FTP mini-HOWTO

Matthew Borowski (*mkb@yahoo.com*)
(*http://tarp.worldserve.net/*) v0.2, 9 Gennaio 2000

Come usare client e server ftp. Traduzione a cura di Fabio Casadei (*fabicas@tin.it*) e revisione di Alessio Rolleri, 25 Ottobre 2003.

Indice

1 Premessa

Questo documento è stato scritto da Matthew Borowski per il Linux Documentation Project. Non ho ancora finito di aggiungere tutto il materiale che voglio inserire. In particolare, manca una sezione con le informazioni su come far funzionare un server FTP.

1.1 Come contattarmi

Vi prego di contattarmi nel caso abbiate delle aggiunte o dei cambiamenti per questo documento. Vi prego altresì di non chiedermi direttamente supporto tecnico: potete acquistare consulenze e supporto tecnico per sistemi Unix presso la mia società, ***WorldServe*** , o potete chiedere un aiuto su newsgroup quali **comp.os.linux.***. Vi prego, infine, di tenere conto del fatto che io capisco solo l'inglese ed il persiano.

1.2 Aspetti legali e distribuzione

1.3 Storia di questo documento

- 991203: prima scrittura del FTP HOWTO. Si tratta ancora di un lavoro in corso.

2 Informazioni su FTP

FTP (File Transfer Protocol) è un protocollo client/server che consente ad un utente di trasferire file da e verso un sito remoto della rete. Funziona con il protocollo TCP ed è comunemente usato su Internet, anche se può essere usato anche su una LAN.

Un sito FTP è un computer che sta eseguendo un software che fa da server FTP (conosciuto anche come demone ftp o `ftpd`). Solitamente chiunque può accedere ad un sito ftp pubblico perché consente di autenticarsi come utente `anonymous` o `ftp`. Ci sono molti siti ftp pubblici eccellenti che fungono da deposito per il software libero Unix disponibile. Imparare l'uso di FTP consente l'accesso a risorse indispensabili.

I siti FTP privati richiedono un nome utente o una password. Se si possiede un account di shell presso il proprio ISP, allora potrebbe essere possibile accedere ai propri file remoti via FTP (si contatti il proprio amministratore di sistema per verificarlo).

Un client FTP è l'applicazione che consente all'utente l'accesso ai server FTP. Ci sono molti client FTP disponibili. Alcuni sono grafici, altri sono testuali.

FTP venne creato presso l'Università della California, a Berkeley, per essere incluso nel BSD4.2 (lo Unix di Berkeley). Il documento RFC (Request for Comments) è disponibile presso .

3 Guida all'uso di FTP per il principiante

Una guida rapida all'uso di ftp.

Il programma ftp standard è il client originale ftp. Esso è compreso nella configurazione standard della maggior parte delle distribuzioni Linux. Apparve per la prima volta nel BSD4.2, sviluppato dall'Università della California, a Berkeley.

3.1 Eseguire il programma `ftp`

Usare ftp è facile. Poniamo che ci si voglia connettere al sito ftp anonimo metalab.unc.edu per scaricare l'ultima versione dei sorgenti del kernel di Linux.

Si digiti, al prompt dei comandi:

```
$ ftp metalab.unc.edu
```

Il programma ftp tenterà di connettersi a metalab.unc.edu. Un altro modo per fare ciò è quello di eseguire `ftp` dal prompt dei comandi senza parametri, usando poi il comando `open`, con il nome del sito come argomento:

```
$ ftp
ftp> open metalab.unc.edu
```

3.2 Autenticarsi presso un server FTP

Quando ci si connette ad un sito FTP, viene avviata una sessione di autenticazione (premendo il tasto invio, ftp cercherà di autenticarvi con lo stesso utente che state usando sulla macchina locale: in questo caso, `foo`). Ci autentichiamo come `anonymous` o `ftp`, per accedere all'archivio pubblico.

```
220 helios.oit.unc.edu FTP server (Version wu-2.6.0(2) Wed Nov 17 14:44:12
EST 1999) ready.
Name (metalab.unc.edu:foo):
```

Ora, si digiti un indirizzo e-mail completo come password (è quello che la maggior parte dei siti FTP pubblici richiede).

