

UNIX-Grundlagen

Blockiertes Terminal

- warten
- Enter, Line Feed oder Ctrl-J
Reset-Befehl: *Jreset* Ctrl-J
aus- und wieder einloggen
- Ctrl-Z falls Jobcontrol möglich ist
⇒ das noch laufende Programm wird suspendiert und man erhält neue Shell-Eingabeaufforderung. Mit **jobs** die laufenden jobs anzeigen lassen; mit **fg** das Programm in den Vordergrund zurückholen oder mit **kill** beenden.
- Ctrl-C oder Delete

Einloggen von einem anderen System aus

Bekannte Terminalprogramme sind: **procomm**, **qmodem**, **kermit**, **telnet**, sowie **cu** und **tip** mittels Modem über Telefonleitung. Starten eines Terminalprogramms unter UNIX *programmname remote_host_name*; Ausstieg mit *exit*;

X-Window und die Windowmanager

Textmode das Terminal läuft ohne **X**;
starten von **X** mit:

xinit

xdm ist nur für den root erlaubt;

Graphikmode ohne Fenster auf dem Terminal läuft **X** aber kein Window-Manager;
einloggen unter **xdm**:

X Window System
Login:
Password:

Graphikmode mit Fenstern auf dem Terminal läuft **X** und ein Windowmanager.

Windowmanager:

mwm Motif Window Manager; starten mit *mwm &*

olwm Open Look Window Manager; starten mit *olwm &*

twm Tab Window Manager; starten mit *twm &*

setzen des Eingabe-Fokus (input focus) das farblich hervorgehobene Fenster ist das aktive, es empfängt die Eingaben (von Tastatur und Maus); je nach Einstellung des Window Managers reicht es den Pointer auf das Fenster zu bewegen, oder man muß zusätzlich noch klicken;

zusätzliches xterm-Fenster öffnen mit *xterm* oder *xterm &*;

Client separates Dienstprogramm in einem eigenen Fenster;

Shells

In den Eingabeaufforderungen der Bourne- und der Korn-Shell steht meist ein \$-Zeichen, in denen der C-Shell meist ein %-Zeichen.

Text mit Hilfe der Maus editieren

Text markieren: entweder

- linke Taste drücken und über den zu markierenden Text ziehen;
- linke Taste am linken Textrand drücken und die rechte Taste am rechten Textrand;

Text an die Stelle des Cursors kopieren mittlere Maustaste drücken (bei 2-Tastenmaus beide gleichzeitig);

Befehlssyntax:

befehlsname *option(en)* *datei(en)*

Optionen oft einzelne Buchstaben, denen ein Minuszeichen vorangestellt ist; sie sind durch 1 oder mehrere Leerzeichen oder Tabs getrennt;

mehrere Befehle in einer Kommandozeile durch Semikolons (;) getrennt;

Kontrollzeichen

Ctrl-X Eingabezeile vom Anfang bis zur Cursorposition löschen;

Ctrl-K Eingabezeile von Cursorposition bis zum Ende löschen;

Ctrl-S Bildschirmausgabe stoppen;

Ctrl-Q Neustart der Ausgabe nach Stopp durch Ctrl-S;

Ctrl-D zurück zur UNIX-Kommandoebene; Beenden bei manchen Programmen; Ausloggen;

Ctrl-C oder **Delete** Abbruch eines Programmes (Endlosschleife, ...);

Fehlermeldungen beim Ausloggen

There are stopped jobs Programme wurden angehalten aber nicht beendet => mit **fg** die angehaltenen Programme wieder in den Vordergrund holen, dann beenden; ...

not login shell logout-Versuch auf einer subshell; subshell mit **exit** beenden, loginshell mit **logout**;

nützliche Befehle

date Datums- und Zeitangabe;

who Angabe wer seit wann wo eingelogged ist;

who am i Antwort: eigener host! username terminal loggin-time

Hilfe

apropos *suchbegriff* Durchsucht die manualpages nach dem angegebenen Suchbegriff.

befehl **--help** Kurzbeschreibung des Befehls und seiner Optionen.

man *befehl* Ausführliche Beschreibung des angegebenen Befehls. Die manualpages sind meist in folgende Abschnitte untergliedert:

NAME Name des Befehls.

SYNOPSIS Befehlswort und die möglichen Optionen und Argumente.

DESCRIPTION Beschreibung des Befehls samt seiner Optionen ähnlich der Ausgabe auf: *befehl -help*, aber ausführlicher.

OPTIONS Genaue Erläuterungen zu den Optionen.

