

# Arquitectura de Redes y Servicios

## Práctica Tema 7: Cliente TFTP

Jesús Cámara y Diego R. Llanos  
Dpto. de Informática, Universidad de Valladolid

6 de noviembre de 2023

### Índice

<b>1. El protocolo TFTP</b>	<b>1</b>
1.1. Diálogo entre Cliente y Servidor . . . . .	1
1.2. Lectura de Fichero Almacenado en Servidor . . . . .	2
1.3. Escritura de Fichero en Servidor . . . . .	2
<b>2. Práctica a Realizar</b>	<b>3</b>
<b>3. Criterios de Evaluación y Condiciones de Entrega</b>	<b>4</b>

## 1. El protocolo TFTP

El protocolo TFTP (*Trivial File Transfer Protocol*) permite el envío y recepción de ficheros entre dos equipos a través de Internet mediante el uso de datagramas UDP. Ofrece un servicio estándar, por lo que está asociado a un puerto bien conocido (69/udp). Define cinco paquetes diferentes para el intercambio de mensajes entre cliente y servidor. El formato de estos paquetes se muestra en la Figura 1.

TFTP es mucho más sencillo que el servicio FTP, dado que ofrece menos funcionalidad (no muestra el directorio remoto, no hace gestión de usuarios, etc). La principal utilidad de TFTP es el envío de un kernel de arranque a máquinas que no dispongan de unidades de almacenamiento. Estas máquinas tienen un cliente TFTP en memoria ROM y lo utilizan para recibir el kernel de un servidor remoto.

### 1.1. Diálogo entre Cliente y Servidor

El diálogo comienza con el envío, por parte del cliente, de un datagrama con una solicitud de lectura (RRQ) o de escritura (WRQ) de un archivo. Como puede verse en la Figura 1, ambas solicitudes constan de un código de operación, una cadena (acabada en 0) que indica el nombre del archivo a leer/escribir y una cadena que especifica el modo de transmisión, que puede ser `netascii` u `octet`. El modo de transmisión `netascii` garantiza el envío y recepción correctos de ficheros de texto, realizando las conversiones entre los diferentes estándares existentes para, por ejemplo, indicar el final de línea (CR o CR+LF). El modo `octet`, en cambio, transmite de forma literal el fichero. Este modo será el que usaremos en esta práctica.

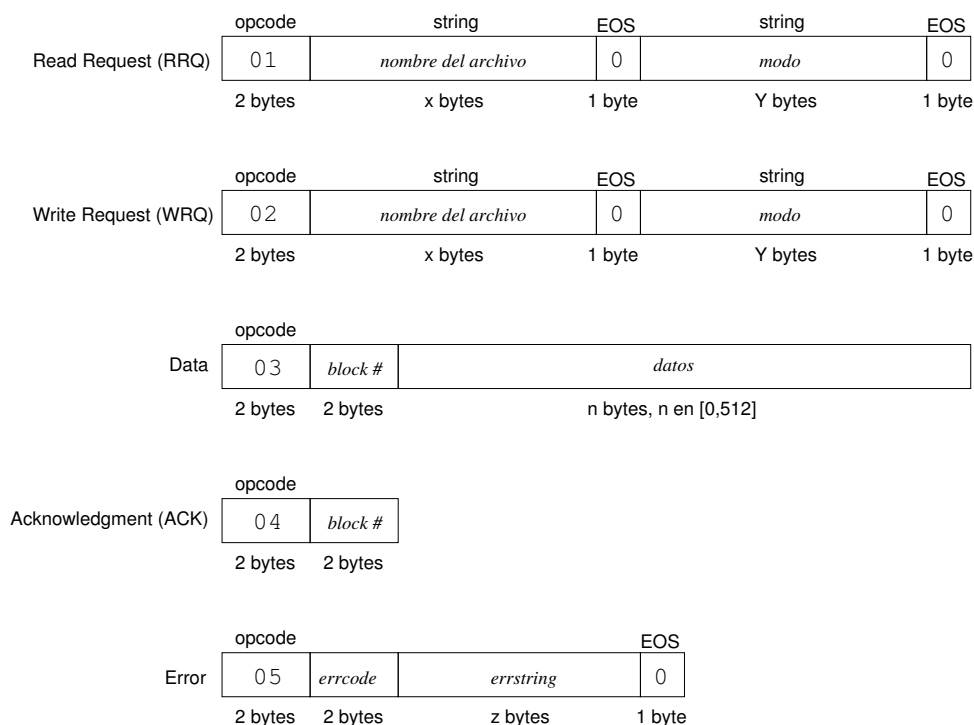


Figura 1: Formato de los paquetes intercambiados en TFTP.

## 1.2. Lectura de Fichero Almacenado en Servidor

La lectura de un fichero almacenado en el servidor se lleva a cabo de la siguiente forma:

1. El cliente solicita la lectura de un fichero mediante envío de un paquete de tipo RRQ.
2. El servidor comprueba que existe un fichero con el nombre indicado y envía el primer bloque de datos mediante un paquete de tipo Data. Este primer datagrama llevará el número de bloque 1 en el campo *block*, que se incrementará en envíos sucesivos. El bloque de datos enviado (en campo *datos*) puede tener una longitud de hasta 512 bytes.
3. El cliente escribe en el fichero el bloque de datos recibido y responde con el envío de un paquete de tipo ACK, indicando en el campo *block* el número de bloque que acaba de recibir.
4. Si el datagrama recibido en el paso 2 contiene un bloque de datos de 512 bytes, el protocolo considera que quedan bloques por recibir → enviar siguiente bloque. En caso contrario (bloque < 512 bytes), se considera que es el último bloque. En este caso, el cliente cierra el fichero almacenado, cierra el socket y finaliza su ejecución.

