Quando il trasferimento di file riguarda un ambito che supera l'estensione di una piccola rete locale, non è conveniente consentire l'utilizzo della condivisione del file system (NFS) o della copia remota. A questo scopo si prestano meglio altri protocolli; storicamente, il più importante è stato il protocollo FTP (*File transfer protocol*). Oggi è però superato, oltre che essere un protocollo problematico per la configurazione dei filtri TCP/IP e dei router NAT. In altri termini: il protocollo FTP è importante e occorre conoscerne le caratteristiche; tuttavia è meglio evitare di predisporre servizi basati su FTP, se si possono utilizzare delle alternative migliori.

Il servizio FTP viene offerto da un demone che funge da servente e viene utilizzato da un programma cliente in grado di comunicare attraverso il protocollo FTP. Il funzionamento di un programma cliente tradizionale è paragonabile a quello di una shell specifica per la copia di file da e verso un sistema remoto.

38.1 Caratteristiche elementari del protocollo

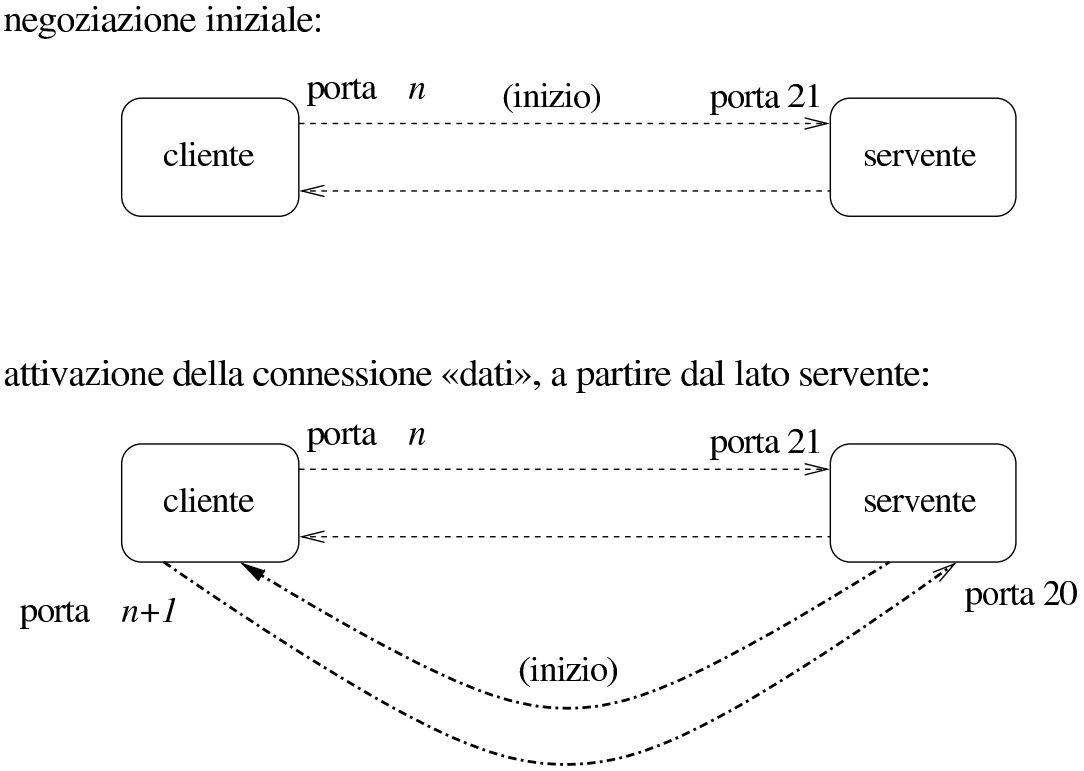
«

In generale, il protocollo FTP si avvale di TCP al livello inferiore, utilizzando precisamente due connessioni TCP per ogni sessione del protocollo FTP. Ciò costituisce un problema molto importante quando si deve controllare in qualche modo il traffico relativo al protocollo FTP, pertanto occorre conoscere come si sviluppa questa connessione. Infatti si distinguono due modalità di utilizzo del protocollo FTP: attiva e passiva. In entrambi i casi, il servente FTP è inizialmente in ascolto della porta 21.

