

Skript zum Kurs
Einführung in die Benutzung der Ausbildungsrechner
der Fachgruppe Mathematik und Informatik

Holger Arndt, Peter Feuerstein, Andre Weinberg

23. Ausgabe, März 2024

Bergische Universität Wuppertal



Fakultät für Mathematik und Naturwissenschaften

Die Vorlesung *Einführung in die Benutzung der Ausbildungsrechner*, die jeweils zu Beginn der Vorlesungszeit jeden Semesters als Blockkurs gehalten wird, ist Voraussetzung für den Zugang zu den Ausbildungsrechnern der Fachgruppe Mathematik und Informatik in der Fakultät für Mathematik und Naturwissenschaften der Bergischen Universität Wuppertal.

Dieser Kurs ist kein vollständiger UNIX-Kurs. Er soll vielmehr eine Einführung in die Benutzung der Ausbildungsrechner der Fachgruppe geben und einige grundlegende UNIX-Kenntnisse vermitteln. Die Darstellung der beschriebenen UNIX-Kommandos beschränkt sich auf die häufig benötigten Varianten. Zur systematischen Einführung in das Betriebssystem UNIX/Linux sei auf entsprechende Lehrangebote in der Fachgruppe Mathematik und Informatik, in der Fakultät für Elektrotechnik, Informationstechnik und Medientechnik und im Zentrum für Informations- und Medienverarbeitung (ZIM) sowie auf die angegebene Literatur verwiesen.

Dieses Skript ist auch online verfügbar unter der URL

<https://www.math.uni-wuppertal.de/de/studium/lehrveranstaltungen/rechnereinfuehrung/>

Inhaltsverzeichnis

1	UNIX–Grundlagen	5
1.1	UNIX	5
1.2	Benutzer	5
1.3	Netz und Maschinen	6
1.4	Dateisystem	6
1.5	Prozesse	7
1.6	Ansprechpartner	7
1.7	Accountvergabe	7
2	Einige Verhaltens–Regeln	8
2.1	Vernünftiger Umgang mit Ressourcen	8
2.2	Was man nicht tut, auch wenn es geht	8
3	Login - Anmelden im System	9
3.1	Anmeldung an der Konsole	9
3.2	Kommandozeilen–Login	10
3.3	Passwort	10
3.4	Login auf einem anderen Rechner übers Netz	10
3.5	Ausloggen	12
4	Graphische Oberflächen	13
4.1	Umgang mit Fenstern, Maus und Tastatur	13
4.1.1	Maus	13
4.1.2	Tastatur	14
4.2	KDE – Ein Linux Desktop	14
4.2.1	KDE konfigurieren und beenden	14
4.3	Konsole–Fenster	15
4.4	Bildschirm sperren	15
4.5	Programme unter graphischen Oberflächen	15
4.5.1	Dateimanager	15
4.5.2	Texteditoren	16
4.6	Entfernte Ausgabe von X–Anwendungen	16
4.7	Entfernte Anmeldung mit xdm/kdm	17
5	Eingaben an der Kommandozeile	18
5.1	Umleitung und Verkettung von Kommandos	18
5.1.1	Umleitung von Ein– und Ausgabe	18
5.1.2	Verkettung von Kommandos durch Filter	19
5.2	Hintergrundprozesse	20
5.3	Starten von Programmen zu einem späteren Zeitpunkt	20
5.4	Starten von Programmen mit nohup und nice	21

6	Grundlegende UNIX-Kommandos	22
6.1	Dateien und Verzeichnisse	22
6.1.1	Abkürzungen und Wildcards	22
6.1.2	Arbeitsverzeichnis bestimmen und wechseln	22
6.1.3	Verzeichnisinhalte anzeigen	22
6.1.4	Kopieren und Umbenennen von Dateien und Verzeichnissen	23
6.1.5	Erstellen und Löschen von Dateien, Verzeichnissen	23
6.1.6	Links	24
6.1.7	Dateiinhalte anzeigen	24
6.1.8	Text in Dateien suchen	25
6.1.9	Zugriffsrechte ändern	25
6.1.10	Dateien suchen	25
6.1.11	Informationen über Plattenbelegung	26
6.2	Dateitransfer zwischen verschiedenen Rechnern übers Netz	27
6.3	Dateiformate und passende Programme	28
7	Hilfen	29
7.1	CIP-Hilfesystem	29
7.2	SuSE-Hilfe / KDE-Hilfe	29
7.3	Hilfe-Optionen der UNIX-Kommandos	29
7.4	Online-Handbuch	29
7.5	Sonstige Dokumentationen	30
8	Was tun bei Problemen	31
8.1	Login schlägt fehl	31
8.2	Probleme beim Programm-Aufruf	31
8.3	Prozesse killen	32
8.4	remote-kill	33
8.5	Wenn nichts mehr hilft	34
9	Drucken	35
9.1	Druckbare Dateien	35
9.2	Absetzen von Druckaufträgen	35
9.3	Abbrechen von Druckaufträgen	36
9.4	Übersicht über die Druckkommandos	36
9.5	Was tun bei Druckerproblemen	36
10	Internet und WWW	37
10.1	Grundsätzliches	37
10.2	Netiquette	37
10.3	E-Mail	37
10.3.1	E-Mail-Adressen	37
10.3.2	Thunderbird / Seamonkey Mail	37
10.3.3	KMail unter KDE	38

