicio.

Por su parte, el funcionamiento bajo TCP es el siguiente:

1. El cliente solicita una conexión al servidor.
2. El servidor acepta la conexión.
3. El servidor realiza una consulta mediante la función `time` de C/C++ y convierte dicha consulta en una cadena de texto, con el formato especificado con anterioridad, mediante una llamada a la función `asctime` de C/C++.
4. El servidor envía la cadena de texto devuelta por `asctime` y cierra la conexión.
5. El cliente recibe la cadena de texto y extrae la información del día y hora del servidor.

Como todo servicio TCP es fiable, los datos siempre llegan del emisor al receptor, o bien se produce un mensaje de error, con lo cual no es necesario tener en cuenta la posibilidad de que los datos no lleguen a su destino. Además, si el servicio no está disponible en un ordenador, el intento de conexión al puerto de daytime de dicho ordenador producirá un error.

Desarrollo de la práctica.

Desarrollar un programa en C/C++ que implemente dos clientes de daytime, uno utilizando el protocolo UDP y otro utilizando TCP. Los clientes a desarrollar, de nombres, por ejemplo, `horaUDP.c` y `horaTCP.c`, deberán responder a las siguientes líneas de comandos:

```
horaUDP <dirección IP del servidor> [timeout]  
horaTCP <dirección IP del servidor>
```

Donde <> indica un parámetro de paso obligatorio y [] un parámetro de paso opcional.

El parámetro opcional `timeout` del cliente UDP servirá para poder incrementar los segundos del `timeout` por defecto que tendrá el programa. Dicho `timeout` por defecto deberá ser, si no se especifica, de 1 segundo.

El análisis básico de desarrollo del programa con el protocolo UDP es:

- 1 Creación de un socket UDP.
- 2 Creación de un datagrama UDP vacío y envío de dicho datagrama al servidor.
- 3 Espera de un tiempo dado por el valor de `timeout` (o el `timeout` por defecto si no se ha especificado) a la recepción del datagrama de respuesta.
- 4 Si dicho datagrama es recibido, se coloca en la pantalla la respuesta obtenida, sino

- se coloca un mensaje indicando que el timeout ha vencido.
- 5 Cierre del socket UDP creado.

El análisis básico de desarrollo del programa con el protocolo TCP es:

- 1 Creación de un socket TCP.
- 2 Conexión del socket creado al puerto daytime del servidor.
- 3 Si la conexión ha sido realizada, se coloca en pantalla la respuesta obtenida, sino se coloca un mensaje indicando que el servicio no esta disponible.
- 4 Cierre del socket TCP creado.

Existen ejemplos compilados de los programas en la página web de la asignatura. Los programas tienen el nombre *horaUDP* y *horaTCP*.

Nota: Debido a que los ordenadores de prácticas tienen configurado el cortafuegos de Linux en un formato muy restrictivo, es posible que los programas, especialmente el de protocolo UDP, no funcione al no recibir nunca la respuesta. Para desactivar el cortafuegos y que funcione correctamente, podéis ejecutar el comando “*service iptables stop*”. Para probar el funcionamiento de la práctica el ordenador lab3inf005 tiene habilitado el servicio de daytime bajo TCP y UDP.