Quando il protocollo FTP viene utilizzato in modalità attiva, il programma cliente apre una porta locale, n, non privilegiata ($n > 1024$), iniziando una connessione TCP con la porta 21 dell'elaboratore che contiene il servente FTP. Nell'ambito di questa connessione vengono inviati dal programma cliente dei comandi al programma servente. Per consentire lo scambio di dati, deve essere aperta una seconda connessione TCP tra i due programmi; per questo il programma cliente apre una seconda porta locale, la quale dovrebbe corrispondere a $n+1$, informando di questo il programma servente attraverso la connessione già attiva. A questo punto, **il programma servente inizia la seconda connessione TCP** utilizzando la propria porta 20, contattando presso l'elaboratore del programma cliente la porta $n+1$ (o qualunque altra porta comunicata dal

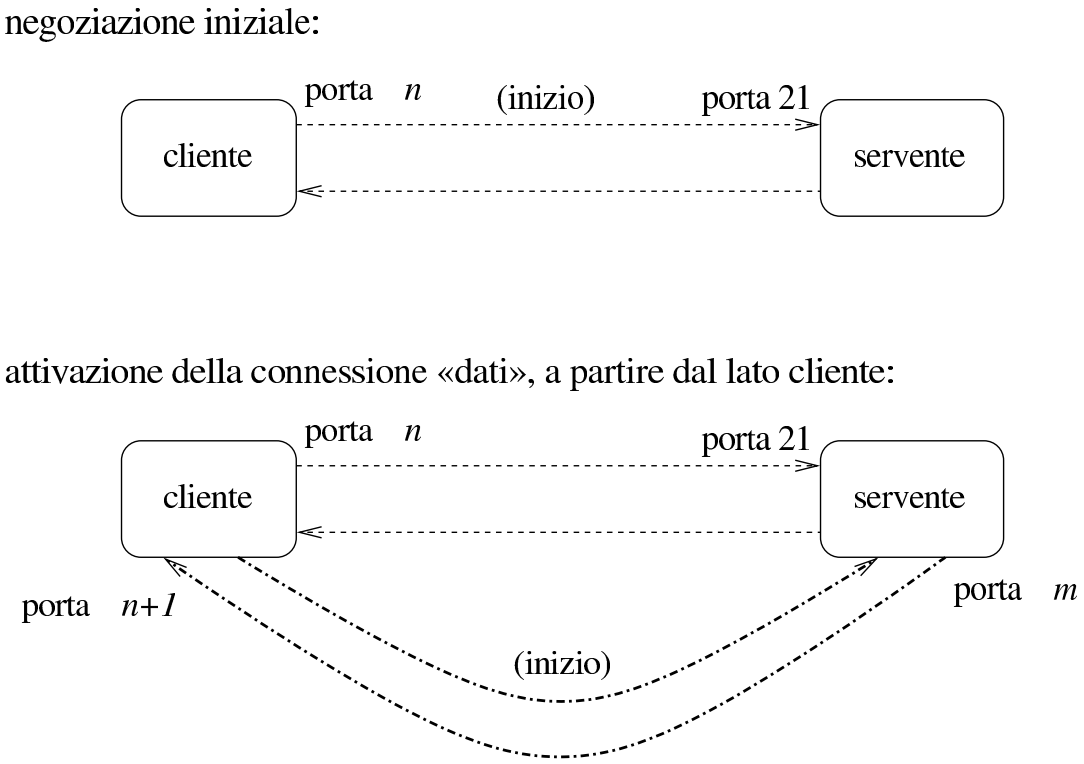
programma cliente).

Figura 38.1. Fasi di una sessione FTP attiva.



Quando il protocollo FTP viene utilizzato in modalità passiva, il programma cliente si comporta inizialmente come nel caso della modalità attiva, iniziando una connessione TCP con la porta 21 dell'elaboratore che contiene il servente FTP. Questa volta, però, chiede al programma servente di operare in modalità «passiva». Così facendo, è il programma servente che apre una porta non privilegiata e comunica al programma cliente il valore di questa, in modo che sia sempre il programma cliente a iniziare tale connessione TCP.

Figura 38.2. Fasi di una sessione FTP passiva.



Quando in una rete si attuano delle tecniche di trasformazione degli indirizzi e delle porte, oppure si intende filtrare il traffico, il controllo del protocollo FTP diventa un problema, proprio a causa dell'apertura di questa connessione secondaria: dal lato servente è più comodo usare la modalità attiva, mentre dal lato cliente è più conveniente la modalità passiva. Purtroppo, nessuna delle due situazioni è equilibrata ed è questo il limite del protocollo FTP.