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

Completata con successo l'autenticazione (login), viene fornita l'informazione seguente:

```
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

3.3 Tipi di trasferimento file

Dopo aver ottenuto accesso ad un sito ftp, `ftp` mostrerà il tipo di trasferimento file. Nel nostro esempio, esso è binario. La modalità binaria trasferisce i file, bit per bit, così come essi si trovano sul server FTP. La modalità ascii, invece, trasferisce direttamente il testo. Per commutare tra le due modalità, basta digitare `ascii` o `binary`.

Si vogliono trasferire i sorgenti del kernel, così si mantiene la modalità di trasferimento file binaria. Si userà tale modalità anche per qualsiasi file non testuale, come immagini, archivi zip/gzip, programmi eseguibili, ecc. Nel dubbio, si usi la modalità binaria.

3.4 Navigare ed elencare le directory

Per vedere un elenco dei file, si digiti `ls`. Il comando `ls` inviato al server ftp è eseguito sul server remoto, così le opzioni da riga di comando che si potranno usare con esso varieranno a seconda del server. Comunque, le opzioni più comuni sono di solito disponibili. Controllare la pagina di manuale di `ls` per i dettagli.

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 33590
-r--r--r--   1 root     other    34348506 Dec 03 03:53 IAFA-LISTINGS
lrwxrwxrwx   1 root     other           7 Jul 15  1997 README -> WELCOME
-rw-r--r--   1 root     other         890 Nov 15 13:11 WELCOME
dr-xr-xr-x   2 root     other         512 Jul 15  1997 bin
dr-xr-xr-x   2 root     other         512 Jul 15  1997 dev
dr-xr-xr-x   2 root     other         512 Jul 18  1997 etc
drwxrwxrwx  11 ftp      20           4608 Nov 28 16:00 incoming
lrwxrwxrwx   1 root     other          13 Jun 04  1998 ls-lR -> IAFA-LISTINGS
dr-xr-xr-x  17 root     root          512 Jun 08 11:43 pub
dr-xr-xr-x   3 root     other         512 Jul 15  1997 unc
dr-xr-xr-x   5 root     other         512 Jul 15  1997 usr
226 Transfer complete.
```

Se il comando `ls` elenca così tanti file che l'elenco sparisce nella parte superiore dello schermo, è possibile usare la combinazione di tasti Shift-PaginaSu per scorrere verso l'alto. Ciò funziona sia nella console Linux che in xterm che in rxvt.

Negli archivi FTP pubblici, le risorse scaricabili sono di solito contenute nella directory `/pub`. Nel nostro esempio, si sa che i sorgenti del kernel si trovano nella directory `/pub/Linux/kernel`, così, per accedere a tale directory, si digiti:

```
ftp> cd pub/Linux/kernel
250-README for kernel
250-
250-What you'll find here: kernel sources and patches
250-
250-
250 CWD command successful.
```

I messaggi mostrati, che iniziano con 250, sono informazioni spedite dal server. In questo esempio, il server ftp è stato configurato per spedire automaticamente il file README non appena si accede alla directory.

3.5 Download e upload dei file

Adesso, dopo aver inviato un altro `ls`, capiamo di dover accedere con `cd` alla directory v2.2. Grazie ad un ulteriore `ls`, si trova il file che si desiderava scaricare. Si tratta di `linux-2.2.13.tar.gz`. Perciò, si digiti:

```
ftp> get linux-2.2.13.tar.gz
local: linux-2.2.13.tar.gz remote: linux-2.2.13.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for linux-2.2.13.tar.gz (15079540
bytes).
```

Il programma ftp ha iniziato a salvare il file remoto `linux-2.2.13.tar.gz` nel file locale `linux-2.2.13.tar.gz`.