10.3.4	Kommandozeilen-Programme	39
10.3.5	Weiterleiten von Mail	39
10.3.6	Spam-E-Mail	39
10.4	Internet-Browser	40
10.4.1	Navigation im WWW (Web-Surfen)	40
10.4.2	Speichern von Dokumenten	41
10.5	News	41
10.6	FTP	41
10.6.1	Kommandozeilen-FTP	41
10.6.2	Secure File Transfer Program	42
10.6.3	Graphische FTP-Programme	42
10.6.4	File Transfer mit Seamonkey und Firefox	42
10.7	Informationssuche	43
10.7.1	Bibliotheks-Recherche	43
10.7.2	WWW-Suchmaschinen	43
11	Archivierung und Dateitransfer	44
11.1	Zugriff auf HOME-Verzeichnisse	44
11.2	Aufräumen im HOME-Verzeichnis	44
11.3	Disketten, CDs/DVDs, USB-Speicher,	45
11.4	Dateiformate zum Dateitransfer und zur Archivierung	46
12	Anpassen der Arbeitsumgebung	47
12.1	Pfade	47
12.2	Aliase	47
13	Installierte Software	48
A	Benutzungsordnung für die Ausbildungsrechner	49

Vorbemerkung

Die Ausbildungsrechner der Fachgruppe (FG) Mathematik und Informatik sind in drei Gruppen (Pools) aufgeteilt:

- Die Rechner des CIP-Pools im Raum G.14.11 sind primär für Bachelor-Veranstaltungen der FG vorgesehen.
- Die Rechner des IT-Pools in G.16.15 werden vorrangig für Veranstaltungen der Bachelor- und Master-Studiengänge Informatik und Informationstechnologie (IT) verwendet.
- Die Rechner des PI-Pools in G.16.13 stehen insbesondere den Tutor*innen und Master-Studierenden der FG für freies Arbeiten zur Verfügung und werden bei Bedarf auch für Lehrveranstaltungen der FG genutzt.

Alle drei Pools bestehen aus Intel-basierten Linux-PCs.

Studierende können je nach Studiengang eine Zugangsberechtigung für die beiden ersten Rechnerpools CIP und IT erhalten. Dieser Zugang gilt stets für beide Rechnerräume. Eine Zugangsberechtigung für die Rechner des PI-Pools kann zusätzlich bei der Systemadministration beantragt werden.

Bei der Beschreibung von Eingaben und Befehlen in diesem Skript sollen folgende Konventionen gelten:

- *Schreibmaschinenschrift* bedeutet Ein- und Ausgaben bei Beispielen,
- **Fettdruck** bedeutet Namen von Befehlen und Optionen oder Benutzereingaben,
- *Kursivdruck* bezeichnet variable Argumente (z. B. den Namen einer Datei),
- In eckigen Klammern [. . .] stehender Text bezeichnet optionale Argumente, die angegeben werden können, aber nicht müssen.

1 UNIX-Grundlagen

1.1 UNIX

UNIX ist ein sog. Multi-Tasking- und Multi-User-Betriebssystem. Ein Multi-User-System verwaltet und unterscheidet verschiedene Benutzer¹, die (gleichzeitig) an einem Rechner arbeiten können. Ein Multi-Tasking-System zeichnet sich dadurch aus, dass jeder an einem System arbeitende Benutzer mehrere Kommandos absetzen und Programme starten kann, die (quasi-)parallel abgearbeitet werden.

Im Laufe der Entwicklung und Portierung des Betriebssystems UNIX auf die unterschiedlichste Hardware sind viele Versionen unter verschiedenen Namen entstanden: Xenix, SunOS, Solaris, FreeBSD, Linux, ...

Auf sämtlichen Ausbildungsrechnern kommt derzeit das Betriebssystem openSUSE Linux zum Einsatz.

1.2 Benutzer

Jeder Benutzer eines UNIX-Systems hat einen eindeutigen Benutzernamen sowie eine Benutzernummer (UID) und gehört (mindestens) einer Benutzer-Gruppe an. Damit niemand unter falschem Namen an einem System arbeiten kann, gehört zu jedem Benutzernamen ein Passwort, das beim Anmelden im System eingegeben werden muss. Dieses Passwort darf nicht weitergegeben, sondern muss geheim gehalten werden.

