# Informationen für FreeBSD- und UNIX®-Einsteiger

Zusammenfassung

Wir gratulieren Ihnen zur Installation von FreeBSD! Diese Einführung ist für alle Personen gedacht, die FreeBSD und UNIX® noch nicht kennen, und enthält daher auch einige grundlegende Informationen. Es wird davon ausgegangen, dass Sie FreeBSD 2.0.5 oder neuer in der von FreeBSD.org veröffentlichten Form installiert haben sowie über gute Kenntnisse von DOS/Windows® oder OS/2® verfügen.

*Übersetzt von Fabio Tosques.*

---

# Inhaltsverzeichnis

1. An- und Abmelden . . . 1
2. Einen neuen Benutzer anlegen, der `root`-Rechte erlangen kann . . . 2
3. Sich im System umsehen . . . 3
4. Hilfe und Informationen erhalten . . . 4
5. Textdateien bearbeiten . . . 5
6. Dateien mit Hilfe von DOS ausdrucken . . . 7
7. Weitere nützliche Befehle . . . 9
8. Die nächsten Schritte . . . 9
9. Die Arbeitsumgebung . . . 11
10. Einige abschließende Bemerkungen . . . 13
11. Kommentare . . . 13

# 1. An- und Abmelden

Melden Sie sich (wenn `login:` am Bildschirm erscheint) als derjenige Benutzer, den Sie während der Installation angelegt haben, oder als `root` an. (Die FreeBSD-Installationsroutine legt in jedem Fall einen Benutzer `root` an; dieser darf auf dem System fast alles tun, einschließlich dem Löschen von wichtigen Dateien. Seien Sie also vorsichtig, wenn Sie sich als `root` anmelden.) Die Zeichen % und # stellen im folgenden die Eingabeaufforderung dar, wobei % für den gewöhnlichen Benutzer und # für den Benutzer `root` steht.

Um sich abzumelden, geben Sie

```
# exit
```

so oft wie nötig ein (und zwar bis wieder `login:` erscheint). Drücken Sie nach jedem Befehl die `enter` -Taste und beachten Sie, dass UNIX® zwischen Groß- und Kleinschreibung unterscheidet - `exit` und `EXIT` sind also zwei verschiedene Befehle.

Um den Rechner herunterzufahren, geben Sie

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

ein.

Um den Rechner neu zu starten, geben Sie

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

ein, oder Sie rufen einfach

```
# /sbin/reboot
```

auf.

Sie können natürlich auch den Rechner mit `Strg` + `Alt` + `Entf` neu starten. Warten Sie danach, bis der Neustart erfolgt ist. Diese Tastenkombination ist in neueren FreeBSD-Versionen mit `/sbin/reboot` identisch und viel besser als das Drücken des Reset-Knopfes. Sie wollen das System ja sicher nicht noch einmal neu installieren, oder?

# 2. Einen neuen Benutzer anlegen, der `root` -Rechte erlangen kann

Falls Sie während der Installation des Systems keine Benutzer angelegt haben und noch als `root` angemeldet sind, sollten Sie mit

```
# adduser
```

einen neuen Benutzer anlegen.

Wenn Sie `adduser` das erste Mal ausführen, werden Sie gefragt, ob Sie einige Standardeinstellungen abspeichern möchten. Vielleicht möchten Sie csh(1) statt sh(1) als Default-Shell auswählen (falls `sh` als Default-Shell angeboten wird). Diese Standardeinstellungen sind in /etc/adduser.conf gespeichert und können von Ihnen verändert werden.

Angenommen, Sie möchten den Benutzer `jack` mit dem vollständigen Namen *Jack Benimble* anlegen. Weisen Sie `jack` auf jeden Fall ein Passwort zu (auch Kinder, die auf der Tastatur spielen, können ein Problem darstellen). Wenn Sie gefragt werden, ob `jack` Mitglied in anderen Gruppen sein soll, geben Sie `wheel` ein.

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

Dadurch können Sie sich als `jack` anmelden, und mit su(1) `root`-Rechte erlangen. Dann werden Sie nie wieder dafür kritisiert werden, dass Sie sich als `root` am System anmelden.

