Per chi è alle Prime Armi sia con FreeBSD che con UNIX®

Annelise Anderson <andrsn@andrsn.stanford.edu >
Revisione: 43184

FreeBSD è un marchio registrato della FreeBSD Foundation.

IBM, AIX, OS/2, PowerPC, PS/2, S/390, e ThinkPad sono marchi della International Business Machines Corporation negli Stati Uniti, in altri paesi, o in entrambi.

Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media e Windows NT sono marchi o marchi registrati della Microsoft Corporation negli Stati Uniti e/o in altri paesi.

Netscape e Netscape Navigator sono marchi registrati della Netscape Communications Corporation negli Stati Uniti e in altri paesi.

Motif, OSF/1, e UNIX sono marchi registrati e IT DialTone e The Open Group sono marchi del The Open Group negli Stati Uniti e in altri paesi.

Molti dei nomi identificativi usati dai produttori e dai venditori per distinguere i loro prodotti sono anche dei marchi. Quando questi nomi appaiono nel libro, e il FreeBSD Project è al corrente del marchio, vengono fatti seguire dal simbolo «™» o «®».

15 Agosto 1997 di .

Sommario

Congratulazioni per aver installato FreeBSD! Questa introduzione é per chi é alle prime armi con FreeBSD *e* UNIX®—perciò comincia dalle basi. Stai certamente usando la versione 2.0.5 o una più recente di FreeBSD distribuita da FreeBSD.org, il tuo sistema ha (per il momento) un solo utente (te stesso)—e sei probabilmente abbastanza bravo con DOS/Windows® o OS/2®.

Traduzione a cura di Massimiliano Stucchi <max@gufi.org >.

Indice

1. Entrare ed Uscire dal Sistema 2
2. Aggiungere un Utente con Privilegi di Root 3
3. Diamoci un'occhiata in giro 3
4. Ottenere Aiuto e Informazioni 4
5. Modificare File di Testo 6
6. Stampa di File da DOS 8
7. Altri Comandi Utili 9
8. Prossimi Passi 10
9. Il tuo Ambiente di Lavoro 11
10. Altro 13
11. I Commenti sono Benvenuti 13

1. Entrare ed Uscire dal Sistema

Entra (quando vedi `login:`) come l'utente che hai creato durante l'installazione oppure come `root`. (La tua installazione di FreeBSD dovrebbe già avere un account di `root`; che può andare ovunque e fare qualsiasi cosa, anche cancellare file essenziali, perciò stai attento!) I simboli `%` e `#` che incontrerai più avanti simboleggiano il prompt (i tuoi potrebbero essere differenti), dove `%` indica un utente ordinario e `#` indica `root`.

Per uscire (e ritrovarsi con un nuovo prompt `login:`) scrivi

```
# exit
```

tante volte quanto serve. Sì, premi invio dopo ogni comando, e ricordati che UNIX® fa distinzione tra maiuscole e minuscole—perciò `exit`, non `EXIT`.

Per spegnere il computer digita

```
# /sbin/shutdown --h now
```

O per riavviarlo digita

```
# /sbin/shutdown --r now
```

oppure

```
# /sbin/reboot
```

Puoi anche riavviarlo premendo Ctrl+Alt+Canc. Lasciagli un po' di tempo per compiere il suo lavoro. Questo equivale a `/sbin/reboot` nelle versioni più recenti di FreeBSD ed è molto meglio che premere il bottone di reset. Non vorrai mica reinstallare tutto da capo, vero?

2. Aggiungere un Utente con Privilegi di Root

Se non hai creato un utente durante l'installazione e quindi sei entrato nel sistema come `root`, dovresti probabilmente crearne uno ora tramite

```
# adduser
```

La prima volta che aggiungi un utente, il sistema dovrebbe chiederti di inserire delle impostazioni di default da applicare. Potresti volere come shell csh(1) invece di sh(1), se ti viene consigliato `sh` come default. Altrimenti premi solo invio per accettare i valori proposti. Questi dati vengono salvati in `/etc/adduser.conf` , un file modificabile successivamente a mano.