Come si può intuire, è il programma cliente che chiede alla controparte di utilizzare una o l'altra modalità. Esistono programmi clienti che in modo predefinito utilizzano la modalità attiva, mentre altri che fanno il contrario; di solito i programmi più recenti sono impostati in modo da usare la modalità passiva se non si specifica diversamente con la configurazione.

38.2 Identificazione e privilegi

Il sistema di trasferimento di file attraverso FTP richiede una forma di autenticazione, in base alla quale il servente può dare privilegi differenti agli utenti.

Generalmente, perché un utente registrato venga accettato per una sessione FTP è necessario che presso il servente abbia una parola d'ordine (non sono quindi ammessi utenti senza parole d'ordine) e una shell valida, cioè compresa nell'elenco del file '`/etc/shells`'. Questo ultimo particolare non è trascurabile, infatti, a volte si sospende l'utilizzo di un'utenza modificando il campo della shell nel file '`/etc/passwd`' con qualcosa di non valido.

Oltre a queste limitazioni, si utilizza solitamente il file '`/etc/ftpusers`' per determinare quali utenti **non** possono essere accettati per una sessione di FTP normale. In questo elenco vanno messi in particolare gli utenti di sistema, come per esempio '**`root`**', '**`bin`**' e '**`mail`**'.

Se si vuole permettere l'accesso a utenti che non sono registrati nel proprio sistema (si parla di utenti che non sono previsti nel file '`/etc/passwd`'), è possibile abilitare l'utilizzo dell'FTP anonimo. Per questo è necessario che sia stato previsto un utente speciale nel file '`/etc/passwd`': '**`ftp`**'.[1]

```
...
ftp:*:101:101::/var/ftp:/bin/false
...
```

A questo utente non viene abbinata alcuna parola d'ordine valida e nemmeno una shell utilizzabile.

Per utilizzare un servizio FTP in modo anonimo si può accedere identificandosi come '**ftp**', oppure '**anonymous**'. Di norma, viene richiesta ugualmente una parola d'ordine che però non viene (e non può essere) controllata: per convenzione si inserisce l'indirizzo di posta elettronica.[2]

Generalmente, un servente FTP che consente l'accesso anonimo, fa sì che tali utenti non identificati possano accedere solo alla directory personale dell'utente fittizio '**ftp**', senza poter esplorare il resto del file system.

38.3 Facilitare le ricerche

«

Il modo più semplice di fornire un indice del contenuto del proprio servizio FTP anonimo è quello di posizionare nella sua directory di partenza un cosiddetto file '`ls-lR`'. Si tratta in pratica del risultato dell'esecuzione del comando '**ls -lR**', che ha quindi suggerito il nome del file indice in questione. Generalmente si comprime questo file con '**gzip**', per cui si usa il nome '`ls-lR.gz`'.

Il comando per generare questo file deve essere eseguito quando la directory corrente è quella di partenza del servizio; in pratica, agendo nel modo seguente:

```
# cd ~ftp [Invio]

# ls -lR | gzip -9 > ls-lR.gz [Invio]
```

38.4 Cliente FTP tradizionale

Il programma cliente tradizionale per accedere a un servizio FTP, è quello originario dei sistemi BSD, del quale esistono comunque diverse varianti.[3] In generale, si tratta semplicemente del programma '**ftp**':

```
ftp [opzioni] [nodo]
```

Quando l'eseguibile '**ftp**' viene avviato con l'indicazione del nome dell'elaboratore remoto, tenta immediatamente di effettuare il collegamento; diversamente si avvia e attende il comando con il quale questo elaboratore deve essere poi specificato. Se esiste il file '`~/.netrc`', questo viene utilizzato per automatizzare l'accesso nell'elaboratore remoto. Quando '**ftp**' è in attesa di un comando da parte dell'utente, presenta l'invito seguente: '**ftp>**'.

Tabella 38.4. Alcune opzioni della riga di comando.

Opzione	Significato mnemonico	Descrizione
`-v`	*verbose*	Vengono visualizzati tutti i messaggi.
`-n`	*no auto*	Disabilita l'accesso automatico.
`-i`	*interactive*	Disattiva la richiesta interattiva durante i trasferimenti multipli di file.
`-d`	*debugging*	Attiva la modalità diagnostica.
`-p`	*passive*	Utilizza la modalità di funzionamento passiva.

