Capítulo 1

Transferencia de archivos entre máquinas con ftp

«**ftp**» es un protocolo de transferencia de archivos. El mismo nombre se le da al programa cliente que permite intercambiar archivos bajo el control de un usuario.

Se inicia «contra» una máquina que dispone de tal servicio, y se hace de la siguiente forma:

sintaxis

`ftp␣[`*opciones*`]␣`*host*	lanza un cliente ftp contra un host concreto

Este nombre de host puede ser remoto o incluso local, lo cual es útil en algunas ocasiones.

Ojo: Cada servidor tiene un límite de usuarios y en él se especifica qué usuarios pueden acceder y a qué archivos.

Otra posibilidad es invocar directamente el mandato

sintaxis

`ftp`	lanza un cliente ftp sin conectar

Tras lo cual tendremos que iniciar la conexión remota por nosotros mismos:

sintaxis

`open␣`*host*	intenta conectar con un host
`user␣`*usuario*`␣[`*passwd*`]`	indica el usuario y el password de la conexión

En caso de no incluir el password, se nos solicitará desde el servidor.

1.1 Usuarios

ftp solicitará una identificación (nombre de usuario + clave de acceso). Algunos servidores disponen de un usuario especial que permite acceder a archivos de uso público bajo el nombre de usuario: «anonymous», solicitando a continuación la dirección de correo electrónico del que se conecta: En ciertas máquinas no es necesario que este sea una dirección de correo real, pero en otras lo contrastan con direcciones reales.

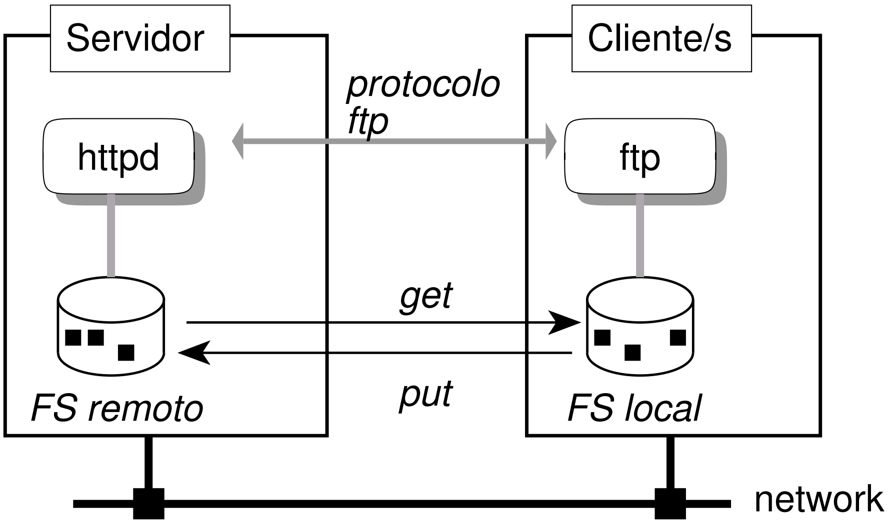

Figura 1.1: diagrama cliente servidor con ftp

ejemplo

`ftp ftp.rediris.es`	
`login: anonymous`	ftp.rediris.es admite conexiones anónimas
`(e-mail):`	el password se teclea a ciegas, como por ejemplo: pedro@infor.uva.es

1.2 Algunos mandatos útiles

Una vez se accede al servidor se dispone de un completo juego de mandatos que depende tanto de las funciones operativas del servidor remoto de ftp, como de las funciones que tiene implementadas el cliente ftp que estemos utilizando.