Ante cualquier problema, el servidor enviará un paquete de tipo Error con un código de error y una descripción del mismo. Los códigos de error disponibles se muestran en la Tabla 1.

## 1.3. Escritura de Fichero en Servidor

Este proceso no es simétrico con respecto a la operación de lectura, por lo que no consiste simplemente en usar el código desarrollado para la lectura de un fichero cambiando los “opcode”

<i>errcode</i>	<b>Descripción</b>
0	No definido (comprobar <i>errstring</i> )
1	Fichero no encontrado.
2	Violación de acceso.
3	Espacio de almacenamiento lleno.
4	Operación TFTP ilegal.
5	Identificador de transferencia desconocido.
6	El fichero ya existe.
7	Usuario desconocido.

Cuadro 1: Códigos de error devueltos por servidor TFTP.

La escritura de un fichero en el servidor se realiza del siguiente modo:

1. El cliente solicita la escritura de un fichero mediante envío de un paquete de tipo WRQ.
2. El servidor comprueba si la operación de escritura está permitida y, en tal caso, envía al cliente un paquete de tipo ACK con número de bloque 0.
3. El cliente envía el primer bloque de datos (hasta 512 bytes), comenzando con número de bloque 1.
4. Cada vez que el servidor recibe un paquete de tipo Data con número de bloque  $x$ , escribe el bloque de datos en el fichero y le responde al cliente con un paquete de tipo ACK con número de bloque  $x$ .
5. Cuando el cliente recibe el ACK del bloque  $x$ , envía el bloque  $x + 1$ , y así sucesivamente hasta que el servidor recibe el último bloque de datos (tamaño  $< 512$  bytes), momento en que almacena en disco el bloque recibido, cierra el fichero y envía al cliente el último ACK.

**Nota:** para almacenar el bloque de datos debe usarse la función `fwrite()`, que permite escribir un número determinado de bytes en disco. No debe usarse `fprintf()`, pues esta función sirve para almacenar cadenas y puede ser que el bloque recibido contenga ceros que provoquen que se almacenen menos bytes de los esperados.

## 2. Práctica a Realizar

La práctica consiste en desarrollar un Cliente UDP que permita hacer uso del servicio TFTP. El cliente se implementará en el fichero `tftp-client-apellidos.c` utilizando el lenguaje C y se iniciará de la siguiente forma desde la terminal de comandos:

```
tftp-client ip_servidor {-r|-w} archivo [-v]
```

Donde:

- `ip_servidor`: dirección IP del servidor TFTP.
- `{-r|-w}`: indica que se desea leer o escribir un fichero del servidor.
- `archivo`: nombre del fichero.
- `-v` (opcional): cliente informa de los paquetes enviados para la lectura/escritura del fichero.

Ejemplo:

```
$> ./tftp-client 127.0.0.1 -r datos-justos.dat -v
Enviada solicitud de lectura de "datos-justos.dat" a servidor tftp (127.0.0.1)
Recibido bloque del servidor tftp.
Es el primer bloque (numero de bloque 1)
Enviamos el ACK del bloque 1.
```

Recibido bloque del servidor tftp.  
Es el bloque numero 2.  
Enviamos el ACK del bloque 2.  
El bloque 2 era el ultimo: cerramos el fichero.

Dado que la red que conecta las máquinas virtuales no pierde datagramas, no es necesario implementar mecanismos de recuperación ante errores en la transmisión, pero hay que **comprobar que los paquetes se reciben en el orden correcto**.

En la máquina virtual del profesor (con IP: 10.0.25.250) hay tres ficheros que se pueden utilizar para probar el cliente desarrollado:

- `datos-cortos.dat`, de 100 bytes.
- `datos-justos.dat`, de 512 bytes.
- `datos-largos.dat`, de 12 345 678 bytes.

Para comprobar que la implementación de la operación de escritura funciona correctamente hay que (a) hacer una copia de un fichero, (b) enviar esa copia, (c) eliminar la copia local, (d) traer el fichero de nuevo y (e) ejecutar el comando `diff` para comparar la copia que acaba de llegar con el fichero original.

### 3. Criterios de Evaluación y Condiciones de Entrega

1. La práctica debe realizarse en la máquina virtual asignada en Matrix.
2. El fichero de código del cliente TFTP debe comenzar con un comentario con el siguiente formato:  
  

```
// Practica Tema 7: Apellido1 Apellido2, Nombre
```
3. El código desarrollado debe estar correctamente documentado, incluyendo comentarios que indiquen cómo se ha implementado la funcionalidad del cliente. De lo contrario, se penalizará con dos puntos la calificación obtenida.
4. El cliente debe compilar sin mostrar advertencias (*warnings*) ni errores. De lo contrario, se penalizará con tres puntos la calificación obtenida.
5. Esta práctica representa un 35 % en la calificación de las prácticas de la asignatura.
6. Cuando esté finalizada, se subirá a la tarea habilitada en el Campus Virtual el fichero de código fuente (sin comprimir) con el siguiente nombre: `tftp-client-apellidos.c`. Un fallo en las condiciones de entrega supondrá un punto menos en la calificación.
7. Se utilizará un sistema automático de detección de copias. En caso de copia, los alumnos involucrados figurarán como suspensos en la convocatoria ordinaria, debiendo enviar todas las prácticas por correo electrónico al profesor para poder presentarse a la convocatoria extraordinaria. En esta situación, todas las prácticas se corregirán sobre 7.
8. Fecha de Entrega: **26 de noviembre de 2023 a las 23:55**.
9. No se admitirán entregas fuera de plazo.