

Die foo-Bibel

Torben Nehmer

Torben.Nehmer@gmx.net, nehmer@foo.fh-furtwangen.de

Erstellt am 14. Dezember 1999

Inhaltsverzeichnis

1	Einführung	5
1.1	Der Computer	5
1.2	Der foo-Pool — Einstieg und Regeln	5
1.3	Was ist Unix?	6
1.4	Unixe im foo-Pool	6
1.4.1	Ultrix	6
1.4.2	DEC Unix (OSF/1)	6
1.4.3	HP Unix (HP-UX)	6
1.4.4	Sun Solaris (SunOS)	6
1.4.5	NetBSD	6
1.4.6	Linux	6
1.5	Das Ziel: Die transparente Umgebung	7
2	Erste Schritte mit Unix	9
2.1	Benutzeraccounts, Login und Paßwort	9
2.1.1	Grundlagen	9
2.1.2	Der Login	9
2.1.3	Das Paßwort	10
2.2	Graphische Benutzeroberflächen	10
2.2.1	Die graphische Benutzeroberfläche mit fvwm	11
2.2.2	Common Desktop Environment (CDE)	13
2.2.3	Beenden der X-Session	15
2.3	Die Shell	15
2.3.1	Starten einer Shell bzw. eines Terminals	16
2.3.2	Der Prompt	16
2.3.3	Grundlegende Kommandos	17
2.3.4	Sonderzeichen in einer Shell	19
2.4	Dateisysteme im Foo-Pool	20
2.5	Zugriffsrechte unter Unix	22
2.5.1	Gruppen und deren Benutzer	22
2.5.2	Dateien und ihre Berechtigungen	22
2.5.3	Ändern der Zugriffsrechte	23
2.5.4	Ändern der Gruppenzuordnung	24
2.6	Dokumentation und Online-Hilfen	24
2.6.1	man-Pages	24
2.6.2	GNU-Dokumentation (info-Pages)	27
2.6.3	Weitere Dokumentation	28
2.6.4	FAQ - Frequently Asked Questions	28

2.6.5	HOWTO - Wie geht das?	29
2.6.6	LDP - Das Linux Documentation Project	29
2.6.7	Dokumentation zu einzelnen Programmpaketen	29
2.7	Environment (Die Umgebung)	29
2.7.1	Anzeigen von Umgebungsvariablen	30
2.7.2	Setzen von Umgebungsvariablen	30
2.7.3	Löschen von Umgebungsvariablen	30
2.7.4	Wichtige Umgebungsvariablen	31
2.8	Prozesse und deren Steuerung	32
2.8.1	Anzeigen von Prozessinformationen	32
2.8.2	Ausser Kontrolle geratene Prozesse beenden	33
2.9	Drucken im Foo-Pool	33
2.9.1	Queues (Warteschlangen)	33
2.9.2	Druckjobs abschicken	34
2.9.3	Wartende Druckjobs anzeigen	34
2.9.4	Eigene Druckjobs aus der Warteschlange löschen	35
2.10	Nutzen anderer Rechner (ssh, rlogin, rsh, telnet)	35
2.10.1	Remoteshell-Typen	35
2.10.2	Grundlagen	36
2.10.3	Direktes Ausführen einzelner Programme	37
2.10.4	Umleiten graphischer Anwendungen	38
2.11	Datenaustausch zwischen (Win)DOS und UNIX	39
2.11.1	Umlaute	39
2.11.2	ASCII-Dateiformat	39
2.11.3	Die mtools	40
2.12	foo-Kommandoübersicht	41
3	Das Internet	43
3.1	WWW: World Wide Web	43
3.1.1	Grundkonfiguration: Proxies, Netscape-Cache, Smart Browsing	45
3.1.2	Bookmarks	46
3.1.3	Interessante URLs	48
3.2	E-Mail	48
3.2.1	Konfiguration des Netscape E-Mail Client	49
3.2.2	E-Mails versenden und empfangen	50
3.2.3	Mehrere Empfänger	53
3.2.4	Attachments (Anhänge)	53
3.2.5	Adreß-Suche mit finger	54
3.2.6	Wichtige E-Mail-Adressen	55
3.3	Internet News (Usenet) - Die Diskussionsforen	55

3.3.1	Konfiguration des Netscape News-Clients	56
3.3.2	News lesen/schreiben	57
3.3.3	Intessante Newsgroups	58
3.4	FTP - File Transfer Protocol	58
3.4.1	FTP in der Shell	59
3.4.2	Anonymous FTP mit Netscape	61
3.4.3	Interessante FTP-Server	61
A	Weitere interessante Programme	63
B	vi Kommandoübersicht	64
C	emacs Kommandoübersicht	65

1 Einführung

Willkommen im foo-Pool. Dieses Buch soll Dir einen Einstieg in den foo-Pool vermitteln.

1.1 Der Computer

Ein funktionsfähiger Computer läßt sich in zwei Sachgebiete unterteilen. Zum einen ist da die Hardware, also die technischen Komponenten eines Rechners. Dies sind beispielsweise Tastatur, Monitor, Maus, Prozessor, Speicher, Diskettenlaufwerke usw. Damit diese ganzen Gerätschaften überhaupt zusammenarbeiten, werden Programme benötigt, die vom Prozessor ausgeführt werden. Diese nennt man Software. Die grundlegenden Koordinationsaufgaben führt das Betriebssystem durch. Es leitet beispielsweise Tastatureingaben an das laufende Programm weiter und schickt dessen Ausgaben auf den Bildschirm.

1.2 Der foo-Pool — Einstieg und Regeln

Im foo-Pool laufen primär Rechner mit Betriebssystemen der Unix-Familie. Bevor jedoch der nähere Umgang mit dem foo-Pool gezeigt wird, möchte ich Dich zuerst auf einige wichtige Verhaltensregeln aufmerksam machen.

- **Die Workstations im foo-Pool dürfen NIE ausgeschaltet werden!!!** Workstations sind für den Dauerbetrieb konfiguriert. Lediglich die Systemadministratoren dürfen in Ausnahmefällen die Workstations herunterfahren.
- Falls Du der Meinung bist, daß eine Maschine defekt ist, versuche bitte nicht, die Maschine selbst zu reparieren. Sollte keiner der Systemadministratoren erreichbar sein, schreibe eine Nachricht via E-Mail (siehe Kapitel 3.2) an `sysadmin@foo.fh-furtwangen.de`.
- Bei einem Teil der im foo-Pool verwendeten Software handelt es sich um freie Software¹. Diese kannst Du nach belieben kopieren und weiterverwenden. Für andere Programme (z.B. Maple) trifft dies nicht zu. Diese dürfen zwar im Labor freizügig verwendet werden, kopieren ist aber nicht erlaubt. Wenn Du Dir nicht sicher bist, ob ein bestimmtes Programm nun frei ist, oder nicht, frage einfach einen der Sysadmins.
- Unterhalte Dich leise und nimm Rücksicht auf andere Pool-Nutzer.
- Belege keine Rechner, wenn Du erst in einer Stunde damit arbeiten willst. Breite Dich nicht soweit aus, daß Du Deinen Nachbarn störst. Verlasse Deinen Arbeitsplatz ordentlich.
- Wenn Du Deinen Arbeitsplatz für kurze Zeit verlassen willst, sperre nicht den Rechner. Im foo-Pool sollte jeder Rücksicht auf andere nehmen. Ebenso sollte keiner einfach andere User ausloggen, nur weil sie gerade kurz weggegangen sind. Sollte tatsächlich mal ein Rechner für längere Zeit belegt sein, ohne daß jemand davor sitzt, sag einfach den Sysadmins bescheid.

¹Freie Software kann von jedem benutzt und frei weitergegeben werden. Sie ist im Quellcode verfügbar. Die Idee der freien Software ist, daß jeder auf den Ergebnissen anderer aufbauen kann, jedoch muß er die Resultate wieder (im Quellcode) der Allgemeinheit zur Verfügung stellen. Siehe www.gnu.org

1.3 Was ist Unix?

Spricht man von Unix, ist eigentlich eine ganze Familie von Betriebssystemen gemeint. Alle besitzen die gleichen Grundzüge, sind untereinander weitestgehend kompatibel, unterscheiden sich jedoch in ihrem internen Aufbau. Der wohl größte Vorteil von Unix-Systemen sind die Netzwerkfähigkeiten, die bereits im innersten Kern des Betriebssystems verankert sind. Unix wurde von Anfang an auf den Einsatz in einem Netzwerk konzipiert. Dies bringt zwei wichtige Fähigkeiten mit sich: Auf einem Rechner können sowohl mehrere Programme gleichzeitig abgearbeitet werden („multiprocessing“, siehe Kapitel 2.8) als auch mehrere Benutzer gleichzeitig arbeiten („multiuser“, siehe Kapitel 2.10). Bekannte Unixe sind SunOS (auch als Sun Solaris bekannt), HP-UX und, für den Privatanwender, Linux.

1.4 Unixe im foo-Pool

1.4.1 Ultrix

Auf vielen Rechnern im Pool läuft derzeit Ultrix 4.5. Dies ist, außer NetBSD, das einzige Unix, das auf dec-mips Prozessoren (alle DECStations) läuft. Ein weiterer Grund für den Einsatz im foo-Pool ist seine relative Unbekanntheit, wodurch diese Systeme recht sicher gegen Angriffe von außen sind. Derzeit sind dies die Rechner foo1, foo2, shire, hanne, timbuktu. Das Hauptproblem von Ultrix ist derzeit das Jahr 2000.

1.4.2 DEC Unix (OSF/1)

Die neueren DEC's mit Alpha-Architektur (grandma, cimba) arbeiten derzeit mit OSF/1. Dieses Betriebssystem ist auch unter DEC Unix oder True64 bekannt.

1.4.3 HP Unix (HP-UX)

Im Pool besitzen wir zwei HP-Workstations, die derzeit mit HP-UX gefahren werden: tutsie und hanoi.

1.4.4 Sun Solaris (SunOS)

Als neuste Errungenschaft darf sich der foo-Pool über zwei Sun Workstations freuen: father und son, beide mit Sun Solaris 7. Deren Rechenleistung wird derzeit genutzt um leistungsintensive Programme wie den Netscape Navigator ins Netz zu exportieren. Weiterhin ist die x86-Version von Sun Solaris auf arath installiert.

1.4.5 NetBSD

Als Arbeitsstation unter NetBSD dienen nettie, k2, marie, tarzan, jane, pat und patachon. Weiterhin arbeiten unsere Network Computer mit NetBSD.

1.4.6 Linux

Auf den im foo benutzten Intel Plattformen arbeitet derzeit RedHat Linux. Dies sind merlin, mascha, balu und goedel. Zusätzlich dienen zwei weitere Linux-Rechner als Boot-Server für die Network Computer.

1.5 Das Ziel: Die transparente Umgebung

Bei einer solchen Vielfalt an verschiedenen Betriebssystemen stellt sich dem OttoNormalAnwender zwangsläufig die Frage nach der Konsistenz der Benutzeroberfläche. Die gesamte Konfiguration der Rechner im foo-Pool ist darauf ausgelegt, einem bestimmten Benutzer auf jedem beliebigen Rechner die gleiche Benutzeroberfläche zu liefern. Dies funktioniert im foo-Pool auch weitestgehend. Als kleines Problem stellt sich immer wieder die CDE-Oberfläche von HP-UX und SunOS heraus. Aus diesem Grunde ist oftmals der Ausweg über eine Kommandozeile hilfreich.

2 Erste Schritte mit Unix

Dieses Kapitel ist für all diejenigen gedacht, die bisher noch nicht mit Unix-Systemen gearbeitet haben. Ich möchte dieses Kapitel besonders denjenigen ans Herz legen, die aus der Windows-Welt kommen. Einige Konzepte mögen dem gewöhnlichen Windows-Nutzer bekannt vorkommen, haben jedoch oft einen völlig anderen Hintergrund.

2.1 Benutzeraccounts, Login und Paßwort

2.1.1 Grundlagen

Bevor man überhaupt mit einem Unix-System arbeiten kann, benötigt man einen Benutzeraccount (Benutzerkonto). Dieser besteht aus Loginname, Paßwort, einem eigenen Verzeichnis (siehe Kapitel 2.4) und einem E-Mail Account (siehe Kapitel 3.2). Deinen Benutzernamen und ein vorläufiges Paßwort erhältst Du vom Systemadministrator. Da man ohne einen Account auf einem Unix-System keine Rechte besitzt, sind diese Daten ziemlich wichtig.

2.1.2 Der Login

```
Red Hat Linux release 5.1 (Manhattan)
Kernel 2.0.34 on an i586

merlin login: nehmer Return
Password: (wird nicht angezeigt) Return

.
(Login wird durchgeführt, einige Meldungen erscheinen, darunter die Message Of The Day, mit deren Hilfe die Sysadmins aktuelle Meldungen verbreiten können. Man sollte also ab und an einen Blick darauf werfen.)
.

merlin:/ce_users/nehmer> startx Return

(Die graphische Benutzeroberfläche wird gestartet)
```

Abbildung 1: Login unter Linux mit Start der graphischen Benutzeroberfläche.

Die Rechner sind immer eingeschaltet. Sollte man trotzdem nur einen dunklen Bildschirm vorfinden, existieren zwei Möglichkeiten: Einerseits könnte der Bildschirm noch ausgeschaltet sein. Andererseits aktiviert sich nach einigen Minuten automatisch der Bildschirmschoner. In letzterem Falle genügt ein Tastendruck oder eine Mausbewegung um den Bildschirm zu aktivieren.

Läuft der Monitor bekommt man vom Rechner die Aufforderung, Loginname und Paßwort einzugeben („login“ und „password“). Das Aussehen der Eingabemaske variiert von Betriebssystem zu Betriebssystem. Ultrix, OSF/1 und NetBSD präsentieren ein einziges Dialogfeld, in dem nacheinander Benutzername und Paßwort eingegeben werden, CDE-Systeme wie HP-UX und SunOS zeigen zwei Dialogfelder nacheinander und Linux zeigt eine Eingabeaufforderung im Textmodus. Geben Sie also Benutzernamen und Paßwort nacheinander jeweils mit Return abgeschlossen ein.

```
timbuktu:/ce_users/nehmer> yppasswd (Return)
Changing yp password for nehmer
Old yp password: (Altes Paßwort eingeben) (Return)
New password: (Neues Paßwort eingeben) (Return)
Retype new password: (Neues Paßwort nochmal eingeben) (Return)
yellow pages passwd changed on foo2
timbuktu:/ce_users/nehmer>
```

Abbildung 2: Ändern des Paßwortes mit yppasswd

Wurde die Eingabe korrekt durchgeführt, wird die graphische Benutzeroberfläche bei allen Systemen außer Linux automatisch geladen. Unter Linux erscheint nach dem Login der Kommandoprompt, an dem mit `startx` die graphische Benutzeroberfläche gestartet werden kann. Dieser Vorgang wird in Abbildung 1 gezeigt.

2.1.3 Das Paßwort

Dein Paßwort ist der einzige Schutz, der zwischen den anderen Benutzern und Deinen Daten steht. Aus diesem Grunde solltest Du diesen Sicherheitsmechanismus ernst nehmen. Als ersten Schritt nachdem Du Deinen Account erhalten hast, solltest Du Dein Paßwort auf eine Zeichenfolge ändern, die Du Dir leichter merken kannst.

Bei der Wahl des neuen Paßwortes sollten folgende Regeln beachtet werden:

- Das Paßwort muß mindestens sechs Zeichen lang sein. Das Betriebssystem vergleicht maximal die ersten acht Zeichen des Paßwortes, wodurch längere Paßwörter, obwohl erlaubt, keinen weiteren Schutz bieten.
- Das Paßwort sollte auch Ziffern, Großbuchstaben und Sonderzeichen wie `_@#!...` enthalten. Beachte aber, daß im foo-Pool sowohl deutsche als auch englische Tastaturen vorhanden sind. Man sollte also nicht gerade ein Paßwort mit deutschen Umlauten wählen.
- Auf keinen Fall den eigenen Namen in irgendeiner Form (rückwärts usw.) als Paßwort verwenden!
- Auch Modenamen und Wörter, die im Wörterbuch stehen sollten vermieden werden. Beim Knacken eines Paßwortes werden im Regelfalle diese Dinge zuerst getestet.

Die Änderung selbst wird mit Hilfe des Kommandos `yppasswd (Return)` durchgeführt. Es fordert zuerst die Eingabe des alten Paßwortes. Ist dies korrekt, muß das neue Paßwort zwei Mal eingegeben werden. Alle Eingaben erfolgen verdeckt und müssen mit `(Return)` abgeschlossen werden. Die Aktualisierung der Paßwortdatenbanken kann bis zu einer Stunde dauern. Abbildung 2 zeigt die Verwendung dieses Kommandos.

2.2 Graphische Benutzeroberflächen

Die Grundidee einer graphischen Benutzeroberfläche ist es, jeder Anwendung einen eigenen, variablen Bildschirmbereich zuzuordnen. Variabel bedeutet hier, daß sowohl die Größe als auch der Ursprung des Bereiches veränderlich ist. Ein solcher Bereich wird Window („Fenster“) genannt. Ein Window kann vom Benutzer verschoben und skaliert werden.

Unix bietet eine äusserst flexible graphische Benutzeroberfläche. Mit ihrer Hilfe lassen sich recht einfach mehrere parallel arbeitende Programme bedienen. Der Hauptbestandteil der

graphischen Benutzeroberfläche ist der Window-Manager, der genau das macht: Er verwaltet die Fenster, die von Anwendungen erzeugt werden. Dies beinhaltet beispielsweise das verschieben oder skalieren von Fenstern. Auch der Rahmen und die Titelleiste werden von ihm erzeugt.

Die eigentliche Basis der graphischen Benutzeroberfläche ist der „X-Server“. Er stellt die Hardwarechnittstellen zur Verfügung, die dann von den Programmen zur Bildschirmausgabe genutzt werden. Ein Fenstermanager setzt auf den X-Server auf und nutzt dessen Funktionen für den Hardwarezugriff.

Die nachfolgenden Seiten beschreiben zuerst den Fenstermanager `fvwm`, da dieser im `foo-Pool` am gebräuchlichsten ist. Das auf HP-UX und SunOS gebräuchliche Common Desktop Environment (CDE) wird in dem darauf folgenden Kapitel kurz erläutert.

Für alle Windowsbenutzer: Der unter Windows übliche Doppelklick auf das Systemmenü sollte unter Unix tunlichst vermieden werden. *Er beendet nicht das Programm sondern zerstört nur das Fenster!* Aus diesem Grunde sollte sich jeder folgende Regel zu eigen machen: „Don't destroy the window, destroy the application.“ Wird diese Regel nicht beachtet, läuft der Prozeß, der dem Fenster zugeordnet war, weiter. Damit belegt er dauerhaft Systemressourcen, da er als Hintergrundprozeß beim Ausloggen nicht automatisch beendet wird.

2.2.1 Die graphische Benutzeroberfläche mit `fvwm`

Abbildung 3 zeigt eine einfache Konfiguration des im `foo-Pool` gebräuchlichen Window-Managers `fvwm`².

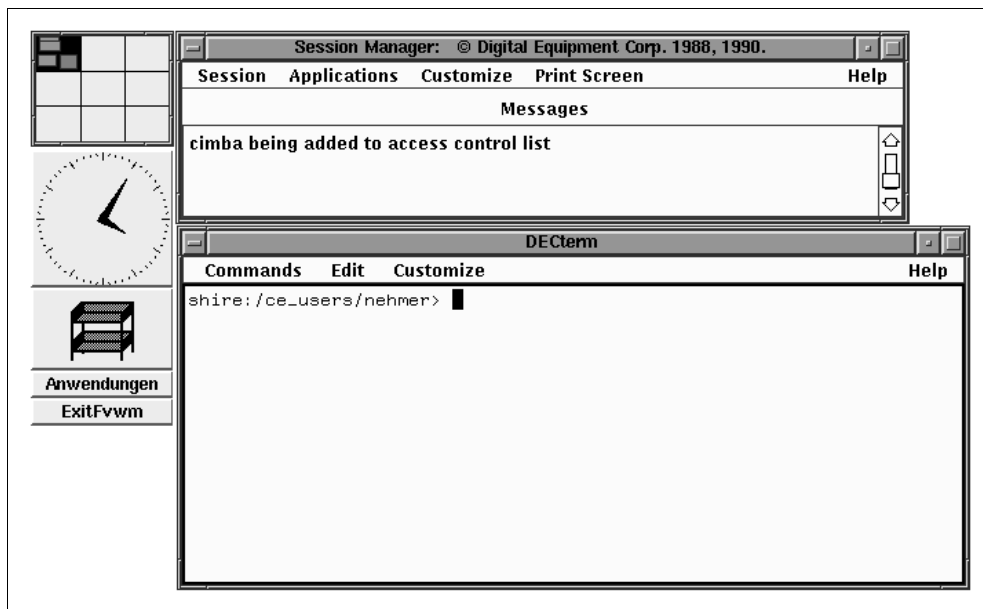


Abbildung 3: Der Window-Manager `fvwm` in einer Standardkonfiguration

In der oberen linken Bildschirmecke befinden sich vier Hilfsprogramme, die nachfolgend erläutert werden. Die Uhr hat keine weitere Funktion und bleibt deshalb hier außen vor.

²Für experimentierfreudige Zeitgenossen: Die Dateien `.xdefaults`, `.xinitrc` und `.fvwmrc` in eurem Homeverzeichnis enthalten alle grundlegenden Informationen. Sichert sie aber bitte vorher ab. :-)

Der fvwm-Pager

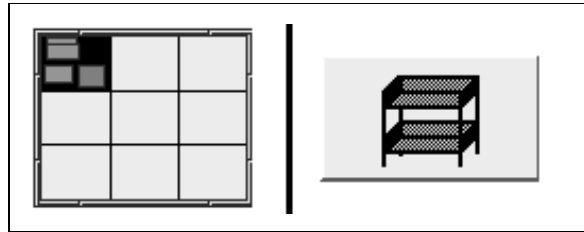


Abbildung 4: Der fvwm-Pager und die automatische Mail-Benachrichtigung

Der fvwm Pager (engl. "page" = Seite, siehe Abbildung 4) ermöglicht es einem, mehrere „Bildschirmseiten“ nebeneinander zu nutzen. Standardmässig werden neun Bildschirmseiten zur Verfügung gestellt, die in einem 3x3 Raster angeordnet sind. Man kann sich den Monitor wie eine Schablone vorstellen, die über einem der neun Quadrate liegt. Klickt man mit der linken Maustaste in eines der neun Quadrate, so wechselt die Anzeige in das betreffende Quadrat. Jedes angezeigte Fenster wird innerhalb des Pagers durch ein farbiges Rechteck symbolisiert. Dieses Merkmal wird als „Virtual Desktop“ bezeichnet.

Automatische Mail-Benachrichtigung

Die Mail-Benachrichtigung (siehe Abbildung 4) ist ein kleines nützliches Tool, das den eigenen Mail-Account überwacht. Treffen neue E-Mails ein, wird dies mit einem Briefumschlag in der Box angezeigt. Hat man die Mails gelesen kann man die Benachrichtigung wieder entfernen, in dem man auf das Symbol klickt.

Menüleiste

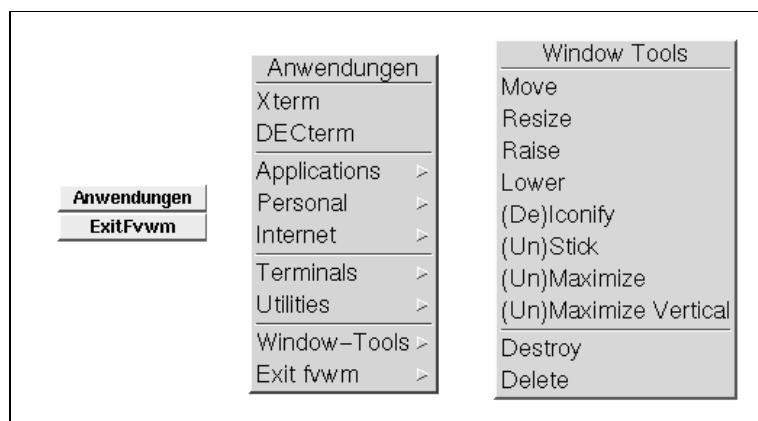


Abbildung 5: Die Menüleiste, das Anwendungsmenü und die Window-Tools

Eine kleine Menüleiste (siehe Abbildung 5) versorgt Dich mit den wichtigsten Programmen im foo-Pool. Klickt man auf den Menüpunkt „Anwendungen“, öffnet sich ein weiteres Menü. Die Anwendungen, die dort zu finden sind, werden in den folgenden Kapiteln beschrieben. Ein Menüpunkt mit einem kleinen Pfeil am Ende zeigt an, daß sich dort ein Untermenü befindet. Der Start einer Anwendung erfolgt mittels Klick auf den dazugehörigen Eintrag im Menü.

Die Window-Tools

Die Window-Tools (Abbildung 5) beinhalten einige Funktionen, die für Fenstermanager typisch sind. Mit Ihnen lassen sich die verschiedenen Funktionen des Fenstermanagers direkt ansteuern. Du findest sie entweder durch einen Klick mit der mittleren Maustaste auf den Desktop oder im Untermenü Window-Tools des Anwendungsmenüs.

Move: Move bewegt ein Fenster.

Resize: Resize verändert die Größe eines Fensters.

Raise: Raise bringt ein Fenster in den Vordergrund.