Supponiamo che tu voglia creare l'utente `jack` di nome reale *Jack Benimble*. Assegna a `jack` una password per ragioni di sicurezza (anche i bambini che gironzolano per casa potrebbero mettere le mani sulla tastiera). Quando ti viene chiesto se vuoi invitare `jack` in un altro gruppo, digita `wheel`

```
Login group is ``jack''. Invite jack into other groups: wheel
```

Questo ti permetterà di entrare come l'utente `jack` e usare il comando su(1) per diventare `root`. A quel punto non sarai più preso in giro per essere entrato direttamente come `root`.

Puoi uscire da `adduser` in qualsiasi momento premendo Ctrl+C, e alla fine avrai l'opportunità di approvare il nuovo utente oppure premere n per non farlo. Potresti voler creare un secondo utente cosicché quando andrai a modificare i file di `jack` avrai un'ancora di salvezza in caso qualcosa vada male.

Una volta fatto questo, usa `exit` per tornare al prompt di login ed entrare come `jack`. In generale è meglio cercare di lavorare da utente normale in modo da non avere il potere—e il rischio—di `root`.

Se hai già creato un utente e vuoi che quell'utente sia in grado di usare `su` per diventare `root`, puoi entrare come `root` e modificare il file `/etc/group` , aggiungendo `jack` alla prima linea (il gruppo `wheel`). Ma prima devi fare pratica con vi(1), l'editor di testo—oppure usa il più semplice ee(1), installato sulle recenti versioni di FreeBSD.

Per cancellare un utente, usa il comando `rmuser`.

3. Diamoci un'occhiata in giro

Una volta avuto accesso come utente normale, guardati in giro e prova alcuni dei comandi che ti daranno accesso alle fonti di aiuto e di informazioni su FreeBSD.

Ecco qui una lista di comandi e le loro funzioni:

`id`
: Ti dice chi sei!

`pwd`
: Ti mostra dove sei—la directory in cui stai lavorando.

`ls`
: Ti mostra una lista dei file contenuti nella directory.

`ls -F`
: Ti mostra un elenco dei file contenuti nella directory ponendo * dopo i file eseguibili, `/` dopo le directory, e `@` dopo i collegamenti simbolici.

`ls -l`
: Mostra un elenco di file nel formato lungo—grandezza, data, permessi.

`ls -a`
: Mostra una lista dei file nascosti, cioè con un «punto» davanti al nome, insieme agli altri. Se sei `root`, i file «puntati» vengono mostrati anche senza l'opzione `-a`.

`cd`
: Cambia la directory di lavoro. `cd ..` torna alla directory superiore; nota lo spazio dopo `cd`. `cd /usr/local` va nella directory specificata. `cd ~` va nella directory home dell'utente collegato in quel momento—per esempio, `/usr/home/jack`. Prova `cd /cdrom`, e poi `ls`, per scoprire se il tuo CDROM è montato e funziona.

`view nomefile`
: Mostra il contenuto del file (chiamato *`nomefile`*) senza modificarlo. Prova `view /etc/fstab`. Digita `:q` per uscire.

`cat nomefile`
: Mostra *`nomefile`* sullo schermo. Se è troppo lungo e ne puoi vedere solo la fine, premi BlocScorr e usa freccia-su per muoverti in alto; puoi usare BlocScorr anche con le pagine man. Premi ancora BlocScorr per uscire dallo scorrimento. Potresti provare `cat` con alcuni dei file nascosti presenti nella tua directory home—`cat .cshrc`, `cat .login`, `cat .profile`.

Noterai degli alias in `.cshrc` per alcuni dei comandi `ls` (sono molto convenienti). Puoi creare degli altri alias modificando `.cshrc`. Puoi far sì che questi alias diventino disponibili a tutti gli utenti mettendoli nel file di configurazione generale di `csh`, `/etc/csh.cshrc`.

4. Ottenere Aiuto e Informazioni

Ecco alcune risorse utili per ottenere aiuto. *`Testo`* è qualcosa che puoi digitare a tuo piacere—normalmente si tratta di un comando o del nome di un file.

`apropos testo`
: Tutto ciò che contiene la stringa *`testo`* nel database whatis.

`man testo`
: Mostra la pagina man di *`testo`*, la maggior risorsa di documentazione per i sistemi Un*x. `man ls` ti dirà tutti i modi possibili per usare il comando `ls`. Premi Invio per muoverti nel testo, Ctrl+B per andare indietro di una pagina, Ctrl+F per andare avanti, q oppure Ctrl+C per uscire.