FILES Die vom Befehl verwendeten oder zu ihm zugehörigen files.

SEE ALSO Ähnliche Befehle.

BUGS Dokumentierte Fehler.

AUTHOR Name des Autors.

info Die auf Texinfo files beruhende online-formatierte GNU-Hilfe. Die einzelnen Menüpunkte sind durch '*' gekennzeichnet. Fortbewegung durch die Menüs:

Auswahl des Menüpunktes Mit dem Cursor darauffahren und **return**.

Auswahl des nächsten folgenden Menüpunktes (next): n

Auswahl des vorhergehenden Menüpunktes (previous): p

Zurückspringen ins übergeordnete Menü (up): u

Ctrl-h alle in info verfügbaren Befehle werden angezeigt;

zgrep *suchwort* \usr\doc\HOWTO* greppen durch die gezippten HOWTO-files;

Dateiverwaltung

absoluter Pfad gesamter Pfad vom root-directory beginnend; beginnt mit einem '/';

relativer Pfad ab einem (in dem man sich gerade befindet) Unterverzeichnis beginnend;

df *file* Gibt die Größe des vorhandenen, benutzten und noch verfügbaren Platzes an. Gibt man ein file an, so werden die Angaben nur fuer das das file enthaltene filesystem ausgegeben, ansonsten fuer alle gemounteten filesystems.

du *file od. directory* Gibt die Groesse des angegebenen files oder directories samt den darin enthaltenen subdirektories und files an.

Wildcards

***** für beliebig viele Zeichen;

? für 1 Zeichen;

[] eines der eingeklammerten Zeichen soll an dieser Stelle stehen; z.B.: **[a,b,c]** a oder b oder c; **[2-5]** Zahlenbereich 2, 3, 4, 5;

[!...] ein beliebiges Zeichen, welches **nicht in der Klammer angegeben ist, exakt einmal an dieser Position**;

Befehle

pwd print working directory;

ls list files;

-a alle, auch die versteckten;

-l long, mit Typ, Zugriffsrechten, Anzahl der links, Eigentümer, zugriffsberechtigter Gruppe, Größe, Datum der letzten Änderung und Name;

-F ausführbare files werden mit einem **'*** markiert und Verzeichnisse mit einem **'/'**

mv move, rename

mv *altdateiname neudateiname* Umbenennen einer Datei;

mv *dateiname pfad* Verschieben der Datei ins durch den Pfad angegebene directory;

cp move, rename

cp *altdateiname neudateiname* kopieren einer Datei;

cp *dateiname pfad* Kopiert eine Sicherungskopie der Datei ins durch den Pfad angegebene directory;

rmp remote copy;

rmp *host-name:pfad/dateiname host-name:pfad/dateiname*

mkdir Verzeichnis(se) erstellen;

rm Verzeichnis(se) oder Datei(en) löschen;

-i mit interaktiver Frage ob man wirklich löschen will;

rmdir löscht Verzeichnisse; Verzeichnisse müssen leer sein;

find Datei- oder Verzeichnissuche;

find [*path*] **-name** '*zusuchender_name*'

find [*path*] **-mtime** *-tage* Dateien die weniger als *tage* Tage alt sind werden gesucht; die *path*-Angabe darf nicht vergessen werden; will man nur unter dem aktuellen Verzeichnis suchen '.', ab dem root-Verzeichnis '/';

Dateizugriffsrechte

Durch die Einstellung von **umask** mittels einer 3-stelligen Oktalzahl in einem hidden file (oft `.login` oder `.cshrc`) sind die Standardzugriffsrechte der von einem selbst erzeugten Dateien eingestellt. Will man Dateien einer Gruppe oder der Öffentlichkeit zur Verfügung stellen, so muß man die entsprechenden Zugriffsrechte an diesen Dateien setzen, aber auch für alle übergeordneten directories die entsprechenden execute-Rechte freigeben.

chmod *3-stellige_oktalzahl dateiname* in der 3-stelligen Oktalzahl sind die jeweiligen Rechte: **read**, **write** und **execute** für **user**, **group** und **public** codiert (1-aus-n-Code bei binärer Darstellung).