Das Programm `adduser` kann jederzeit mit `Strg` + `C` beendet werden. Am Schluß haben Sie die Möglichkeit, die Einstellungen für den neuen Benutzer nochmals zu kontrollieren oder diese mit `n` zu verwerfen. Sie sollten noch einen zweiten Benutzer anlegen, damit Sie auch dann auf Ihr System zugreifen können, wenn Sie die Anmeldedaten von `jack` bearbeitet und dabei einen Fehler gemacht haben.

Haben Sie alle Benutzer angelegt, geben Sie `exit` ein, damit Sie sich als `jack` anmelden können. Es ist generell sinnvoll, wann immer möglich, als gewöhnlicher Benutzer und nicht als der allmächtige Benutzer `root` zu arbeiten.

Falls Sie schon einen Benutzer angelegt haben und diesem die Möglichkeit einräumen wollen, mit `su root`-Rechte zu erlangen, können Sie sich als `root` anmelden und die Datei /etc/group anpassen. Fügen Sie in der ersten Zeile (der Gruppe `wheel`) den Benutzer `jack` ein. Sie sollten auch den Umgang mit dem Texteditor vi(1) erlernen. Alternativ können Sie auch den einfacher zu bedienenden Texteditor ee(1) verwenden, der auf aktuellen FreeBSD-Systemen automatisch installiert wird.

Mit `rmuser` können Sie einen Benutzer löschen.

# 3. Sich im System umsehen

Nachdem wir uns als gewöhnlicher Benutzer angemeldet haben, sehen wir uns nun im System um und testen einige Befehle, die uns den Zugriff auf die Hilfefunktion ermöglichen sowie einige Informationen über unser System liefern.

`id`

: Sagt uns, wer wir sind!

`pwd`

: Zeigt uns, wo wir uns befinden, gibt also das aktuelle Verzeichnis (working directory) aus.

`ls`

: Listet die Dateien im aktuellen Verzeichnis auf.

`ls -F`

: Listet Dateien im aktuellen Verzeichnis auf und kennzeichnet ausführbare Dateien mit einem `\*`, Verzeichnisse mit einem `/`, sowie symbolische Links mit `@`.

`ls -l`

: Listet Dateiem im Langformat auf, dabei werden Größe, Datum und Rechte angezeigt.

`ls -a`

Listet auch versteckte Dateien auf (so genannte "dot"-Dateien). Falls Sie als `root` angemeldet sind, werden die "dot"-Dateien auch ohne die Option `-a` angezeigt.

`cd`

Wechselt das Verzeichnis. `cd ..` wechselt in der Verzeichnishierarchie eine Stufe nach oben (beachten Sie das Leerzeichen nach `cd`!). `cd /usr/local` wechselt dorthin. `cd ~` wechselt wechselt in des Heimatverzeichnis des angemeldeten Benutzers, in unserem Fall nach /usr/home/jack. Geben Sie `cd /cdrom`, gefolgt von `ls`, ein, um herauszufinden, ob Ihr CD-ROM-Laufwerk eingebunden ist und funktioniert.

`view Dateiname`

Zeigt den Inhalt von *Dateiname* auf dem Bildschirm an. Falls der Inhalt der Datei zu lang ist, und Sie nur das Ende sehen, aktivieren Sie die `Rollen`-Taste und drücken dann die `Bild hoch`-Taste, um nach oben zu scrollen. Sie können die `Rollen`-Taste auch bei Manualpages verwenden. Drücken Sie die `Rollen`-Taste nochmals, um das Scrollen zu beenden. Sie sollten sich auch einige der versteckten Dateien in Ihrem Heimatverzeichnis mit `cat` ansehen: `cat .cshrc`, `cat .login`, `cat .profile`.

In .cshrc sind einige nützliche Aliase für den Befehl `ls` definiert. Dort können Sie auch zusätzliche Aliase definieren. Um diese Aliase allen Benutzern zur Verfügung zu stellen, müssen Sie diese in /etc/csh.cshrc, der systemweiten Konfigurationsdatei von `csh`, eintragen.

# 4. Hilfe und Informationen erhalten

Im folgenden Abschnitt erhalten Sie einige Tipps zur Verwendung der Hilfefunktion. *Text* steht dabei für etwas, das Sie eingeben müssen - meistens einen Befehl oder einen Dateinamen.

`apropos text`

Jeder Eintrag, der die Zeichenkette *text* in der `whatis`-Datenbank enthält, wird aufgelistet.