`which testo`
: Ti dice dove si trova il comando *`testo`* nel path dell'utente.

`locate testo`
: Ti dice tutte le directory nei path dell'utente in cui si trova il comando *`testo`*.

`whatis testo`
: Ti dice che cosa fa il comando *`testo`* e la sua pagina man. Digitando `whatis *` ti verranno presentate tutte le pagine man associate agli eseguibili presenti nella directory corrente.

`whereis testo`
: Trova il file *`testo`*, dandoti il suo percorso completo.

Potresti voler provare ad usare `whatis` con alcuni comandi utili come `cat`, `more`, `grep`, `mv`, `find`, `tar`, `chmod`, `chown`, `date`, e `script`. `more` ti permette di leggere una pagina alla volta come in DOS, ad esempio, `ls -l | more` oppure `more nomefile`. `*` ha valore assoluto—per esempio, `ls w*` mostra tutti i file che cominciano con `w`.

Per caso alcuni di questi comandi non funzionano correttamente? Sia locate(1), sia whatis(1) dipendono da un database che viene ricostruito settimanalmente. Se la tua macchina non sarà lasciata accesa per il fine settimana (usando FreeBSD), può darsi che tu voglia usare i comandi per la manutenzione giornaliera, settimanale, e mensile ogni tanto. Falli partire come `root` e lascia loro il tempo di finire il lavoro prima di farne partire un altro.

```
# periodic daily
output tralasciato
# periodic weekly
output tralasciato
# periodic monthly
output tralasciato
```

Se ti stufi di aspettare, premi Alt+F2 per avere un'altra *console virtuale*, e poterti loggare nuovamente. Dopotutto è un sistema multi-utente, e multi-tasking. Probabilmente questi comandi produrranno dei messaggi sullo schermo quando lavorano; puoi digitare `clear` per pulire lo schermo. Quando hanno finito, dovresti dare un'occhiata a `/var/mail/root` e `/var/log/messages` .

Usare tali comandi fa parte dell'amministrazione di sistema—e come utente singolo di un sistema UNIX®, sei tu l'amministratore del sistema. Praticamente l'unica cosa per la quale è necessario che tu sia `root` è l'amministrazione. Queste responsabilità non vengono trattate bene nemmeno in quei grossi libri su UNIX®, che sembrano dedicare troppo spazio all'uso dei menu nei windows manager. Potresti voler leggere uno dei più interessanti libri sull'amministrazione di sistema, come *UNIX System Administration Handbook* di Evi Nemeth et.al. (Prentice-Hall, 1995, ISBN 0-13-15051-7)—la seconda edizione con la copertina rossa; oppure *Essential System Administration* di Æleen Frisch (O'Reilly & Associates, 2002, ISBN 0-596-00343-9). Io ho usato quello di Nemeth.

5. Modificare File di Testo

Per poter configurare il tuo sistema, devi modificare dei file. Molti di questi saranno in `/etc`; e avrai bisogno del comando `su` per diventare `root` e poter così modificarli. Puoi usare il semplice editor `ee`, ma alla lunga risulta più utile imparare `vi`. C'é un eccellente tutorial su `vi` in `/usr/src/contrib/nvi/docs/tutorial` se hai installato i sorgenti di sistema.

Prima di modificare un file, dovresti farne una copia. Supponiamo tu voglia modificare `/etc/rc.conf`. Puoi semplicemente usare `cd /etc` per andare in `/etc` e fare:

```
# cp rc.conf rc.conf.orig
```

Questo copierà `rc.conf` in `rc.conf.orig`, e potrai successivamente copiare `rc.conf.orig` in `rc.conf` per tornare all'originale. Ma ancora meglio sarà spostare (rinominare) il file per poi ricopiarlo con il nome originale:

```
# mv rc.conf rc.conf.orig
# cp rc.conf.orig rc.conf
```

perché il comando `mv` mantiene la data e il proprietario originali del file. Puoi ora modificare `rc.conf`. Se vuoi tornare all'originale, potresti fare **`mv rc.conf rc.conf.myedit`** (assumendo che vuoi tenere la versione modificata) e quindi fare

```
# mv rc.conf.orig rc.conf
```

per tornare allo stato iniziale.