Opzione	Significato mnemonico	Descrizione
`-g`	*globbing*	Disabilita l'uso dei metacaratteri (caratteri jolly) per l'indicazione di gruppi di file.

Come già accennato, quando '**`ftp`**' è in attesa di un comando da parte dell'utente, presenta l'invito '**`ftp>`**'. La tabella che segue elenca alcuni dei comandi che possono essere utilizzati. Se i parametri dei comandi contengono il carattere spazio, questi devono essere delimitati da una coppia di apici doppi ('`"`').

Alcuni comandi di maggiore utilità.

Comando	Descrizione
`get` ***file_remoto*** [***file_locale***] `recv` ***file_remoto*** [***file_locale***]	'**`get`**' e '**`recv`**' sono sinonimi. Riceve il file remoto indicato, eventualmente rinominandolo come indicato.
`mget` ***file_remoti***	Esegue un '**`get`**' multiplo, cioè su tutti i file che si ottengono dall'espansione del nome indicato utilizzando i metacaratteri (caratteri jolly).
`put` ***file_locale*** [***file_remoto***] `send` ***file_locale*** [***file_remoto***]	'**`put`**' e '**`send`**' sono sinonimi. Copia il file specificato nel sistema remoto eventualmente rinominandolo come indicato.
`mput` ***file_locali***	Espande il nome indicato se contiene dei metacaratteri ed esegue un '**`put`**' per tutti questi file, trasmettendoli in sostanza nel sistema remoto.

Comando	Descrizione
`reget` ***file_remoto*** [***file_locale***]	Permette di riprendere il '**`get`**' di un file remoto quando l'operazione precedente è stata interrotta involontariamente. L'operazione non è sicura e si basa solo sul calcolo della dimensione del file locale per determinare la parte mancante ancora da trasferire.
[*Ctrl c*]	L'operazione di trasferimento può essere interrotta utilizzando la combinazione [*Ctrl c*].
`passive`	Richiede di utilizzare la modalità «passiva» per il protocollo FTP.
`binary`	Imposta il tipo di trasferimento in modalità binaria. Questa modalità è adatta al trasferimento di qualunque tipo i file.
`type` [***tipo_di_trasferimento***]	Attiva o visualizza il tipo di trasferimento dei dati. Il valore predefinito è '**`ascii`**'. I tipi a disposizione sono: '**`ascii`**', '**`ebcdic`**', '**`image`**' (trasferimento binario), '**`local byte size`**'.
`prompt`	Attiva o disattiva la modalità di conferma. Se è attiva, durante le operazioni di trasferimento di gruppi di file, viene richiesta la conferma per ogni file.

Comando	Descrizione

`bye`
 	
`quit`
	'**`bye`**' e '**`quit`**' sono sinonimi. Termina il collegamento e termina l'attività di '**`ftp`**'.

`close`
 	
`disconnect`
	Termina la connessione senza uscire dal programma.

`open` ***nodo*** [***porta***]	Apre una connessione con l'elaboratore remoto indicato ed eventualmente anche specificando la porta di comunicazione. Se la modalità di accesso automatico è attiva, '**`ftp`**' tenta anche di effettuare l'accesso nel sistema remoto.
`cd` [***directory_remota***]	Cambia la directory corrente nel sistema remoto.
`chmod` ***permessi file_remoto***	Cambia i permessi sul file remoto.
`delete` ***file_remoto***	Cancella il file indicato nel sistema remoto.

Comando	Descrizione
`dir` [*directory_remota*] ↩ ↪ [*file_locale*] `ls` [*directory_remota*] ↩ ↪ [*file_locale*] `nlist` [*directory_remota*] ↩ ↪ [*file_locale*]	'**`dir`**', '**`ls`**', '**`nlist`**' sono sinonimi. Elencano il contenuto della directory remota specificata, oppure di quella attuale se non viene indicata. L'elenco viene emesso attraverso lo standard output, quando non viene specificato il file locale all'interno del quale si vuole immettere questo elenco. L'aspetto dell'elenco dipende dal sistema con il quale si sta comunicando. Di solito è molto simile a quello di un '**`ls -l`**'.
`mdelete` [*file_remoti*]	Cancella i file remoti espandendo i metacaratteri prima di procedere.
`mkdir` *directory_remota*	Crea una directory nel sistema remoto.
`pwd`	Visualizza il nome della directory corrente del sistema remoto.
`rename` *origine destinazione*	Permette di cambiare il nome di un file nel sistema remoto.
`rmdir` *directory_remota*	Cancella una directory nel sistema remoto.
`status`	Visualizza lo stato attuale del sistema remoto.
`help` [*comando*] `?` [*comando*]	'**`help`**' e '**`?`**' sono sinonimi. Visualizza una breve guida dei comandi.
`remotehelp` [*comando*]	Permette di richiedere la guida dei comandi al sistema remoto.