Lower: Lower stellt ein Fenster in den Hintergrund.

(De)Iconify: Mit Hilfe dieses Menüpunktes kann ein Fenster zu einem Icon verwandelt werden. Dies bedeutet, daß das eigentliche Anwendungsfenster durch ein kleines Symbol ersetzt wird. Ein erneutes Ausführen stellt das Anwendungsfenster wieder her.

(Un)Stick: Mit Hilfe dieses Menüpunktes kann die Sticky-Eigenschaft eines Fensters an- und ausgeschaltet werden. Sticky bedeutet, daß ein Fenster vom virtuellen Desktop losgelöst wird. Egal, welcher Bereich des virtuellen Desktops gerade auf dem Bildschirm angezeigt wird, das betroffene Fenster bleibt immer sichtbar.

(Un)Maximize: Diese Funktion vergrößert ein Anwendungsfenster auf die komplette Bildschirmfläche.

(Un)Maximize Vertical: *Maximize Vertical* vergrößert ein Anwendungsfenster auf die maximal mögliche Größe in vertikaler Richtung.

Destroy: Destroy vernichtet ein Anwendungsfenster. Da die Anwendung, die dieses Fenster benutzt, diesen Vorgang nicht unbedingt bemerken muß, sollte dies nur als allerletzte Rettung benutzt werden. Anwendungen sollten immer regulär über die von der Anwendung vorgesehene Funktion beendet werden.

Delete: Delete verhält sich wie Destroy, geht dabei intern jedoch etwas sanftere Wege. Auch dieser Menüpunkt sollte soweit möglich vermieden werden.

2.2.2 Common Desktop Environment (CDE)

Das CDE geht als graphische Benutzeroberfläche grundsätzlich andere Wege als die herkömmlichen Window-Manager von Unix. CDE ist kein reiner Window-Manger sondern integriert den „Session-Manager“. Während andere Systeme Session- und Window-Manager strikt trennen versucht CDE diese Trennung zu Gunsten einer einfacheren Bedienung zu vermeiden. CDE kann den Zustand der aktuellen Session, also alle geöffneten Anwendungen und deren Positionen, speichern und beim nächsten Login wieder herstellen. Abbildung 6 zeigt eine solche mögliche Startkonfiguration.

Die zentrale CDE-Konfiguration³ des foo-Pools stellt die wichtigsten Anwendungen in einem eigenen Menü zur Verfügung. Dazu will ich hier kurz die wichtigsten Grundzüge des CDEs vorstellen.

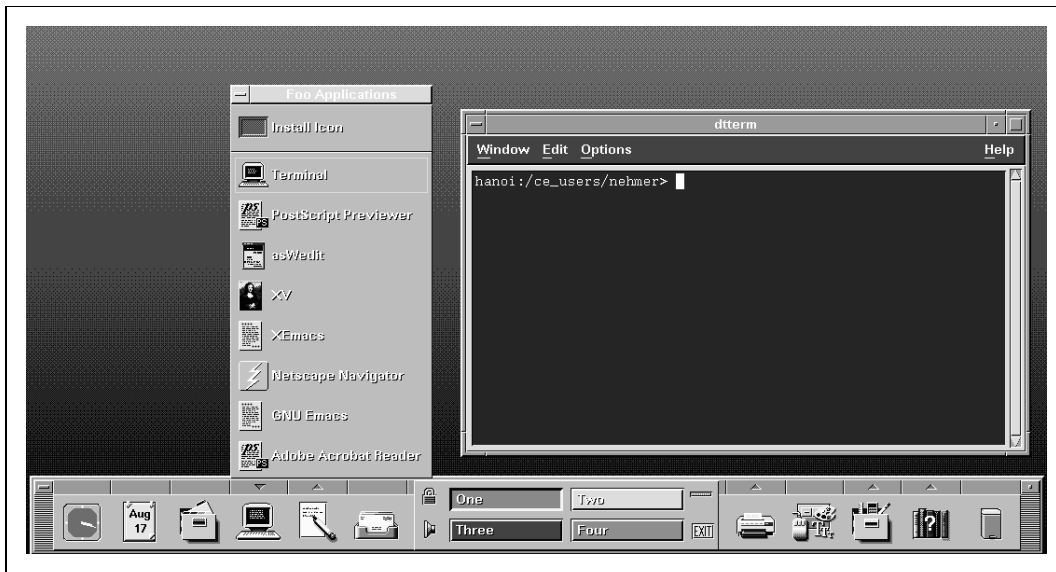


Abbildung 6: Das CDE in einer möglichen Standardkonfiguration



Abbildung 7: Die Toolbar des CDE

Die Toolbar des CDE

Von Ihr aus werden die wichtigsten Elemente des CDE gesteuert. Sie läßt sich in zwei Teile untergliedern: Einerseits den Mittelteil (mit den Schaltflächen „One“ bis „Four“), das zur Steuerung des Desktops dient und andererseits die beiden Seitenteile, die vom Benutzer definierbare Anwendungsmenüs enthalten.

Die Toolbar stellt mit den Schaltflächen „One“ bis „Four“ vier völlig unabhängige Desktops (genannt „Workspace“) zur Verfügung. Anwendungen starten immer im aktuellen Desktop und können gezielt mit Hilfe des Systemmenüs (ganz oben links in einem Anwendungsfenster) auf einen anderen Desktop verschoben werden. Ein Klick auf eine der Schaltflächen wechselt in den entsprechenden Desktop. Weiterhin dient die „Exit“-Schaltfläche dazu, die aktuelle Session zu beenden. Die restlichen Symbole repräsentieren verschiedene Anwendungen, die per Klick gestartet werden können. Ein kleiner, nach oben gerichteter, Pfeil über einem Symbol zeigt ein vorhandenes Untermenü, das ebenfalls per Klick angezeigt werden kann. Das vierte Menü von links (mit dem kleinen Computersymbol) stellt dabei die Anwendungen des Foo-Pools zur Verfügung. Dies ist das in Abbildung 6 aufgeklappte Menü.

Das Workspace Menü

Das Workspace Menü kann mit einem Rechts-Klick auf den Desktop-Hintergrund geöffnet werden. Es arbeitet nach den gleichen Prinzipien wie das Startmenü des fvwm (siehe Kapitel 2.2.1).

³Jeder Benutzer besitzt eine private CDE-Konfiguration. Sie befindet sich in der Datei `.dtprofile` und im Unterverzeichnis `.dt/`, jeweils direkt im Homeverzeichnis.

2.2.3 Beenden der X-Session

Hat man alle Arbeit erledigt, sollte man sich wieder ordnungsgemäss ausloggen, also die Session schliessen. Erster Schritt ist es, alle lauffenden Anwendungen zu beenden. Benutzt man nun einen Rechner *ohne* CDE (CDE wird unten gesondert behandelt), wird als nächstes der Fenstermanager `fvwm` beendet. Dies geschieht über den Menüpunkt „Exit fvwm“ der Menüleiste. Nun trennen sich die Wege erneut.

Unter **Ultrix** und **OSF/1** kann die Aktuelle Sitzung über das Menü des Sessionmanagers: Session→Quit (Ultrix) bzw. Session→End Session (OSF/1). Abbildung 8 zeigt diesen Vorgang.

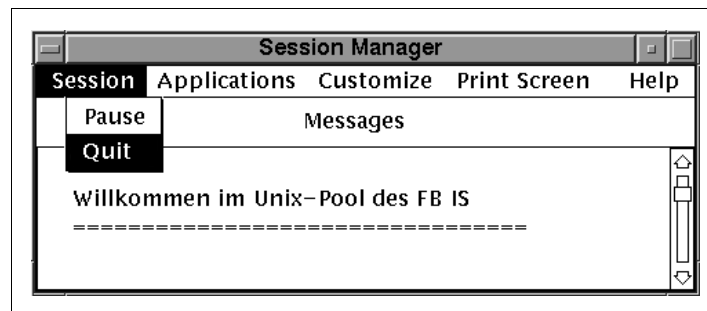


Abbildung 8: Beenden einer X-Session unter Ultrix und OSF/1

Linux und **NetBSD** hingegen nutzen in der aktuellen Konfiguration keinen Sessionmanager. Hat man den `fvwm` beendet, landet man unter Linux wieder am Kommandoprompt und kann sich mit Hilfe des Befehles `exit` (Return) ausloggen, während NetBSD direkt den Login-Schirm anzeigt.

Im **Common Desktop Environment** ist keine offensichtliche Trennung zwischen Fenstermanager und X-Session mehr vorhanden. Dadurch genügt ein Klick auf die `EXIT`-Schaltfläche in der Kontrolleiste (siehe Abbildung 9).

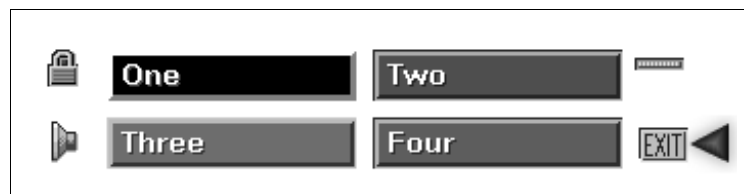


Abbildung 9: Beenden einer X-Session unter CDE

2.3 Die Shell

In jedem Terminalfenster (DecTerm, xterm) und unter Linux nach dem Login wartet ein Kommandozeileninterpreter auf Deine Eingaben. Dieser Interpreter wird Shell genannt. Jede eingegebene Kommandozeile wird von der Shell interpretiert. Erweist sich die Syntax als korrekt, wird das entsprechende Kommando gestartet.

Unix bietet die Möglichkeit, verschiedene Shells zu verwenden. Historisch gibt es zwei Entwicklungslinien: Die Bourne-Shells (`sh` und `bash`) und die *c*-Shells (`csh` und `tcsh`). Welche Shell ein Benutzer letztendlich verwendet, hängt von seinen persönlichen Vorlieben ab. Im foo-Pool bekommt jeder neue Benutzer standardmässig eine *tc*-Shell (`tcsh`) zugeordnet. Für diese Shell existiert auch eine regelmässig gewartete globale Standardkonfiguration. Alle Skripte, die im


```
hanoi:/ce_users/nehmer>
```

Abbildung 10: Der Standardprompt der tcsh am Beispiel des Rechners „hanoi“

foo-Pool verwendet werden, basieren auf der Bourne-Shell *sh*, da diese garantiert in jeder Unix-Umgebung vorhanden ist.

Alle Beispiele in diesem Buch beziehen sich auf die Syntax der **tcsh**, es sei denn, es wird ausdrücklich auf eine andere Shell hingewiesen.

Anwender aus der Windows-Welt argumentieren jetzt vermutlich damit, das es dort seit Jahren Tools gibt, um Kontakte mit einer Shell zu vermeiden. Nur ist eine Unix-Shell bei weitem leistungsfähiger, als alle „DOS-Tools“ zusammen. Hat man sich erst einmal an die vielfältigen Möglichkeiten der großen Unix-Shells gewöhnt, wird man diese in der Windows-Welt sehr vermissen.

2.3.1 Starten einer Shell bzw. eines Terminals

Eine Shell benötigt immer ein Terminal, in dem sie abläuft. Diese Terminals, unter Windows als Eingabeaufforderung bekannt, stellen eine textbasierte Umgebung zur Verfügung. Im foo-Pool sind grundsätzlich zwei verschiedene Terminals vorhanden, die beide über das Anwendungsmenü geöffnet werden können: Das xTerm und das DecTerm. Beide Terminals besitzen ihre Vor- und Nachteile, so daß es jedem überlassen bleibt, welches Terminal man nun verwendet.

Jedes Mal, wenn eine Shell gestartet wird werden (bei der (t) csh) die Dateien `.cshrc` und `.login` in Deinem Homeverzeichnis abgearbeitet. Speziell die erste Datei ist ein Anlaufpunkt zum Anpassen der Benutzerumgebung an Deine Bedürfnisse.

2.3.2 Der Prompt

Jede Shell zeigt mit einer Eingabeaufforderung (Prompt) ihre Bereitschaft an. Abbildung 10 zeigt den Standardprompt der tcsh des foo-Pools, der sich aus Rechnernamen und aktuellem Verzeichnis zusammensetzt.

Wie die meisten Unix-Shells bietet die tcsh umfangreiche Möglichkeiten an, die aktuelle Kommandozeile zu bearbeiten und zu ergänzen. Mit \leftarrow und \rightarrow kann der Eingabecursor innerhalb der Eingabezeile bewegt werden. Dadurch wird die Korrektur von Tippfehlern zum Kinderspiel. Ergänzt wird dies durch den Kommandozeilenspeicher („Command History“). Mit \uparrow und \downarrow lässt sich durch die zuletzt eingegebenen Befehle blättern, diese wieder ausführen und auch verändern. Dadurch wird das Ausführen von komplexen Kommandos in leichten Variationen sehr einfach.

Ein weiteres hilfreiches Werkzeug ist die automatische Textergänzung. $\overline{\text{Tab}}$ weist die Shell an, nach möglichen Komplettierungen des aktuellen Wortes zu suchen. Dies geschieht für Kommandonamen im aktuellen Suchpfad (siehe Umgebungsvariable `$PATH`, Kapitel 2.7) und für Argumente im aktuellen Verzeichnis. Wird etwas gefunden, wird die längste, eindeutige Lösung angezeigt. Abbildung 11 zeigt hierzu ein Beispiel.

Oftmals will man mehrere Dateien auf einmal ansprechen. Dies wird mit Jokerzeichen (auch als „Wildcards“ bekannt) erreicht. Diese Zeichen können in der Regel überall dort verwendet werden, wo Dateinamen erwartet werden. Ein `*` kann beliebig viele Zeichen repräsentieren. `m*` steht damit für alle Dateien, die mit `m` beginnen. Will man hingegen die Dateien `doc1.txt`


```

Alle Dateien im aktuellen Verzeichnis anzeigen:
hanne:/ce_users/nehmer/templates> ls (Return)
latex-large.tex latex.tex
latex-large.tex anzeigen:
hanne:/ce_users/nehmer/templates> cat l(Tab)
(Tab) ergänzt das Kommando jetzt soweit möglich. Da beide vorhandenen Dateien mit
„latex“ beginnen, weiß die Shell nicht, welche der beiden Dateien gemeint ist. Deshalb
liefert sie
hanne:/ce_users/nehmer/templates> cat latex
und wartet auf weitere Eingaben. Jetzt muß die gesuchte Datei näher spezifiziert werden:
hanne:/ce_users/nehmer/templates> cat latex-(Tab)
Wieder sucht die Shell nach möglichen Lösungen. Diesesmal ist es eindeutig und der
gesuchte Dateiname ist gefunden:
hanne:/ce_users/nehmer/templates> cat latex-large.tex

```

Abbildung 11: Verwendung der Textkomplettierung mit Tab innerhalb der Shell

doc2.txt doc3.txt ansprechen, kann man ? verwenden, das ein einzelnes Zeichen vertritt. Man erhält also mit doc?.txt die gesuchten Dateien.

2.3.3 Grundlegende Kommandos

Bevor wir weiter in die Tiefen von UNIX einsteigen, will ich zuerst einige grundlegende Kommandos vorstellen. Wie sie über bestimmte Befehle weitere Informationen erhalten, beschreibt Kapitel 2.6.

Die meisten Kommandos können mit Hilfe von Parametern beeinflusst werden. Beispielsweise kann `ls` (zeigt alle Dateien im aktuellen Verzeichnis an) so beeinflusst werden, daß es eine detaillierte Liste anstelle einer kurzen Übersicht ausgibt. Parameter beginnen, wenn nichts anderes angegeben ist, mit einem `-`. In unserem Fall heißt der Parameter `-l`. Zusammengesetzt ergibt dies dann `ls -l`. Will man nun noch weitere Parameter verwenden, kann man diese nacheinander angeben: `ls -l -a . . .`. Der Einfachheit halber lassen sie sich aber auch zu `ls -la . . .` zusammenfassen.

```
ls [-parameter] [pfad]
```

`ls` zeigt eine Liste aller Dateien und Verzeichnisse an. Damit entspricht es im wesentlichen dem DOS-Befehl `dir`. Nun zu den wichtigsten Parameter von `ls`:

`-a` bzw. `--all`

zeigt auch Dateien an, die mit `.` beginnen. Dies sind die Unix-Pendants zu den versteckten Dateien der Windows-Welt: Sie werden im normalen Betrieb dem Benutzer nicht angezeigt. Nähere Informationen zu versteckten Dateien finden Sie im Kapitel 2.4 „Die Verzeichnishierarchie“.

`-d` bzw. `--directory`

zeigt nur den Namen der angezeigten Verzeichnisse an. Die Option ist beispielsweise dann sinnvoll, wenn als Pfad ein Verzeichnisname angegeben wird und die Zugriffsrechte des Verzeichnisses (siehe Kapitel 2.5) überprüft werden sollen (und nicht sein Inhalt).

`-l` bzw. `--format=long` oder `--format=verbose`

zeigt zusätzlich zum Dateinamen weitere Informationen an: Zugriffsrechte, Besitzer, Dateigrö-

ße und Änderungsdatum. Zur Anzeige jeder Datei wird eine eigene Zeile verwendet. Genauere Informationen zum Ausgabeformat dieser Zeile findet sich im Kapitel 2.5.

Beispiel

```
timbuktu:/ce_users/nehmer> ls *.tex
```

listet alle *.tex-Dateien im aktuellen Verzeichnis auf.

```
cd [pfad]
```

cd wechselt in das angegebene Verzeichnis. Dabei kann cd .. zum Wechsel in die nächst höhere Verzeichnisebene und cd ~ zum Wechsel ins eigene Homeverzeichnis benutzt werden. cd ohne Pfadangabe zeigt das aktuelle Verzeichnis an.

Beispiel

```
timbuktu:/ce_users/nehmer> cd ..
```

wechselt eine Verzeichnisebene nach oben. Man landet also in:

```
timbuktu:/ce_users>
```

```
cp [-parameter] quelle ziel
cp [-parameter] dateien zielverzeichnis
```

cp dient zum Kopieren von Dateien. Einzelne Dateien können dabei umbenannt werden. Wenn mehrere Dateien (z.B. durch die Angabe von Jokerzeichen) bearbeitet werden sollen, können diese lediglich in ein anderes Verzeichnis kopiert, nicht aber umbenannt werden. cp unterstützt unter anderem folgenden Parameter:

-r bzw. --recursive
kopiert auch Unterverzeichnisse und die darin enthaltenen Dateien.

Beispiel

```
timbuktu:/ce_users/nehmer> cp buch/*.tex .
```

kopiert alle *.tex-Dateien aus dem Unterverzeichnis buch in das aktuelle Verzeichnis. Der Punkt gibt dabei als Zielverzeichnis das aktuelle Verzeichnis an.

```
mv [-parameter] quelle ziel
mv [-parameter] dateien zielverzeichnis
```

mv benennt eine Datei um bzw. verschiebt ein oder mehrere Dateien in ein anderes Verzeichnis.

Beispiel

```
timbuktu:/ce_users/nehmer> mv *~ backup/
```

verschiebt alle Dateien, deren Namen mit ~ enden, ins das Unterverzeichnis backup/.

```
rm [-parameter] dateien
```

rm löscht die angegebenen Dateien. Verzeichnisse werden – sofern nicht die Option -r verwendet wird – nicht gelöscht. Beim Verwenden dieses Kommandos ist ein wenig Vorsicht angebracht, da es, im Gegensatz zu dem Windows-Kommando del auch Unterverzeichnisse mitverarbeiten kann. *Achtung: Einmal gelöschte Dateien können nicht wiederhergestellt werden!!!* Wichtige Parameter von rm sind:

-f
löscht die angegebenen Dateien ohne Rückfrage. Vor allem dieser Parameter ist mit sehr viel

Bedacht einzusetzen!

`-i`

frägt bei jeder einzelnen Datei nach, ob sie gelöscht werden soll. Im foo-Pool wurde dies mit Hilfe eines Alias⁴ als Standard gesetzt, da `rm` sonst `-f` als Standard setzen würde.

`-r` bzw. `-R` bzw. `--recursive`

löscht auch Dateien in allen angegebenen Unterverzeichnissen. Wenn dabei der gesamte Inhalt des Unterverzeichnisses gelöscht wird, wird auch das Unterverzeichnis selbst eliminiert. Vorsicht!

Beispiele

```
timbuktu:/ce_users/nehmer> rm *~
```

löscht im aktuellen Verzeichnis alle Dateien, die mit `~` enden.

```
timbuktu:/ce_users/nehmer> rm -rf *
```

würde den *kompletten Homebereich des Users nehmer löschen!* Dies ist die gefährlichste `rm`-Kombination, da sie ohne Nachfrage alle Dateien im aktuellen Verzeichnis inklusive aller Unterverzeichnisse löscht.

```
mkdir verzeichnis
```

`mkdir` erstellt ein neues Verzeichnis.

```
rmdir verzeichnis
```

`rmdir` löscht ein Verzeichnis. Dieser Befehl wird nur dann ausgeführt, wenn das Verzeichnis selbst leer ist.

```
more datei
```

`more` zeigt den Inhalt einer Textdatei seitenweise an. Ist eine Bildschirmseite voll, wartet `more` auf eine Tastatureingabe. Die wichtigsten Eingabemöglichkeiten sind `(Return)` (eine Zeile nach unten), `(Leertaste)` (eine Seite nach unten), `(B)` (eine Seite nach oben) und `(Q)` (Beenden).

```
uname -a
```

`uname -a` liefert den Namen des aktuellen Betriebssystems.

Beispiel

```
hanne:/ce_users/nehmer> uname -a
```

```
ULTRIX hanne 4.5 0 RISC
```

`uname` zeigt uns hier, daß auf `hanne` derzeit ein Ultrix 4.5 arbeitet.

2.3.4 Sonderzeichen in einer Shell

Damit eine Shell die Benutzereingaben richtig versteht, gibt es feste Regeln, an denen die verschiedenen Teile eines Kommandos erkannt werden. Dies geschieht mit Hilfe von Sonderzeichen, die jeweils bestimmte Funktionen erfüllen.

Die Shell muß einzelne Teile einer Kommandozeile separieren können. Dies sind das aufzuru-

⁴Mit Hilfe von Aliassen können in einer Shell Makros definiert werden. Die Shell erweitert ein `rm` automatisch nach `rm -i`, wenn `alias "rm=rm -i"` gesetzt ist.

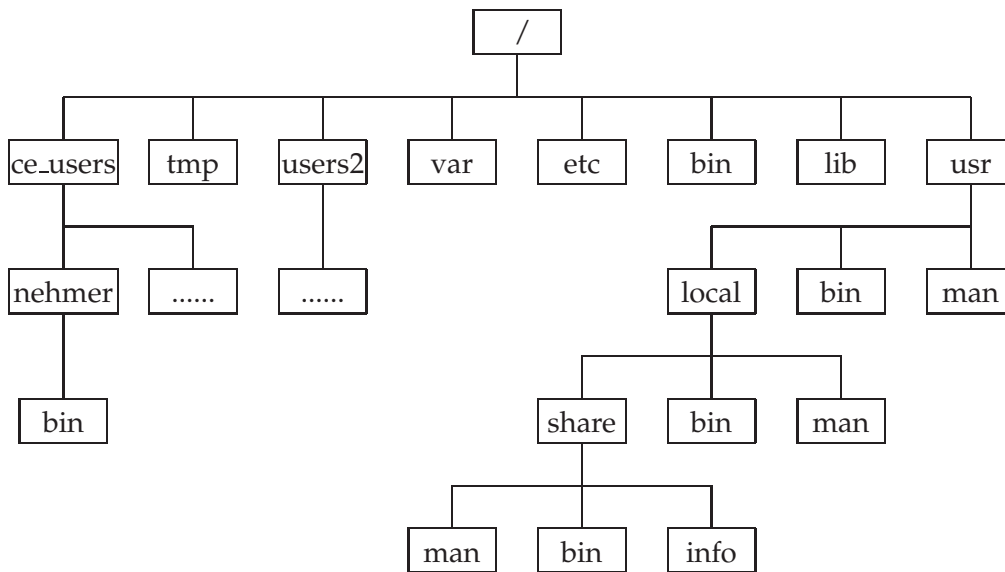


Abbildung 12: Die Verzeichnishierarchie im foo-Pool

fende Programm und dessenen Parameter, von denen es beliebig viele (auch keine) haben darf. Jeder dieser Parameter wird durch mindestens ein Leerzeichen von den anderen getrennt. Art und Form eines Parameters sind von Programm zu Programm unterschiedlich. Parameter werden oft auch als Argumente bezeichnet.

Will man nun *innerhalb* eines Parameters ein Leerzeichen angeben, so muß dieses vor der Shell versteckt werden. Man nennt dies *maskieren*. Dies macht man mit Hilfe eines weiteren Sonderzeichens, dem Backslash (`\`). Man stellt ihn einfach dem Zeichen voran, daß von der Shell nicht interpretiert werden soll. Will man also in das Verzeichnis „private daten“ wechseln, so kann man dies mit Hilfe des Befehls „`cd private\ daten`“ erreichen.

Dies gilt generell für alle Sonderzeichen innerhalb einer Shell, selbst für `/` (`\`), oder dem Backslash selbst (`\\`).

Will man eine gesamte Zeichenkette maskieren, so kann man sie auch in Hochkommata bzw., Akzente einschliessen. Unser Verzeichniswechsel hätte also mit „`cd 'private daten'`“ genauso funktioniert. (Achtung! UNIX unterscheidet die beiden Akzentarten `'` und ```. Die Maskierung funktioniert nur mit dem ersten Typ. (Der zweite hat eine völlig andere Bedeutung, die im Rahmen der Foo-Bibel zu weit ginge⁵.)