`man text`

Die Manualpage von *text*. Manualpages sind die Hauptinformationsquelle für UNIX®-Systeme. `man ls` zeigt alle Möglichkeiten, wie `ls` verwendet werden kann. Drücken Sie `Enter` um weiterzublättern, `Strg` + `B`, um eine Seite zurück zu blättern, `Strg` + `F`, um eine Seite vorwärts zu blättern oder `Strg` + `C` zum Beenden.

`which text`

Sagt Ihnen, in welchem Verzeichnis der Befehl *text* im Pfad des Benutzers gefunden wird.

`locate text`

Alle Verzeichnisse, welche die Zeichenkette *text* enthalten, werden ausgegeben.

`whatis text`

Gibt aus, wofür das Kommando *text* steht und welche Handbuchseite(n) es dafür gibt. `whatis *` liefert Ihnen Informationen zu allen ausführbaren Dateien des aktuellen Verzeichnisses.

`whereis text`

Gibt den absoluten Pfad der Datei *text* aus.

Rufen Sie nun `whatis` mit einigen nützlichen Befehlen wie `cat`, `more`, `grep`, `mv`, `find`, `tar`, `chmod`, `chown`, `date`, und `script` auf. `more` funktioniert genau so wie unter DOS und zeigt den Inhalt der Datei seitenweise an. `ls -l | more` entspricht daher `more Dateiname`. Das *-Zeichen dient als Platzhalter, `ls w*` zeigt daher alle Dateien an, die mit `w` beginnen.

Funktioniert der eine oder andere Aufruf bei Ihnen nicht? Sowohl locate(1), als auch whatis(1) sind von einer Datenbank abhängig, die wöchentlich aktualisiert wird. Falls Ihr Rechner nicht ständig läuft, können Sie die täglichen, wöchentlichen und monatlichen Aktualisierungen auch manuell starten. Melden Sie sich dazu als `root` an. Warten Sie jeweils auf das Ende eines Befehls, bevor Sie den nächsten Befehl eingeben.

```
# periodic daily
output omitted
# periodic weekly
output omitted
# periodic monthly
output omitted
```

Diese Befehle sind zwar Teil der Systemverwaltung, als alleiniger Benutzer eines UNIX®-Systems sind Sie aber auch der Administrator des Systems. Sie müssen sich nur als `root` anmelden, wenn Sie das System verwalten möchten. Eine Information, die leider in vielen dicken Büchern über UNIX® vergessen wird, die sich lieber den Pulldown-Menüs und Window-Managern widmen. Empfehlenswerte Bücher zur UNIX®-Systemadministration sind das UNIX System Administration Handbook von Evi Nemeth et.al. (Prentice-Hall, 1995, ISBN 0-13-15051-7, die zweite Ausgabe hat ein rotes Cover), oder Essential System Administration von Æleen Frisch (O'Reilly & Associates, 2002, ISBN 0-596-00343-9).

# 5. Textdateien bearbeiten

Um Ihr System konfigurieren zu können, müssen Sie häufig Textdateien bearbeiten. Die meisten Konfigurationsdateien befinden sich im Verzeichnis /etc. Um diese editieren zu können, müssen Sie mit `su` zu `root` werden. Sie können den einfachen Editor `ee` verwenden, auf lange Sicht lohnt es sich aber, die Bedienung von `vi` zu erlernen. Falls Sie die Systemquellen installiert haben, befindet sich unter /usr/src/contrib/nvi/docs/tutorial ein ausgezeichnetes Tutorial zu `vi`.

Bevor Sie eine Datei bearbeiten, sollten Sie eine Sicherheitskopie der Datei anlegen. Wenn Sie beispielsweise /etc/rc.conf bearbeiten möchten, wechseln Sie mit `cd /etc` nach /etc und geben Folgendes ein:

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

Dadurch wird eine Kopie von rc.conf mit dem Namen rc.conf.orig angelegt, mit der Sie notfalls das Original wiederherstellen können, indem Sie etc.conf.orig nach etc.conf kopieren. Noch besser ist

es, die Datei zuerst zu verschieben (umzubenennen) und dann zu kopieren:

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

weil bei der Verwendung von `mv` das Datum und der Besitzer der Datei erhalten bleiben. Danach können Sie die Datei rc.conf bearbeiten. Möchten Sie die Originaldatei wiederherstellen, geben Sie `mv rc.conf rc.conf.myedit` (falls Sie die bearbeitete Version erhalten möchten), gefolgt von:

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

ein, um den Originalzustand wiederherzustellen.