Per modificare un file, digita

```
# vi nomefile
```

Muoviti nel testo con i tasti freccia. Esc mette `vi` in modalità comando. Ecco qui alcuni dei comandi:

x
: cancella la lettera su cui si trova il cursore

`dd`
: cancella l'intera riga (anche se va a capo sullo schermo)

`i`
: inserisci del testo nella posizione del cursore

`a`
: inserisci del testo dopo il cursore

Quando digiti `i` o `a`, puoi inserire del testo. `Esc` ti riporta in modalità comando dove puoi digitare

`:w`
: per salvare le modifiche sul disco e continuare a modificare il file

`:wq`
: per salvare le modifiche e uscire

`:q!`
: per uscire senza salvare le modifiche

`/testo`
: per spostare il cursore su *`testo`*; `/Invio` per trovare la prossima occorrenza di *`testo`*.

`G`
: per andare alla fine del file

`nG`
: per andare alla riga *n* del file, dove *n* è un numero

Ctrl+L
: per ridisegnare lo schermo

Ctrl+b e Ctrl+f
: vai avanti e indietro di una pagina, come succede con `more` e `view`.

Fai un po' di pratica con `vi` nella tua directory home creando un nuovo file digitando `vi nomefile` e aggiungendo e cancellando del testo, salvando il file, e riaprendolo di nuovo. `vi` è pieno di sorprese perché è abbastanza complesso, e ti capiterà di digitare un comando che farà di sicuro qualcosa che non ti aspetti. (Alcune persone preferiscono `vi`—è più potente dell'EDIT del DOS—scopri il comando `:r`) Usa Esc una o più volte per essere sicuro di essere in modalità comando e continua da lì quando hai dei problemi, salva spesso con `:w`, e usa `:q!` per uscire e ricominciare (dal tuo ultimo `:w`) quando ne hai bisogno.

Ora puoi usare `cd` per andare in `/etc`, `su` per diventare `root`, `vi` per modificare il file `/etc/group` , e aggiungere un utente al gruppo `wheel` cosicché possa avere privilegi di `root`. Aggiungi solo una virgola e il nome di login dell'utente alla fine della prima riga

del file, premi Esc, e usa `:wq` per salvare il file su disco e uscire. La modifica ha effetto immediato. (Non hai lasciato uno spazio dopo la virgola, vero?)

6. Stampa di File da DOS

A questo punto la tua stampante non funzionerà ancora sotto FreeBSD, ecco quindi un sistema per creare un file da una pagina man, metterlo su un floppy, e quindi stamparlo da DOS. Supponiamo che tu voglia leggere attentamente come cambiare i permessi sui file (abbastanza importante). Puoi usare `man chmod` per leggere come fare. Il comando

```
% man chmod -| col --b > chmod.txt
```

toglierà gli elementi di formattazione e manderà il tutto sul file `chmod.txt` al posto di mostrare il contenuto sullo schermo. Ora metti un dischetto formattato DOS nel lettore, digita `su` per diventare `root`, e scrivi

```
# /sbin/mount --t msdos -/dev/fd0 -/mnt
```

per montare il floppy su `/mnt`.

Ora (non hai più bisogno di essere `root`, e puoi digitare `exit` per tornare ad essere l'utente `jack`) puoi andare nella directory in cui hai creato `chmod.txt` e copiare il file sul floppy digitando:

```
% cp chmod.txt -/mnt
```

e usare `ls /mnt` per vedere il contenuto di `/mnt`, che dovrebbe contenere il file `chmod.txt`.

In particolare potresti voler creare un file con l'output di `/sbin/dmesg` digitando

```
% /sbin/dmesg > dmesg.txt
```

e copiare `dmesg.txt` sul floppy. `/sbin/dmesg` è il file di log di avvio, ed è importante comprenderlo perché ti mostra cosa ha trovato FreeBSD all'avvio. Se poni delle domande sulla mailing list per le domande generiche su FreeBSD o su un gruppo USENET—del tipo «FreeBSD non trova il mio drive per i nastri, che cosa faccio?»—la gente vorrà sapere cosa mostra il tuo `dmesg`.

Ora devi smontare il floppy (da `root`) per poter togliere il disco

```
# /sbin/umount -/mnt
```

e riavviare per tornare in DOS. Copia questo file in una directory DOS, richiamali con l'EDIT del DOS, Notepad o Wordpad di Windows®, o un editor di testi, fai una piccola modifica in modo che il file debba essere salvato, e stampa come faresti da DOS o Windows®. Spera che funzioni! Le pagine man vengono meglio se stampate con il comando DOS

`print`. (Copiare i file da FreeBSD su una partizione DOS montata è ancora in alcuni casi rischioso.)