38.4.1 Esempi

«

L'uso di un cliente FTP può essere anche semplice, se si lasciano da parte raffinatezze non indispensabili. Seguono alcuni esempi di sessioni FTP.

38.4.1.1 Prelievo di file

«

```
daniele@roggen:~$ ftp dinkel.brot.dg[Invio]
```

Si richiede la connessione FTP all'elaboratore *dinkel.brot.dg*.

```
Connected to dinkel.brot.dg.
220 dinkel.brot.dg FTP server (Version wu-2.4.2-academ[BETA-12]) ready.
Name (roggen.brot.dg:daniele):
```

anonymous [*Invio*]

Si utilizza una connessione anonima e per correttezza si utilizza il proprio indirizzo di posta elettronica abbreviato al posto della parola d'ordine.

```
331 Guest login ok, send your complete e-mail address as
password.
Password:
```

daniele@ [*Invio*]

```
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using ascii mode to transfer files.
```

Come si vede, la modalità di trasferimento predefinita è ASCII (almeno così succede di solito). Generalmente si deve utilizzare una modalità binaria. Questa viene selezionata tra un po'; per ora si richiede la guida interna dei comandi a disposizione:

```
ftp> help[Invio]

  Commands may be abbreviated.  Commands are:

  !           debug       mdir        sendport    site
  $           dir         mget        put         size
  account     disconnect  mkdir       pwd         status
  append      exit        mls         quit        struct
  ascii       form        mode        quote       system
  bell        get         modtime     recv        sunique
  binary      glob        mput        reget       tenex
  bye         hash        newer       rstatus     tick
  case        help        nmap        rhelp       trace
  cd          idle        nlist       rename      type
  cdup        image       ntrans      reset       user
  chmod       lcd         open        restart     umask
  close       ls          prompt      rmdir       verbose
  cr          macdef      passive     runique     ?
  delete      mdelete     proxy       send
```

```
ftp> binary[Invio]
```

Come accennato, viene richiesto di passare alla modalità di trasferimento binario.

```
  200 Type set to I.
```

```
ftp> prompt[Invio]
```

Anche la modalità interattiva viene disattivata per evitare inutili richieste.

```
  Interactive mode off.
```

La struttura delle directory di un normale servizio FTP anonimo prevede la presenza della directory ‘`pub/`’ dalla quale discendono i dati

accessibili all'utente sconosciuto.

Anche se dal punto di vista del cliente FTP, che accede al servizio remoto, si tratta della prima directory dopo la radice, in realtà questa radice è solo la directory iniziale del servizio FTP anonimo. Di conseguenza, è quasi impossibile che corrisponda realmente con la directory radice del file system remoto. Tutto questo serve solo a spiegare perché il comando '**cd /pub**' potrebbe non funzionare quando ci si collega a serventi configurati male. Ecco perché nell'esempio che segue non si utilizza la barra obliqua davanti a '**pub**'.

```
ftp> cd pub[Invio]

  250 CWD command successful.
ftp> pwd[Invio]

  257 "/pub" is current directory.
ftp> ls[Invio]

  200 PORT command successful.
  150 Opening ASCII mode data connection for /bin/ls.
  total 4
  dr-xr-sr-x   3 root     ftp        1024 Nov 12 21:04 .
  drwxr-xr-x   6 root     root       1024 Sep 11 20:31 ..
  -rw-r--r--   1 root     ftp          37 Nov 12 21:04 esempio
  drwxrwsrwx   2 root     ftp        1024 Nov  2 14:04 incoming
  226 Transfer complete.
```