E-/A-Umleitung

befehl < *datei* Der Befehl verwendet die Datei als Eingabe; meist ist die Tastatur die Standardeingabeeinheit;

datei > *befehl* Der Befehl verwendet die Datei als Ausgabeeeinheit; meist ist der Bildschirm die Standardausgabeeeinheit; um zu vermeiden, daß eine Datei dadurch überschrieben wird kann man in der Datei `.profile` den befehl **set noclobber** für die C-Shell setzen, für die Korn-Shell lautet der Befehl: **set -o noclobber**

datei >> *befehl* Der Befehl hängt die Ausgabe an die Datei an;

Pipelines und Filter

befehl | *befehl* Pipeline; der 1. Befehl gibt seine Ausgabe an den 2. Befehl als dessen Eingabe weiter; der 2. Befehl wird als Filter bezeichnet;

Befehle die oft als Filter verwendet werden

Diese Befehle sind natürlich auch als selbständige Programme nützlich.

grep "*muster*" *datei* (Globally search for a Regular Expression and Print all lines containing it) Läßt man die Dateiangabe weg, so wird die Standardeingabe (Tastatur) bearbeitet;

–**f** fixed string; der string kann auch Shell-Sonderzeichen enthalten;

–**v** gibt die Zeilen aus in denen das Muster nicht enthalten ist;

–**n** gibt die das Muster enthaltenen Zeilen mit vorangestellter Zeilennummer aus;

–**l** gibt die Namen der das Muster enthaltenen Dateien aus;

–**c** gibt die Anzahl der das Muster enthaltenen Zeilen aus;

–**i** unterscheidet bei der Suche nicht zwischen Groß- und Kleinbuchstaben;

sort sortiert, falls nicht anders angegeben alphabetisch;

–**n** sortiert numerisch;

–**r** kehrt die Reihenfolge um;

–**f** sortiert gemischt Groß- und Kleinschreibung;

+*x* ignoriert beim Sortieren die ersten *x* Felder;

pr Dateiformatierung für Probeausdrucke

- x* *x*-spaltige Ausgabe der Datei;
- h** "**title**" schreibt den title in die Titelzeile;
- t** unterdrückt header, oberen- und unteren Seitenrand;
- e** wandelt Tabs in Leerzeichen um;

Druckbefehle

groff startet troff und einen zum Drucker passenden Postprozessor;

troff wird normalerweise von troff gestartet;

nroff Datei für den Druck formatieren;

System-V Druckerkonzept:

lp Druckbefehl für System-V Unixsysteme (Solaris, HP-UX).

- dprinter** Drucken auf dem Drucker *printer*;
- nx** die Datei *x*-mal ausdrucken;
- m** mail-Antwort wenn der Ausdruck beendet ist;
- cancel** Abbruch des Druckjobs mit Hilfe des Druckernamens oder der Prozeß-ID;

lpstat Ansehen der Druckerwarteschlange;

BSD-Druckerkonzept

lpr Druckbefehl für BSD-Unixsysteme wie z.B.: Linux

- Pprinter** Drucken auf dem Drucker *printer*;
- x* die Datei *x*-mal ausdrucken;
- m** mail-Antwort wenn der Ausdruck beendet ist;

lprm Abbruch des Druckjobs mit Hilfe des Druckernamens oder der Prozeß-ID;

lpq Ansehen der Druckerwarteschlange;

- o** zeigt alle, nicht nur die eigenen Druckjobs an;

lprm Job_Nr Abbrechen eines Printer-Jobs;

Multitasking

befehl & einen Befehl im Hintergrund laufen lassen; die Shell weist dem Prozeß eine Prozeß-ID (Identifikationsnummer) zu und zeigt diese auch gleich an;

befehl_1; *befehl_2* & mehrere Befehle in der C-Shell im Hintergrund laufen lassen;

(*befehl_1*; *befehl_2*) & mehrere Befehle in der Bourne- oder Korn-Shell im Hintergrund laufen lassen;

bg einen Prozeß in den background befördern;

fg einen Prozeß in den foreground befördern;

ps processes; gibt die laufenden Prozesse mit Prozeß-ID, Terminal von dem aus gestartet wurde, Laufzeit und Namen aus;

kill *PID(s)* Prozess(e) abbrechen;

kill-9 *PID(s)* Prozess(e) abbrechen, auch die die sich mit kill alleine nicht abbrechen lassen; der Befehl beendet sogar die eigene (ihn aufrufende) Shell;

sleep *n* läßt den Befehl *n*-sec schlafen;

Der Editor vi

Der Editor vi (sprich: "wie-ei") ist der älteste und verarbeitetste unter UNIX. Nach dem Aufruf mit **vi** *dateiname* befindet er sich im **Command Mode**. Um Text schreiben zu können muß man mittels eines Befehls in den **Edit Mode** wechseln. Zurück in den Command Mode kommt man mit **Esc**. Die **ex**-Befehle werden vom Command Mode aus mit vorangestelltem ':' aufgerufen. Man kann auch die **ex**-Befehle zu einem Befehls-Mode zusammenfassen, in den man vom Command Mode aus mit ':' gelangt.