Um eine Datei zu bearbeiten, geben Sie

```
# vi filename
```

ein.

Innerhalb des Textes können Sie sich mit den Pfeiltasten bewegen. Mit der `Esc`-Taste wechselt `vi` in den Kommandomodus. Einige der wichtigsten Kommandos sind:

`x`
: löscht den Buchstaben links vom Cursor

`dd`
: löscht die komplette Zeile, auch wenn diese am Bildschirm umgebrochen wird.

`i`
: fügt Text an der Cursorposition ein.

`a`
: fügt Text rechts vom Cursor ein.

Nachdem Sie `i` oder `a` eingegeben haben, können Sie also die Datei bearbeiten. Mit `Esc` kommen Sie wieder in den Kommandomodus, wo Sie beispielsweise

`:w`
: eingeben, um Ihre Änderungen zu speichern und mit `vi` weiterarbeiten können.

`:wq`
: Speichern und Beenden.

`:q!`
: Beenden, ohne die Änderungen zu speichern.

`/text`

bewegt den Cursor an die Stelle des ersten Vorkommens von *text*; `/Enter` (die Enter-Taste) findet dann das nächste Vorkommen von *text*.

`G`

bewegt den Cursor ans Ende der Datei.

`nG`

bewegt den Cursor in die Zeile *n* der Datei, wobei *n* für eine Zahl steht.

`Ctrl` + `L`

zeichnet den Bildschirm neu.

`Ctrl` + `b` **und** `Ctrl` + `f`

eine Bildschirmseite zurück oder vorwärts gehen, analog zu `more` und `view`.

Am besten sammeln Sie praktische Erfahrungen mit `vi` in Ihrem Heimatverzeichnis, indem Sie eine Datei anlegen und danach bearbeiten, die Datei speichern und wieder aufrufen. `vi` birgt wirklich einige Überraschungen, da es sich um ein komplexes Programm handelt. Manchmal werden Sie versehentlich ein Kommando ausführen, das etwas anderes tut, als Sie erwarten. (`vi` ist viel leistungsfähiger als der DOS-Editor EDIT. Rufen Sie doch einmal das Kommando `:r` auf.) Drücken Sie die `Esc`-Taste ruhig öfter als einmal, um sicher zu gehen, dass Sie sich im Kommandomodus befinden und arbeiten Sie in diesem Modus weiter, falls Probleme auftreten. Speichern Sie die Datei immer wieder mit `:w`, und geben Sie `:q!` ein, um von vorne zu beginnen (genauer gesagt, vom letzten `:w`-Kommando), falls dies nötig sein sollte.

Danach können Sie mit `cd` nach /etc wechseln, mit `su` zu `root` werden, und mit `vi` /etc/group bearbeiten und einen Benutzer zur Gruppe `wheel` hinzufügen, damit dieser mit `su root`-Rechte erlangen kann. Fügen Sie dafür einfach ein Komma und den Login-Namen des Benutzers an das Ende der ersten Zeile an, drücken Sie `Esc` und `:wq` (um die Datei zu speichern und den `vi` zu verlassen). Sehr effizient, nicht wahr? (Sie haben hoffentlich kein Leerzeichen nach dem Komma eingefügt?)

# 6. Dateien mit Hilfe von DOS ausdrucken

Bis zu diesem Zeitpunkt haben Sie wahrscheinlich noch keinen Drucker eingerichtet, daher zeigen wir Ihnen, wie man eine Datei aus einer Manualpage erzeugt, diese auf eine Diskette kopiert und dann unter DOS ausdruckt. Möchten Sie etwa nachlesen, wie Sie Dateirechte verändern können (was sehr wichtig ist), rufen Sie mit `man chmod` die entsprechende Manualpage auf. Der Befehl

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

entfernt alle Formatierungen und leitet die Ausgabe der Manualpage nach chmod.txt um, statt diese auf dem Bildschirm anzuzeigen. Legen Sie danach eine DOS-formatierte Diskette in Ihr Diskettenlaufwerk a ein und geben Sie `su` ein, um zu `root` zu werden. Tippen Sie nun

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

ein, um das Diskettenlaufwerk unter /mnt einzuhängen.