Se si fosse voluto salvarlo, piuttosto, nel file locale `foo.tar.gz`, si sarebbe dovuto specificare nel modo seguente:

```
ftp> get linux-2.2.13.tar.gz foo.tar.gz
local: foo.tar.gz remote: linux-2.2.13.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for linux-2.2.13.tar.gz (15079540
bytes).
```

Se si desidera scaricare più di un file alla volta, si può usare il comando `mget` (multiple get). Si possono specificare i file che `mget` deve scaricare come una lista di nomi di file separati da spazi, oppure tramite l'uso dei caratteri jolly. Per esempio:

```
ftp> mget linux*
```

Scaricherà tutti i file che iniziano con la stringa linux. Normalmente, `mget` avverte, per ogni file, che sta per scaricarlo. Si può variare questo comportamento usando il comando `prompt`.

Ora, supponiamo di aver scritto una parte di un programma e supponiamo di volerla spedire alla Meta-Lab affinché la includa nell'archivio di software per Linux. Per prima cosa, ci si sposterà verso la directory `/incoming` (la maggior parte dei server FTP pubblici prevede una directory, chiamata solitamente "incoming" o "uploads", in cui si può eseguire l'upload dei file), poi si userà il comando `put`:

```
ftp> cd /incoming
ftp> put foo.tar.gz
local: foo.tar.gz remote: foo.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for foo.tar.gz.
226 Transfer complete.
10257 bytes sent in 0.00316 secs (3.2e+03 Kbytes/sec)
```

Il comando `put` funziona allo stesso modo di `get`, così è possibile usare `mput` per l'upload di più file. Si può anche effettuare l'upload di un file locale cambiandogli il nome sul server: basta specificare il nome del file remoto e/o il nome del percorso come argomenti.

Cosa succede se il file `foo.tar.gz` non si trova nella directory locale corrente al momento del tentativo di upload? Si può usare il comando `lcd` (local change directory) per cambiare posizione tra le directory locali:

```
ftp> lcd /home/foo/
Local directory now /home/foo
```

3.6 Eseguire comandi di shell

Il client ftp supporta l'uso del punto esclamativo (!) per eseguire comandi locali. Per esempio, per ottenere un elenco dei file presenti nella propria directory locale corrente, si digiti:

```
ftp> !ls
```

Ciò che accade è che ftp chiama la shell (specificata nella variabile d'ambiente `$SHELL`), ed è questa che esegue `ls`. In questo modo, si può eseguire qualsiasi riga di comando che funziona con la propria shell, semplicemente facendola precedere da "!" (la shell predefinita nella maggior parte delle distribuzioni Linux è la bash: Bourne Again SHell). Si noti che `!cd` non funziona nel modo atteso: da qui l'esigenza del comando `lcd`.

3.7 hash e tick

Non sarebbe carino poter monitorare il progresso dello scaricamento di un file con ftp? Il comando `hash` mostra a video dei caratteri cancelletto (ovvero '#') durante lo scaricamento di un file:

```
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
```

Come si intuisce, ftp mostrerà un carattere cancelletto per ogni 1024 byte di dati scaricati.

Esiste anche la possibilità di usare `tick`:

```
ftp> tick
Tick counter printing on (10240 bytes/tick increment).
```

Quando si scaricherà un file, questo comando stamperà un messaggio simile al seguente:

```
Bytes transferred: 11680
```

3.8 Altri comandi ftp

Esistono molti altri comandi ftp. Se si possiede l'autorizzazione (e, connettendosi al proprio account di shell privato, si dovrebbe possederla) è possibile anche creare directory sul server remoto usando il comando `mkdir`. La rimozione di un file o di una directory sul server remoto è possibile usando i comandi `delete` e `rmdir` rispettivamente. Si possono anche cambiare i permessi associati a file e directory remote usando il comando `chmod`.

Per informazioni più avanzate sull'uso di ftp si veda l'help in linea del programma ftp (accessibile digitando `help` senza argomenti per una lista dei comandi disponibili, o seguito da <`nome_comando`> per aiuto su un comando specifico. Un'altra fonte è la pagina di manuale Unix di ftp, che si ottiene digitando `man ftp` al prompt dei comandi.

4 Client FTP in modalità console

Il programma ftp originale fu il primo client ftp, ed è un buon investimento il tempo che si spende ad impararlo, essendo anche l'unico che si è certi di trovare sulla maggior parte dei sistemi (persino Win32 offre il programma ftp, sebbene in una versione arcaica e piuttosto stupida).