Command Mode-Commands

Cursor Movements

h Cursor nach links;

j Cursor nach unten;

k Cursor nach oben;

l Cursor nach rechts;

0 an den Zeilenanfang;

\$ an das Zeilenende;

Ctrl-D halbe Seite vorwärts;

Ctrl-U halbe Seite rückwärts;

Ctrl-F eine Seite vorwärts;

Ctrl-B eine Seite rückwärts;

1G zum Dateianfang (Zeile 1);

xG zur Zeile *x*;

b Cursor zum Beginn des aktuellen Wortes; Wort ist Folge von Buchstaben, Ziffern und underline;

w Cursor zum Beginn des nächsten Wortes; Wort ist Folge von Buchstaben, Ziffern und underline;

e Cursor zum Ende des aktuellen Wortes;

fc zum nächsten Zeichen *c*;

Suchbefehle

% springt zur korrespondierenden Klammer; es wird zwischen den verschiedenen Klammerformen unterschieden;

/*muster* suche vorwärts nach dem angegebenen *muster*;

?*muster* suche rückwärts nach dem angegebenen *muster*;

n Wiederholung der letzten Suche;

N Wiederholung der letzten Suche in umgekehrter Richtung;

Löschbefehle

x Löschen des Zeichens auf dem sich der Cursor befindet;

D Löschen bis zum Zeilenende;

dd aktuelle Zeile löschen;

undo

u letzten Befehl rückgängig machen (mehrmals möglich);

Wechsel vom Command– in den Edit Mode

a append nach der aktuellen Cursorposition;

i insert vor der aktuellen Cursorposition;

o open new line; Einfügen nach der aktuellen Zeile;

Wechsel vom Edit Mode in den Command–

Esc

ex–Befehle

:w write file; file speichern;

:wq write and quit; file speichern und vi beenden;

:q! vi beenden ohne file zu speichern;

:r *dateiname* nach der aktuellen Zeile die Datei einfügen;

:n editieren der nächsten Datei (falls beim Aufruf von vi mehrere Dateinamen angegeben wurden);

:e *dateiname* verlassen der alten Datei und editieren der angegebenen;

Nützliche Befehle

cat (concatenate) hängt die angegebenen Dateien zusammen; für kurze Dateien ist es als schneller viewer geeignet;

more *datei(en)* ist auch als Filter praktisch zur Ausgabe von Dateien, **less** ist aber praktischer;

pg *datei(en)* ähnlich wie more;

less *datei(en)* Ausgabe von Dateien, oder als Filter;
Bewegen in der Datei:

Cursor down oder **e** eine Zeile weiter;

Cursor up oder **y** eine Zeile zurück;

d halbe Seite vorwärts;

u halbe Seite rückwärts;

f 1 Seite vorwärts;

b 1 Seite rückwärts;

/pattern suchen des Suchwortes nach vorne;

?pattern suche des Suchwortes nach hinten;

n weitere Suche des Suchwortes in der gleichen Richtung;

N weitere Suche des Suchwortes in umgekehrter Richtung;

:n zum nächsten file das in der Kommandozeile angegeben wurde;

:p (previus) zum davor angegebenen file;

q (quit) beenden;

Emails

Emails können mit komfortablen mail-Programmen wie z.B.: **pine** und **elm** empfangen und gesendet werden, allerdings ist auf jedem Unix-System das primitive mail-Programm namens **mail** verfügbar.

mail nachsehen ob man mail(s) bekommen hat und lesen; die einzelenen Nachrichten sind durch **&** getrennt; zur nächsten email mit **return**

x beenden des email-Lesens, die emails bleiben in der inbox enthalten;

q beenden des email-Lesens; die gelesen emails werden in der Datei *mbx* gespeichert;

mail *emailadresse* **return** eine email schicken; in der nächsten Zeile wird nach dem subject gefragt; danach den Text des emails eingeben; Abschließen des emails mit **.** **return**; danach können noch carboncopies angegeben werden;

ftp

ftp Aufruf des Programms; grundsätzlich alle Befehlswoorte mit return abschließen, dann wird man automatisch weitergefragt;

ftp *host* Aufruf des Programmes und Aufbau der Verbindung;

? Ausgabe der Befehlsliste;

get remote-file holen;
put local-file schicken;
close Verbindung zu remote-host abbrechen;
quit Programm beenden;