Ein weiteres nützliches Sonderzeichen ist der Semikolon. Mit seiner Hilfe können mehrere Kommandos in Folge abgeschickt werden. Dies ist besonders beim Nutzen anderer Rechner (siehe Kapitel 2.10) von Bedeutung. Die Kommandofolge „`cd test ; ls`“ wechselt folglich ins Unterverzeichnis `test` und zeigt gleich die dort vorhandenen Dateien an.

2.4 Dateisysteme im Foo-Pool

Die Verzeichnishierarchie eines Unix-Systems ist auf den ersten Blick recht komplex. Doch diese verschachtelte Struktur dient dazu, dem Benutzer die Orientierung innerhalb eines Netzwerkes zu erleichtern. Der Grundsatz ist, Dateisysteme verschiedener Unix-Rechner nach einem Standard aufzubauen. So wird dem Benutzer der Wechsel zwischen verschiedenen Unix-Rechnern stark erleichtert.

⁵Ich empfehle hier einen Blick in eine Shell-Manpage unter dem Kapitel „Command Substitution“

Im Gegensatz zur Windows-Welt kennt Unix keine Laufwerksbuchstaben. Das Unix-Filesystem beginnt im Rootverzeichnis `/`. Dies ist das Wurzelverzeichnis, von dem alle vorhandenen Verzeichniszweige abgehen. Die Nutzung eines solchen Dateisystems geschieht auch dann für den Nutzer völlig transparent, wenn es sich über mehrere physikalische Datenträger erstreckt. Unix besitzt aus diesem Grunde eine sehr hohe Flexibilität, denn ein Systemadministrator kann die Hardware eines Rechners verändern, ohne das Benutzer davon betroffen sind.

Abbildung 12 zeigt die im foo-Pool verwendete Verzeichnishierarchie. Dies sind die wichtigsten Verzeichnisse, die man als Benutzer im tagtäglichen Betrieb benötigt.

- `/ce_users`** In diesem Verzeichnis befinden sich die Homebereiche aller CE-Studenten.
- `/users2`** Hier finden sich die Homebereiche der ER- und IK-Studenten. Im folgenden gelten alle Unterverzeichnisse von `/ce_users` auch für `/users2`.
- `/erlab`** Alle Homebereiche der Mitarbeiter, Assistenten und Professoren befinden sich an dieser Stelle. Auch für `/erlab` gelten die Unterverzeichnisse von `/ce_users`.
- `/ce_users/Nutzername`** Dies ist der Homebereich eines bestimmten Nutzers. Ein Synonym für den eigenen Homebereich ist `~`, womit ein `cd ~` in das eigene Homeverzeichnis wechselt.
- `/ce_users/Nutzername/bin`** Jeder Benutzer kann in diesem Verzeichnis seine eigenen, privaten Programme unterbringen. Dieses Verzeichnis wird von den Startskripten automatisch in den Pfad mit aufgenommen. (siehe Umgebungsvariable `$PATH`, Kapitel 2.7)
- `/tmp`** Da der verfügbare Speicherplatz für die Benutzer eingeschränkt ist, stellen UNIX-Systeme mit `/tmp` einen Bereich zur Verfügung, der für jeden Benutzer frei zugänglich ist. Allerdings kann man nicht damit rechnen, dass dort abgelegte Dateien morgen noch vorhanden sind, da dieses Verzeichnis vom System regelmässig gesäubert wird.
- `/scratch`** Wie `/tmp` stellen wir im foo-Pool in `/scratch` einen frei benutzbaren Bereich zur Verfügung, für den die gleichen Regeln gelten. Im Unterschied zu den lokalen `/tmp`-Verzeichnissen liegt `/scratch` zentral auf dem Fileserver und bietet dadurch deutlich mehr Platz an.
- `/bin`** Die wichtigsten Programme des Betriebssystems (z.B. `ls`, `cp` usw.) sind in `/bin` abgelegt.
- `/lib`** beinhaltet die wichtigsten System-Bibliotheken („Libraries“).
- `/etc`** Die Konfiguration eines UNIX-Rechners wird in diesem Verzeichnis abgelegt.
- `/usr/bin`** hat sich als Standardposition für Anwendungsprogramme („Binaries“) herausgebildet.
- `/usr/lib`** Die Bibliotheken der Anwendungsprogramme in `/usr/bin` werden hier abgelegt.
- `/usr/man`** Alle systemabhängigen Onlinehilfen („manual-pages“, siehe Kapitel 2.6) landen an dieser Stelle.
- `/usr/local`** Unter `/usr/local` befinden sich die meisten Anwendungsprogramme, die im foo-Pool installiert sind. Der `/usr/local` - Bereich befindet sich jeweils auf einem bestimmten Rechner des betreffenden Betriebssystems und wird von dort an die gleichartigen Rechner verteilt. (Linux-Programme laufen nicht unter SunOS usw.) Der `/usr/local` Bereich besitzt analog zu `/usr` die Unterverzeichnisse `bin/`, `lib/` und `man/`.
- `/usr/local/share`** Viele Anwendungsprogramme besitzen Daten, die plattformunabhängig verwendet werden können. Dadurch bietet sich eine zentrale Ablage für solche Dateien an. Dies sind vor allem Hilfedateien in `man/` und `info/` (siehe Kapitel 2.6).

`/usr/local/share/bin` Globale Skripte, die betriebssystemabhängig Programme (z.B. Netscape) starten.

2.5 Zugriffsrechte unter Unix

Eingangs wurde bereits auf die Mehrbenutzer-Fähigkeiten von Unix hingewiesen. Aus diesem Umfeld entsteht das Problem der Zugriffsrechte. Um den unautorisierten Zugriff auf Daten zu unterbinden, wird ein System zur Kontrolle der Zugriffsrechte benötigt. Nur so kann das Betriebssystem sicherstellen, das keine unerwünschten Zugriffe geschehen.

2.5.1 Gruppen und deren Benutzer

Jedes Netzwerk setzt sich aus vielen Benutzern zusammen. Auf dieser Basis setzt die Kontrolle der Zugriffsrechte auf: Es wird für jede Datei festgelegt, wer was darf. Um dies ein wenig zu vereinfachen, können mehrere Benutzer zu einer Gruppe zusammengefasst werden, die dann geschlossen Rechte zugewiesen bekommt. Im foo-Pool sind alle Studenten der Gruppe `users` zugeordnet. Ein Benutzer kann auch mehreren Gruppen zugeordnet werden. Das Kommando `id` gibt darüber Aufschluß:

```
timbuktu:/ce_users/nehmer> id Return
uid=3817(nehmer) gid=15(users) groups=81(www_adm) ,10(staff)
```

Aus dieser Meldung kann man folgende Informationen entnehmen: Man ist aktuell als Benutzer `nehmer` eingeloggt. Dieser Benutzer trägt die interne Nummer (ID) 3817. `nehmer` ist Mitglied der Gruppe `users` mit der Group-ID 15. Dies ist die primäre Gruppenzuordnung, mit der der Benutzer arbeitet. Ist man zusätzlich noch anderen Gruppen (zum Beispiel für Jahresprojekte oder ähnliches) zugeordnet, kann man diese Information in der Ausgabe von `id` unter `groups` zu finden: `nehmer` ist derzeit folglich noch Mitglied in den Gruppen `www_adm` und `staff`, jeweils mit den Group-IDs 81 und 10.

Die IDs sind für den normalen Benutzer in der Regel nicht von Bedeutung. Hier sei nur erwähnt, das Unix intern immer nur die ID-Nummern abgleicht und das neue Dateien immer mit der primären Gruppenzuordnung erstellt werden.

2.5.2 Dateien und ihre Berechtigungen

Unix ordnet jeder Datei zwei Informationsblöcke zu: Die eigentlichen Zugriffsrechte und den Besitzer. Diese Informationen können direkt mit dem Kommando `ls -l` angezeigt werden:

```
timbuktu:/ce_users/nehmer/ce3/foo-bibel> ls -l Return
drwxr-xr-x 2 nehmer users 512 Apr 13 14:58 screenshots
-rw-r--r-- 1 nehmer users 4818 Apr 14 17:30 master.tex
```

Die ersten zehn Zeichen jeder Zeile zeigen die Zugriffsrechte an. Die beiden Namen `nehmer` und `users` geben an, wem die Datei gehört. `nehmer` ist der Eigentümer. Er hat die Datei angelegt und kann auch die Rechte der betroffenen Datei ändern. Zusätzlich ist die die Datei der Gruppe `users` zugeordnet.

Aus diesen Informationen lassen sich nun die Zugriffsberechtigungen ableiten. Es existieren grundsätzlich drei Rechte die der Eigentümer vergeben kann:

Read Permission (Leseberechtigung, „r“) Das Recht, eine Datei lesen (und somit kopieren)

zu dürfen. Im Falle eines Verzeichnisses darf die betreffende Person den Inhalt des Verzeichnisses anzeigen.

Write Permission (Schreibberechtigung, „w“) Das Recht, die Datei inhaltlich zu verändern, sie zu löschen oder sie umzubenennen. Bei Verzeichnissen berechtigt es den Nutzer dort Dateien anzulegen, zu kopieren, umzubenennen oder zu löschen.

Execute Permission (Ausführungsberechtigung, „x“) Das Recht, eine Datei auszuführen. Ist dieses Recht bei einem Verzeichnis nicht gesetzt, dann darf nicht in dieses gewechselt werden.

Diese Rechte können jetzt in drei Ebenen verteilt werden:

User (Eigentümer) Der Eigentümer hat die betroffene Datei bzw. das betroffene Verzeichnis erstellt. Dieser Nutzer kann auch die Zugriffsberechtigungen festlegen. Der Eigentümer kann nur vom Systemadministrator geändert werden.

Group (Gruppe) Diese Rechte gelten für alle Benutzer, die sich in der angegebenen Gruppe befinden. Ausser dem Eigentümer darf jedoch niemand die Zugriffsberechtigungen ändern.

Others (Andere) Alle Benutzer des Netzwerkes.

Unix kodiert diese Rechte nun zu einer zehn Zeichen langen Zeichenkette:

Typ	User			Group			Others		
d/l/-	r	w	x	r	w	x	r	w	x

Der Typ gibt dabei an, ob es sich um ein Verzeichnis (d), einen Link (l) oder eine Datei (-) handelt. Danach folgen die drei Gruppen mit den Zugriffsberechtigungen für die jeweilige Ebene. Ein gesetztes Zugriffsrecht wird dann mit dem Kürzel (r, w oder x) repräsentiert, während ein nicht gesetztes Recht mit - angezeigt wird.

Oft werden Zugriffsrechte mit drei Oktalzahlen⁶ dargestellt. Jede der drei Zahlen repräsentiert eine Ebene der Zugriffsrechte. Die Reihenfolge entspricht der Ausgabe des ls-Befehls. `rwrx-rx-x` würde also als `755` codiert werden: 7 oktal = 111 binär = `rwX`, 5 oktal = 101 binär = `r-x`. Analog lässt sich jede beliebige Kombination an Zugriffsrechten bequem mit Oktalzahlen darstellen..

2.5.3 Ändern der Zugriffsrechte

```
chmod [-R bzw. --recursive] änderungen dateien
chmod [-R bzw. --recursive] rechte dateien
```

`chmod` ändert die neun Zugriffsbits von Dateien. Dabei akzeptiert `chmod` in der einfachsten Form direkt die neuen Dateirechte als Oktalzahl. Alternativ kann auch eine Änderung angegeben werden: Die drei möglichen Gruppen werden mit „u“ (user) „g“ (group) und „o“ (others) angegeben. Darauf folgt entweder ein + zum hinzufügen einer Berechtigung oder ein - zum entfernen einer Berechtigung. Nun können entsprechend r, w oder x angegeben

⁶Oktalzahlen werden mit der Basis 8 aufgebaut. Somit stehen die Ziffern 0..7 für die Darstellung zur Verfügung. 21 oktal wäre also dezimal 17. Oktalzahlen werden oft zur vereinfachten Darstellung von Binärzahlen verwendet, da sich drei Binärzahlen zu einer Oktalzahl zusammenfassen lassen: 21 oktal = 010 001 binär.

werden. `g+rx` setzt für alle Gruppenmitglieder Read, Write und Execute-Rechte. Weitere Beispiele siehe unten.

`-R` bzw. `--recursive` verändert die Rechte von Dateien auch in allen Unterverzeichnissen.

Beispiele

```
atlantis:/ce_users/nehmer> chmod -R 755 bin/
```

Erlaubt jedem, Dateien im Verzeichnis `/ce_users/nehmer/bin` zu lesen und auszuführen, aber lediglich der Besitzer hat Schreibrechte. Dies entspricht den Rechten `rwxr-xr-x`. Hier wird der Befehl rekursiv angewandt, es werden also auch die Berechtigungen der Unterverzeichnisse geändert.

```
atlantis:/ce_users/nehmer> chmod g+rxw todo.txt
```

Erteilt der Gruppe die Rechte `read`, `write` und `execute` auf die Datei `todo.txt`.

```
atlantis:/ce_users/nehmer> chmod -w todo.txt
```

Wird lediglich eine Rechteänderung angegeben, so gilt sie für alle drei Ebenen. Hier würden also dem Eigentümer, der Gruppe und dem Rest des Netzes die Schreibrechte entzogen.

```
atlantis:/ce_users/nehmer> chmod g+x-rw todo.txt
```

Es können auch gleichzeitig Rechte gesetzt und entfernt werden. Dieses Kommando würde der Gruppe `execute`-Rechte zuweisen, jedoch gleichzeitig `read`- und `write`-Rechte entziehen.

2.5.4 Ändern der Gruppenzuordnung

```
chgrp [-R bzw. --recursive] gruppe dateien
```

`chgrp` ändert die Gruppenzuordnung der angegebenen Dateien. Dabei kann eine Datei natürlich nur in Gruppen gestellt werden, in denen man auch Mitglied ist.

`-R` bzw. `--recursive` verändert die Gruppenzuordnung auch in allen angegebenen Unterverzeichnissen.

2.6 Dokumentation und Online-Hilfen

Für Unix-Systeme gibt es Dokumentationen in den verschiedensten Formen. Durchgesetzt hat sich die Online-Dokumentation, da eine vollständige Unix-Dokumentation sehr schnell ganze Regalwände füllen kann.

2.6.1 man-Pages

Die wohl wichtigsten Unix-Hilfsmittel sind die sogenannten `man`-Pages (`man` = manual). Mit deren Hilfe hat jeder Benutzer Zugriff auf eine äusserst umfangreiche Online-Dokumentation zu vielen Programmen, Konfigurationsdateien und C-Funktionen. Gesteuert wird die Anzeige über das Programm `man`. Da `man` als einfaches Kommandozeilen-Utility konzipiert ist, ermöglicht es eine schnelle Suche nach Begriffsdefinitionen während der laufenden Arbeit.

Der größte Nachteil der `man`-Pages ist die fehlende Strukturierung der Texte. Dies macht sich vor allem bei langen Texten negativ bemerkbar. Aus diesem Grunde erfolgt die Dokumentation zu GNU-Software häufig durch `info` (siehe Kapitel 2.6.2), daß auch Querverweise innerhalb von Hilfetexten ermöglicht.

Um das Suchen von `man`-Pages zu vereinfachen, wurden sie in Themenbereiche untergliedert. Dies wurde vor allem dadurch notwendig, da für einen Begriff hin und wieder mehrere `man`-

Pages existieren. Ohne weitere Angaben zeigt man nur den ersten gefundenen Hilfetext an. man kennt folgende Themenbereiche (Auszug aus `man man` der Linux-Welt):

- 1 Executable programs or shell commands
- 2 System calls (functions provided by the kernel)
- 3 Library calls (functions within system libraries)
- 4 Special files (usually found in `/dev`)
- 5 File formats and conventions eg `/etc/passwd`
- 6 Games
- 7 Macro packages and conventions eg `man(7)`, `groff(7)`.
- 8 System administration commands (usually only for root)
- 9 Kernel routines [Non standard]

In vielen Unix-Büchern werden zusammen mit den Kommandos auch gleich die Themenbereiche angegeben – zum Beispiel `find(1)`. Da sich dies auch innerhalb von man-Pages als Standard für Referenzen etabliert hat, will auch ich dies im folgenden beibehalten. Das Kommando `man(1)` selbst besitzt folgende Syntax:

```
man [optionen] [bereich] thema
```

Die optionale Angabe eines Bereiches schränkt die Suche nach man-Pages auf einen Themenbereich ein – etwa `man 1 man`.

-a

zeigt der Reihe nach alle gleichnamigen man-Pages an. Dieser Parameter ist nur unter Linux und unter SunOS verfügbar! Linux zeigt dabei eine man-Page nach der anderen getrennt an, während SunOS alle man-Pages zu einer einzigen zusammenstellt.

-k

zeigt eine Liste aller vorhandenen man-Pages an, in denen das Schlüsselwort vorkommt. Dabei wird allerdings keine Volltextsuche durchgeführt, sondern lediglich in einer kleinen Übersicht gesucht.

Verschiedene Tastaturkommandos stehen zur Anzeige von man-Pages zur Verfügung:

<code>Return</code>	Scrollt um eine Zeile nach unten
<code>Space</code>	Scrollt um eine Bildschirmseite nach unten
<code>b</code>	Scrollt um eine Bildschirmseite nach oben
<code>h</code>	Zeigt einen (plattformabhängigen) Hilfetext an
<code>/</code> muster <code>Return</code>	Sucht den Begriff <code>muster</code> nach unten.
<code>q</code>	Beendet man

Eine man-Page kann sich unter anderem aus folgenden Abschnitten zusammensetzen:

NAME Kurzbeschreibung des Befehls

SYNOPSIS Die Syntax des Befehls mit allen möglichen Optionen

DESCRIPTION Ausführliche Beschreibung des Befehls

OPTIONS Erklärung der Optionen

ENVIRONMENT Liste aller Umgebungsvariablen (siehe Kapitel 2.7), die den Befehl beeinflussen

EXAMPLES Einige Beispiele zu dem behandelten Thema

FILES Dateien, die mit dem behandelten Thema zusammenhängen

```
User Commands ls(1)

NAME
  ls - list contents of directory

SYNOPSIS
  /usr/bin/ls [ -aAbcCdFgILmnoPqrRstux1 ] [ file ... ]

  /usr/xpg4/bin/ls [ -aAbcCdFgILmnoPqrRstux1 ] [ file ...]

DESCRIPTION
  For each file that is a directory, ls lists the contents
  of the directory; for each file that is an ordinary file,
  ls repeats its name and any other information requested.
  ...

OPTIONS
  The following options are supported:

  -a      List all entries, including those that begin with
          a dot (.), which are normally not listed.
  ...

EXAMPLES
  Example 1: Using the ls command.

  An example of a file's permissions is:
  ...

ENVIRONMENT VARIABLES
  See environ(5) for descriptions of the following environment
  variables that affect the execution of ls: LC_COLLATE,
  LC_CTYPE, LC_TIME, LC_MESSAGES, NLSPATH, and TZ.
  ...

FILES
  /etc/group
          group IDs for ls -l and ls -g
  ...

SEE ALSO
  chmod(1), cp(1), setfacl(1), terminfo(4), attributes(5),
  environ(5), largefile(5), XPG4(5)
```

Abbildung 13: Auszug aus ls(1) einer Sun Solaris

SEE ALSO Andere man-Pages, die mit dem aktuellen Sachgebiet verwandt sind.

BUGS Hier finden sich bekannte Fehler und Tips zur Nutzung

Je nach Sachgebiet kann es vorkommen, das noch andere Abschnitte hinzukommen oder einige der oben genannten fehlen. Gerade die man-Pages von Konfigurationsdateien (zum Beispiel `hosts.allow(5)` oder `nfs(5)`) beinhalten oft nur Name, Description, Examples und See Also. Ein Beispiel einer man-Page findet sich in Abbildung 13.

Da jedes Unix-System seine eigenen man-Pages verwaltet, sind die man-Pages verschiedener Plattformen in der Regel unterschiedlich. Besonders die man-Pages aus der Linux-Welt fallen recht kurz aus, vergleicht man sie mit den selben man-Pages von Ultrix oder Sun Solaris.

Im folgenden werden einzelne Programme, die angesprochen werden, nicht mehr detailliert erläutert. Lediglich die im speziellen Kontext notwendigen Parameter werden angesprochen, weiterführende Informationen könnt Ihr dann aus den man-Pages zu den entsprechenden Befehlen erhalten.

2.6.2 GNU-Dokumentation (info-Pages)

```

...
=====
==          FOO-Pool Info Pages START          ==
=====

* Info: (info).          Documentation browsing system.
...

=====
==          FOO-Pool Info Pages ENDE          ==
=====
...

```

Abbildung 14: Abtrennung der Info-Pages des Foo-Pools

Da man gerade bei umfangreichen Programmen (beispielsweise der Editor `emacs`) sehr schnell an die Grenze der man-Pages stösst, existiert parallel zu den man-Pages eine weitere Informationsquelle: Die info-Pages. Im Gegensatz zu den linear aufgebauten man-Pages nutzt das info-System ein Hypertext-Medium⁷. info-Pages existieren hauptsächlich für Software innerhalb des GNU-Projektes.

Als bestes Tool zum Betrachten der info-Datenbestände findet sich im `foo-Pool` der Editor `emacs(1)`. An dieser Stelle soll `emacs` selbst noch nicht weiter behandelt werden, da dies ein äusserst umfangreiches Programmpaket ist. Anhang C geht kurz auf die Tastaturbelegung dieses äusserst flexiblen Editors ein. Gestartet wird `emacs` entweder über das `fvwm`-Menü „Anwendungen → Applications → emacs“, oder direkt über die Kommandozeile:

```
atlantis:/ce_users/nehmer> emacs (Return)
```

Nach dem Start von `emacs` kann auf die info-Pages durch den Menüpunkt „Help → Manuals → Browse Manuals with Info“ aktiviert. Daraufhin wird die info-Datenbank angezeigt. In

⁷Ein Hypertext-Medium charakterisiert sich dadurch, daß Textstellen direkt auf andere Texte verweisen können. Erhält man in einem gedruckten Buch Begriffserklärungen über den Index, kann man in (elektronischen) hypertext-medien einfach den betreffenden Begriff anwählen und bekommt die entsprechenden Seiten angezeigt.

```

/usr/doc/.
|-- FAQ
|   |-- html
|   |-- ps
|   |-- txt
|-- HOWTO
|   |-- mini
|   |-- other-formats
|       |-- sgml
|       |-- ps
|-- LDP
|   |-- install-guide
|   |-- lpg
|   |-- nag
|   |-- sag
|-- AfterStep-1.7.90
|-- AnotherLevel-0.9
|-- ElectricFence-2.1
|-- gv-3.5.8
|-- ImageMagick-4.2.2
|-- ..

```

Abbildung 15: Verzeichnisstruktur /usr/doc auf Linux-Systemen

der obersten Zeile befindet sich die Navigationsleiste, mit der man sich recht bequem im aktuellen Themenbereich bewegen kann. Alle Links sind im Text in Fettdruck hervorgehoben. Ein Link kann sowohl über Tastatur als auch über die Maus ausgewählt werden: Entweder man klickt den Link mit der mittleren Maustaste an, oder man bewegt den Cursor auf den Link und drückt **(Return)**. Beendet wird emacs durch den Menüpunkt „File → Exit emacs“.

Wie bei man-Pages unterscheidet sich auch der Bestand an info-Pages von Plattform zu Plattform. Aus diesem Grunde gibt es einen zentralen info-Bestand, der auf jedem Rechner verfügbar ist. Man erkennt ihn an den in Abbildung 14 zu sehenden Textzeilen.

2.6.3 Weitere Dokumentation

Während die man- und info-Pages auf allen Plattformen verfügbar sind, findet sich auf Linux-Systemen eine ganze Reihe weiterer Online-Dokumente im Verzeichnis /usr/doc. Die Verzeichnisstruktur dieses Teils ist in Abbildung 15 abgebildet. Im folgenden sollen die einzelnen Teile hiervon ein wenig näher erläutert werden.

2.6.4 FAQ - Frequently Asked Questions

Frequently Asked Questions sind, wie der Name schon vermuten lässt, Sammlungen mit häufig gestellten Fragen und den Antworten dazu. FAQ's existieren zu jedem nur denkbarem Thema. In der Regel werden FAQ's regelmässig in den dazugehörigen Newsgroups (siehe Kapitel 3.3) gepostet oder sind auf den Homepages der betreffenden Programme im WWW (siehe Kapitel 3.1) zu finden. Speziell auf Linux-Systemen werden unter /usr/doc/FAQ die FAQ's der installierten Software untergebracht.

Vom Dateiformat abhängig, kommen vier Programme in Frage um eine FAQ zu betrachten.

Liegt die FAQ als reiner Text vor (Keine Dateierdung oder `.txt`), kann man sich die FAQ direkt in einem Terminal mit Hilfe von `more` (1) anzeigen lassen. Befindet man sich auf einem Linux-System, kann auch anstelle von `more` das etwas komfortablere `less` (1) eingesetzt werden. Die Bedienung dieser Programme ähnelt dem man-Page Viewer (siehe Kapitel 2.6.1).

Weiterhin existieren viele FAQ's im HTML-Format, dem Format des WWW (siehe Kapitel 3.1). Betrachtet werden diese mit dem Browser Netscape, der im WWW-Kapitel näher erläutert wird.