Far funzionare la stampante sotto FreeBSD consiste nel creare un opportuno elemento in `/etc/printcap` e creare una directory di spool corrispondente in `/var/spool/output` . Se la tua stampante è su lpt0 (ciò che DOS chiama LPT1), devi solo andare in `/var/spool/output` e (da `root`) creare la directory `lpd` digitando: `mkdir lpd`, se non è già presente. A quel punto la stampante dovrebbe rispondere quando il sistema parte, e `lp` o `lpr` dovrebbero mandare un file alla stampante. Che il file venga stampato o meno è solo questione di configurazione, che è discussa nel Manuale di FreeBSD.

7. Altri Comandi Utili

`df`
: mostra lo spazio disponibile e tutte le partizioni montate.

`ps aux`
: mostra i processi in esecuzione. `ps ax` è una forma contratta.

`rm nomefile`
: cancella *`nomefile`*.

`rm -R dir`
: cancella la directory *`dir`* e tutte le sottodirectory—attenzione!

`ls -R`
: mostra il contenuto della directory e delle sue sottodirectory; io usavo una variante, `ls -AFR > where.txt` , per avere una lista dei file in `/` e (separatamente) `/usr` prima che scoprissi dei metodi migliori per cercare i file.

`passwd`
: per cambiare la password dell'utente (o di `root`)

`man hier`
: pagina man sul file system di UNIX®

Usa `find` per trovare `nomefile` in `/usr` o nelle sue sottodirectory digitando

```
% find -/usr --name -"nomefile"
```

Puoi usare * come identificatore universale in "*`nomefile`*" (che dovrebbe essere tra virgolette). Se dici a `find` di cercare in `/` anziché `/usr` cercherà il/i file su tutti i file system montati, inclusi i CDROM e le partizioni DOS.

Un libro eccellente che tratta i comandi e le utility di UNIX® è *Unix for the Impatient* di Abrahams & Larson (2nd ed., Addison-Wesley, 1996). Ci sono anche un sacco di informazioni su UNIX® su Internet.

8. Prossimi Passi

Dovresti ora avere gli strumenti necessari per girare nel sistema e modificare i file, così da poter rendere tutto funzionante. Ci sono un sacco di informazioni nel Manuale di FreeBSD (che è probabilmente sul tuo disco rigido) e sul sito web di FreeBSD. Una grande scelta di package e port è presente sul CDROM così come sul sito web. Il manuale ti spiega come usarli (prendi il package se esiste, con `pkg_add /cdrom/packages/All/nomepackage`, dove *`nomepackage`* è il nome del file del package). Il CDROM ha una lista di package e di port con delle brevi descrizioni in `cdrom/packages/index`, `cdrom/packages/index.txt`, e `cdrom/ports/index`, e con descrizioni più ampie in `/cdrom/ports/*/*/pkg/DESCR`, dove * rappresenta rispettivamente sottodirectory di tipi di programmi e nomi di programmi.

Se trovi il manuale troppo difficile su come installare i port dal CDROM (con il sistema di `lndir` e altro), ecco come funziona normalmente:

Trova il port che vuoi, supponiamo `kermit`. Ci sarà una directory per lui sul CDROM. Copia la sottodirectory in `/usr/local` (un buon posto perché il software che aggiungi sia disponibile a tutti gli utenti) con:

```
# cp --R -/cdrom/ports/comm/kermit -/usr/local
```

Questo dovrebbe portarti ad avere la sottodirectory `/usr/local/kermit` che contiene tutti i file presenti nella sottodirectory `kermit` del CDROM.

Ora, crea la directory `/usr/ports/distfiles` se non esiste ancora, usando `mkdir`. Poi controlla `/cdrom/ports/distfiles` cercando un file con il nome che indica che si tratta del port esatto. Copia quel file in `/usr/ports/distfiles`; nelle versioni più recenti puoi saltare questo passo, perché FreeBSD lo farà per te. Nel caso di `kermit`, non c'è nessun distfile.