Attraverso il comando '**ls**' si vede che la directory '`pub/`' contiene solo il file '`esempio`' e la directory '`incoming/`'. Si decide di

prelevare il file.

```
ftp> get esempio[Invio]

  local: esempio remote: esempio
  200 PORT command successful.
  150 Opening BINARY mode data connection for esempio (37 bytes).
  226 Transfer complete.
  37 bytes received in 0.00155 secs (23 Kbytes/sec)
```

Il file scaricato viene messo nella directory in cui si trovava l'utente quando avviava il programma '**ftp**'.

```
ftp> quit[Invio]

  221 Goodbye.
```

38.4.1.2 Invio di dati

```
daniele@roggen:~$ ftp dinkel.brot.dg[Invio]
```

Si richiede la connessione FTP all'elaboratore *dinkel.brot.dg* e si danno dei comandi per raggiungere la directory '`pub/incoming`'.

```
  Connected to dinkel.brot.dg.
  220 dinkel.brot.dg FTP server ↩
↪(Version wu-2.4.2-academ[BETA-12](1) ↩
↪Wed Mar 5 12:37:21 EST 1997) ready.
  Name (dinkel.brot.dg:daniele):
```

anonymous[*Invio*]

```
  331 Guest login ok, send your complete e-mail address as
  password.
  Password:
```

daniele@[*Invio*]

```
  230 Guest login ok, access restrictions apply.
  Remote system type is UNIX.
  Using ascii mode to transfer files.
ftp> binary[Invio]

  200 Type set to I.
ftp> prompt[Invio]

  Interactive mode off.
ftp> cd pub/incoming[Invio]

  250 CWD command successful.
ftp> pwd[Invio]
```

Si verifica la posizione in cui ci si trova.

```
  257 "/pub/incoming" is current directory.
ftp> mput al-1*[Invio]
```

Dal momento che la directory è giusta, si inizia la trasmissione di tutti i file che nella directory locale corrente iniziano per '**al-1**'.

```
  local: al-1 remote: al-1
  200 PORT command successful.
  150 Opening BINARY mode data connection for al-1.
  226 Transfer complete.
  2611649 bytes sent in 1.38 secs (1.9e+03 Kbytes/sec)
  local: al-15 remote: al-15
  200 PORT command successful.
  150 Opening BINARY mode data connection for al-15.
  226 Transfer complete.
  2612414 bytes sent in 2.51 secs (1e+03 Kbytes/sec)
```

```
local: al-16 remote: al-16
200 PORT command successful.
150 Opening BINARY mode data connection for al-16.
226 Transfer complete.
2612414 bytes sent in 2.16 secs (1.2e+03 Kbytes/sec)
local: al-17 remote: al-17
200 PORT command successful.
150 Opening BINARY mode data connection for al-17.
226 Transfer complete.
2612420 bytes sent in 2.17 secs (1.2e+03 Kbytes/sec)
local: al-18 remote: al-18
200 PORT command successful.
150 Opening BINARY mode data connection for al-18.
226 Transfer complete.
2612409 bytes sent in 2.4 secs (1.1e+03 Kbytes/sec)
local: al-19 remote: al-19
200 PORT command successful.
150 Opening BINARY mode data connection for al-19.
226 Transfer complete.
2612431 bytes sent in 2.35 secs (1.1e+03 Kbytes/sec)
```

`ftp>` **`ls`**[*Invio*]

Si controlla il risultato nell'elaboratore remoto. A volte, i servizi FTP impediscono la lettura del contenuto di questa directory.

```
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 15379
drwxrwsrwx   2 root     ftp          1024 Dec 11 20:40 .
dr-xr-sr-x   3 root     ftp          1024 Nov 12 21:04 ..
-rw-rw-r--   1 ftp      ftp       2611649 Dec 11 20:40 al-1
-rw-rw-r--   1 ftp      ftp       2612414 Dec 11 20:40 al-15
-rw-rw-r--   1 ftp      ftp       2612414 Dec 11 20:40 al-16
-rw-rw-r--   1 ftp      ftp       2612420 Dec 11 20:40 al-17
-rw-rw-r--   1 ftp      ftp       2612409 Dec 11 20:40 al-18
-rw-rw-r--   1 ftp      ftp       2612431 Dec 11 20:40 al-19
226 Transfer complete.
```

ftp> **quit** [*Invio*]

```
221 Goodbye.
```

38.4.2 Midnight Commander

«

Midnight Commander (a cui corrisponde l'eseguibile '**mc**') è un programma che offre le funzionalità di un gestore di file abbastanza completo, includendo la capacità di utilizzare il protocollo FTP. Con Midnight Commander è sufficiente utilizzare il comando '**cd**' in modo appropriato per accedere a un servizio FTP remoto:

$ **cd ftp://tizio@dinkel.brot.dg** [*Invio*]

In questo caso si accede al servizio FTP dell'elaboratore *dinkel.brot.dg* con il nominativo utente '**tizio**'. Trattandosi di un accesso che non è anonimo, prima di iniziare, Midnight Commander chiede l'inserimento della parola d'ordine.

