

Ejercicios Obligatorios

Cristina Díaz García

Diciembre 2018



Índice

Índice general	1
1. II. Desarrollo de servicios estándar	2
1.1. Ejercicio 1	2
2. III. Patrones de diseño	3
2.1. Ejercicio 3	3

1. II. Desarrollo de servicios estándar

1.1. Ejercicio 1

1. Protocolo de transferencia de ficheros trivial (TFTP). Implementar un proceso servidor y un proceso cliente que realicen la transferencia de ficheros sobre UDP de acuerdo al protocolo estándar TFTP definido en el RFC 1350. Tener en cuenta los siguientes puntos.

- Usar otro puerto diferente al estándar 69 de TFTP para el servidor.
- La transferencia podrá hacerse en cualquiera de los dos sentidos (get o put). La fase de transferencia de datos y su finalización es igual para ambos procesos cliente y servidor, variando sólo el establecimiento de la comunicación.
- El fichero que se transfiera debe ser textual ya que se mostrará por pantalla (opcionalmente pueden almacenarse los datos transferidos en un fichero).
- El proceso cliente tendrá un pequeño interfaz de usuario para pedir el tipo de la transferencia y el nombre del fichero a transferir. Emular los comandos del cliente tftp estándar (como mínimo: connect, get, put y quit).
- Tanto el cliente como el servidor comprobarán que todos los mensajes que le llegan han sido enviados por el proceso correcto (comprobando el identificador de transferencia que será la IP y el puerto), sino contestará con un mensaje de error. El servidor podrá ser concurrente o iterativo.
- Ambos procesos implementarán los mecanismos para garantizar que la transferencia se haga de forma fiable. Para ello se comprobará que los números de secuencia de los paquetes sean consecutivos y ante la pérdida de un paquete se retransmitirá después de esperar 1 segundo. Poner un número máximo de reintentos (ej: 3 o 5).
- Cada proceso tendrá dos modos funcionamiento: el correcto y con pérdida de paquetes. El modo de funcionamiento con pérdida de paquetes consiste en que durante el inicio y la transferencia de un fichero se podrán perder paquetes (acks o datos) hasta un 5 %. La pérdida de paquetes se hará simulada, cada vez que un proceso cliente o servidor tenga que enviar un mensaje, de forma aleatoria decidirá si lo envía o no. Al finalizar se dará un resumen estadístico con el número/porcentaje de paquetes perdidos y retransmitidos.
- Por simplicidad no se considerará que el primer paquete enviado (READ o WRITE) pueda perderse.

- El proceso cliente (y/o el servidor) dará la traza de la transferencia con el formato que se sigue en el ejemplo:
 - >RRQ “fichero.dat” “octet” <— ACK 0
 - >DATA 1 512 bytes <— ACK 1
 - >DATA 2 10 bytes <— ACK 2 (P)
 - >DATA 2 10 bytes (R) <— ACK 2
- % P: paquetes perdidos; % R: paquetes retransmitidos

2. III. Patrones de diseño

2.1. Ejercicio 3

3. Desarrollar un cliente de smtp simplificado que sea capaz únicamente de enviar mensajes de correo electrónico. El protocolo SMTP está definido en el RFC 821 que se adjunta.

- Esta aplicación debe probarse en la máquina virtual de Linux, con el servidor de correo estándar instalado en dicha máquina.
- De los comandos del protocolo propuestos en el RFC sólo tendrán que implementar: HELO, MAIL, RCPT, DATA, QUIT (implementación mínima, pág. 41).
- La interfaz de usuario será muy sencilla:
 - Compose
 - Introducir una dirección de correo destino (opcionalmente podrían ser varias)
 - Introducir el asunto del mensaje (el *Subject*)
 - Introducir el cuerpo del mensaje (podrá ser una o varias líneas, hasta finalizar con un punto)
 - Quit
 - Cerrar la conexión
- Para comprobar que los mensajes enviados con tu aplicación han llegado correctamente utiliza el cliente de correo estándar de Linux Mail. Un ejemplo sería:

```
[alumno@localhost ~]$ Mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/alumno": 1 message 1 new
>N 1 alumno@localhost.loc  Thu Oct 17 13:57  22/759  "Prueba"
& 1
Message 1:
From: alumno@localhost.localdomain  Thu Oct 17 13:57:30 2013
Date: Thu, 17 Oct 2013 13:57:29 +0200
From: alumno@localhost.localdomain
To: alumno@localhost.localdomain
Subject: Prueba

Estimados alumnos:
espero que la práctica no les resulte difícil.

Saludos,
Lidia
&
```

- Nota: No usar la API javax.mail
- Nota: La dirección de correo de este usuario es nombre__usuarionombre__dominio. Para obtener el nombre del usuario actual, usar los siguientes métodos de Java:
 - nombre_usuario: System.getProperty("user.name")
 - nombre_dominio: InetAddress.getLocalHost().getHostName()

El cliente acepta la conexión en localhost y el puerto 25. Gestiona cuatro instrucciones básicas del protocolo SMTP para poder realizar el servicio: HELO, MAIL FROM:, RCPT TO: y DATA (que acaba cuando se introduce un punto). La información necesaria se introduce con un scanner por pantalla y comienza a enviar información.