Quindi entra con `cd` nella sottodirectory di `/usr/local/kermit` che contiene il file `Makefile`. Digita

```
# make all install
```

Durante questo processo il port userà FTP per scaricare i file compressi che non ha trovato sul CDROM o in `/usr/ports/distfiles`. Se la tua connessione non funziona ancora e non c'è nessun file per il port in `/cdrom/ports/distfiles`, dovrai recuperare il distfile usando un'altra macchina e poi copiarlo in `/usr/ports/distfiles` da un dischetto o dalla partizione DOS. Leggi `Makefile` (usando `cat` o `more` oppure `view`) per scoprire dove andare (il sito principale di distribuzione) per trovare il file e conoscere il suo nome. Il nome verrà troncato quando scaricato da DOS, e dopo averlo trasferito in `/usr/ports/distfiles` dovrai rinominarlo (usando il comando `mv`) nel suo nome originale cosicché possa essere trovato. (Usa il trasferimento di file binario!) Quindi torna in `/usr/local/kermit`, trova la directory contenente `Makefile`, e digita `make all install`.

Un'altra cosa che può succedere quando si installa un port o un package è che questi abbiano bisogno di un altro programma. Se l'installazione si ferma con un messaggio can't find unzip o simile, potresti dover installare il package o il port di unzip prima di proseguire.

Una volta installato, digita `rehash` per far sì che FreeBSD rilegga i file contenuti nel path e sappia quali sono presenti. (Se trovi un sacco di messaggi path not found quando usi `whereis` o which, dovresti fare delle aggiunte all'elenco delle directory nel file `.cshrc` nella tua directory home. L'elenco dei path in UNIX® fa la stessa cosa che fa in DOS, tranne che la directory corrente (di default) non si trova nel path per ragioni di sicurezza; se il comando che vuoi eseguire è nella directory in cui ti trovi, devi digitare `./` prima del nome del comando; niente spazio dopo la barra.)

Potresti volere la versione più recente di Netscape® dal loro sito FTP. (Netscape® necessita dell'X Window System.) Ora c'é una versione per FreeBSD, quindi dà un'occhiata in giro. Usa solo `gunzip nomefile` e `tar xvf nomefile` sul file, sposta il binario in `/usr/local/bin` o qualche altro posto in cui vengono tenuti i binari, esegui `rehash`, e quindi aggiungi le seguenti linee a `.cshrc` in tutte le directory home degli utenti oppure (più semplicemente) in `/etc/csh.cshrc` , il file di configurazione globale di `csh`:

```
setenv XKEYSYMDB -/usr/X11R6/lib/X11/XKeysymDB
setenv XNLSPATH -/usr/X11R6/lib/X11/nls
```

Questo assume che il file `XKeysymDB` e la directory `nls` siano in `/usr/X11R6/lib/X11` ; se non lo sono, trovale e mettile lì.

Se hai originariamente installato Netscape® dal CDROM (o via FTP), non sostituire `/usr/local/bin/netscape` con il nuovo binario di netscape; questo è solo uno script di shell che imposta le variabili di ambiente per te. Rinomina invece il nuovo binario in `netscape.bin` e rimpiazza il vecchio binario, che dovrebbe essere `/usr/local/netscape/netscape` .

9. Il tuo Ambiente di Lavoro

La shell è la parte più importante del tuo ambiente di lavoro. In DOS, la shell è solitamente command.com. La shell è ciò che interpreta i comandi che digiti sulla linea di comando, e quindi comunica con il resto del sistema operativo. Puoi anche scrivere script di shell, che sono come i file batch di DOS: una serie di comandi che devono essere eseguiti senza il tuo intervento.

Due shell vengono normalmente installate con FreeBSD: `csh` e `sh`. `csh` è buona per lavoro da linea di comando, ma gli script dovrebbero essere scritti usando `sh` (o `bash`). Puoi scoprire che shell hai digitando `echo $SHELL` .

`csh` è una buona shell, ma `tcsh` fa tutto ciò che `csh` fa e anche altro. Ti permette di richiamare i comandi usando le frecce e ti permette di modificarli. Ha l'auto-completamento

dei nomi di file con tab (`csh` usa Esc), e ti permette di tornare alla directory in cui eri digitando `cd -`. È anche più semplice alterare il prompt con `tcsh`. Ti rende la vita più facile.