La configurazione predefinita di Midnight Commander prevede l'uso della modalità passiva, ma se lo si vuole si può ripristinare l'uso

della modalità attiva intervenendo attraverso la voce *Virtual FS* del menù *Options*.

Figura 38.27. La maschera di modifica della configurazione relativa alle funzionalità FTP di Midnight Commander. Si può osservare che in questo caso è previsto il funzionamento in modalità passiva.

```
.----------- Virtual File System Setting ------------.
|                                                     |
|  Timeout for freeing VFSs:          [60      ] sec  |
|                                                     |
|  ftp anonymous password:                            |
|  [tizio@                                       ]    |
|  ftpfs directory cache timeout:     [1800    ] sec  |
|  [ ] Always use ftp proxy                           |
|  [gate                                         ]    |
|  [x] Use ~/.netrc                                   |
|  [x] Use passive mode                               |
|                                                     |
|          [< OK >]           [ Cancel ]              |
`-----------------------------------------------------'
```

Midnight Commander è descritto nella sezione 22.16.

38.5 Servente OpenBSD FTP

Il servente OpenBSD FTP[4] è un programma molto semplice da installare e configurare, anche in un sistema GNU. Come altri serventi FTP mette a disposizione l'eseguibile '**`in.ftpd`**' (o '**`ftpd`**', a seconda della distribuzione). Questo demone può funzionare in modo autonomo, oppure sotto il controllo del supervisore dei servizi di rete. Nel primo caso si avvia con l'opzione '**`-D`**', mentre nel secondo si usa l'opzione '**`-q`**'.

In generale, l'opzione '**-q**' sta per *quiet*, nel senso di non inviare informazioni al programma cliente sulla versione del servente. L'opzione '**-q**' dovrebbe andare bene anche quando si avvia il programma in modo indipendente dal supervisore dei servizi di rete; in ogni caso, dalle prove eseguite, quando è sotto il controllo del supervisore dei servizi di rete sembrerebbe che senza l'opzione '**-q**' il programma non possa funzionare.

```
in.ftpd -D [opzioni]
```

```
in.ftpd -q [opzioni]
```