Da Sie ab nun keine `root`-Rechte mehr benötigen, werden Sie durch die Eingabe von `exit` wieder zu `jack` und wechseln dann in das Verzeichnis, in dem sich chmod.txt befindet, und kopieren diese Datei mit

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

auf Ihre Diskette. Zeigen Sie mit `ls /mnt` den Inhalt von /mnt an. Als Ergebnis sollten Sie chmod.txt erhalten.

Leiten Sie nun die Ausgabe von `/sbin/dmesg` in eine Datei um, indem Sie

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

eingeben und diese Datei ebenfalls auf die Diskette kopieren. Mit `/sbin/dmesg` können Sie alle während des Systemstarts angezeigten Meldungen ausgeben. Es ist wichtig, dass Sie diese Meldungen verstehen, da hier angezeigt wird, welche Hardware von FreeBSD beim Systemstart gefunden wurde. Falls Sie Fragen auf der Mailingliste Fragen und Antworten zu FreeBSD <de-bsd-questions@de.FreeBSD.org> oder im USENET stellen (etwa "FreeBSD erkennt mein Bandlaufwerk nicht, was soll ich tun?"), wird man Sie fragen, was `dmesg` ausgibt.

Anschließend können Sie das Diskettenlaufwerk wieder aus dem Verzeichnisbaum aushängen (unmounten), um die Diskette zu entfernen. Dies funktioniert natürlich nur als `root`:

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

Danach starten Sie DOS (Windows®). Kopieren Sie die Dateien in ein Verzeichnis. Nun öffnen Sie die Dateien mit DOS EDIT, Windows® Notepad, Wordpad oder einem anderen Schreibprogramm, speichern die Datei, falls Sie kleinere Änderungen vornehmen wollen, und drucken die Datei unter DOS oder Windows aus. Manualpages werden unter DOS am besten mit `print` ausgedruckt.

Um Ihren Drucker unter FreeBSD einzurichten, muss ein entsprechender Eintrag in /etc/printcap und ein Spool-Verzeichnis unter /var/spool/output angelegt werden. Falls der Drucker an `lpt0` (entspricht LPT1 unter DOS) angeschlossen ist, müssen Sie wahrscheinlich nur nach /var/spool/output wechseln und (als `root`) das Verzeichnis lpd mit `mkdir lpd` anlegen, falls es nicht schon vorhanden ist. Danach sollte sich der Drucker, wenn er eingeschaltet ist, beim Booten melden und `lp` oder `lpr` sollte eine Datei zum Drucker schicken und ausdrucken. Ob die Datei schließlich ausgedruckt wird, hängt von der Konfiguration des Druckers ab, die im FreeBSD Handbuch ausführlich beschrieben ist.

# 7. Weitere nützliche Befehle

`df`

zeigt den freien Speicherplatz und eingehängte (gemountete) Partitionen an.

`ps aux`

zeigt laufende Prozesse an. `ps ax` schränkt diese Ausgabe etwas ein.

`rm dateiname`

löscht *dateiname*.

`rm -R dir`

löscht das Verzeichnis *dir* und alle Unterverzeichnisse - Vorsicht!

`ls -R`

listet Dateien im aktuellen Verzeichnis und allen Unterverzeichnissen auf. Die Variante `ls -AFR > where.txt` kann dazu verwendet werden, um eine Liste aller Dateien unter / und (separat) unter /usr zu erhalten. Allerdings gibt es bessere Möglichkeiten, bestimmte Dateien zu finden.

`passwd`

zum Ändern des Passworts eines Benutzers (oder von `root`).

`man hier`

die Manualpage des UNIX®-Dateisystems.

Nutzen Sie `find`, um unter /usr oder anderen Verzeichnissen nach dateiname zu suchen:

```
% find /usr -name "dateiname>"
```

Als Platzhalter können Sie bei `"dateiname"` einen `\*` verwenden. (*dateiname* muss dabei in Hochkommata stehen.) Weisen Sie `find` an, in / statt in /usr zu suchen, werden alle eingehängten Dateisysteme durchsucht, CD-ROM und DOS-Partitionen eingeschlossen.

Ein ausgezeichnetes Buch, in dem sämtliche UNIX®-Befehle beschrieben werden, ist Unix for the Impatient von Abrahams & Larson (2nd ed., Addison-Wesley, 1996). Auch im Internet finden Sie viele Informationen zu UNIX®.