Will man FAQ's drucken, so existiert oft eine dritte Möglichkeit: FAQ's im PostScript Format. Dies sind fertige FAQ's in Buch-Form, die direkt auf unseren Drucker geschickt werden können (siehe Kapitel 2.9).

2.6.5 HOWTO - Wie geht das?

HOWTO's besitzen eine ähnliche Grundlage wie FAQ's. Sie existieren zu vielen, häufig gefragten Themen. Sie unterscheiden sich von FAQ's dadurch, das sie sehr viel systematischer aufgebaut sind. Eine HOWTO ist am ehesten mit einer kurzen Einführung in ein bestimmtes Thema gleichzusetzen. Die German-HOWTO beschreibt beispielsweise die Anpassung eines Linux-Systems an eine Deutsche Sprachumgebung. Viele dieser HOWTO's sind zwar speziell für Linux-Systeme geschrieben, lassen sich aber hier und da auch auf andere UNIXe anwenden.

Wie die FAQ's existieren von den HOWTO's bis zu drei Versionen: Reiner Text, HTML und PostScript.

2.6.6 LDP - Das Linux Documentation Project

Verglichen mit man- oder info-pages beinhaltet das LDP ganze Bücher. Das Linux Documentation Project beruht auf einer Initiative einiger Linux-Benutzer, eine komplette Linux-Dokumentation auf die Beine zu stellen. Sie gehören mit zum besten, was an Linux-Literatur zu finden ist. Zum LDP gehörten die Linux Programmers' Guide (LPG), die Network Administrators' Guide (NAG) und die System Administrators' Guide (SAG). Diese Bücher gibt es lediglich als HTML und als PostScript.

2.6.7 Dokumentation zu einzelnen Programmpaketen

Ebenfalls unter `/usr/doc` befinden sich die Dokumentationen zu einzelnen Programmpaketen, wobei jedes Paket sein eigenes Unterverzeichnis besitzt. Der Name eines solchen Verzeichnisses setzt sich aus dem Namen des Programmpaketes und seiner Versionsnummer zusammen. Der PostScript-Betrachter `gv` (1), Version 3.5.8, besitzt seine Dokumentation so unter `/usr/doc/gv-3.5.8`. Der Inhalt dieser Verzeichnisse variiert von Programm zu Programm, in der Regel findet sich aber eine FAQ, eine Liste mit bekannten Fehlern (BUGS) und eine kurze Beschreibung des Programmes (README).

2.7 Environment (Die Umgebung)

Viele UNIX-Programme, selbst die Shell, brauchen Informationen über Dich, was Du willst und über die Umgebung in der sie ablaufen.

Welche Informationen? Textbasierte Programme wie man müssen wissen, in was für einem Terminal-Typ sie laufen. Eine Textkonsole von Linux verhält sich anders als beispielsweise ein `xterm`. Oft wird auch ein Editor innerhalb eines Programmes gestartet (beispielsweise E-Mail

Software), und diese Programme benötigen dadurch die Information über Deinen Lieblingseditor, mit dem Du Mails bearbeiten willst.

Natürlich könnte man diese Informationen jedem Programm als Parameter mitteilen. Allerdings ist das nicht sehr komfortabel. Dein Lieblingseditor wird sich wohl nicht jeden Tag ändern, und ein Terminal verändert sich während seiner Laufzeit ebensowenig.

Um diesem Problem entgegenzuwirken, nutzt UNIX Umgebungsvariablen um diese Informationen zu speichern. Jede Umgebungsvariable besteht dabei aus einem Namen und einem Wert. Beispielsweise sagt Dir die Variable `$TERM`, was für einen Terminal-Typus Du gerade verwendest. Auf diese Weise können alle Programme, denen der Terminal-Typ von Bedeutung ist, diesen Abfragen und entsprechend reagieren.

Auf eine ähnliche Weise werden in der Variable `$PATH` die Verzeichnisse abgespeichert, in denen ausführbare Programme gesucht werden. Die hier angegebenen Pfade werden von der Shell der Reihe nach abgearbeitet.

Umgebungsvariablen können vom Benutzer beliebig definiert werden. Jedoch gibt es einige vordefinierte Variablen, die alle eine bestimmte Funktion haben und weiter unten beschrieben werden.

2.7.1 Anzeigen von Umgebungsvariablen

Natürlich ist eine Umgebungsvariable auch für den Benutzer lesbar. Dies erledigt das Kommando `printenv (1)`, daß entweder alle oder eine bestimmte Variable ausgibt:

```
atlantis:/ce_users/nehmer> printenv
HOME=/ce_users/nehmer
SHELL=/bin/tcsh
TERM=xterm
USER=nehmer
LOGNAME=nehmer
PATH=./ce_users/nehmer/bin:/usr/local/bin: ...
...
EDITOR=vi
atlantis:/ce_users/nehmer> printenv TERM
xterm
```

2.7.2 Setzen von Umgebungsvariablen

Analog zu `printenv` lässt sich mit `setenv`⁸ eine Umgebungsvariable auf einen bestimmten Wert setzen:

```
atlantis:/ce_users/nehmer> setenv LOVER Judy
atlantis:/ce_users/nehmer> setenv EDITOR /usr/local/bin/emacs
```

2.7.3 Löschen von Umgebungsvariablen

Selbstverständlich lassen sich gesetzte Umgebungsvariablen auch wieder löschen. Dies erledigt der Befehl `unsetenv`.

⁸`setenv` selbst besitzt keine man-Page, da es ein Bestandteil der `tcsh (1)` ist. Genauere Informationen finden sich dort. Man beachte, daß Bourne-Shells (`sh (1)`, `bash (1)`) den Befehl `export` erwarten. Da im foo-Pool die standardmäßig `tcsh` verwendet wird, soll dies hier aussen vor bleiben.

```
atlantis:/ce_users/nehmer> unsetenv LOVER
```

2.7.4 Wichtige Umgebungsvariablen

Wie bereits erwähnt gibt es einen Standardbestand an Umgebungsvariablen. Standard in sofern, daß ihre Namen festgelegt sind. Sie werden nicht automatisch von UNIX-Systemen gesetzt – Dies erledigen die Login-Scripte. Wenn Du Deine Umgebungsvariablen auf eigene Spezialitäten anpassen willst, dann mußt Du die `setenv`-Anweisungen in Deine `.cshrc` aufnehmen, die bei jedem Start einer Shell automatisch ausgeführt wird.

Hier sind die wichtigsten:

\$PATH beinhaltet den Kommando-Suchpfad. Es ist eine durch Doppelpunkte getrennte Liste von Verzeichnissen, in der die Shell nach Programmen sucht. Dabei wird das erste gefundene Programm ausgeführt.

\$EDITOR, \$VISUAL kann mit dem Namen Deines Lieblingseditors belegt werden. Einige Programme unterscheiden nach folgendem Kriterium zwischen `$EDITOR` und `$VISUAL`: `$EDITOR` bezeichnet einen Zeilenorientierten Editor (z.B. `ed(1)`) während `$VISUAL` einen Vollbildeditor (wie `vi(1)` oder `emacs(1)`) festlegt. Viele Leute, unter anderem auch wir hier im foo-Pool, nehmen diese Unterscheidung nicht vor und setzen beide Variablen auf den gleichen Editor. Hier im foo-Pool ist standardmässig `vi(1)` (siehe Anhang B) als `$EDITOR` und `$VISUAL` festgelegt.

\$PRINTER kann auf den Namen Deines Standarddruckers gesetzt werden. Dadurch brauchst Du `lpr(1)` nicht jedesmal den Namen mitteilen. Diese Variable wird im den Standard foo-Scripten nicht gesetzt. Nähere Informationen zum Thema Drucken findest Du in Kapitel 2.9.

\$HOME enthält den Pfad zu Deinem Homeverzeichnis.

\$USER, \$LOGNAME enthält Deinen Benutzernamen.

\$TERM enthält den aktuellen Terminal-Typ. Er wird von den verschiedenen Terminals automatisch gesetzt. Unter bestimmten Bedingungen kann jedoch ein manuelles Setzen des Terminals notwendig werden. Am besten eignet sich `vt100` als absolutes Basis-System und `xterm` für ein graphisches Terminal.

\$MANPATH enthält den Suchpfad für `man(1)`. Diese Verzeichnisliste arbeitet genau wie `$PATH` und dient dazu, verschiedene man-Verzeichnisse zu ermöglichen.

\$DISPLAY enthält den Namen des Rechners, auf dem X-Anwendungen (siehe Kapitel 2.2) angezeigt werden. Mit Hilfe dieser Variablen können auch graphische Anwendungen auf entfernten Rechnern ausgeführt werden. Wie man Anwendungen auf anderen Rechnern ausführt, wird in Kapitel 2.10 näher beschrieben.

Wer sich das Standard-Environment des foo-Pools einmal näher ansieht, wird noch einige andere, hier nicht erläuterte, Variablen finden. Viele davon sind selbsterklärend. Man halte sich hier an die man-Pages der verschiedenen Programme des foo-Pools, die im Environment-Teil alle Umgebungsvariablen, auf die das Programm zugreift, auflisten. Je nach dem kann auch die Umgebungsvariable selbst eine man-Page besitzen.

2.8 Prozesse und deren Steuerung

Wie jedes Multiprocessing System besitzt auch Unix verschiedene Möglichkeiten, laufende Prozesse zu beeinflussen. Man kann sich Prozessinformationen anzeigen lassen und „abschieden“, sollten sie ausser Kontrolle geraten.

2.8.1 Anzeigen von Prozessinformationen

Unix stellt mit dem Kommando `ps (1)` ein recht umfangreiches Tool zur Verfügung. Ein einfacher Aufruf von `ps` führt dazu, daß alle innerhalb des aktuellen Terminals laufende Prozesse aufgelistet werden. Es handelt sich dabei um ein recht einfaches Listing, das aus Prozeß-ID (PID), Terminal Code, CPU-Zeit und Prozeß-Name besteht. Da man aber gerade unter X nicht immer alle Prozesse aus dem gleichen Terminal startet, gibt es auch die Möglichkeit, alle laufenden Prozesse auszugeben.

Hier trennen sich jetzt die Wege. Sun Solaris und OSF/1⁹ benötigen hier den Aufruf `ps -edf`, während NetBSD, Ultrix und Linux `ps -aux` wollen. (Linux will genau genommen `ps aux`, erkennt aber auch die andere Syntax). Danach wird man von einer riesengroßen Tabelle überfallen. Wichtigster Vorteil dieser Ansicht ist die Anzeige des Benutzernamens. So lassen sich recht einfach die eigenen Prozesse herausfinden.

An dieser Stelle werden wir mal einen kleinen Abstecher in das Kapitel Shells machen, das Grundlage ganzer Bücherserien ist: Es besteht die Möglichkeit, eine Ausgabe nach bestimmten Begriffen zu durchsuchen und sich nur die Zeilen anzeigen zu lassen, die den Suchbegriff enthalten. Dies erreicht man mit Hilfe des Programmes `grep (1)` und der Möglichkeit, die Ausgabe eines Programmes mit dem Zeichen `|` an ein anderes weiterzugeben. Man nennt dieses Zeichen die „Pipe“:

```
atlantis:/ce.users/nehmer> ps aux | grep nehmer
Die Ausgabe von "ps aux" wird durch grep nach Zeilen mit dem Begriff "nehmer" durchsucht. Nur diese Zeilen werden tatsächlich ausgegeben.
```

USER	PID	%CPU	%MEM	...	START	TIME	COMMAND
nehmer	1588	0.0	1.0	...	15:22	0:00	-csh
nehmer	2597	74.1	10.9	...	15:28	55:06	./setiathome
nehmer	4895	0.0	0.7	...	16:29	0:06	ps aux
nehmer	3857	0.0	0.9	...	16:29	0:04	grep nehmer

Auf diese Weise lässt sich recht einfach nach einem bestimmten Prozeß oder nach den Prozessen eines bestimmten Users suchen. Der Aufruf `ps aux | grep setiathome` hätte die zweite Zeile geliefert, natürlich wieder die Zeile des Prozesses `grep` und eventuell weitere `setiathome`-Prozesse, die von anderen Benutzern gestartet worden sind.

Die Ausgabe von `ps` liefert einige interessantere Angaben. Ausser dem Benutzernamen kommen in der ausführlichen Angabe noch CPU- (%CPU) und Speicherlast (%MEM), CPU-Zeit (TIME), Startzeitpunkt (START), der Prozeß-Status (STAT) und einige andere Dinge, die in der man-Page beschrieben werden, hinzu. Man erhält also einen recht umfassenden Überblick über den aktuellen Systemzustand.

Eine etwas interaktivere Lösung bietet das Programm `top (1)`. Es liest in regelmässigen Abständen die aktuelle Prozeßtabelle ein und bereitet die Informationen tabellarisch auf. Dabei werden nur so viele Prozesse angezeigt, wie auf den Bildschirm passen. So lassen sich sehr schnell ressourcenfressende Prozesse finden. *Wichtig: Die DEC-OSF/1 Rechner besitzen dieses Programm nicht.*

⁹Man kann das aktuelle Betriebssystem mit `uname -a` anzeigen lassen (siehe Kapitel 2.3.3)

`top` liefert auf den Tastendruck (H) eine Übersicht der vorhandenen Kommandos. Die wichtigsten sind:

(Shift) + (P)	Sortiert nach Prozessorlast, dies ist die Standardeinstellung
(Shift) + (M)	Sortiert nach Speicherlast
(Shift) + (T)	Sortiert nach Prozessorzeit
(U)	Nach einem Benutzernamen filtern, <code>top</code> fordert zur Eingabe eines Benutzernamens auf.
(Q)	Beendet <code>top</code>

2.8.2 Ausser Kontrolle geratene Prozesse beenden

In jedem Betriebssystem kann es vorkommen, daß ein laufender Prozeß abstürzt. Deshalb muß eine Möglichkeit bestehen, solche Prozesse gewaltsam zu beenden. UNIX stellt dafür einen speziellen Mechanismus zur Verfügung: Signale. Ein Signal wird an einen Prozeß gesendet. Dieser kann nun darauf reagieren. Tut er es nicht, dann wird vom Kernel eine Standardreaktion ausgelöst, die in der Regel darin besteht, den Prozeß zu beenden.

Das Programm `kill(1)` kann zum Senden eines Signales verwendet werden. Die Syntax ist `kill -Signalnummer Prozeß-ID`. Gibt man keine Signalnummer an, wird das TERM-Signal gesendet, das einen Prozeß sanft zu beenden versucht. Wenn ein Prozeß sich komplett im „Nirwana“ befindet, reicht oft das TERM-Signal nicht mehr aus. Als letzte Hintertür besitzt Unix deshalb das Signal KILL mit der Nummer 9. Diese Signale können von einem Programm nicht abgeblockt werden, sondern veranlassen das Betriebssystem direkt, das Programm zu beenden.

Als weitere Möglichkeit, einen laufenden Prozeß zu beenden, dient innerhalb von Terminals die Tastenkombination (CTRL)+(C).

2.9 Drucken im Foo-Pool

Im foo-Pool steht allen Benutzern ein HP LaserJet zur Verfügung. Wie für die Workstations gilt auch für diesen Drucker: **Er darf nicht ausgeschaltet werden!** Treten während des Druckens Fehler auf, sagt uns bitte bescheid.

Der LaserJet akzeptiert zwei verschiedene Datenformate: Adobe PostScript¹⁰ und ASCII. Bei Adobe PostScript handelt es sich um eine vektororientierte Seitenbeschreibungssprache, die sich in der gesamten Branche als der Standard etabliert hat. Größter Vorteil von PostScript ist die Plattformunabhängigkeit. PostScript-Dateien sehen auf jedem postScriptfähigen Gerät gleich aus, egal von welcher Plattform oder welchem Programm sie erstellt wurden. ASCII-Daten dagegen sind reine Textdaten, die ohne weitere Konvertierung wie auf einer Schreibmaschine ausgegeben werden.

2.9.1 Queues (Warteschlangen)

In einer Mehrbenutzerumgebung ist es notwendig, Druckjobs (= Druckaufträge) in einer bestimmten Reihenfolge abzuarbeiten, da sich ansonsten mehrere Druckjobs gegenseitig blockieren könnten. Dazu dienen die Warteschlangen, genannt Queues, eines Druckservers. Dieser nimmt die Jobs der verschiedenen Clients anstelle des Druckers selbst in Empfang und

¹⁰Das bekanntere Adobe Portable Document Format PDF ist eine Erweiterung des PostScript Standards um Kompression, Inhaltsverzeichnisse, Hyperlinks u.a., das hauptsächlich für das Internet konzipiert ist.

Anwendung	Druckformat	Von HP-UX gestartet?
Adobe Acrobat Reader	PostScript	Ja
PostScript Previewer	PostScript	Nein
Emacs PostScript Printing	PostScript	Nein
Emacs Printing	ASCII-Text	Nein
Netscape Navigator 4.x	PostScript	Nein

Abbildung 16: Druckformate von typischen foo-Pool Anwendungen

schreibt sie in seine Queue hinein. Von dort aus wird kontrolliert ein Job nach dem anderen an den Drucker gesandt.

Ein Druckserver ordnet nun jedem angeschlossenen Drucker eine eigene Queue zu, wobei auch mehrere Queues auf den gleichen Drucker führen können. Im foo-Pool sind zwei Queues für den Zugriff auf den Laserdrucker eingerichtet. `fast` nimmt PostScript Ausgaben entgegen, während `lpr1` ASCII-Daten an den Drucker sendet. Generell gilt: Alle Programme, die ihre Ausgabe graphisch vornehmen (und sei es nur ein einfacher Fettdruck) liefern ihre Druckdaten im PostScript Format ab und können direkt auf die Queue `fast` geschickt werden. Ist man sich nicht sicher, kann als Notlösung immer noch über die ASCII-Queue `lpr1` gedruckt werden, da diese eventuelle PostScript Ausgaben erkennen kann und sie korrekt bearbeitet. Allerdings ist der Druck über `lpr1` erheblich langsamer, da er nicht über die Netzwerkanbindung des Druckers, sondern über dessen serielle Anbindung läuft. Tabelle 16 zeigt eine kleine Übersicht der am häufigsten verwendeten Programme im Foo-Pool.

2.9.2 Druckjobs abschicken

Das Programm `lpr (1)` ist für das Einschleusen eines Druckjobs in die entsprechende Queue zuständig. An dieser Stelle wird die heterogene Umgebung des foo-Pools wieder zu einer kleinen Stolperfalle: Die im HP-UX vorhandene `lpr`-Implementation erwartet den Namen der Queue über den Parameter `-d`, während alle anderen Systeme des foo-Pools die Queue per `-P` erwarten. Der Name der Queue wird in beiden Fällen direkt ohne Leerzeichen an den Parameter gehängt: z.B. `lpr -Pfast` oder `lpr -dfast`.

Dieses Problem der unterschiedlichen `lpr`-Befehle macht sich besonders bemerkbar, wenn Anwendungen auf anderen Rechnern ausgeführt werden (siehe Kapitel 2.10). Dadurch kann es vorkommen, das man selbst nicht an einer HP-Maschine sitzt, aber in der betroffenen Anwendung trotzdem die HP-Syntax verwenden muß. Tabelle 16 gibt deshalb ebenfalls darüber Aufschluß, von welchem System aus die Programme gestartet werden.

Hat man sich für eine Queue entschieden, gibt man noch die zu druckende Datei an, fertig. Wenn man von einem Programm nach einem Druckerkommando gefragt wird, muß man keinen weiteren Dateinamen angeben!

Beispiel: Will man die Postscript-Datei `ausarbeitung.ps` drucken, so kann man dies (auf nicht-HP-Rechnern) mit dem Kommando `lpr -Pfast ausarbeitung.ps` erreichen.

2.9.3 Wartende Druckjobs anzeigen

Auch hier verhalten sich die beiden HP-Systeme wieder ein wenig anders, als der Rest des Pools. Wer also gerade vor einer HP sitzt, sollte sich noch einen Moment gedulden.

Die Druckerwarteschlange wird mit Hilfe des Programmes `lpq (1)` ausgegeben. Auch hier ist wieder die Angabe der Queue mit Hilfe des Parameters `-P` notwendig. Ein `lpq -Pfast` listet also alle am PostScript-Queue anliegenden Druckaufträge auf.

HP's benötigen im Gegensatz hierzu lediglich ein `lpstat(1)` mit dem Parameter `-t`. Dadurch werden alle Queue's nach Druckaufträgen abgesucht. Die Angabe eines bestimmten Druckers ist nicht notwendig.

2.9.4 Eigene Druckjobs aus der Warteschlange löschen

Ihr ahnt es schon? Richtig. Beim Löschen kommt Ihr auch nicht um das HP-Problem drumherum. Also wieder das gleiche Spiel: Wer vor einer HP sitzt...

Die Ausgabe von `lpq` liefert zu jedem Druckjob eine ID. Diese bezeichnet, ähnlich wie Prozeß-IDs einen bestimmten Druckjob. Mit Hilfe dieser ID und des Programmes `lprm(1)` lassen sich bereits abgeschickte Druckjobs wieder abrechnen. Es ist ebenfalls wieder die Druckerqueue über den Parameter `-P` anzugeben. Man erhält also: `lprm -P<Queue> <Job-ID>`.

Wie bereits angedeutet, die HPs nehmen hier völlig getrennte Wege: Das Kommando `cancel(1)` ist hier das Werkzeug: `cancel <Queue>-<Job-ID>` löscht den angegebenen Druckjob.

2.10 Nutzen anderer Rechner (ssh, rlogin, rsh, telnet)

Der wohl grösste Vorteil von UNIX-Umgebungen gegenüber der Windows-Welt ist die Möglichkeit, Programme wahlweise auf entfernten Rechnern auszuführen. Dabei wird die Ein- und Ausgabe über das Netzwerk auf das lokale System umgeleitet. Diese Funktionalität ist ein integraler Bestandteil jedes Unix-Systems und wird hier im foo-Pool an vielen Stellen verwendet. Beispielsweise werden ressourcenintensive Programme wie der Netscape Navigator grundsätzlich auf einem leistungsstarken System (derzeit ist es die Sun Workstation `son`) gestartet. Genauso können Programme die nur auf einer Plattform verfügbar sind, auf diese Weise auch an anderen Rechnern verwendet werden.

2.10.1 Remoteshell-Typen

Die Secure Shell (`ssh(1)`) Die Secureshell ist die neuste Variante der Remoteshells. Die `ssh` merzt das größte Manko der anderen Remoteshells aus: Sie verschlüsselt die komplette Session von Anfang bis Ende. Zur Authentifizierung kennt die `ssh` zwei Möglichkeiten. Ohne weiteres zutun fordert sie Benutzernamen und Paßwort. Alternativ kann man sich mit einem sogenannten Identity-Key authentifizieren. Dieses Schlüsselpaar muß auf beiden Seiten der Verbindung vorhanden sein, dann wird die Paßwortabfrage übersprungen. Da im foo-Pool standardmässig für jeden neuen User ein solches Schlüsselpaar angelegt wird, kann innerhalb des foo-Pools ohne Paßwortabfrage gewechselt werden. Als einziger Dienst mit User-Identifizierung ist die Secureshell auch von ausserhalb des FH-Netzes nutzbar. Alle anderen Dienste werden nur innerhalb des FH-Netzes akzeptiert, um die Gefahr des Paßwortklaus zu minimieren.

Die Remoteshell (`rlogin(1)`, `rsh(1)`) Die Remoteshell existiert schon seit langer Zeit. Sie führt direkt Befehle auf anderen Rechnern aus, und leitet die Eingabe zum aktuellen Rechner um. Dabei ist aber alles unverschlüsselt. Normalerweise fragt auch die Remoteshell nach dem Paßwort. Da dies aber auf Grund der Klartextübertragung nicht wünschenswert ist, kann die Authentifizierung eines anderen Systems übernommen werden. Dadurch wird die Paßwortabfrage überflüssig, denn es wurde auf dem lokalen System ja bereits eingegeben. Man kann also (nur) unter seiner Benutzer-ID ohne Paßwortabfrage auf anderen Systemen arbeiten. Welchen Rechnern nun vertraut wird, legt die Datei `~/ .rhosts` fest. Auf dieser Basis werden im foo-Pool viele Programme entfernt gestartet. Die `rsh` ist dabei nichts anderes, als ein gekapselter `rlogin`-

Aufruf. Während `rlogin` immer eine Eingabeshell öffnet, kann man der `rsh` direkt die auszuführende Kommandozeile mitgeben. `rsh` wird daher primär dazu verwendet, Programme auf anderen Rechnern direkt zu starten, während `rlogin` zum starten von Shells auf anderen Rechnern genutzt wird. Da `rlogin` durch die unverschlüsselte Paßwortübertragung aber recht große Sicherheitslücken öffnete, wird der `rlogin`-Aufruf automatisch durch die `ssh` ersetzt. Innerhalb des Pools war dieses Problem nicht allzu schwerwiegend, da sich `rlogin` ja auch auf die `.rhosts`-Datei verlässt, jedoch wurden Aufrufe von und nach aussen immer mehr zum Problem. *Wichtig*: Auf den HP-Rechnern im `foo`-Pool nennt sich die `rsh` `remsh(1)`! Dies werde ich in den folgenden Beispielen nicht mehr gesondert erwähnen.