Ecco tre semplici passi per installare una nuova shell:

1. Installa la shell tramite port o package, come faresti con un qualsiasi altro port o package. Usa `rehash` e `which tcsh` (assumendo che tu stia installando `tcsh`) per essere sicuro di averla installata.

2. Da `root`, modifica `/etc/shells` , aggiungendo una riga nel file per la nuova shell, in questo caso `/usr/local/bin/tcsh` , e salva il file. (Alcuni port lo fanno per te.)

3. Usa il comando `chsh` per cambiare permanentemente la tua shell in `tcsh`, o digita `tcsh` al prompt per cambiare la shell senza dover uscire dal sistema per poi rientrare.

Nota

Può essere pericoloso cambiare la shell di `root` in qualcosa di diverso da `sh` o `csh` su versioni più recenti di FreeBSD e di UNIX®; potresti non avere una shell funzionante se il sistema entra in modalità singolo utente. La soluzione è usare `su -m` per diventare `root`, che ti dà `tcsh` come shell di `root`, poiché la shell è parte del tuo ambiente. Puoi rendere tutto ciò permanente aggiungendo al tuo `.tcshrc` un alias con:

```
alias su su --m
```

Quando `tcsh` parte, legge i file `/etc/csh.cshrc` e `/etc/csh.login` , come farebbe `csh`. Leggerà anche il file `.login` nella tua directory home ed anche `.cshrc` , a meno che tu non abbia un file `.tcshrc`. Puoi crearlo copiando `.cshrc` in `.tcshrc`.

Ora che hai installato `tcsh`, puoi sistemare il tuo prompt. Puoi trovare i dettagli nella pagina man di `tcsh`, ma ecco qui una linea da mettere nel tuo `.tcshrc` che ti dirà quanti comandi hai digitato, che ore sono, e in che directory ti trovi. Produce anche un `>` se sei un utente normale e un `#` se sei `root`, ma `tcsh` lo farebbe in ogni caso:

set prompt = "%h %t %~ %# "

Questa dovrebbe andare nella stessa posizione della linea di prompt corrente se ce n'è una, o sotto "if($?prompt) then" in caso contrario. Commenta la vecchia riga; così potrai tornare a quella vecchia se la preferirai. Non dimenticare gli spazi e le virgolette. Puoi far rileggere `.tcshrc` digitando `source .tcshrc`.

Puoi avere una lista delle variabili di sistema che sono state impostate digitando `env` al prompt. Il risultato ti mostrerà il tuo editor di default, il pager, e il tipo di terminale, tra

le altre possibili variabili. Un comando utile se ti connetti al sistema da una postazione remota e non riesci ad eseguire un programma perché il terminale non ne è capace è `setenv TERM vt100`.

10. Altro

Da `tcsh` puoi smontare il CDROM con `/sbin/umount /cdrom` , toglilo dal lettore, inseriscine un altro, e montalo con `/sbin/mount_cd9660 /dev/cd0a /cdrom` assumendo che cd0a sia il nome di dispositivo del tuo lettore di CDROM. La versione più recente di FreeBSD ti permette di montare il CDROM solo con `/sbin/mount /cdrom` .

Usare il «live file system»—il secondo cd del set di FreeBSD—è utile se hai poco spazio a disposizione. Ciò che si trova sul live file system cambia da release a release. Potresti provare ad eseguire dei giochi dal CDROM. Questo comporta l'uso di `lndir`, che viene installato con l'X Window System, per dire ai programmi dove trovare i file necessari, poiché questi si trovano nel file system `/cdrom` anziché `/usr` e le sue sottodirectory, che è dove dovrebbero essere. Leggi `man lndir` per avere più informazioni.

11. I Commenti sono Benvenuti

Se usi questa guida, sarei interessata a sapere dove non è chiara, ciò che è stato tralasciato e che vorresti venisse incluso, e sapere se tutto ciò è stato utile. I miei ringraziamenti vanno a Eugene W. Stark, professore di informatica a SUNY-Stony Brook, e John Fieber per i suoi utili commenti.

Annelise Anderson, <andrsn@andrsn.stanford.edu>

Per questioni legate alla traduzione, o se avete commenti da poter esprimere solo in italiano, non esitate a contattarmi. Come per l'autrice originale, ogni genere di commenti è ben accetto.

Massimiliano Stucchi, <stucchi@willystudios.com>