# 8. Die nächsten Schritte

Sie sollten nun alle wichtigen Werkzeuge kennen, um sich auf Ihrem System zurechtzufinden, Dateien zu editieren und alles am Laufen zu halten. Viele nützliche Informationen finden sich im FreeBSD-Handbuch (das vielleicht schon auf Ihrer Festplatte ist) und auf der FreeBSD Webseite. Viele Softwarepakte und Ports sind sowohl auf der FreeBSD-CD-ROM als auch auf der Webseite verfügbar. Im Handbuch finden Sie Hinweise, wie Sie mit diesen arbeiten können. Pakete installieren Sie mit `pkg_add /cdrom/packages/All/Paketname`, wobei *Paketname* für den Dateinamen des Paketes steht. Auf der CD-ROM finden Sie auch eine Liste der Pakete und Ports inklusive einer

kurzen Beschreibung. Lesen Sie dazu cdrom/packages/index, cdrom/packages/index.txt sowie cdrom/ports/index. Eine ausführliche Beschreibung den jeweiligen Ports finden Sie in /cdrom/ports/*/*/pkg-descr, wobei die *-Zeichen für die Kategorie und das Unterverzeichnis des jeweiligen Programms stehen.

Falls Ihnen die Beschreibung zur Installation von Ports von der CD-ROM im Handbuch zu kompliziert erscheint (mit Programmen wie `lndir` und anderen), finden Sie hier eine alternative Beschreibung, die in der Regel problemlos funktioniert:

Suchen Sie zuerst den Port, den Sie installieren möchten, etwa `kermit`. Auf der CD-ROM sollte dafür ein entsprechendes Verzeichnis vorhanden sein. Kopieren Sie dieses Unterverzeichnis nach /usr/local (ein guter Platz für Programme, die hinzugefügt werden und allen Benutzern zugänglich sein sollen):

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

Danach wird das Verzeichnis /usr/local/kermit angelegt, in dem sich alle Dateien des Unterverzeichnisses kermit der CD-ROM finden.

Danach legen Sie mit `mkdir` das Verzeichnis /usr/ports/distfiles an, falls dieses noch nicht existiert. Suchen Sie dann in /cdrom/ports/distfiles nach der Datei, die dem Namen des Ports gleicht, den Sie installieren möchten. Kopieren Sie diese Datei nach /usr/ports/distfiles. Unter neueren Versionen können Sie diesen Schritt überspringen, da FreeBSD dies automatisch erledigt.

Wechseln Sie nun mit `cd` nach /usr/local/kermit. In diesem Verzeichnis befindet sich bereits ein Makefile. Geben Sie hier Folgendes ein:

```
# make all install
```

Während der Installation wird sich der Port alle gepackten Dateien via FTP holen, die zur Installation benötigt werden, aber nicht auf der CD-ROM oder unter /usr/ports/distfiles vorhanden sind. Falls Sie noch keine Internetverbindung haben und die benötigten Dateien nicht in /cdrom/ports/distfiles vorhanden sind, müssen Sie sich die Dateien über einem anderen Rechner besorgen und manuell nach /usr/ports/distfiles kopieren (entweder über eine Diskette oder durch das Einhängen einer DOS-Partition. Sehen Sie im Makefile (mit `cat`, `more` oder `view`) nach, wo die Dateien zu finden sind (meist auf der Hauptseite des Programms) und besorgen Sie sich die Datei. Laden Sie die Datei unter DOS herunter, wird der Dateiname möglicherweise gekürzt. In diesem Fall müssen Sie, nachdem die Datei nach /usr/ports/distfiles kopiert wurde, den ursprünglichen Namen mit `mv` wiederherstellen, damit die Datei von der Installationsroutine gefunden werden kann. Laden Sie die Datei mit FTP herunter, müssen Sie den Binär-Modus verwenden. Nun wechseln Sie zurück nach /usr/local/kermit (hier befindet sich ja das zur Installation nötige Makefile) und installieren das Programm mit `make all install`.

Beim Installieren von Ports oder Paketen kann es auch passieren, dass andere Programme benötigt werden. Falls die Installation beispielsweise mit `can't find unzip` oder einer ähnlichen Meldung abbricht, müssen Sie zuerst das unzip-Paket oder den entsprechende Port installieren, bevor Sie mit der Installation fortfahren können.