Telnet (`telnet(1)`) Telnet war das erste Remoteshell-System, daß für das Internet konzipiert ist. Es arbeitet ausschliesslich über TCP/IP Verbindungen. Wie die herkömmlichen Remoteshells besitzt auch Telnet keine Verschlüsselungsmechanismen. Dazu kommt noch der Nachteil, daß Telnet sich nicht auf andere Systeme verlässt, sondern die Authentifizierung immer selbst durchführt. Im `foo`-Pool selbst spielt Telnet eine untergeordnete Rolle. Es wird nur in seltenen Fällen, z.B. zur Wartung, verwendet.

Faßt man dies zusammen, lassen sich die verschiedenen Remoteshells in zwei Kategorien unterteilen. Die erste Gruppe führt direkt Kommandos auf anderen Rechnern aus und nutzt eine interne Authentifizierung, so daß kein Paßwort übertragen werden muß. Die zweite Gruppe dient mehr zur Administration, denn sie öffnet eine neue Shell im Zielsystem und wird häufig auch ausserhalb des lokalen Netzes verwendet. Gerade in letzterem Falle ist heute eine Verschlüsselung sinnvoll. Aus diesem Grunde kommen von den vier vorhandenen Programmen in der Regel nur zwei zur Nutzung: Die `ssh` und die `rsh`.

Die `rsh` wird von vielen Skripten dazu verwendet, automatisiert Software auf anderen `foo`-Rechnern zu starten¹¹. Oft will man aber komplexe Operationen auf einem anderen System ausführen. In diesem Falle wäre das einzelne Absetzen von `rsh`-Kommandos doch sehr mühselig. Deshalb nutzt man die `ssh`, um sich erst einmal eine Shell auf dem anderen Rechner zu öffnen, und dort dann Schritt für Schritt die gewünschten Operationen durchzuführen.

2.10.2 Grundlagen

Nach der ganzen Theorie nun mal ein wenig Praxis. Vermutlich wirst Du zu Hause viel unter Windows arbeiten. Dadurch kommst Du natürlich in die Verlegenheit, Dateien irgendwie in den `foo`-Pool zu schaffen. Ein Weg wäre der Datenaustausch über Disketten. Nun sind nur in wenigen Rechnern des `foo`-Pools PC-fähige Diskettenlaufwerke zugänglich. Die sind die Linux-Rechner, die mit den `mttools(3)` ausgestattet sind. Dies ist ein spezielles Programmpaket, mit dem Disketten aus der DOS-Welt genutzt werden können. Gerade weil es aber so wenig Linux-Rechner gibt, ist es nicht immer ganz leicht, einen freien Platz zu ergattern. Mit unserem Wissen über Remoteshells ist das auch gar nicht mehr nötig. Hier im folgenden soll uns folgendes Problem als Beispiel dienen: Wie bekomme ich die Text-Datei `readme.txt` in den `foo`-Pool. Genutzt wird das Diskettenlaufwerk von `goedel`. In Abbildung 17 ist dieses Beispiel zu sehen.

Wir loggen uns zunächst einmal auf einem Linux-System ein. Dies erledigen wir mit Hilfe der `ssh`. Die `ssh` muß natürlich noch wissen, auf welchem System wir ankommen wollen, deshalb geben wir ihr den Rechnernamen als normales Argument mit. Während des Loginprozesses (1) tauchen verschiedenste Meldungen auf: Erst einmal kommt die sogenannte „Message of the day“ mit deren Hilfe die Systemadministration wichtige Informationen verbreiten können.

¹¹Diese Skripte befinden sich im Verzeichnis `/usr/local/share/bin`

```

atlantis:/ce_users/nehmer> ssh goedel
Loginmeldungen von goedel erscheinen jetzt. (1)
goedel:/ce_users/nehmer> mdir a: (2)
Volume in drive A has no label
Volume Serial Number is 4C13-BA21
Directory for A:/
test      cpp      1205 06-25-1999  8:14a
readme   txt      1172 06-25-1999  8:14a
          2 file(s)                2 377 bytes
                               1 426 442 bytes free
goedel:/ce_users/nehmer> mcopy a:readme.txt . (3)
Copying readme.txt
goedel:/ce_users/nehmer> exit (4)
atlantis:/ce_users/nehmer>

```

Abbildung 17: Zugriff auf einen anderen Rechner via ssh

```

atlantis:/ce_users/nehmer> rsh goedel 'mdir a:' (1)
Volume in drive A has no label
Volume Serial Number is 4C13-BA21
Directory for A:/
test      cpp      1205 06-25-1999  8:14a
readme   txt      1172 06-25-1999  8:14a
          2 file(s)                2 377 bytes
                               1 426 442 bytes free
atlantis:/ce_users/nehmer> rsh goedel 'mcopy a:readme.txt .' (2)
Copying readme.txt
atlantis:/ce_users/nehmer>

```

Abbildung 18: Zugriff auf einen anderen Rechner via rsh

Danach folgen einige Meldungen der Loginskripte¹².

Als nächster Schritt sollte nun die Diskette in goedels Diskettenlaufwerk eingelegt werden. Dazu sollte man sich sicherheitshalber kurz mit demjenigen kurzschliessen, der goedel gerade nutzt – für den Fall das er das Diskettenlaufwerk selbst braucht. Zum Zugriff auf die DOS-Diskette dienen nun die Programme `mcopy` (1), `mdir` (1) und `mdel` (1). All diese Programme funktionieren wie ihre DOS-Geschwister. Wir lassen uns also den Inhalt der Diskette einmal anzeigen (2), und kopieren anschliessend `readme.txt` in unser Home-Verzeichnis (3):

Jetzt befindet sich die Datei `readme.txt` in unserem Homeverzeichnis¹³. Natürlich sollten wir die neue Shell wieder sauber beenden. Dies geschieht, wie immer, mit `exit` (4).

2.10.3 Direktes Ausführen einzelner Programme

Ich habe in der Übersicht bereits auf die Möglichkeit hingewiesen, Programme mit Hilfe von `rsh` direkt zu starten, ohne den Umweg über eine Loginshell. Dies können wir natürlich auch auf obiges Beispiel anwenden:

¹²Hier kann es je nach Kombination zu der einen oder anderen Warnung kommen, die aber an dieser Stelle getrost ignoriert werden können.

¹³Kapitel 2.11 geht auf die speziellen Besonderheiten ein, die hier eigentlich noch zu beachten wären und aus den unterschiedlichen Text-Formaten von DOS und UNIX resultieren.

```
atlantis:/ce_users/nehmer> xhost +hanoi (1)
hanoi being added to access control list
atlantis:/ce_users/nehmer> ssh hanoi
hanoi:/ce_users/nehmer> setenv DISPLAY atlantis: (2)
hanoi:/ce_users/nehmer> /usr/local/acrobat3/bin/acroread &
[1] 7054 (3)
hanoi:/ce_users/nehmer>
```

Abbildung 19: Umleiten einer graphischen Anwendung via ssh

Wie in Abbildung 18 zu sehen, lassen wir uns zuerst wieder den Inhalt des Diskettenlaufwerks anzeigen (1). Dies erledigt der erste `rsh`-Aufruf. Dabei gilt die einfache Syntax `rsh <rechner> <kommando>`. Wichtig ist dabei, das Kommando in Hochkommas einzuschliessen. Damit wird verhindert, daß die Shell den Befehl bereits auf dem lokalen System interpretiert. Würdest Du sie weglassen, würde die Shell auf Grund des Leerzeichens ein drittes Argument generieren.

Interessanter ist der zweite Aufruf, der den eigentlichen Kopiervorgang einleitet. Hier kommt eine ganz wichtige Regel zum Tragen: *Auf dem Zielsystem wird eine neue Shell geöffnet, welche die üblichen Login-Prozeduren durchläuft.* Dadurch ist auf dem Zielrechner das Arbeitsverzeichnis grundsätzlich erst einmal das eigene Homeverzeichnis, unabhängig vom der aktuellen Position im Dateisystem! Die Pfadangabe `.` bezeichnet in diesem Fall also das eigene Homeverzeichnis und nicht etwa das aktuelle Arbeitsverzeichnis. (Auch wenn Home- und Arbeitsverzeichnis in diesem Beispiel identisch waren.)

2.10.4 Umleiten graphischer Anwendungen

Während sich eine Umleitung auf Terminal-Ebene doch recht einfach gestaltet, müssen bei graphischen Anwendungen einige Dinge beachtet werden. Im Gegensatz zur Textausgabe sorgt UNIX hier nicht für eine automatische Einstellung aller nötigen Parameter. Zwei Schritte sind hierbei notwendig: Zuerst muß auf dem lokalen System der Zugriff für das entfernte System freigegeben werden. Danach wird auf dem entfernten System der Ursprung als Ziel angegeben. Abbildung 19 demonstriert diese, im folgenden genauer erklärten, Schritte.

Als Beispiel wollen wir den Adobe Acrobat Reader von hanoi aus starten. Zuerst müssen wir der hanoi erlauben, Anwendungen an uns zu exportieren. Das Kommando `xhost` (1) nimmt uns diese Aufgabe dankend ab (1). Als Argument erwartet `xhost +hostname | -hostname`. Damit kann entweder der Zugriff für einen Rechner erlaubt (+) oder verboten (-) werden. Standardmässig ist der Zugriff für alle Rechner erst einmal verboten.

Nachdem dieser Teil erledigt ist, wechseln wir mit der `ssh` auf den entfernten Rechner Als nächster Schritt müssen wir dem X-Server von hanoi nun mitteilen, auf welchem System die Ausgabe erscheinen soll. Dies geschieht über die Umgebungsvariable `$DISPLAY`. Ihr Wert setzt sich aus Rechnername und Monitornummer zusammen. Letzere ist im foo-Pool (bisher) immer 0, da wir keine Systeme mit mehr als einem Monitor besitzen. Hostname und Monitornummer werden durch einen Doppelpunkt getrennt. Die Umgebungsvariable wird, wie gewohnt, mit `setenv` gesetzt. Danach lässt sich jede graphische Anwendung über das Netzwerk nutzen. Dieser Vorgang wird auch als „Umlenken des Displays“ bezeichnet (2).

In (3) zeigt UNIX uns wieder die Prozeß-ID des neuen Prozesses an, mit Hilfe des `&` wird der Prozeß im Hintergrund gestartet, damit die Shell nicht blockiert wird. Dieser Prozeß lässt sich selbstverständlich wieder ein wenig vereinfachen, indem man die `rsh` nutzt. An dieser Stelle wird es zum ersten mal notwendig, mehrere Kommandos in Folge mit Hilfe des Semikolons abzusetzen. Der Befehl, der dem `rsh` Aufruf übergeben wird, muß ja sowohl das Display


```
atlantis:/ce_users/nehmer> xhost +hanoi
hanoi being added to access control list
atlantis:/ce_users/nehmer> rsh hanoi ' setenv DISPLAY atlantis:0 ;
/usr/local/acrobat3/bin/acroread & ' &
[1] 7054
```

Abbildung 20: Umleiten einer graphischen Anwendung mit rsh

umgelenken, als auch die eigentliche Anwendung starten. Selbstverständlich muß auch in dieser Variante mit `xhost` der entfernte Rechner freigeschaltet werden, wie in Abbildung 20 zu sehen ist.

Auf dieser Basis arbeiten viele Anwendungs-Startskripte¹⁴ im `foo-Pool`. Sie starten, je nach Anwendung, den Prozeß auf einem bestimmten Rechner. Auf diese Weise wird eine bessere Ressourcenauslastung im `foo-Pool` erreicht: Eine zweite Instanz desselben Programmes braucht auf Grund geteilter Ressourcen wesentlich weniger Speicher als die erste.

2.11 Datenaustausch zwischen (Win)DOS und UNIX

In diesem Kapitel soll eher auf die konzeptionellen Unterschiede der beiden Welten eingegangen werden. Der Datenaustausch via Diskette wurde im letzten Kapitel bereits angesprochen. Als häufig benutzte Alternative kommt dann noch der Dateitransfer via Internet, FTP (File Transfer Protocol), dazu. FTP wird im Kapitel 3.4 besprochen. Aber auch für FTP gelten die hier genannten Probleme.

2.11.1 Umlaute

Das grösste Problem vieler Unix-Systeme sind die fehlenden Deutschen Sonderzeichen. Besonders in Dateinamen wird dies häufig zum Problem. Ebenfalls können viele Programme der traditionellen Unix-Welt Umlaute oft nur schwer verarbeiten. Man sollte also generell versuchen, Deutsche Sonderzeichen zu vermeiden.

2.11.2 ASCII-Dateiformat

Ein ebenfalls sehr lästiges Problem sind die verschiedenen Textformate von DOS und UNIX. Während DOS Zeilenenden mit den Zeichen CR/LF (Carriage Return, Line Feed) kennzeichnet, verwendet UNIX nur LF. Auf dem Weg von DOS nach UNIX ist das noch nicht so dramatisch. Der Editor zeigt an jedem Zeilenende `^M` an, das dem überflüssigen CR entspricht, ansonsten sieht die Datei aber gleich aus. Der umgekehrte Weg sorgt da für etwas mehr Ärger. Beispielsweise kann Window's Notepad Zeilenumbrüche nur anhand einer CR/LF Kombination erkennen. UNIX' LF erscheint als Sonderzeichen und man sieht die komplette Datei in einer einzigen Zeile. (Mein Tip: Die Schnellansicht kann auch UNIX-ASCII Dateien anzeigen)

Um diesem Problem entgegenzuwirken, besitzen FTP-Programme die Möglichkeit zur automatischen Umwandlung CR - CR/LF. FTP stellt hierfür eigene Kommandos bereit. Wichtig ist aber, daß Binärdateien (zum Beispiel Bilder) nicht mit dieser Option kopiert werden, da sie durch die Zeichenumwandlung in der Regel unbrauchbar werden. Wie FTP-Programme entsprechend konfiguriert werden, wird im Kapitel 3.4 beschrieben.

¹⁴Die Startskripte befinden sich im Verzeichnis `/usr/local/share/bin`. Eine Warnung aber gleich vorweg: sie sind bereits recht komplex.

2.11.3 Die mtools

Oft werdet Ihr Dateien via Diskette zwischen dem Foo-Pool und dem heimischen Rechner austauschen. Auch hier macht sich das CR/LF-Problem bemerkbar. Linux stellt hierfür die `mtools(3)` zur Verfügung. Dies sind einige Hilfsprogramme, die den Zugriff auf Disketten mit MS-DOS Format ermöglichen und wie Ihre MS-DOS Geschwister arbeiten: `mcopy(1)`, `mdir(1)` und `mdel(1)`

Ein `mdir a:` zeigt beispielsweise den Inhalt des Diskettenlaufwerkes in gewohnter DOS-Manier an. Einige Fallstricke verbergen sich jedoch bei den Programmen `mcopy(1)` (Dateien kopieren) und `mdel(1)`. (Dateien löschen) Beide akzeptieren Jokerzeichen nach den gleichen Regeln wie die DOS-Befehle `copy` und `del`.

Leider sorgt aber ein `mcopy a:*. *` grundsätzlich für die Meldung `No such file or directory`. Woran liegt das? Jokerzeichen wie `*` und `?` werden unter UNIX im Gegensatz zu DOS nicht von dem aufgerufenen Programm ausgewertet, sondern bereits von der Shell ersetzt. In unserem Beispiel setzt die Shell also für beide `*` jeweils alle Dateinamen aus dem aktuellen Verzeichnis ein – was natürlich keine sinnvolle Anweisung für `mcopy(1)` übriglässt. Abhilfe schafft auch hier wieder das Maskieren der betreffenden Zeichen. Einfachster Weg ist es, das komplette Argument in Anführungszeichen einzufassen (`mcopy "a:*. *" .`). Diese Regel gilt grundsätzlich für alle `mtools(3)`.

Leider wird die CR-CR/LF Umwandlung von `mcopy(1)` nicht automatisch durchgeführt. Dies muß explizit durch den Parameter `-t` aktiviert werden (z.B. `mcopy -t "a:*.txt" .`).

2.12 foo-Kommandoübersicht

Hier eine kurze Übersicht über die grundlegenden UNIX-Kommandos. Einige davon habe ich in diesem Kapitel bereits erwähnt, andere, wie `find(1)` sind hier der Vollständigkeit halber genannt, da sie im täglichen Betrieb sehr nützliche Werkzeuge darstellen. Die einzelnen Zeilen wurden mit Hilfe von `man -f` erstellt.

<code>cd(1)</code>	Change working directory
<code>chgrp(1)</code>	change group ownership
<code>chmod(1)</code>	change file access permissions
<code>cp(1)</code>	copy files and directories
<code>df(1)</code>	report filesystem disk space usage
<code>du(1)</code>	estimate file space usage
<code>echo(1)</code>	display a line of text
<code>find(1)</code>	search for files in a directory hierarchy
<code>grep(1)</code>	print lines matching a pattern
<code>hostname(1)</code>	show or set the system's host name
<code>id(1)</code>	print real and effective UIDs and GIDs
<code>ln(1)</code>	make links between files
<code>lpq(1)</code> , HP: <code>lpstat(1)</code>	spool queue examination program
<code>lpr(1)</code>	off line print
<code>lprm(1)</code> , HP: <code>cancel(1)</code>	remove jobs from the line printer spooling queue
<code>ls(1)</code>	list directory contents
<code>man(1)</code>	format and display the online manual pages
<code>mcopy(1)</code>	copy MSDOS files to/from Unix
<code>mdel(1)</code>	delete an MSDOS file
<code>mdir(1)</code>	display an MSDOS directory
<code>mkdir(1)</code>	make directories
<code>more(1)</code>	file perusal filter for crt viewing
<code>mv(1)</code>	move (rename) files
<code>printenv(1)</code>	print all or part of environment
<code>ps(1)</code>	report process status
<code>pwd(1)</code>	print name of current/working directory
<code>rlogin(1)</code>	remote login
<code>rm(1)</code>	remove files or directories
<code>rmdir(1)</code>	remove empty directories
<code>rsh(1)</code>	remote shell
<code>setenv(1)</code>	change or add an environment variable
<code>ssh(1)</code>	secure shell client (remote login program)
<code>tcsh(1)</code>	C shell with file name completion and command line editing
<code>telnet(1)</code>	user interface to the TELNET protocol
<code>top(1)</code>	display top CPU processes
<code>uname(1)</code>	print system information
<code>yppasswd(1)</code>	change your password in the NIS database

3 Das Internet

Während des Studiums werdet ihr sehr oft mit dem Internet in Berührung kommen. Als weltumspannendes Computernetzwerk bietet es schier endlos erscheinende Möglichkeiten. Ihr könnt euch Informationen für Reparate beschaffen, Diskussionen auf elektronischen, schwarzen Brettern führen, elektronische Post innerhalb von Sekunden um die halbe Welt schicken, und ... und ... und ...

Im folgenden sollen die wichtigsten Dienste einer nach dem anderen eingeführt werden. Dabei wird zunächst alles mit Hilfe des Netscape Navigators gemacht, da dieses Programm die wichtigsten Internetdienste unterstützt. Ein solches Programm wird Web-Browser genannt. Der Netscape Navigator kann auf zwei Arten gestartet werden: Direkt in einer Shell mit dem Befehl `netscape` oder über das Anwendungsmenü unter „Internet“. Nach dem Start wartet Netscape mit der Startseite auf Deine Anfragen. Abbildung 21 zeigt die Homepage des Studiengangs Communication Engineering. In den folgenden Kapiteln werden wir eine angepasste Grundkonfiguration für Deinen Netscape Navigator erstellen.

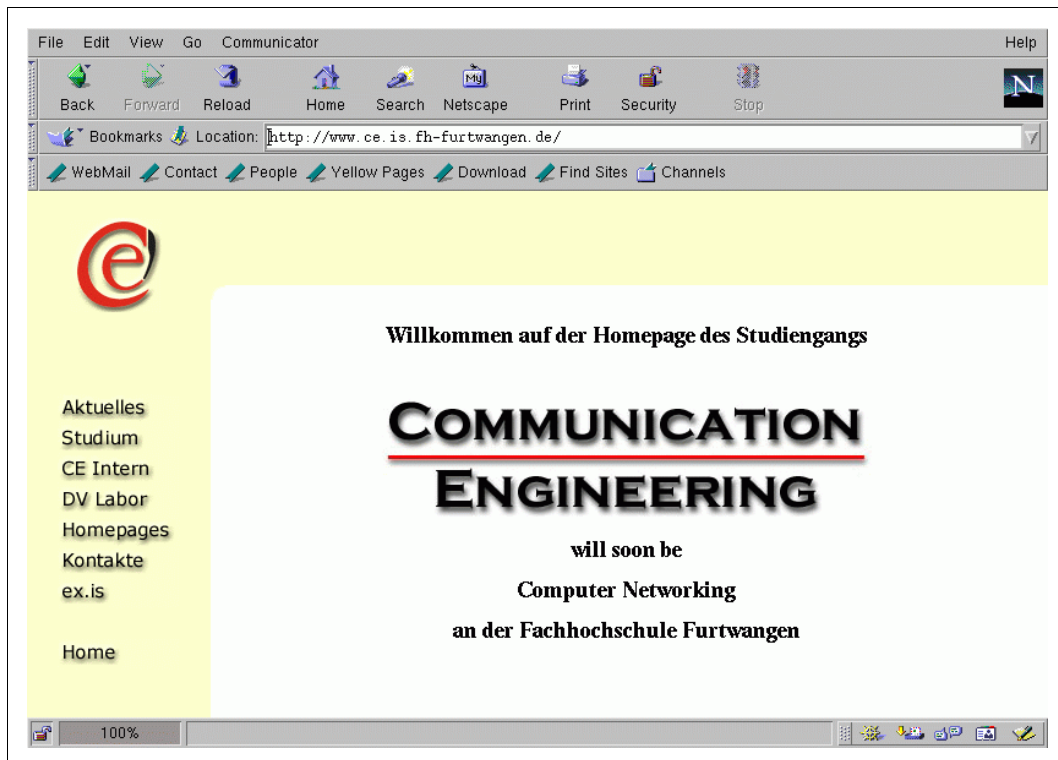


Abbildung 21: Der Netscape Navigator

3.1 WWW: World Wide Web

Das WWW kann man sich am ehesten als eine riesige Bibliothek vorstellen, in der man wirklich alles finden kann. Das WWW besteht aus einer Ansammlung von Seiten, die in der Beschreibungssprache HTML (Hypertext Markup Language)¹⁵ abgelegt sind. Wie der Name

¹⁵HTML ist eine rein textorientierte Beschreibung einer Seite, die sich mehr an Konzepten, als an exakten Angaben orientiert. Anstatt „Ich hätte gerne eine 16 Punkt Serifenschrift in fett, das ganze zentriert und in Blau.“ sagt man lediglich „Ich hätte gerne eine Überschrift zweiter Ebene.“ Dadurch erreicht man auf Kosten einer genauen Seitenformatierung eine sehr plattformunabhängige Beschreibung eines Dokumentes.

Kürzel	Standort
.de	Deutschland
.com	meistens USA - Kommerzielles Netz (commercial)
.org	meistens USA - Nichtkommerzielle Organisation (organisation)
.edu	meistens USA - Bildungseinrichtung (educational)
.gov	meistens USA - Regierungseinrichtung (governmental)
.net	meistens USA - Diverses (network)
.uk	Großbritannien
.au	Australien
.at	Österreich (Austria)
.tw	Taiwan

Abbildung 22: Übersicht über die gebräuchlichsten Länderkürzel in WWW-Adressen

schon suggeriert, ist HTML ein Hypertextmedium, daß auf ähnlichen Prinzipien basiert, wie GNU's Info-Dokumente (siehe Kapitel 2.6.2). Bestimmte Textstellen können mit Verknüpfungen auf andere HTML-Dokumente versehen werden. Solche Verknüpfungen werden Hyperlinks genannt. Dadurch wird eine sehr einfache Navigationsmöglichkeit geschaffen, die sehr große Datenmenge übersichtlich präsentieren kann. Hyperlinks können an mehreren Merkmalen erkannt werden: Sie sind in der Regel farblich hervorgehoben (oftmals blau) und meistens unterstrichen. Dazu ändert Netscape den Mauszeiger zu einer Hand, wenn der Mauszeiger über einem Hyperlink steht. Ein einfacher Mausklick genügt dann, um den Inhalt des verknüpften Dokuments anzuzeigen. Tip: Auch Bilder (oder Teile davon) können als Hyperlink verwendet werden.

Diese Bibliothek hat aber ein großes Problem: Es existiert derzeit nicht ein einziger, vollständiger Index über die im WWW vorhandenen Informationen. Sucht man nun Informationen über ein bestimmtes Thema, so bemüht man in der Regel eine der vielen Suchmaschinen im Internet. Im Februar 1999 umfasste das Web nach einer Studie von NEC Research etwa 800 Millionen Seiten. Die beste Suchmaschine „fast Search“ hingegen hat aber gerade mal 200 Millionen Seiten indiziert. Man benötigt also ein wenig Geduld, um über die Seiten im Suchergebnis, weitere themenverwandte Seiten zu finden.