Nell'esempio seguente viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd -q
```

Tabella 38.29. Alcune opzioni della riga di comando.

Opzione	Significato mnemonico	Descrizione
`-d`	*debugging*	Vengono aggiunte informazioni diagnostiche all'interno del registro di sistema.
`-l`	*log*	Ogni sessione FTP viene annotata all'interno del registro di sistema; se viene usata due volte, le indicazioni sono più dettagliate.

Opzione	Significato mnemonico	Descrizione
`-t` ***n***	*timeout*	Permette di specificare la durata espressa in secondi (***n***) del tempo di inattività oltre il quale la sessione FTP viene conclusa automaticamente. Questo parametro è negoziabile anche da parte del cliente. Il valore predefinito è di 15 minuti (900 s).
`-T` ***n***	*max timeout*	Permette di specificare la durata espressa in secondi (***n***) del tempo massimo di inattività. In questo modo, un cliente non può negoziare una durata superiore.
`-A`	*anonymous*	Consente solo l'accesso anonimo, oppure solo le utenze elencate nel file '`/etc/ftpchroot`'.
`-u` ***maschera***	*umask*	Definisce un valore particolare della maschera dei permessi; altrimenti, il valore predefinito è pari a 0027_8.
`-p`	*no passive*	Disabilita la modalità «passiva», in modo da non accettare la creazione di connessioni verso porte indicate dai clienti. Ciò serve a facilitare l'attraversamento di un firewall (purché il firewall consenta questo passaggio), ma può creare difficoltà ad alcuni programmi clienti.

Opzione	Significato mnemonico	Descrizione
`-q`	*quiet*	Non mostra informazioni sulla versione al cliente che si collega.
`-M`	*multihome*	Consente di gestire directory differenti per l'accesso anonimo, in base al nome a dominio presso cui giunge la richiesta, secondo la forma '`~ftp/`***nome_a_dominio/***'.

38.5.1 Configurazione

La configurazione di OpenBSD FTP è molto semplice. Per prima cosa, l'accesso anonimo è consentito solo se nel sistema è previsto l'utente fittizio '**`ftp`**', assieme alla sua directory personale e a una shell valida.[5] Convenzionalmente, una shell è valida quando è indicata nel file '`/etc/shells`'.

Teoricamente, OpenBSD FTP non richiede nemmeno la predisposizione di una struttura particolare della directory '`~ftp/`', secondo la tradizione, perché gestisce internamente il comando '**`ls`**' e di tutto il resto si può fare a meno.

Nel caso si utilizzi l'opzione '**`-M`**', si deve provvedere a dividere la directory '`~ftp/`' in sottodirectory corrispondenti ai nomi a dominio con cui si può accedere al servizio. Per esempio, se l'elaboratore che ospita il servente OpenBSD FTP è raggiungibile con i nomi *dinkel.brot.dg* e *weizen.mehl.dg*, ci possono essere le directory '`~ftp/dinkel.brot.dg/`' e '`~ftp/weizen.mehl.dg/`'; chi ac-

cede a *ftp://dinkel.brot.dg* in modo anonimo, vede la prima directory, mentre chi accede a *ftp://weizen.mehl.dg* vede la seconda.

Si rammenta che l'utente anonimo accede solo alla porzione di file system che inizia da '`~ftp/`', come se questa fosse la radice.

Dopo la sistemazione dell'accesso anonimo, conviene occuparsi del file '`/etc/ftpchroot`', all'interno del quale si possono elencare gli utenti che, pur potendo accedere con il proprio nominativo, possono entrare solo nella propria directory personale, come avviene per gli utenti anonimi con la directory '`~ftp/`'.

```
tizio
caio
```

L'esempio che si vede sopra è molto breve e serve a fare in modo che gli utenti '**`tizio`**' e '**`caio`**' possano accedere limitatamente alla propria directory personale; tutti gli altri utenti hanno accesso a tutto il file system, con le limitazioni normali date dai permessi dei file e delle directory.

OpenBSD FTP riconosce anche il file '`/etc/ftpusers`', all'interno del quale vanno elencati i nominativi degli utenti a cui **non** si consente l'accesso. Generalmente si tratta di utenti fittizi, compreso '**`root`**' per questioni di sicurezza, come nell'esempio seguente:

```
root
bin
daemon
adm
lp
sync
shutdown
```

```
halt
mail
news
uucp
operator
games
nobody
```

Naturalmente, per compilare correttamente questo file, è bene analizzare il file '`/etc/passwd`' del proprio sistema. Si osservi che per impedire l'accesso agli utenti anonimi, ovvero '**`ftp`**' e '**`anonymous`**', occorre estendere questo file con tali nomi:

```
root
bin
daemon
adm
lp
…
nobody
ftp
anonymous
```

Infine, OpenBSD FTP riconosce anche il file '`/etc/nologin`', in presenza del quale rifiuta gli accessi; inoltre, è possibile definire un messaggio di benvenuto nel file '`/etc/ftpwelcome`' e anche il contenuto di '`/etc/motd`' viene visualizzato all'accesso.

38.6 Riferimenti

«

- Jay Rabak, *Active vs. passive FTP, a definitive explanation*, *http://slacksite.com/other/ftp.html*

• J. Postel, J. Reynolds, *RFC 959, File transfer protocol (FTP)*, 1985, *http://www.ietf.org/rfc/rfc959.txt*

[1] I numeri UID e GID dipendono dall'organizzazione del proprio sistema.

[2] Di solito, quando si inserisce il proprio indirizzo di posta elettronica come parola d'ordine per accedere a un servizio FTP anonimo, è sufficiente indicare la parte che precede il dominio, fino al simbolo '`@`' incluso. Quindi, se l'indirizzo fosse *daniele@dinkel.brot.dg*, basterebbe inserire '**`daniele@`**'.

[3] **FTP** UCB BSD

[4] **OpenBSD FTP** UCB BSD

[5] Il particolare della shell valida va tenuto in considerazione perché altri serventi FTP si comportano diversamente.