Esistono molti altri client ftp disponibili in modalità console (ossia testuale o da riga di comando). L'elenco seguente non intende essere esaustivo, ma include i client più usati. Per trovarne altri, si cerchi presso *FreshMeat* .

4.1 NcFTP

NcFTP è il client ftp preferito da sempre da molti utenti Unix. Si trova incluso nella maggior parte delle distribuzioni Unix ed offre molte caratteristiche avanzate come, ad esempio, il completamento tramite tasto tab ed i segnalibri (`bookmark`). La versione 2 di NcFTP aveva una modalità a tutto schermo basata su curses, che è stata tolta di mezzo nella versione 3 (adesso in fase di beta testing).

NcFTP non è compatibile al 100% con i comandi presenti in ftp standard. Per esempio, `get` e `put` in NcFTP agiscono come `mget` e `mput` nel ftp standard. Così, volendo salvare un file remoto come un file locale con nome diverso, si dovrà digitare 'get -z nome_remoto nome_locale'. Fortunatamente, NcFTP possiede un valido sistema di help in linea che assiste nell'apprendimento dei comandi.

Si può ottenere la versione più recente di NcFTP presso .

4.2 lukemftp

Porting del client FTP di NetBSD su altri sistemi, lukemftp prende il suo nome dall'autore della maggior parte delle caratteristiche avanzate, le quali includono: editing della riga di comando, ricerca di URL FTP e HTTP da riga di comando (anche via proxy), completamento automatico delle parole in base al contesto, barra di progresso dinamica, supporto dell'IPv6, conservazione della data di modifica, paginazione dei file locali e remoti, supporto della modalità passiva (con ricaduta sulla modalità attiva), supporto SOCKS, supporto di server gate-ftp TIS FWTK e regolazione della velocità di trasferimento.

Raccomando vivamente lukemftp agli utenti che, pur non volendo adattarsi a qualcosa di totalmente differente dal client ftp standard, richiedono caratteristiche più avanzate.

Si può ottenere la versione più recente di lukemftp presso .

4.3 lftp

lftp è un sofisticato client FTP basato su riga di comando. Come la bash, lftp offre un controllo dei job. Utilizza la libreria GNU `readline` per l'input, in modo da offrire completamento ed editing della riga di comando. Implementa anche i segnalibri (bookmark), il supporto al mirroring e permette di trasferire diversi file in parallelo.

Si può ottenere la versione più recente di lftp presso .

Sono disponibili anche i pacchetti Debian d'installazione presso .

4.4 cftp

Comfortable FTP (cftp) è un client in modalità a tutto schermo. Sopperisce alle proprie carenze in fatto di caratteristiche con la facilità d'uso. Basta, infatti, navigare attraverso le directory usando le frecce direzionali ed il tasto invio.

La versione più recente di cftp dovrebbe essere disponibile presso .

4.5 yafc

Yafc è un client ftp molto carino, le cui caratteristiche comprendono: directory cache, completamento automatico dei nomi dei file remoti, alias, ls con supporto dei colori, get/put/ls/rm ricorsivi, trasferimenti in modalità nohup, tagging (accodamento), connessioni multiple, supporto ai proxy e altro. Supporta anche autenticazione Kerberos4.

Si può ottenere la versione più recente di yafc presso .

Sono anche disponibili i pacchetti Debian d'installazione presso .

Sono anche disponibili i pacchetti Redhat d'installazione presso .

5 Client FTP per X Window

Ci sono diversi client FTP grafici, progettati per essere eseguiti sul sistema X Window. Questi client offrono facilità d'uso agli utenti che sono abituati agli ambienti grafici e a volte offrono opzioni versatili che sarebbe arduo implementare in un client ftp testuale.

5.1 gFTP

gFTP è un client FTP per X Windows scritto usando Gtk. L'interfaccia assomiglia al famoso software WS_FTP, usato comunemente su un tale sistema operativo instabile.

gFTP prevede la possibilità di effettuare download simultanei, riprendere i trasferimenti precedentemente interrotti di file, accodare i trasferimenti di file, scaricare intere directory. Supporta il proxy ftp, il caching delle directory remote, il trasferimento passivo e non passivo dei file, il drag-n-drop, possiede un gestore delle connessioni molto carino e altro ancora.