Seiten im Internet werden immer anhand ihrer Internetadressen identifiziert. Dies sind Namen, die aus mehreren Teilen bestehen. Die verschiedenen Segmente sind durch Punkte getrennt. Da eine ausführliche Erklärung über das Zustandekommen dieser Namen den Rahmen der Foo-Bibel sprengen würde, will ich es bei einer kurzen Übersicht belassen: Die Adresse *www.fh-furtwangen.de* lässt sich in zwei grobe Teile zerlegen: *www.fh-furtwangen* und *.de*. Der letzte Teil gibt immer den Standort des Web-Servers an. *.de* bedeutet dabei Deutschland. Abbildung 22 zeigt eine kleine Übersicht mit den wichtigsten Codes¹⁶. Der Rest gibt den Namen der Seite selbst an. Normalerweise kennzeichnet ein vorangestelltes *www* einen WWW-Server, dies ist aber nicht Pflicht. Vor eine Adresse wird deshalb noch der Dienstyp vorangestellt. Bei HTML-Seiten über das WWW ist dies *http://* und steht für „Hypertext Transfer Protocol“. Eine vollständige Internet-Adresse besteht also aus Dienstyp und Ziel. Das obige Beispiel müsste genau genommen also *http://www.fh-furtwangen.de* heißen. Glücklicherweise wird von Web-Browsern *http://* als Standarddienst vorausgesetzt, wodurch dieses Präfix auch weggelassen werden kann. Vollständige Adressen dieser Art werden auch als URLs (Uniform Resource Locator) bezeichnet. Soll jetzt eine bestimmte Internetseite angezeigt werden, so muß lediglich ihre Adresse in das Eingabefeld „Location:“ eingegeben und mit Return bestätigt werden.

¹⁶Zunächst scheint es ein wenig zu verwirren, daß die USA keinen eigenen Ländercode hat. Das liegt daran, daß das Internet in den USA entstanden ist, und die anderen Länder erst später dazu kamen, gab es zunächst nur die Aufteilung in *.com*, *.org*, *.edu* usw. Server in den USA haben aus diesem Grund noch immer diese Codes. Jedoch haben inzwischen auch einige nicht-Amerikanische Firmen Adressen mit diesen Kürzeln, was eine Zuordnung ein wenig erschwert.

Im folgenden wird Netscapes Konfiguration und Benutzung beschrieben. Dabei werden alle Konfigurationsdaten in `~/ .netscape` abgespeichert. Editiert wird die Konfiguration in einem recht komfortablen Dialog. Er verbirgt sich hinter dem Menüpunkt *Edit* → *Preferences*. Die Bedienung dieses Dialogs gestaltet sich recht einfach. Links befindet sich ein Menü, in dem die verschiedenen Optionsgruppen zusammengefasst sind. Ein Klick auf die kleinen Pfeile öffnet oder schliesst eine solche Gruppe, während ein Klick auf einen Namen das entsprechende Dialogfeld einblendet.

3.1.1 Grundkonfiguration: Proxies, Netscape-Cache, Smart Browsing

Bevor wir hier weitermachen will ich zunächst einmal den Begriff „Proxy“ klären. Ein Proxy ist ein Rechner, der WWW-Anfragen zwischenspeichert um so die Netzlast zu reduzieren. Rufen zwei Nutzer nacheinander die selbe Seite ab, wird sie im zweiten Aufruf nicht mehr vom eigentlichen Server heruntergeladen, sondern wird vom Proxy ausgeliefert, was zu einem erheblichen Geschwindigkeitsvorteil führt. Der Proxy prüft natürlich in regelmässigen Abständen, ob sich die gespeicherten Seiten geändert haben. Man nennt dieses Prinzip „caching“.

Wir rufen also zunächst einmal Netscape's Optionsdialog auf und wählen dort unter „Advanced“ den Eintrag „Proxies“. Dort muß das Optionskästchen „Automatic Proxy Configuration“ aktiviert werden und im Feld „Configuration Location (URL)“ folgende Adresse eingetragen werden: `http://www.foo.fh-furtwangen.de/proxy.pac` Das ganze ist in Abbildung 23 zu sehen.

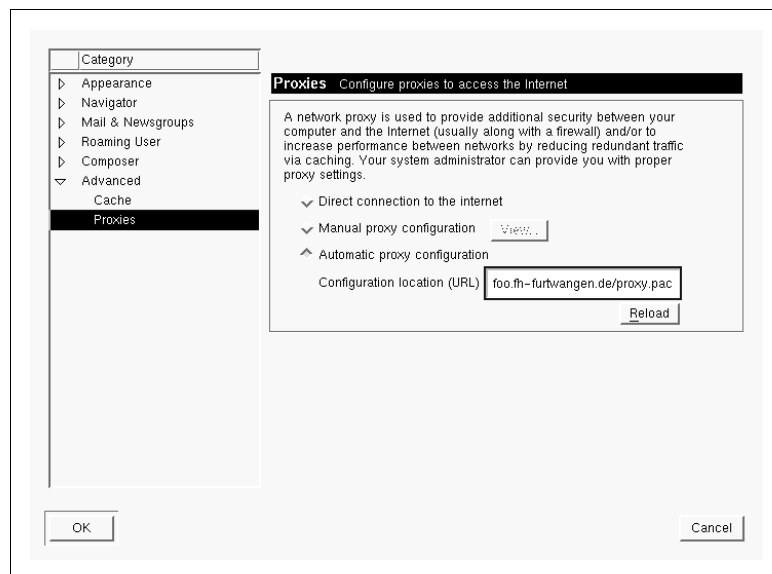


Abbildung 23: Netscape: Proxy-Konfiguration

Als Ergänzung zu den Proxies stellt Netscape einen internen Cache zur Verfügung. Hier im Foo-Pool macht die Benutzung dieses Caches aber wenig Sinn, denn Netscape speichert die Seiten in Deinem Homeverzeichnis - und diese Daten kommen über das LAN, genauso wie die Seiten aus dem Proxy. Deshalb ist es sinnvoll, den Netscape-Cache zu deaktivieren. Dies spart jede Menge Plattenplatz.

Als nächstes bemühen wir also, ebenfalls unter „Advanced“, den Punkt „Cache“ und tragen unter „Disk Cache“ den Wert 0 ein, wie man das in Abbildung 24 sieht.

Smart Browsing wiederum ist eine neue Errungenschaft von Netscape: Die Adresse jeder Sei-

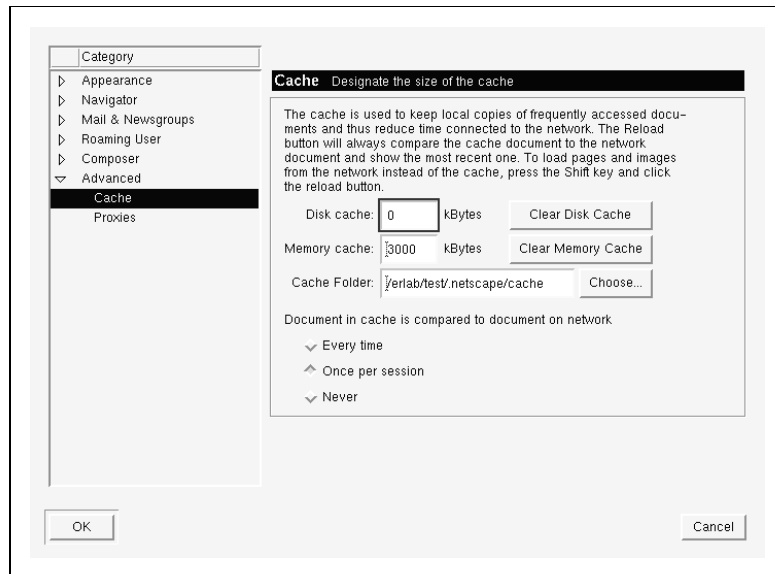


Abbildung 24: Netscape: Cache-Konfiguration

te, die Du besuchst, wird an Netscape übermittelt. Dort sucht ein Server in einer Datenbank nach verwandten Seiten. Beispiel: Du bist auf den BMW-Seiten, und bekommst alternativ die Adressen von Mercedes, Audi und ein paar anderen angezeigt. Das ganze hat nur ein paar Haken: Es funktioniert nur bei wirklich bekannten Seiten, die man sowieso ohne viel Aufwand findet. Bei den Bereichen, in denen es tatsächlich interessant wäre, bleibt die Suche oft erfolglos. Auch muß man sich darüber im Klaren sein, das jede Seite, die man besucht, im Klartext durch das halbe Internet übermittelt wird. Es ist somit ein nicht zu unterschätzender Eingriff in die Privatsphäre.

Wer also dieses Feature lieber nicht nutzen will, kann es unter „Navigator, Smart Browsing“ deaktivieren, indem er „Enable What’s Related“ wegklickt. Auch dafür habe ich natürlich einen wunderschönen Screenshot: Abbildung 25.

3.1.2 Bookmarks

Im Laufe der Zeit findest Du mit Sicherheit immer mehr Seiten im WWW, die Du regelmässig besuchst. Da es recht mühsam wäre, sich die ganzen Adressen von Hand irgendwo zu notieren, bietet Netscape die Möglichkeit, WWW-Adressen in einer internen Liste, den Bookmarks¹⁷, zu speichern.

Kontrolliert wird diese Liste über das Menü „Bookmarks“ links neben dem Eingabefeld „Location“. Klickt man darauf, erscheint das in Abbildung 26 gezeigte Menü. In diesem Fall handelt es sich um die Standard-Bookmarks des Netscape Navigators 4.5. In diesem Menü gibt es drei Arten von Einträgen: Die ersten drei Punkte (Add, File und Edit Bookmarks) sind für die Verwaltung der Bookmarks zuständig. Alle anderen Punkte repräsentieren entweder direkt ein Bookmark (Beispiel „Netscape Netcenter“) oder ein Untermenü, genannt Folder, mit weiteren Bookmarks. In beiden Fällen genügt ein Klick auf den entsprechenden Eintrag um die WWW-Seite beziehungsweise den Folder anzuzeigen.

„Edit Bookmarks“ erlaubt es, die Liste der Bookmarks durchzusortieren, neue Folder anzulegen, Bookmarks zu löschen und gewisse Standardwerte zu setzen. Die Bookmarks können

¹⁷Genauer gesagt speichert Netscape die Bookmarks in Form einer regulären HTML-Datei im Verzeichnis `~/ .netscape/bookmarks.html` ab.

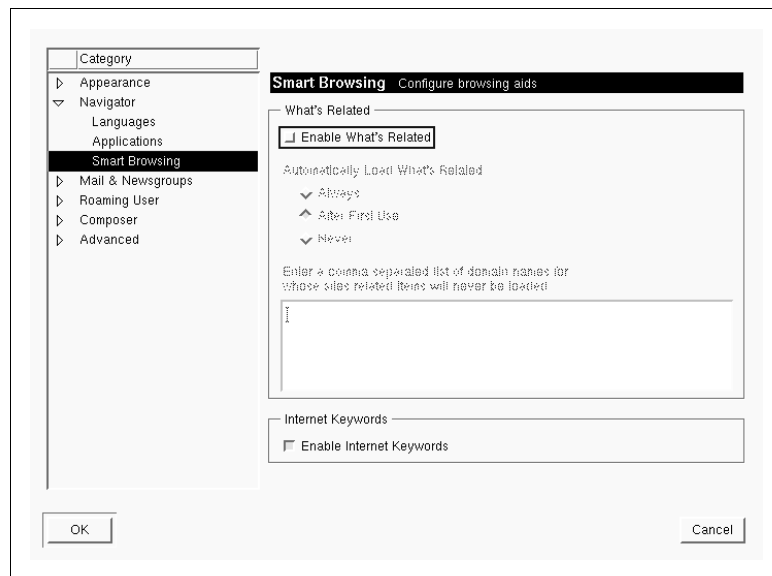


Abbildung 25: Netscape: Smart Browsing Konfiguration

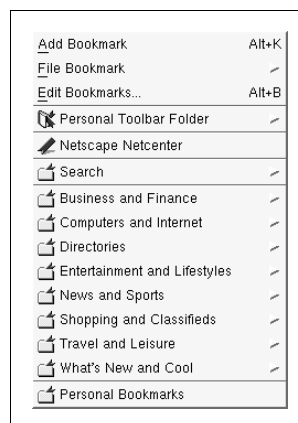


Abbildung 26: Netscape: Bookmarks

mit Hilfe von Drag&Drop sortiert werden. Alle weiteren Funktionen sind über die Kontextmenüs¹⁸ der verschiedenen Einträge der Bookmark-Liste zu erreichen:

New Bookmark Ein neues Lesezeichen kann von Hand eingegeben werden. Dies wird in der Regel nicht benötigt, da Lesezeichen direkt auf der Basis der gerade besuchten Seite erstellt werden können.

New Folder Ein neuer Folder wird erstellt.

New Separator Ein Separator ist eine Trennungslinie, die nur die Funktion hat, die Bookmarks ein wenig zu untergliedern. In Abbildung 26 ist zum Beispiel zwischen „Search“ und „Business and Finance“ ein Separator eingefügt worden.

Open link in New Window Hier wird der aktuell angewählte Bookmark in einem neuen Navigator-Fenster geöffnet.

¹⁸Ein Kontextmenü ist ein spezielles Menü, das grundsätzlich mit Hilfe eines Rechtsklicks geöffnet werden kann. Dabei verändert sich der Inhalt abhängig von dem angeklickten Programmelement. Aktionen, die aus einem Kontextmenü heraus ausgeführt werden, beziehen sich immer auf das angeklickte Objekt.

Set as Toolbar Folder Unterhalb des Eingabefeldes „Location“ befindet sich eine Toolbar mit mehreren Schaltflächen. Auch hier handelt es sich um Bookmarks: In dieser Leiste werden all diejenigen Bookmarks angezeigt, die sich in dem mit „Set as Toolbar Folder“ ausgewählten Folder befinden. Auf diese Art lassen sich sehr häufig besuchte Bookmarks schnell verfügbar machen.

Set as Folder for New Bookmarks Wird ein Bookmark ohne Angabe eines Zieles hinzugefügt (siehe unten), dann landet es in diesem Folder.

Properties Properties ruft einen Dialog mit allen Eigenschaften, wie Zieladresse oder Name, des angewählten Bookmarks oder Folders auf.

Hast Du nun eine Seite gefunden, die Du in Deine Bookmarks aufnehmen willst, hast Du zwei Möglichkeiten: Der schnellste Weg ist ein Klick auf „Add Bookmark“, der einen neuen Eintrag in Deinem New Bookmark Folder erstellt. Willst Du den Bookmark hingegen an einer ganz bestimmten Stelle in Deinen Bookmarks ablegen, wählst Du den Menüpunkt „File Bookmark“ aus. Dieser zeigt Dir daraufhin eine Liste mit allen Folders in Deinen Bookmarks an, in der Du den gewünschten Ziel-Folder auswählst.

3.1.3 Interessante URLs

Die folgende Tabelle bietet ein paar interessante Einstiegsstellen ins WWW.

FH-Interne Homepages

http://www.foo.fh-furtwangen.de	Homepage des Fachbereichs Informationssysteme
http://www.ce.is.fh-furtwangen.de	Homepage des Studiengangs Communication Engineering
http://www.fh-furtwangen.de	Homepage der FH Furtwangen
http://www.ghb.fh-furtwangen.de/intern/netiquette.html	Ein kleiner Auszug aus der Netiquette: Die Etikette für das Internet (betrifft vor allem E-Mail und News)

Suchmaschinen

http://www.alltheweb.com	FastSearch bietet den derzeit umfangreichsten Index
http://www.yahoo.com	Yahoo! hat ein sehr gutes Branchenverzeichnis
http://www.altavista.com	Altavista beherrscht eine sehr flexible Anfragesyntax

3.2 E-Mail

Die Infrastruktur des Internets bietet vielfältige Möglichkeiten zur Kommunikation. Die wohl am stärksten genutzte ist Electronic Mail, kurz E-Mail. Eine E-Mail ist im Grunde nichts weiteres als eine reine ASCII-Datei. In dieser können mit Hilfe verschiedener Verfahren auch binäre Daten wie Bilder untergebracht werden. Ergänzt werden die Daten innerhalb der E-Mail durch einige spezielle Zeilen, genannt Header, in denen wichtige Daten wie Empfänger, Sender, Betreff und so weiter gespeichert werden.

Eine E-Mail-Adresse wird immer nach dem Schema *user@host* aufgebaut. *user* ist dabei der Login-Name des Benutzers, der die E-Mail erhalten soll, und *host* der Rechner, auf dem der Benutzeraccount liegt. Bei Netzen in der Art des Foo-Pools, kann das Mailsystem so konfiguriert werden, daß auch ohne Angabe eines bestimmten Rechners die E-Mails immer zum Mail-

Server gelangen. Man muß also nicht *nehmer@father.foo.fh-furtwangen.de* schreiben (obwohl das auch funktionieren würde), sondern lediglich *nehmer@foo.fh-furtwangen.de*.

Hierraus ergibt sich, das (in der UNIX-Welt) mit einem Benutzer-Account auch immer ein Mail-Account verknüpft ist. Username und Paßwort für beide Accounts sind gleich.

Eine komplette Einführung in das Thema E-Mail würde auch hier den Rahmen sprengen, daher konzentriere ich mich hier auf Netscapes E-Mail Schnittstelle. Bevor diese allerdings richtig genutzt werden kann, muß Netscape mit Deinen Benutzerdaten konfiguriert werden.

3.2.1 Konfiguration des Netscape E-Mail Client

Wir öffnen zunächst einmal Netscape's Konfigurations-Dialog (Edit → Preferences) und öffnen die Gruppe „Mail & Newsgroups“. Der erste Schritt ist es, den eigenen Namen und die eigene E-Mail-Adresse unter „Identity“ einzutragen. Für beide Felder existiert bereits eine Vorgabe, die Netscape aus Deinen Account-Informationen erhält. Abbildung 27 zeigt diesen Dialog.

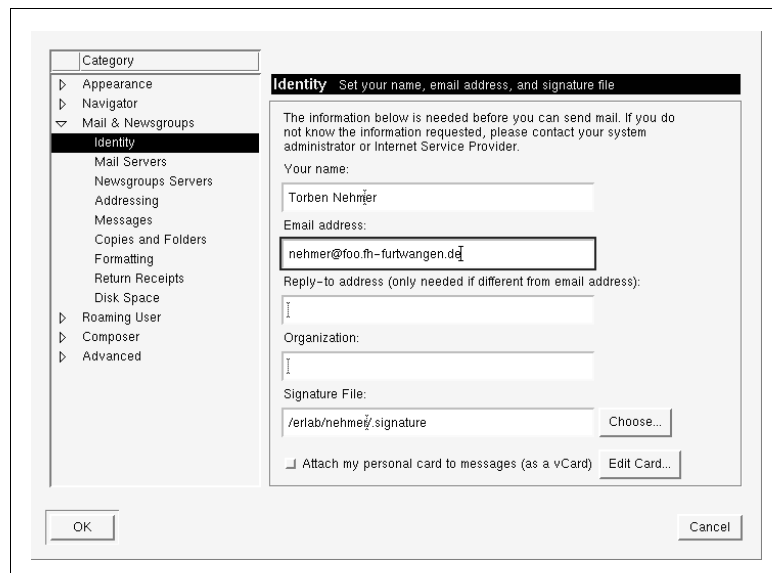


Abbildung 27: Netscape: Konfiguration eigener Name und E-Mail-Adresse

Dein Name landet im Eingabefeld „Your Name:“. Er dient dazu, Deinen Realnamen zur E-Mail-Adresse angeben zu können. Dies kommt daher, daß E-Mail-Adressen unter anderem aus dem Benutzernamen aufgebaut werden. Und dieser ist bei uns auf acht Zeichen limitiert. Dadurch können, gerade bei häufig vorkommenden Nachnamen wie „Mueller“, ein wenig verwirrende Benutzernamen wie muelerjo oder muellju entstehen. Um dem ein wenig entgegenzuwirken, kann Dein Realname mit angegeben werden. Wichtig ist aber, daß trotzdem vom Empfänger einer E-Mail die Adresse bekannt sein muß.

Deine E-Mail-Adresse in „Email Address:“ solltest Du um ein *@foo.fh-furtwangen.de* ergänzen, da Netscape nur Deinen Loginnamen hier einsetzt. Ausserhalb des Foo-Pools würde diese Adresse nicht funktionieren.

Als nächstes öffnen wir die Seite „Mail Servers“. Die dortigen Angaben benötigt Netscape, um überhaupt auf Deine E-Mails zugreifen zu können. Als allererstes müssen wir den Netscape vorgegebenen „Incoming Mail Server“ *pop* löschen. Dazu klicken wir ihn in der Liste an und wählen „Delete“. Nun legen wir mit „Add“ einen neuen Server an. In dem nun folgenden Eingabedialog gibst Du als „Server Name“ *mailhost.foo.fh-furtwangen.de* und als „User Name“

Deinen normalen User-Namen an. Über „Remember password“ legst Du fest, ob das Paßwort zu Deinem Mail-Account von Netscape gespeichert werden soll oder nicht. Das Paßwort wird verschlüsselt gesichert, was das Sicherheitsrisiko aber nicht aus der Welt schafft. Netscape's Verschlüsselung ist nämlich nicht allzu effektiv. Sind beide Angaben gemacht (das Paßwort wird später abgefragt), kann dieser Dialog wieder mit OK geschlossen werden. Bei „Outgoing mail (SMTP) server:“ muß dann noch einmal ein *mailhost.foo.fh-furtwangen.de* eingetragen werden. Abbildung 28 zeigt diesen Dialog.

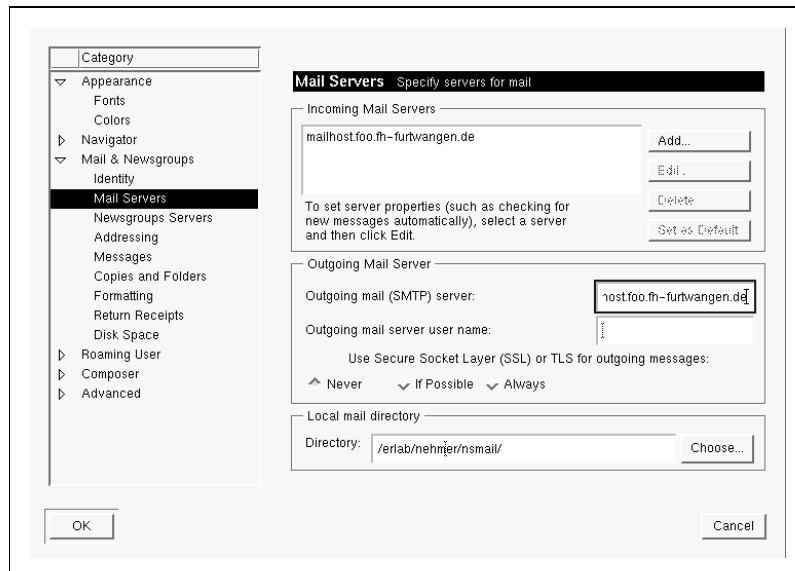


Abbildung 28: Netscape: Konfiguration der Mailserver

Ein letzter Punkt betrifft die Netiquette. Standardmässig sendet Netscape mit HTML formatierte E-Mails. Dies sollte man aber vermeiden, da viele Leute nicht unbedingt Netscape zum lesen Ihrer E-Mail verwenden. Oft sind dies dann Programme, die im Textmodus an Konsolen arbeiten und entsprechend keine HTML-Formatanweisungen vertragen. In solchen Programmen tauchen dann die HTML-Codes als ganz normaler E-Mail Text auf, was natürlich sehr lästig ist. Abbildung 29 zeigt den Dialog „Formatting“ der regelt, wie ausgehende E-Mails gehandhabt werden. Die Einstellung „Use the plain text editor to compose messages“ sorgt dafür, daß man die HTML-Formate während des editierens schon gar nicht erst zur Verfügung bekommt. Jedoch muß dazu noch „Convert the message into plain text“ aktiviert werden, damit Netscape auch garantiert nur ASCII-Nachrichten verschickt.

Nachdem alle diese Einstellungen aktiviert sind, kannst Du den Dialog mit OK schliessen.

3.2.2 E-Mails versenden und empfangen

Aufgerufen wird Netscapes E-Mail Schnittstelle entweder mit dem Menüpunkt „Communicator → Messenger“ oder mit einem Klick auf das zweite Symbol (Briefkasten) in Netscapes Launcher (siehe Abbildung 30). Daraufhin erscheint ein neues Fenster mit Deinen verschiedenen Mailboxen. Eine Mailbox ist eine Art Ordner, in dem E-Mails abgelegt werden. Der wichtigste ist hierbei die Inbox. In ihr landen alle neuen E-Mails. Das neue Fenster ist in drei Bereiche unterteilt. Links befindet sich eine Liste mit Mail-Foldern, darunter beispielsweise auch die Inbox. Rechts oben befindet sich eine Übersicht, in der alle Mails angezeigt werden, die in der aktuell gewählten Mailbox liegen. Hier werden natürlich keine Inhalte, sondern lediglich Absender, Betreff und ähnliches angezeigt. In dieser Liste können nun Mails ausgewählt werden, die im darunterliegenden Fenster angezeigt werden.