Para conocer las funcionalidades se utiliza:

sintaxis

`help`	muestra las palabras clave disponibles
`help␣`*mandato*	ayuda en un mandato concreto

Una de las funciones primordiales es poder recorrer el árbol de carpetas remoto, para lo cual se utilizan habitualmente los siguientes mandatos

sintaxis

`cd`	para conocer la posición actual
`cd␣`*carpeta*	para acceder a una carpeta concreta (admite ..)
`dir` o `ls`	para conocer los contenidos de la carpeta

ejemplo

`cd␣pub/software`	cambia de carpeta
`dir`	inspecciona la carpeta

Incluso, si se dispone de los permisos adecuados se pueden utilizar los mandatos siguientes:

sintaxis

`mkdir␣`*nombre*	crear directorios remotos
`rmdir␣`*nombre*	borrado de directorios remotos
`delete␣`*nombre*	borrar 1 archivo remoto
`mdelete␣`*reg-exp*	borrar varios archivos remotos (m: multiple)
`rename␣`*n_ ant*`␣`*n_ nuevo*	cambiar el nombre de un archivo

En el sistema de archivos local también se puede realizar operaciones como

sintaxis

`lcd␣`*carpeta*	establece *carpeta* en el entorno local

el resto de los mandatos que pueden afectar al sistema de archivos locales, se pueden efectuar mediante la siguiente sintaxis:

sintaxis

`!␣`*mandato local*	ejecuta localmente un mandato

1.3 Transferencia de archivos

El el momento de realizar la transferencia se establece un canal de comunicación (en realidad dos). Este canal puede configurarse para transmitir archivos de texto o archivos binarios.

En el caso de los archivos binarios, éstos se transfieren tal cual byte a byte. Y en el caso de la transferencia de archivos de texto, y en función de la representación de los archivos de texto en las máquinas local y remota se suele modificar el archivo en la medida correspondiente para que sea legible en el destino.

sintaxis

`binary` o `image`	para transferencias binarias
`ascii` o `text`	para transferencias de texto

La operación de transferencia puede ser de depósito en la máquina remota o de traida de datos. En el primer caso (entre otros mandatos), se utiliza:

sintaxis

`put␣`*nombre*	deposita el archivo *nombre*
`mput␣`*reg-exp*	deposita varios archivos concordantes con la expresión regular *reg-exp*

ejemplo

`put␣readme.txt`	deposita el archivo `readme.txt`
`mput␣*.c`	deposita los archivos que terminan en `.c`

Para la descarga o traida de archivos a la máquina local se suele utilizar

sintaxis

`get␣`*nombre*	descarga el archivo *nombre*
`mget␣`*reg-exp*	descarga los archivos concordantes con la expresión regular *reg-exp*

ejemplo

`get␣index.html`	descarga el archivo remoto `index.html`
`mget␣Ab*.gif`	descarga los archivos que comienzan con `Ab` y terminan con `.gif`

1.4 Despedida de la sesión

Es una buena costumbre despedir la conexión con la máquina remota. Para ello se suele utilizar

sintaxis

`bye` o `<^D>`	despedida y cierre del programa cliente
`close`	despedida canal

En este último caso es necesario también cerrar el programa cliente ftp si es que deseamos finalizar las operaciones.

En todo caso, tras un tiempo prolongado (que se establece en la máquina servidora) sin utilizar ningun mandato se produce una desconexión automática del servidor, con lo que se libera el canal para otros posibles clientes.

1.4.1 Ejercicios

- Realice alguna conexión remota anónima con alguna máquina remota como
  - `ftp.rediris.es`
  - `ftp.dante.de`
  - `luna.gui.uva.es`
  - `ftp.uva.es`
- Inspeccione la estructura habitual y vea que algunos de los directorios que se pueden ver no tienen ninguna utilidad concreta. Suelen estar preparados solamente para la descarga de software y datos de utilidad.
- Realice alguna conexión remota en alguna máquina con algun usuario de una máquina, remota o local (por ejemplo: sobre la cuenta del compañero de prácticas):
  - `ftp␣duero.lab.fi.uva.es`
  - `ftp␣l104b1.lab.fi.uva.es`

  Puede utilizar esta conexión para compartir datos entre diversas máquinas en las que tenga acceso, e incluso descargar y cargar programas y datos en cuentas de usuario de compañeros suyos.