Ist die Installation abgeschlossen, geben Sie `rehash` ein, damit FreeBSD den Pfad neu einliest und weiß, wo welche Programme zu finden sind. (Falls beim Ausführen von `whereis` oder `which` viele Fehlermeldungen wie `path not found` auftreten, müssen Sie die Pfad-Anweisungen in Ihrer .cshrc im Heimatverzeichnis erweitern. Die Pfad-Anweisung tut unter UNIX® dasselbe wie unter DOS. Das aktuelle Verzeichnis ist allerdings aus Sicherheitsgründen nicht im voreingestellten Ausführungspfad enthalten. Befindet sich die auszuführende Datei im aktuellen Verzeichnis, muss ./ vor dem Befehl eingegeben werden, (wobei nach dem Slash kein Leerzeichen stehen darf), damit die Datei ausgeführt wird.)

Vielleicht möchten Sie auch die neueste Version des Netscape®-Browsers (zu finden auf der FTP-Seite von Netscape®) installieren, von der es auch eine FreeBSD-Version gibt. Dazu müssen Sie allerdings das X Window-System installiert haben. Nach dem Herunterladen verwenden Sie `gunzip dateiname` und `tar xvf dateiname`, um die Datei zu entpacken. Danach kopieren Sie die Binärdatei nach /usr/local/bin oder ein anderes Verzeichnis Ihres Pfades, und geben abschließend `rehash` ein. Fügen Sie dann in /etc/csh.cshrc, der systemweiten Konfigurationsdatei von `csh`, folgende Zeilen ein:

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

Dabei wird davon ausgegangen, dass XKeysymDB und das Verzeichnis nls sich unter /usr/X11R6/lib/X11 befinden. Ist dies auf Ihrem System nicht der Fall, müssen Sie diese finden und dorthin kopieren.

Falls Sie Netscape® schon als Port von der CD-ROM oder über FTP installiert haben, achten Sie unbedingt darauf, nicht versehentlich /usr/local/bin/netscape durch die neue Binärdatei zu ersetzen, da es sich bei dieser Datei nur um ein Shell-Skript handelt, das die Umgebungsvariablen für Sie setzt. Nennen Sie die neue Binärdatei stattdessen besser netscape.bin und ersetzen Sie die alte Version /usr/local/netscape/netscape.

# 9. Die Arbeitsumgebung

Die Shell ist das wichtigste Programm Ihrer Arbeitsumgebung. Unter DOS heißt die Shell command.com. Die Shell interpretiert die Befehle, die Sie auf der Kommandozeile eingeben und kommuniziert so mit dem Rest des Betriebssystems. Sie können auch Shellskripte schreiben, die den DOS-Batchdateien ähnlich sind. Es handelt sich dabei um eine Folge von Befehlen, die hintereinander ausgeführt werden.

Zwei Shells werden von FreeBSD automatisch installiert: `csh` und `sh`. `csh` ist für die Arbeit auf der Kommandozeile komfortabler, Skripten sollten allerdings für `sh` (oder für `bash`) geschrieben werden. Mit `echo $SHELL` können Sie herausfinden, welche Shell benutzt wird.

Die `csh`-Shell ist zwar nicht schlecht, aber die `tcsh`-Shell kann alles, was die `csh` kann und einiges mehr. Sie erlaubt es Ihnen, ausgeführte Befehle mit den Pfeiltasten wieder aufzurufen und diese sogar zu verändern. Dateinamen oder Befehle können mit der Tabulator-Taste automatisch vervollständigt werden (`csh` nutzt dafür `ESC`) und mit `cd -` können Sie schnell in das zuletzt verwendete Verzeichnis wechseln. Auch der Prompt lässt sich unter der `tcsh`-Shell wesentlich einfacher anpassen. All diese Eigenschaften erleichtern den Umgang mit dem Betriebssystem.

Wollen Sie eine neue Shell installieren, gehen Sie wie folgt vor:

1. Installieren Sie die Shell als Port oder Paket genau so, wie Sie jedes andere Programm als Port oder Paket installieren würden. Rufen Sie dann `rehash` sowie `which tcsh` auf (wir nehmen hier an, dass Sie `tcsh` installieren wollen), um sicher zu gehen, dass alles korrekt installiert wurde.
2. Als `root` editieren Sie nun /etc/shells, und fügen eine Zeile für die neue Shell ein, in unserem Fall also /usr/local/bin/tcsh. Danach speichern Sie die Datei (Einige Ports erledigen diesen Schritt automatisch für Sie.).
3. Nutzen Sie `chsh`, um `tcsh` dauerhaft als Ihre Shell zu definieren. Alternativ geben Sie einfach `tcsh` ein, ohne sich neu am System anzumelden.