Se si possiede un'installazione Linux Red Hat con il gestore di finestre GNOME, allora probabilmente si possiede già gFTP. Se non è così, allora si può scaricare gFTP dalla sua homepage, presso .

5.2 WXftp

WXftp è un client FTP per il sistema X Window, progettato per essere usato principalmente sulle postazioni di lavoro Linux. È stato scritto usando il toolkit WXWindows, quindi può essere compilato usando sia Motif che GTK+.

Include un'interfaccia utente intuitiva (come WS_FTP), un gestore delle sessioni, help in linea, una barra di progresso ed altro.

Si visiti la homepage di WXftp presso .

5.3 LLNL XDIR e XFTP

LLNL XFTP fu uno dei primi client FTP grafici per Linux. Supporta FXP (trasferimento di file tra due host remoti) ed ha un'interfaccia basata su Motif.

Maggiori informazioni sono disponibili presso .

5.4 guiftp

Guiftp è un semplice client ftp scritto con il toolkit GTK+. Lo si consiglia a chi non necessita di tante caratteristiche e vuole un aspetto semplice e pulito.

La homepage di Guiftp si trova presso .

6 Server FTP

6.1 Come funziona un server FTP

Un server FTP tradizionale viene eseguito da inetd (il demone che avvia le applicazioni server per internet). La porta standard del servizio FTP è la numero 21. Quando un utente prova ad autenticarsi, il server FTP usa una chiamata di sistema standard per controllare il nome utente e la password, confrontando questi dati con quanto salvato nel file di sistema per le password o con le tabelle NIS se si sta usando NIS. Se l'autenticazione è corretta, viene consentito all'utente l'accesso al sistema.

L'FTP anonimo funziona diversamente. L'utente, in questo caso, si autentica con il nome utente 'anonymous' o 'ftp' (come definito nel file di configurazione). Quindi, gli viene fornito accesso ad un albero delle directory designato con il servizio chroot(). In questo modo, si garantisce che l'utente non possa ottenere l'accesso ad alberi delle directory per i quali non possiede l'autorizzazione. L'albero delle directory designato da chroot() generalmente contiene un filesystem fittizio, con le directory bin/, etc/ e lib/. I file disponibili per lo scaricamento sono solitamente resi disponibili nella directory pub/.

La motivazione per l'uso di un filesystem fittizio in un albero riservato a FTP anonimo risiede nel fatto che il demone FTP può dover eseguire comandi esterni per le richieste di tipo ls. Si possono anche aggiungere programmi nella directory bin/ e un utente può eseguirli con il comando SITE nel suo client ftp. Per esempio, l'FTP di Red Hat include il comando RPM (per consentire agli utenti di cercare pacchetti RPM sul sito).

Alcuni server FTP funzionano diversamente. Per esempio, alcuni consentono di configurare gli account utente indipendentemente dal file di sistema usato globalmente per le password (tali account prendono il nome di account FTP-only). Alcuni server (ad esempio ProFTPD e NcFTPd) implementano al loro interno i comandi di tipo ls e non necessitano di un albero delle directory speciale all'interno di quello designato con chroot(). Altri server ftp deviano completamente dal concetto di ftp standard. FTP4ALL, ad esempio, non usa per niente il sistema di password. Esso utilizza i propri file di utenti e di gruppi ed implementa caratteristiche quali il rapporto upload/download e la personalizzazione dei messaggi del server.

6.2 Aiuto riguardo ai server FTP

6.2.1 WU-FTPD

WU-FTPD è il demone ftp incluso in molte distribuzioni Linux, incluse Red Hat e Caldera. Per sapere di più su WU-FTPD si consulti .

Le FAQ riguardo WU-FTPD possono essere reperite presso .

6.2.2 ProFTPD

ProFTPD è un potente server FTP che include una configurazione in stile Apache, ampio supporto agli host virtuali ed ls interno.

Un riferimento completo ai comandi ed ai download si trova presso .