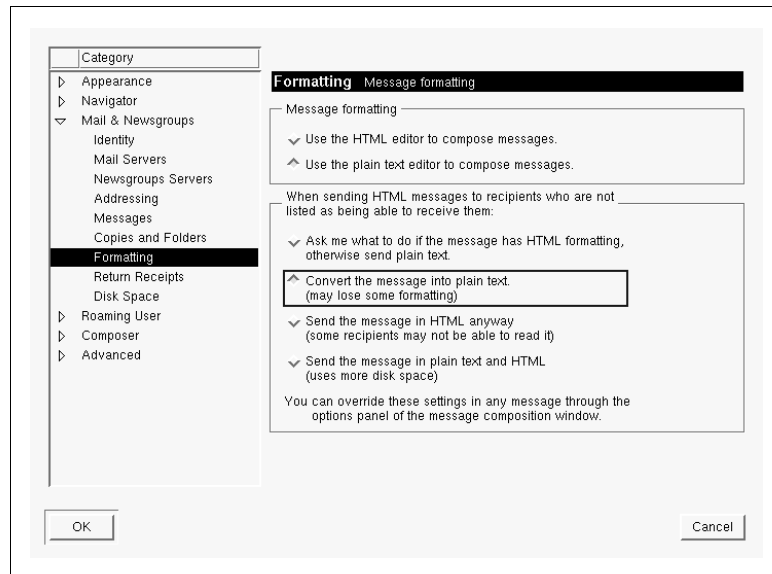


Abbildung 29: Netscape: Konfiguration der Mail-Formatierung (HTML/Plain Text)



Abbildung 30: Netscape's Launcher

Wichtig sind grundsätzlich erst einmal zwei Schaltflächen in der Toolbar: „Get Msg“ ganz links und direkt daneben „New Msg“. Erstere fragt den Mailserver ab, ob neue Nachrichten vorhanden sind. Klickst Du darauf, fragt Netscape Dich nach dem Paßwort für den Zugriff auf den Mailserver. Dies ist Dein normales Login-Paßwort. Hast Du es korrekt eingegeben, beginnt Netscape, neue Mails vom Mailserver runterzuladen. Es kann natürlich gut sein, daß auf Deinem nagelneuen E-Mail Account noch keine Mails angekommen sind. Um das ganze einmal auszuprobieren ist es das einfachste, sich selbst eine E-Mail zu schicken:

Nach einem Klick auf „New Message“ erhältst Du ein neues Fenster, in dem Du die E-Mail verfassen kannst. Erster Schritt ist es, hinter „To:“ Deine eigene E-Mail Adresse als Empfänger einzugeben, in meinem Fall wäre dies *nehmer@foo.fh-furtwangen.de*. Unter „Subject“ mußt Du nun eine kurze Betreffzeile angeben, anhand derer der Empfänger schnell sehen kann, was du von ihm willst. Wir belassen es hier bei einem einfachen „Test“. Jetzt kommen wir erst zum eigentlichen Teil der E-Mail, dem Body. Dies ist der große Textbereich unter der Subject-Zeile. Dort kannst Du nun nach Herzenslust loslegen. Abbildung 31 zeigt dieses Eingabeformular noch einmal komplett ausgefüllt. Ist man mit der E-Mail zufrieden, kann sie mit Hilfe des Buttons „Send“ ganz links in der Toolbar abgeschickt werden.

Diese E-Mail müsste innerhalb von Sekunden wieder in Deinem Mail-Account landen. Klick einfach noch einmal auf „Get Msg“ und Netscape lädt die neue Mail runter und sortiert sie in die (vermutlich noch leere) Inbox ein. Normalerweise zeigt Netscape ungelesene Mails im Fettdruck an. Lässt man sich die Mail anzeigen, setzt Netscape sie automatisch auf gelesen (= normale Schrift). Wenn dies die erste Mail ist, die in Deiner Inbox landet, wird sie gleich angezeigt und folglich auch nicht im Fettdruck hervorgehoben. Wenn Du also ausser unserer Test-Mail weitere Mails in Deiner Mailbox haben solltest, dann klicke die Test-Mail in der Liste im oberen, rechten Fensters des E-Mail Interface' an.

Mailer bieten einige spezielle Funktionen zum Beantworten von empfangenen E-Mails, die

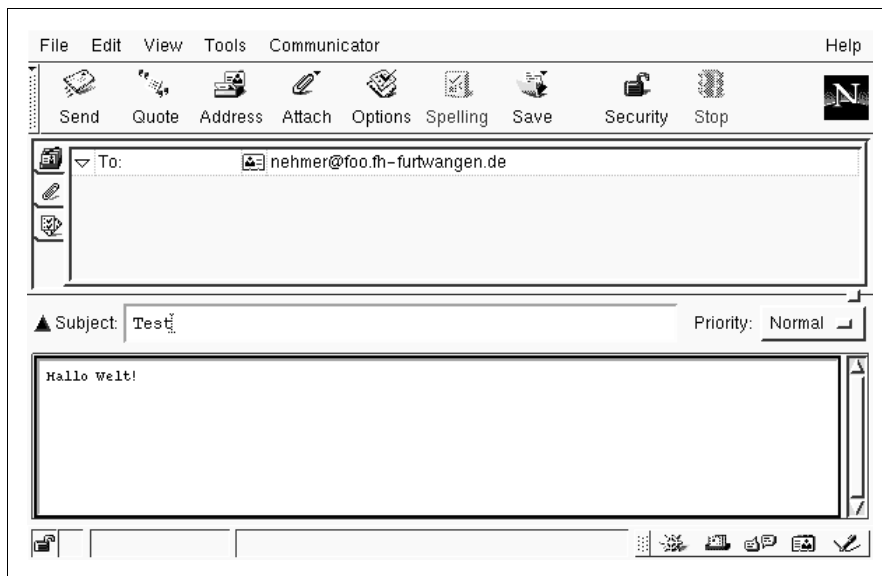


Abbildung 31: Netscape: Eine erste Test-E-Mail

Du über die Funktion „Reply“ in der Toolbar nutzen kannst. Netscape öffnet ein neues Fenster mit der neuen E-Mail. Dabei füllt er aber, im Gegensatz zu vorher, einige Felder mit Standardwerten aus der empfangenen E-Mail aus. Dies sind Empfänger-Adresse, die unverändert übernommen wird, Subject und Body, die ein wenig modifiziert werden. Dem Subject wird ein „Re:“ vorangestellt, was für Reply (also Antwort) steht. Im Body findest Du den Text der empfangenen E-Mail, wobei jede Zeile ein „>“ vorangestellt bekommt. Dadurch kannst Du Dich sehr leicht auf Teile der empfangenen Mail beziehen. Dazu fügt er am Anfang noch die Zeile „Torben Nehmer wrote:“ ein. Dies geschieht, weil eine E-Mail auch mehrere Empfänger (siehe Kapitel 3.2.3) haben kann und man so weiß, von wem welcher Teil der E-Mail stammt. Eine Antwort auf unsere Test-Mail könnte wie in Abbildung 32 aussehen. Wie eben kann die Mail nun mit „Send“ verschickt werden.

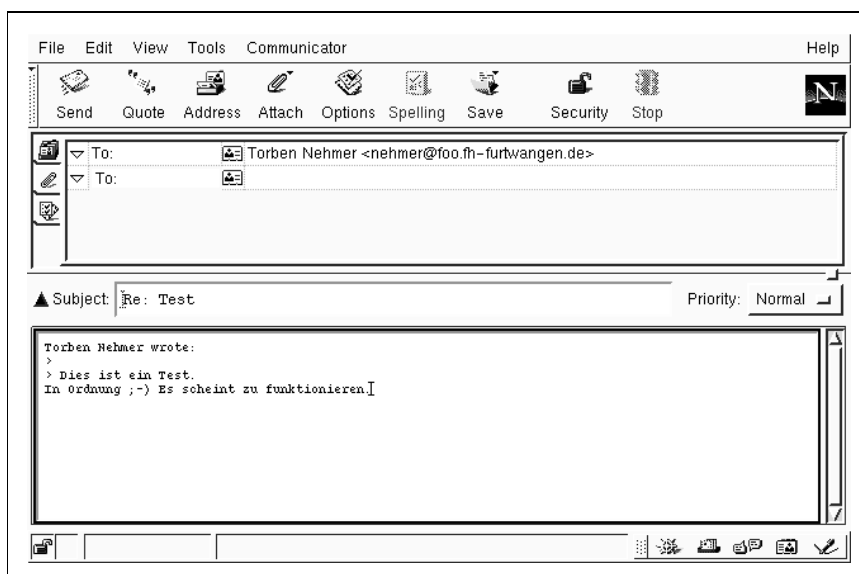


Abbildung 32: Netscape: Die Antwort auf die Test-E-Mail

Willst Du lediglich eine E-Mail an einen anderen weiterleiten, beispielsweise weil Du selbst nicht zuständig bist, dann kannst Du die gleichnamige Funktion nutzen: „Forward“. Du bekommst wiederrum ein neues E-Mail Fenster, daß, wie unter Reply, verschiedene Vorgaben hat: Das Subject mit vorangestelltem „Fwd:“ und die ursprüngliche E-Mail als Attachment (= Anhang, siehe Kapitel 3.2.4). Letzteres sorgt dafür, daß die E-Mail exakt so weitergeleitet wird, wie sie empfangen wurde.

3.2.3 Mehrere Empfänger

Es gibt mehrere Möglichkeiten, eine Email an mehrere Empfänger zur versenden. Netscape's Eingabe-Routine löst dies über dynamische Adress-Eingabefelder. Vor dem „To:“ ist ein kleiner Pfeil nach unten, mit dem Du den Typ des Empfängers auswählen kannst. Hast Du in der ersten Zeile eine Adresse eingegeben, kannst Du mit Return eine neue Adresszeile öffnen. „Follow-Up“ und „Newsgroup“ werden im Kapitel 3.3 erklärt, ansonsten sind folgende Empfänger-Typen für E-Mails relevant:

To: Dies sind die ganz normalen E-Mail Empfänger und damit der einfachste Weg, eine E-Mail an mehrere Stellen zu schicken.

Cc: Cc steht für „Carbon-Copy“, also Durchschlag oder Kopie. Hiermit lassen sich Kopien einer Mail an andere Empfänger schicken. Beispielsweise „zur Kenntnisname“ oder ähnliches. Die eigentlichen Empfänger können diese Einträge sehen.

Bcc: Bcc bedeutet „Blind Carbon-Copy“ und funktioniert mit einer Ausnahme analog zu Cc: Die Empfänger bekommen von den Kopien, die an Bcc-Empfänger gehen nichts mit. Dies kann beispielsweise dazu verwendet werden, eine Kopie aller vom Foo-Pool aus versendete E-Mails an einen anderen, eigenen E-Mail Accounts weiterzuleiten. Damit man auch dann den Überblick behält, wenn man mehrere, unabhängige E-Mail Accounts besitzt. Da es umständlich ist, ein solches Bcc jedes mal von Hand einzugeben, kann in den Netscape-Optionen unter „Mail & Newsgroups → Copies and Folders → (Mail) Bcc other adresses“ eine Adresse eingegeben werden, an die immer ein Bcc geschickt werden soll. Ansonsten sollte man berücksichtigen, das Bcc's nicht gerade die feine, Englische Art sind. Man sollte sie vorsichtig verwenden.

Reply-To: Normalerweise antwortet ein E-Mail Client auf die in der E-Mail eingetragene Empfänger-Adresse. Unter Umständen ist dies nicht erwünscht, beispielsweise wenn man mehrere E-Mail Accounts besitzt. In einem solchen Falle kann hier eine alternative E-Mail Adresse angegeben werden, auf die geantwortet werden soll. Auch dieses Feld kann mit einem Standardwert versehen werden: „Mail & Newsgroups → Identity → Reply-to adress“.

Ein Beispiel einer Mail mit jedem der vier Empfänger-Typen findest Du in Abbildung 33. Obwohl im Beispiel pro Empfänger-Typ jeweils nur eine Adresse angegeben wurde, können natürlich auch mehrere Adreß-Zeilen mit dem gleichen Empfänger-Typ erstellt werden. Soll auf eine solche Mail geantwortet werden, ist es wichtig, nicht „Reply“, sondern „Reply All“ zu wählen, damit auch alle Empfänger eine Antwort erhalten. Ansonsten würde nur derjenige, der die Mail weggeschickt hat, eine Antwort bekommen.

3.2.4 Attachments (Anhänge)

E-Mails müssen nicht ausschliesslich aus Text-Nachrichten bestehen. An eine E-Mail kann auch eine Datei angehängt werden. Diese Anhängsel werden Attachments genannt. Netscape bietet auch hier eine einfache Schnittstelle an. Rechts neben dem Eingabe-Bereich für die E-

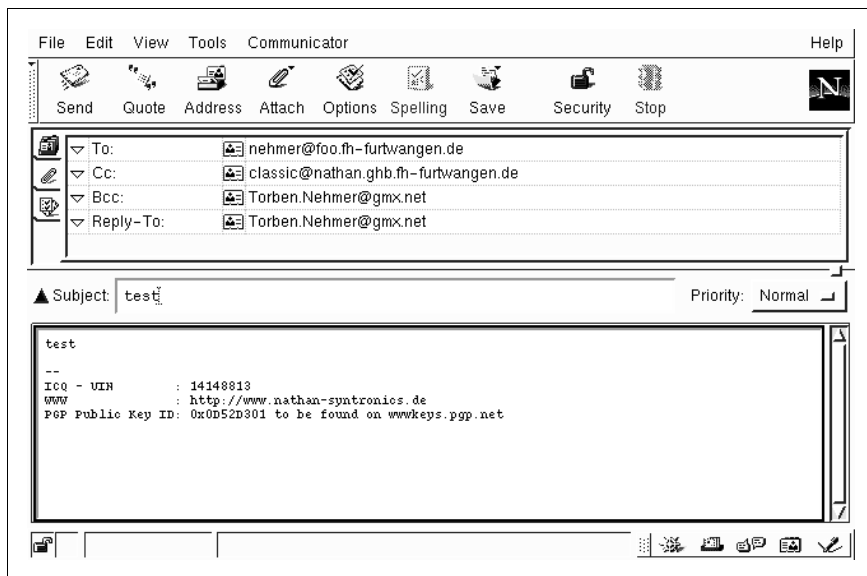


Abbildung 33: Netscape: Eine E-Mail mit mehreren Empfängern

Mail Adressen befinden sich drei Schaltflächen. Die Erste (Karteikasten) wird automatisch ausgewählt und repräsentiert die angegebenen Empfänger. Die zweite Schaltfläche (Büroklammer) blendet die Attachements ein. In einer neuen Mail (Ausnahme: Forward) ist dieser Bereich immer leer. Klickt man mit der rechten Maustaste in diesen Bereich, kann man mit „Attach File...“ eine Datei in die Mail einfügen. Dabei sollte man beachten, daß E-Mail Accounts in der Regel eine Größenbeschränkung besitzen. Ein guter Richtwert ist es, Mails kleiner als ein Megabyte zu halten. Gerade bei Attachments ist diese Regel aber mit Vorsicht zu genießen: Attachements werden codiert, da in E-Mails grundsätzlich nur der sieben Bit breite ASCII Satz erlaubt ist, Binärdaten aber auf acht Bit breiten Daten betehen. Durch diese Codierungsverfahren wachsen Attachements auf ungefähr 150% ihrer ursprünglichen Größe an.

3.2.5 Adreß-Suche mit finger

Mit Hilfe des Befehles `finger(1)` kannst Du die E-Mail-Accountes eines Netzes durchsuchen. Als Suchbegriffe kannst Du nicht nur die Usernamen, sondern auch die Realnamen der vorhandenen Benutzer verwenden. Es eignet sich also hervorragend dazu, die Email-Adresse anhand eines Namens herauszufinden. Selbst Vornamen lassen sich verwenden.

```
atlantis:/ce_users/nehmer/> finger torben
Login: nehmer                               Name: Torben Neh-
mer CE1 SS98 krm
Directory: /ce_users/nehmer                 Shell: /bin/tcsh
On since Thu Aug 19 09:05 (CEST) on tty1    38 minutes 39 se-
conds idle
Mail forwarded to Torben.Nehmer@gmx.net
No mail.
No Plan.
```

In dieser Liste findest Du unter „Login“ den Usernamen, der ja gleichzeitig auch die E-Mail Adresse bildet: `user@foo.fh-furtwangen.de`. Sollten mehrere Einträge zutreffen, werden natürlich auch alle angezeigt.

3.2.6 Wichtige E-Mail-Adressen

Selbstverständlich haben alle wichtigen Ansprechpartner, sowohl im Foo-Pool, als auch in der FH im Allgemeinen, E-Mail-Adressen:

sysadmin@foo.fh-furtwangen.de Alle Anfragen, die den foo-Pool betreffen, können an diese Adresse gerichtet werden. Vor allem Probleme technischer Natur sollten hier gemeldet werden.

fachschaft@foo.fh-furtwangen.de Die Fachschaft ist für alle studentischen Probleme verantwortlich. Vor allem bei Problemen mit Professoren soll sie als erste Vermittlungsstelle dienen. Genaugenommen vertritt die Fachschaft die Studenten im Fachbereich.

hanne@foo.fh-furtwangen.de Frau Professor Frank ist Studiengangsleiterin CE. Sie ist für den Studiengang zuständig und beantwortet gerne alle aufkommenden Fragen. Wichtig ist, daß sich der Studiengangsleiter jederzeit ändern kann. Jeweils aktuelle Informationen finden sich auf der CE-Homepage.

mueller@foo.fh-furtwangen.de Herr Professor Dr. Müller ist momentan Studiengangsleiter IK und ist dort für die studiumsbezogenen Fragen verantwortlich. Auch dieser Name kann sich jederzeit ändern, weshalb man sich vorher kurz auf der IK-Homepage anschauen sollte.

karger@foo.fh-furtwangen.de Professor Karger leitet den Studiengang ER. Wie gehabt ist auch diese Belegung nicht für die Ewigkeit. Aktuelle Informationen stehen natürlich auf der ER-Homepage.

sauerburger@fh-furtwangen.de Herr Professor Dr. Sauerburger ist der Dekan des Fachbereiches IS und für alle Fragen über den Fachbereich selbst, seine Struktur, den Haushalt und das Personal zuständig.

Es existiert noch eine weitere Möglichkeit, eine E-Mail an einen Professor zu senden: Jeder Professor hat ein Kurzzeichen. Eine Liste mit allen Kurzzeichen findet sich im Briefkasten-Raum im B-Bau. Aus diesem Kurzzeichen ergeben sich dann die E-Mail Adressen nach dem Muster `kurzzeichen@fh-furtwangen.de`.

3.3 Internet News (Usenet) - Die Diskussionsforen

Internet News, in der Regel einfach nur News genannt, kann man sich am ehesten als elektronisches Pendant zu „Schwarzen Brettern“ vorstellen. Überall auf der Welt verteilt befinden sich News-Server, die Nachrichten von Nutzern annehmen und an andere News-Server weiterleiten. Dadurch entsteht ein weltumspannendes Diskussions-System, daß in sogenannte Newsgroups eingeteilt ist. Eine Newsgroup ist ein Diskussionforum zu einem ganz bestimmten Thema. Insgesamt existieren zehntausende von Newsgroups, die aber nicht zwingend von jedem News-Server verbreitet werden müssen. Der News-Server der FH beispielsweise stellt momentan etwa 6.000 Newsgroups zur Verfügung.

Damit die Orientierung in einem solchen Informations-Overkill halbwegs möglich bleibt, sind Newsgroups hierarchisch aufgeteilt: Ein Name einer Newsgroup setzt sich aus mehreren, durch Punkte getrennten, Wörtern zusammen. Als Beispiel soll uns hier die Newsgroup `de.comp.os.unix` dienen. `de.` steht für Newsgroups in Deutscher Sprache, `comp.` für den allgemeinen Bereich Computer. Dieser führt weiter in das Sachgebiet Betriebssysteme `os.` (Operating Systems), im Speziellen `unix`. Auf diese Weise sind alle Newsgroups strukturiert. Auch die FH hat ihren eigenen Oberbegriff: Alle FH-Internen Gruppen beginnen mit `fhf.` und sind ausserhalb des FH-Netzes nicht abrufbar, denn sie werden nicht nach aussen weitergeleitet.

Netscape bietet auch eine News-Schnittstelle, was das ganze ein wenig erleichtert. Besonders bei News solltest Du Dir immer bewußt sein, daß die von Dir verfassten Artikel auf der ganzen Welt gelesen werden können. Dies ist die eigentliche Stärke der Internet News. Nirgendwo sind die Chancen grösser, innerhalb von kurzer Zeit Antworten auf aktuell anstehende Probleme zu bekommen. Gerade diese Verbreitung bringt aber einen wichtigen Punkt mit sich: Du solltest Dir ganz genau überlegen, was Du in welche Newsgroup setzt. Vor allem hier ist es sehr wichtig, die Netiquette (siehe Kapitel 3.1.3) einzuhalten. Zu schnell kann man sich im Usenet unbeliebt machen.

3.3.1 Konfiguration des Netscape News-Clients

Natürlich muß Netscape auch hier erst einmal konfiguriert werden. Diesmal benötigen wir „Mail & Newgroups → Newsgroups Servers“. Dort muß zunächst einmal mit „Add...“ der News-Sever `news.ai-lab.fh-furtwangen.de` hinzugefügt und mit „Set as Default“ als Standard festgelegt werden. Danach kann Netscape's Standardserver mit „Delete“ entfernt werden. Abbildung 34 zeigt den Endzustand dieses Dialogs, der nun geschlossen werden kann.

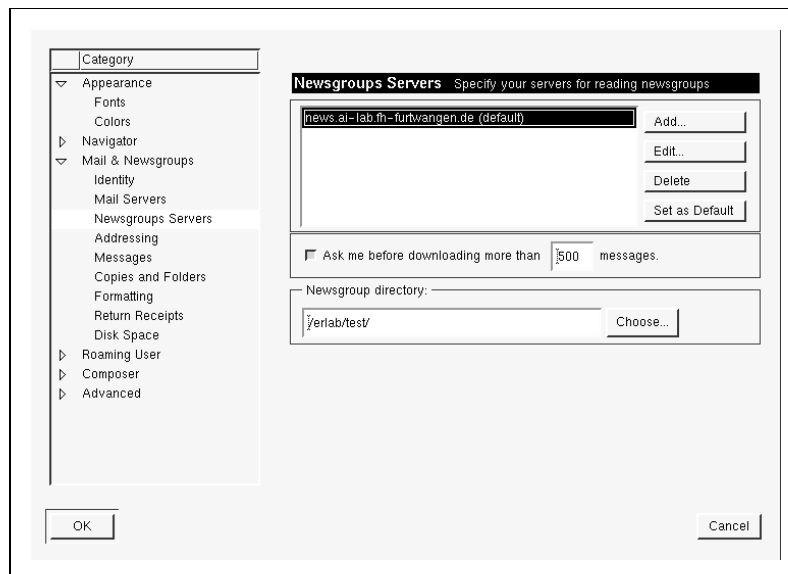


Abbildung 34: Netscape: Konfiguration der News-Komponente

Der nächste Schritt besteht darin, sich ein paar Newsgroups zum mitlesen auszusuchen. Diesen Vorgang nennt man abonnieren (engl. subscribe). Du öffnest zuerst einmal die E-Mail Schnittstelle, von der aus Du auch Newsgroups lesen kannst. Danach klickst Du mit der rechten Maustaste auf „news.ai-lab.fh-furtwangen.de“, dem letzten Eintrag in der Mailboxliste. Im Kontextmenü benötigen wir dann den Punkt „Subscribe to Newsgroups...“. Daraufhin wird eine Verbindung zum News-Server hergestellt und eine Liste aller aktiven Newsgroups heruntergeladen.

Nun erscheint die Hierarchie des News-Servers und man kann sich die Newsgroups aussuchen, die man lesen will. Am Ende dieses Kapitels findest Du ein paar, meiner Meinung nach, interessante Newsgroups. Für unseren Testbetrieb wollen wir uns auf die Newsgroup `fhf.test` beschränken. Wir öffnen also zunächst einmal den Baum `fhf.*`, in dem wir dann `fhf.test` finden. Hinter jeder abonnierbaren Newsgroup befindet sich ein kleiner Punkt. Klickt man auf diesen, verwandelt er sich in ein Häkchen und die Newsgroup ist abonniert. Natürlich funktioniert dies auch rückwärts: Das Häkchen wegklicken und die Newsgroup wird

wieder abbestellt. Abbildung 35 zeigt diesen Dialog mit abonniertes fhf.test.

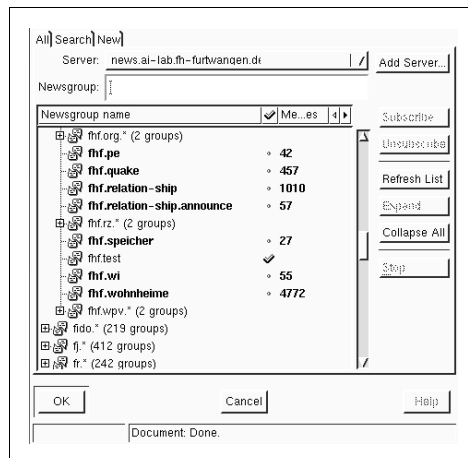


Abbildung 35: Netscape: Newsgruppen abonnieren

Damit wäre die Konfiguration des News-Clients abgeschlossen, und der Subscribe-Dialog kann mit „OK“ geschlossen werden.

3.3.2 News lesen/schreiben

Von hier an bedient sich Netscape's News-Client in etwa wie Netscape's E-Mail-Client. Du kannst jetzt unter den Mailboxen den Bereich „news.ai-lab.fh-furtwangen.de“ öffnen. (Wenn die Spalte zu schmal ist, kürzt Netscape diesen Namen nach dem Muster „news.ai-la...angen.de“ ab.) Klickst Du die Newsgroup „fhf.test“ an, dürfte sich ein Bild ähnlich wie in Abbildung 36 ergeben. Natürlich werdet Ihr mit großer Wahrscheinlichkeit andere, neuere Nachrichten in dieser Newsgroup finden. Gelesen wird jetzt wie im E-Mail Bereich durch einfaches anklicken der Mail in der Übersicht oben rechts.