Bei früheren Versionen von FreeBSD und einigen anderen Versionen von UNIX® traten Probleme auf, wenn `root` eine andere Shell als `sh` oder `csh` zugewiesen wurde; sie hatten dann keine funktionierende Shell, wenn das System im Single-User-Modus ausgeführt wurde. Die Lösung besteht darin, `su -m` einzugeben, um `root` zu werden, weil damit `root` die `tcsh` zugewiesen wird, da diese Shell Teil der Umgebung ist. Dies kann auch fest als Alias in .tcshrc eingetragen werden:

```
alias su su -m
```

Wenn `tcsh` startet, werden /etc/csh.cshrc und /etc/csh.login eingelesen (analog zur `csh`). Ebenfalls eingelesen werden .login und .cshrc des Heimatverzeichnisses, falls .tcshrc nicht gefunden wurde. Sie können auch einfach .cshrc nach .tcshrc kopieren.

Da `tcsh` nun installiert ist, kann der Prompt angepasst werden. Details dazu finden Sie in tcsh(1). Die hier vorgeschlagene Zeile können Sie in Ihre .tcshrc eintragen. Dadurch wird angezeigt, wie viele Befehle bereits eingegeben wurden. Außerdem erhalten Sie die aktuelle Uhrzeit sowie das aktuelle Arbeitsverzeichnis. Für den gewöhnlichen Benutzer wird ein `>` und für `root` ein `#` ausgegeben. Dabei handelt es sich um die Standardeinstellung von `tcsh`:

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

Diese Zeile sollte an derselben Stelle eingefügt werden, wie die vorhandene "set prompt" Zeile oder nach "if($?prompt) then". Kommentieren Sie die alte Zeile aus, damit diese notfalls wieder benutzt werden kann. Vergessen Sie auch auf keinen Fall die Leerzeichen und Hochkommata. .tcshrc wird neu eingelesen, wenn Sie `source .tcshrc` eingeben.

Eine Liste aller Umgebungsvariablen erhalten Sie, wenn Sie am Prompt `env` eingeben. Sie sehen dann den Default-Editor, -Pager, -Terminaltyp und viele andere Umgebungsvariablen. Falls Sie sich von einem entfernten Rechner anmelden und ein Programm nicht starten können, weil das Terminal dazu nicht in der Lage ist, kann die Eingabe von `setenv TERM vt100` sehr hilfreich sein.

# 10. Einige abschließende Bemerkungen

Als `root` können Sie das CD-ROM-Laufwerk mit `/sbin/umount /cdrom` abhängen, eine andere CD-ROM einlegen und mit `/sbin/mount_cd9660 /dev/cd0a /cdrom` wieder einhängen (falls `cd0a` der Devicename Ihres CD-ROM-Laufwerkes ist). Bei neueren FreeBSD-Versionen muss dafür nur `/sbin/mount /cdrom` eingegeben werden.

Das Live-Filesystem (die zweite FreeBSD-CD-ROM) kann hilfreich sein, wenn auf Ihrem System nur begrenzt Speicherplatz vorhanden ist. Was genau auf dieser Live-CD enthalten ist, variiert von Ausgabe zu Ausgabe. Vielleicht können verschiedene Spiele von CD-ROM gespielt werden. Dazu muss aber auch `lndir` vorhanden sein, damit diese Programme wissen, wo sie die von ihnen benötigten Dateien finden, da sich diese unter /cdrom und nicht unter /usr und den entsprechenden Unterverzeichnissen befinden, wie es diese Programme erwarten. Lesen Sie dazu auch lndir(1).

# 11. Kommentare

Falls Sie diese einführende Anleitung benutzen, würde es mich interessieren, ob Dinge schlecht erklärt sind, was noch ergänzt werden sollte, und ob Ihnen diese Einführung geholfen hat. Ich danke Eugene W. Stark, Informatikprofessor am SUNY-Stony Brook, sowie John Fieber für ihre hilfreichen Kommentare.

Annelise Anderson, andrsn@andrsn.stanford.edu