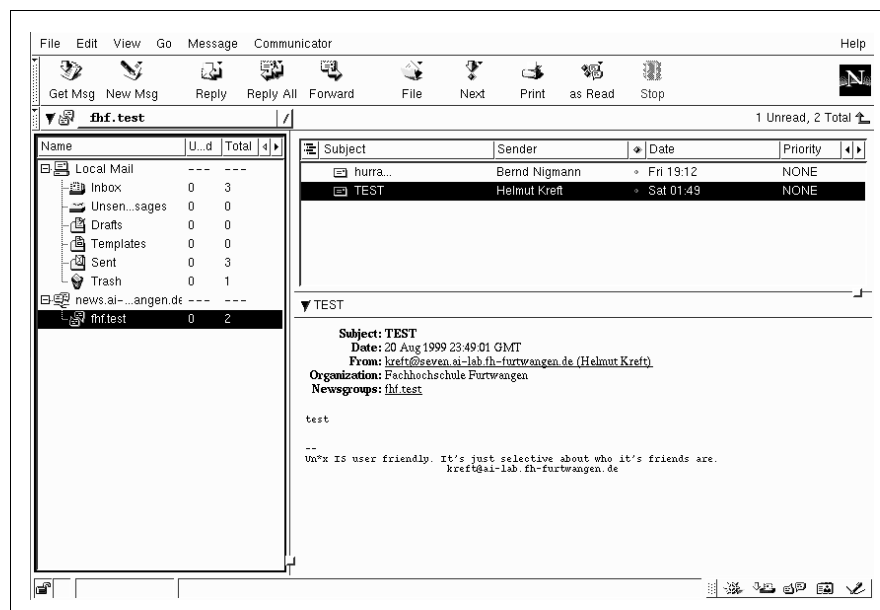


Abbildung 36: Netscape: News lesen

Beim schreiben oder beantworten einer Nachricht kommen hier jetzt die Empfängertypen „Newsgroup“ und „Followup-To“ ins Spiel:

Unter „Newsgroup“ stehen die Newsgroups, in denen das Posting erscheinen soll. Sei jedoch mit Crosspostings (versenden einer Nachricht in mehrere Newsgroups) vorsichtig. Meistens gibt es genau eine Newsgroup, in die Deine Nachricht hingehört, auch wenn fünf andere ungefähr zu passen scheinen. Solche ähnlichen Gruppen sollten erst als letzte Notlösung angesprochen werden. Crosspostings erzeugen zudem unnötige Netzlast und machen Dich damit unbeliebt.

Ein „Followup-To“ wird benötigt, wenn sich während einer Diskussion das Thema auf ein anderes Sachgebiet (und damit eine neue Newsgroup) verlagert. Beispielsweise könnte ursprünglich eine Frage zur Hardwarekompatibilität gestellt worden sein und das Thema ist auf Betriebssysteme abgeglitten. Dann kannst Du folgendes Prinzip verwenden: Du postest eine Antwort, am besten mit neuem Betreff (z.B. „Betriebssystemvergleich (war: Hardwarekompatibilität)“) in beide betroffenen Newsgroups und setzt dann aber das Followup-To auf die neue Newsgroup. Dadurch landen alle Antworten auf Deine Nachricht in dieser Newsgroup. Followup-To entspricht also im weiteren Sinne dem Feld Reply-To in den E-Mails. Wichtig ist, daß nicht automatisch das Posting mit dem Followup-To schon in der neuen Newsgroup landet. Dies muß von Hand mit einer zweiten Ziel-Newsgroup gemacht werden. Ausserdem sollte man am Ende des Postings kurz erwähnen, das ein Followup_to gesetzt ist. Für gewöhnlich da reicht ein `f'up to <newsgroup>`.

Neue Nachrichten werden wie üblich über die Schaltfläche „New Msg“ verfasst, die den bereits bekannten Dialog zum Nachrichtenschreiben zeigt. Es wird automatisch die Newsgroup, in der man sich gerade befindet, als Ziel angegeben. Weitere Empfänger (auch E-Mail!) können dann nach Wahl angegeben werden. Ebenso funktionieren auch die Schaltflächen „Reply“ (Antwort in aktuelle Newsgroup), „Reply All“ (Antwort an alle angegeben Newsgroups und E-Mails Empfänger) und „Forward“ (Forward des Postings via E-Mail). Wer das jetzt mal ausprobieren will, kann sich in der `fhf.test` austoben. Denn in den verschiedenen `.test`-Gruppen kann man gefahrlos die verschiedenen Mechanismen ausprobieren.

3.3.3 Intessante Newsgroups

Die folgende Übersicht enthält einige Newsgroups, die meiner bescheidenen Meinung nach wichtig oder interessant sind.

FH-Interne Newsgroups

<code>fhf.is</code>	Newsgroup des Fachbereichs Informationssysteme
<code>fhf.is.ce</code>	Newsgroup des Studiengangs CE
<code>fhf.markt</code>	Das Pendant zu den Schwarzen Brettern
<code>fhf.lists.*</code>	Verschiedene Mailinglisten als Newsgroup, darunter die Linux-Kernel-Mailingliste und die NT-Bugtraq Liste
<code>fhf.org.unfug</code>	Unix Freak and User Group

Externe Newsgroups

<code>de.comp.os.unix.*</code>	Viele Newsgroups rund um Unix
<code>de.comp.lang.*</code>	Verschiedene Programmiersprachen
<code>comp.*</code>	Die internationalen Pendants zu <code>de.comp.*</code> . Hier wird nur Englisch gesprochen.

3.4 FTP - File Transfer Protocol

Mit Hilfe des File Transfer Protocols lassen sich Dateien zwischen zwei Rechnern übertragen. Dabei wird zwischen FTP mit und ohne Authentifizierung unterschieden. Ersterer, auch

Anonymous FTP genannt, dient hauptsächlich für den anonymen Zugriff auf öffentliche Fileserver, zum Beispiel den des Gnu-Projektes. Im Gegensatz zu Anonymous FTP erfordert der reguläre FTP einen gültigen Account auf dem Zielsystem. Somit können über das Internet Daten zwischen Deinem Foo-Account und dem heimischen Rechner ausgetauscht werden.

FTP ist ein Tool, daß am besten von der Kommandozeile aus verwendet wird. Netscape bietet zwar auch eine Schnittstelle für FTP, jedoch ist diese recht umständlich, wenn man mehr machen will, als Dateien von Anonymous FTP Servern herunterzuladen.

Egal ob anonym oder normal, nach dem Login bietet FTP eine shellartige Umgebung, mit deren Hilfe der Dateitransfer stattfindet. Dabei werden zwei Übertragungsarten unterschieden: Binär und ASCII. Binär bedeutet eine Eins-zu-Eins-Übertragung der Daten, während ASCII eine Umwandlung zwischen CR/LF und LF durchführt. Dies resultiert wieder aus den Unterschieden zwischen DOS und Unix, die in Kapitel 2.11 bereits beschrieben wurden. Normalerweise versucht FTP, den Transfertyp selbstständig korrekt einzustellen. Dies wird mit Hilfe der Dateiendung festgestellt. Da diese Unterscheidung nicht immer ganz eindeutig ist, sollte man immer ein Auge auf den Übertragungstyp haben, und ihn im Zweifelsfalle selbst einstellen. Mir sind Probleme dieser Art speziell im Zusammenhang mit den Paketen des Red-Hat Package Manager (* .rpm) bekannt, die von verschiedenen Clients als ASCII übertragen werden und dadurch nach der Übertragung nicht mehr lauffähig sind.

FTP-Zugriffe laufen dabei immer von Deinem lokalen FTP-Client zu dem von Dir angegeben FTP-Server. Für unsere Beispiele dient der FTP-Server des Foo-Pools, der sowohl anonyme, als auch authentifizierte Logins erlaubt. Der anonyme Login führt dabei momentan auf Kopien der FTP-Server von GNU und NetBSD, während der normale Login Dich direkt in Dein Homeverzeichnis bringt. Dieser Server ist unter `ftp.foo.fhfurtwangen.de` zu erreichen.

3.4.1 FTP in der Shell

Das Programm `ftp(1)` dient unter allen Unix-Plattformen als textorientierter FTP-Client. Gestartet wird er mit `ftp <ftp-server-name>`. Daraufhin fordert er ein interaktives Login. Hier entscheidet sich nun, ob Du anonymous FTP willst, oder nicht. Gibst Du als Name `anonymous` und als Paßwort Deine E-Mail Adresse¹⁹ ein, dann arbeitest Du anonym, also als Gast-Nutzer auf dem FTP-Server, gibst Du Deine normalen Accountdaten ein, landest Du mit Deinen normalen User-Rechten in Deinem Homeverzeichnis.

Abbildung 37 zeigt einen Anonymous Login auf `ftp.foo.fh-furtwangen.de`, während Abbildung 38 den normalen Login beschreibt. Du kannst hier auch leicht die Verwandtschaft zu einer normal Unix-Shell erkennen. `ls` zeigt hier das aktuelle Verzeichnis in gewohnter Form an. Das aktuelle Verzeichnis wird ganz normal mit `cd` gewechselt. Alle Up- und Downloads beziehen sich ohne weitere Aktion auf das Verzeichnis, in dem der FTP-Client gestartet wurde. Dieses Verzeichnis wird vom FTP-Client als „Local Directory“ bezeichnet und kann mit Hilfe des Befehls `lcd` beeinflusst werden. Dieser Befehl arbeitet wie `cd`: Ohne Parameter gibt er das aktuelle, lokale Verzeichnis aus, mit Parameter wechselt er in das angegebene Verzeichnis.

Für den Dateitransfer selbst gibt es sieben wichtige Befehle. Dateinamen, die Sonderzeichen, beispielsweise Leerzeichen, enthalten und als Parameter übergeben werden sollen, müssen in Anführungszeichen eingeschlossen werden.

auto: Mit `auto` überlässt Du dem FTP-Client die Wahl des Datentyps (Binär oder ASCII). Dies ist die Standardeinstellung und arbeitet in den meisten Fällen auch einwandfrei. Wie Eingangsbereich bereits erwähnt, kann es unter Umständen jedoch vorkommen, daß dieser Da-

¹⁹Es ist so üblich, als Paßwort für Anonymous FTP eine E-Mail Adresse zu fordern, damit der Administrator Dich im Falle eines Falles erreichen kann. Natürlich besitzt der FTP-Server keinerlei Möglichkeit, die Gültigkeit einer E-Mail Adresse zu überprüfen.

```
atlantis:/ce.users/nehmer> ftp ftp.foo.fh-furtwangen.de
Connected to hanne.foo.fh-furtwangen.de.
220 hanne.foo.fh-furtwangen.de FTP server ready.
Name (ftp.foo.fh-furtwangen.de:nehmer): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 4
drwxr-xr-x 2 0 0 512 Aug 4 09:28 bin
drwxr-xr-x 4 0 0 512 Jul 9 13:56 disk4
drwxr-xr-x 3 0 0 512 Jul 9 13:42 disk5
drwxr-xr-x 2 0 0 512 Aug 9 11:51 mirror
226 Transfer complete.
ftp>
```

Abbildung 37: Anonymous FTP Login

```
atlantis:/ce.users/nehmer> ftp ftp.foo.fh-furtwangen.de
Connected to hanne.foo.fh-furtwangen.de.
220 hanne.foo.fh-furtwangen.de FTP server ready.
Name (ftp.foo.fh-furtwangen.de:nehmer): nehmer
331 Password required for nehmer.
Password:
230 User nehmer logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 4
drwxr-xr-x 2 nehmer users 512 Aug 23 13:30 tmp
...
drwxr-xr-x 5 nehmer users 512 Aug 17 13:20 www
-rw-r----- 1 nehmer users 174 Mar 25 15:46 zitate
226 Transfer complete.
ftp>
```

Abbildung 38: Normaler FTP Login

tentyp nicht korrekt erkannt wird. In einem solchen Falle können die beiden folgenden Befehle verwendet werden, um den Datentyp von Hand festzulegen.

binary: Binary erzwingt eine binäre Datenübertragung. Eine CR/LF-LF Umwandlung wird nicht durchgeführt.

ascii Ascii hingegen erzwingt eine textuelle Datenübertragung, so daß die CR/LF-LF Konvertierung immer durchgeführt wird.

get: Mit Hilfe von get können Daten vom FTP-Server heruntergeladen werden. Als Parameter erwartet get den Dateinamen der zu übertragenden Datei. Jokerzeichen wie * oder ? werden nicht unterstützt. Optional kann als zweiter Parameter ein neuer Dateiname angegeben werden, unter dem die Datei gespeichert werden soll.

mget: Mget ist eine Erweiterung zu get, die mehrere Dateien auf einmal übertragen kann und somit Jokerzeichen unterstützt. Standardmässig wird vor jeder Datei, die auf das Suchmuster zutrifft, noch einmal nachgefragt, ob sie wirklich heruntergeladen werden soll.

put: Put ist das Gegenstück zu get: Eine Datei kann auf den Server hochgeladen werden. Als Parameter wird wiederum der lokale Name und, optional, der Zielname der Datei erwartet. Auch hier werden keine Jokerzeichen unterstützt.

mput: Wie mget erweitert mput put um Joker-Funktionen. Vor jeder Datei wird auch hier nachgefragt, ob sie tatsächlich gesendet werden soll.

Es existiert parallel zu den FTP-Befehlen die Möglichkeit, Befehle an eine lokale Shell abzusetzen, beispielsweise um lokal ein Verzeichnis zu erstellen oder ein Verzeichnis anzuzeigen. Dazu wird dem Befehl, der normalerweise an der Shell eingegeben würde, ein ! vorangestellt. !ls zeigt also den Inhalt des lokalen Verzeichnisses an. Wichtig ist, daß diese Befehle in einer eigenen Shell ausgeführt werden, die nach dem Aufruf wieder beendet wird. Ein Verzeichniswechsel mit !cd Verzeichnis ist deshalb nicht möglich!

Abbildung 39 zeigt ein Beispiel einer FTP-Session, in der ein paar gebräuchliche Befehle kurz vorgeführt werden. FTP bietet noch eine ganze Menge anderer Befehle, die man einerseits über den Befehl help und andererseits über die man-Page nachlesen kann.

3.4.2 Anonymous FTP mit Netscape

Auch wenn ich Netscape nicht unbedingt als FTP-Client empfehle, soll er doch der Vollständigkeit halber kurz angesprochen werden. Bisher hatten wir in Netscape beim Eingeben der URL's immer den Dienstyp http:// verwendet. Wenn Du nun auf einen FTP-Server zugreifen willst, kannst Du den Dienstyp ftp:// nutzen. ftp://ftp.foo.fh-furtwangen.de führt demnach zu unserem FTP-Server. Netscape nutzt jetzt automatisch Anonymous FTP. Für den einfachen Anonymous-Betrieb reicht diese Schnittstelle durchaus, sie ist trotz allem mit Vorsicht zu geniessen, da sie in vielen Fällen Probleme mit symbolischen Links oder Dateinamen mit Sonderzeichen (vornehmlich das Leerzeichen) hat. Dieses Problem lässt sich dann nur mit dem Kommandozeilen-FTP umgehen.

3.4.3 Interessante FTP-Server

Hier noch eine kurze Liste interessanter FTP-Server:

ftp.foo.fh-furtwangen.de	Foo-Pool: GNU Software, NetBSD
ftp.ai-lab.fh-furtwangen.de	AI-Labor: RedHat Linux
ftp.ghb.fh-furtwangen.de	Wohnheim GHB: Linux Kernels, Diverse Tools
ftp.informatik.rwth-aachen.de	TH Aachen: Alles (grundsätzlich).

```
atlantis:/ce_users/nehmer/tmp> ftp ftp.foo.fh-furtwangen.de
Connected to hanne.foo.fh-furtwangen.de.
220 hanne.foo.fh-furtwangen.de FTP server
Name (ftp:nehmer): nehmer
331 Password required for nehmer.
Password:
230 User nehmer logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> lcd ce3
Local directory now /amd/server-erlab/erlab/nehmer/ce3
ftp> !ls
add-new-2-all                               new-user-info.tex
new-user-info.tex.backup                    protokoll
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 2
-rw-r----- 1 nehmer  users          2008 Aug 24 10:20 benchmarks
226 Transfer complete.
ftp> get benchmarks
local: benchmarks remote: benchmarks
200 PORT command successful.
150 Opening BINARY mode data connection for bench-
marks (2008 bytes).
226 Transfer complete.
2008 bytes received in 0.418 secs (4.7 Kbytes/sec)
ftp> mput new*
mput new-user-info.tex? y
200 PORT command successful.
150 Opening BINARY mode data connection for new-user-info.tex.
226 Transfer complete.
6929 bytes sent in 0.0181 secs (3.7e+02 Kbytes/sec)
mput new-user-info.tex.backup? n
ftp> quit
atlantis:/ce_users/nehmer/tmp>
```

Abbildung 39: Beispiel einer FTP Session

A Weitere interessante Programme

Diese Liste soll einen Ansatz zu weiterer, im Foo-Pool vorhandener Software geben. Diese Programme verfügen über umfangreiche Man- und Info-Pages, so daß dort ausreichend Informationen über die Bedienung zu finden ist. Mitunter sind nicht alle Programme überall direkt zu erreichen, da die automatische Umleitung von Kommandozeilen-Utilities problematisch ist. Solche Programme sind durch in Klammern gesetzte Rechnernamen oder Betriebssysteme gekennzeichnet. (Zum Zeitpunkt der Drucklegung!)

\LaTeX	\LaTeX ist <i>das</i> Textsatzsystem. Ein wenig gewöhnungsbedürftig aber sehr mächtig. Die Foo-Bibel wurde ebenfalls mit \LaTeX erstellt. (Ultrix 4.2, Linux) (Anm.: LaTeX wird „Latech“ ausgesprochen.)
emacs	Emacs ist der König der Editoren. Sehr komplex, mächtig, aber anfangs nicht ganz einfach zu bedienen. Wichtigste Informationsquelle: Info-Pages. Eine kleine Kommandoübersicht findet sich im Anhang C
xemacs	Der ein wenig graphischer orientierte Bruder des regulären Emacs.
vi	Manche leute schwören auf diesen, sehr eigenwilligen, Texteditor. Ideal für kleine, schnelle Änderungen. Die Kommandoübersicht im Anhang B sei jedem empfohlen ;)
a2ps	Mit a2ps können ASCII-Text in PostScript umgewandelt werden. Dabei wird zweispaltig im Querformat gedruckt, was sowohl übersichtlich als auch papiersparend ist.
elm	Ein einfacher Mail-Reader. (Globales Startkommando: elmfoo, Linux)
pine	Ein komplexer Mail-Reader, der nicht ganz so komfortabel wie Netscape ist, dadurch aber sehr viel flexibler und mächtiger wird. (Linux)
gcc, g++	Der GNU C / C++ Compiler
gdb	Der GNU Debugger
acroread	Adobe Acrobat Reader (liest *.PDF Dateien)

B vi Kommandoübersicht

vi benötigt lediglich **(Esc)** als spezielle Taste. Alle Kommandos können durch Kombination von **(Esc)** und normalen, alphanumerischen Zeichen erreicht werden. Dadurch ist vi ein Editor, der in so ziemlich jeder Umgebung arbeitet, aber von sämtlichen Bedienungskonzepten anderer Editoren abweicht. Er unterscheidet zwischen Kommando- und Eingabemodus. Im Kommandomodus können die unten angegebenen Kommandos *direkt* benutzt werden. Befindet man sich im Eingabemodus (siehe unten), kann dieser mit **(Esc)** verlassen werden. Sitzt Du vor einem Rechner, dessen Tastatur keine Escape-Taste aufweist, kannst Du stattdessen **(F11)** verwenden. vi unterscheidet bei Kommandos zwischen Groß- und Kleinschreibung! Je nach Plattform kann ausserdem der Cursor nur im Kommandomodus bewegt werden.

:q!	vi beenden und nicht speichern
:wq	Speichern und vi beenden
i	Eingabemodus aktivieren: Einfügen vor Cursorposition (Insert)
a	Eingabemodus aktivieren: Einfügen nach Cursorposition (Append)
o	Neue Zeile unter dem Cursor einfügen und Eingabemodus in dieser Zeile aktivieren
O	Neue Zeile über dem Cursor einfügen und Eingabemodus in dieser Zeile aktivieren
R	Eingabemodus aktivieren: Ab Cursorposition überschreiben
[anzahl]yy	Kopiert die nächsten [anzahl] Zeilen in den Puffer
P	Puffer vor Cursor einfügen
p	Puffer nach Cursor einfügen
J	Aktuelle und nachfolgende Zeile aneinanderhängen
:help	Anzeige der Hilfeseiten mit allen Tastenkombinationen (Viel Spaß!)

C emacs Kommandoübersicht

Eine richtige Einführung in Emacs würde wahrscheinlich nicht unter 200 Seiten wegkommen. Und für die ist hier nicht der richtige Ort. Ich will Dich deshalb auf die Info-Pages verweisen, die Emacs sehr ausführlich dokumentieren. Damit Du aber schonmal damit arbeiten kannst, will ich hier die wichtigsten Kommandos ansprechen. Emacs benutzt immer Control- und Meta- Tastenkombinationen. Da es in der Regel zweistufige Kombinationen sind, werden in der Statuszeile die bisher eingegeben Tastenkombinationen angezeigt. $\overline{\text{Ctrl}}+\overline{\text{x}}$, $\overline{\text{Ctrl}}+\overline{\text{c}}$ würde als C-x C-c auftauchen. Genau so werden in den verschiedenen Emacs-Referenzen auch die Tastenkombinationen geschrieben. C-x für $\overline{\text{Control}}+\overline{\text{x}}$, M-w für $\overline{\text{Meta}}+\overline{\text{w}}$. Die Meta-Taste ist auf den verschiedenen Plattformen unterschiedlich belegt. „Alt“, „Compose Character“ oder „Ext Char“ sind da gute Favoriten. Dazu kann die Eingabe von Dateinamen oder Funktionsnamen in der Statuszeile wie in einer Shell mit $\overline{\text{Tab}}$ komplettiert werden.

C-x C-c	Emacs beenden
C-x C-s	Datei speichern
C-x C-f	Datei öffnen
C-x s	Alle Dateien speichern
C-g	Teilweise begonnene Kommandoeingabe abbrechen
C-SPACE	Markierung an aktueller Cursorposition setzen
C-w	Den Bereich zwischen Markierung und Cursorposition löschen und in die Zwischenablage kopieren
M-w	Den Bereich zwischen Markierung und Cursorposition in die Zwischenablage kopieren
C-s	Suche vorwärts
C-r	Suche rückwärts
M-%	Suchen und ersetzen (interaktiv: SPC ersetzen, DEL nicht ersetzen, ! alles ersetzen, RETURN abbrechen)
C-x b	Zu einem anderen Puffer wechseln. (Eingabe des Puffernamens in der Statuszeile.)
C-x k	Puffer entfernen
C-h i	Info-Reader starten

Abbildungsverzeichnis

1	Login unter Linux mit Start der graphischen Benutzeroberfläche.	9
2	Ändern des Paßwortes mit <code>yppasswd</code>	10
3	Der Window-Manager <code>fvwm</code> in einer Standardkonfiguration	11
4	Der <code>fvwm</code> -Pager und die automatische Mail-Benachrichtigung	12
5	Die Menüleiste, das Anwendungsmenü und die Window-Tools	12
6	Das CDE in einer möglichen Standardkonfiguration	14
7	Die Toolbar des CDE	14
8	Beenden einer X-Session unter Ultrix und OSF/1	15
9	Beenden einer X-Session unter CDE	15
10	Der Standardprompt der <code>tcsh</code> auf <code>hanoi</code>	16
11	Verwendung der Textkomplettierung mit <code>Tab</code> innerhalb der Shell	17
12	Die Verzeichnishierarchie im <code>foo</code> -Pool	20
13	Auszug aus <code>ls (1)</code> einer Sun Solaris	26
14	Abtrennung der Info-Pages des <code>foo</code> -Pools	27
15	Verzeichnisstruktur <code>/usr/doc</code> auf Linux-Systemen	28
16	Druckformate von typischen <code>foo</code> -Pool Anwendungen	34
17	Zugriff auf einen anderen Rechner via <code>ssh</code>	37
18	Zugriff auf einen anderen Rechner via <code>rsh</code>	37
19	Umleiten einer graphischen Anwendung via <code>ssh</code>	38
20	Umleiten einer graphischen Anwendung mit <code>rsh</code>	39
21	Der Netscape Navigator	43
22	Übersicht über die gebräuchlichsten Länderkürzel in WWW-Adressen	44
23	Netscape: Proxy-Konfiguration	45
24	Netscape: Cache-Konfiguration	46
25	Netscape: Smart Browsing Konfiguration	47
26	Netscape: Bookmarks	47
27	Netscape: Konfiguration eigener Name und E-Mail-Adresse	49
28	Netscape: Konfiguration der Mailserver	50
29	Netscape: Konfiguration der Mail-Formatierung (HTML/Plain Text)	51
30	Netscape's Launcher	51
31	Netscape: Eine erste Test-Email	52
32	Netscape: Die Antwort auf die Test-Email	52
33	Netscape: Eine E-Mail mit mehreren Empfängern	54
34	Netscape: Konfiguration der News-Komponente	56
35	Netscape: Newsgruppen abonnieren	57
36	Netscape: News lesen	57
37	Anonymous FTP Login	60

38	Normaler FTP Login	60
39	Beispiel einer FTP Session	62