Jeder Benutzer hat einen Teil des Verzeichnisbaumes (siehe 1.4) zur Verfügung, in dem er seine Daten ablegen kann. Dieser Teil beginnt im sog. HOME-Verzeichnis des Benutzers. (Wo genau sich dieses Verzeichnis im Dateisystem befindet, steht in Abschnitt 11.1). Hier hat ein Benutzer sein Arbeitsverzeichnis und kann dieses strukturieren, wie er es für richtig hält.

Da Festplattenplatz beschränkt und in der Regel knapp ist, kann der einem Benutzer zur Verfügung stehende Speicherplatz beschränkt werden. Dieser Quota-Mechanismus ist derzeit auf den Ausbildungsrechnern deaktiviert. Alle Benutzer sind daher verpflichtet, ihren genutzten Plattenspeicher selbst zu kontrollieren und zu beschränken (vgl. Abschnitt 6.1.11).

¹ „Benutzer“ (engl. ‘User’) wird im Folgenden geschlechtsneutral genutzt, meint also immer „Benutzerin und/oder Benutzer“

1.3 Netz und Maschinen

Die Rechner sind vernetzt und so konfiguriert, dass es keine Rolle spielt, an welchem Rechner eines Pools ein Benutzer arbeitet. Er hat auf jedem Rechner eines Pools immer dieselben Daten und Einstellungen vorliegen.

Jeder Rechner im Netz hat einen eindeutigen Maschinen-Namen, anhand dessen er identifiziert werden kann. Die einzelnen Pools bestehen aus folgenden Rechnern:

- Der CIP-Pool besteht aus den Rechnern l01 bis l23,
- der IT-Pool aus den Rechnern l101 bis l115,
- der PI-Pool aus den Rechnern l201 bis l210.

Um diese Rechner von außerhalb der Fachgruppe zu erreichen, ist an den jeweiligen Rechnernamen der Domainname `studs.math.uni-wuppertal.de` anzuhängen, wodurch er zu einem Internet-weit eindeutigen Namen (FQDN) wird:

- beim CIP-Pool z. B.: `l15.studs.math.uni-wuppertal.de`,
- beim IT-Pool z. B.: `l103.studs.math.uni-wuppertal.de`.

1.4 Dateisystem

Das UNIX-Dateisystem hat eine Baumstruktur und besteht aus Verzeichnissen (Ordern) und Dateien. Ein Verzeichnis ist dabei mit einem Behälter zu vergleichen, der weitere Verzeichnisse und/oder Dateien enthalten oder auch leer sein kann. Jedes Verzeichnis und jede Datei wird mit einem Namen identifiziert, der lokal eindeutig ist. Das bedeutet, dass gleiche Namen für Dateien und Verzeichnisse nur innerhalb eines Verzeichnisses nicht erlaubt sind (so wie in einer Straße jede Hausnummer eindeutig sein sollte).

Zuoberst ist das `root`- oder Wurzel-Verzeichnis, von dem aus der ganze Verzeichnisbaum ausgeht. Dieses Verzeichnis hat den Namen `/`. Die Verzeichnisse im Dateisystem liegen in der Regel tatsächlich auf verschiedenen Festplatten, die auch an verschiedene vernetzte Rechner angeschlossen sein können. Für den Benutzer eines Rechners spielt es aber keine Rolle, wo die Daten physisch liegen; entscheidend ist die Position im (logischen) Dateisystem.

Jedes Verzeichnis und jede Datei hat einen Benutzer und eine Gruppe als Eigentümer, denen die Datei/das Verzeichnis gehört, und Zugriffsrechte, die besagen, welche Benutzer welche Rechte zum Zugriff auf das Verzeichnis/die Datei haben.

Bei Zugriffsrechten wird wie folgt unterschieden:

- U** User – Der Eigentümer des Verzeichnisses/der Datei;
- G** Group – Die Mitglieder der Gruppe, der die Datei gehört;
- O** Other – Alle Benutzer, die weder der **User** sind noch **Group** angehören.

Die Zugriffsrechte für Verzeichnisse sind:

- R** Read – Lesen des Verzeichnis-Inhaltes;
- W** Write – Schreiben im Verzeichnis, also Anlegen und Löschen von Dateien und Unterverzeichnissen;
- X** eXecute – Wechseln in das Verzeichnis als aktuelles Arbeitsverzeichnis.

Für Dateien gibt es folgende Rechte:

- R** Read – Lesen der Datei, also Betrachten des Dateiinhaltes;
- W** Write – Schreiben in die Datei, also Ändern des Dateiinhaltes;
- X** eXecute – Ausführen der Datei als Programm.

Mit Hilfe des Kommandos `ls -l` (vgl. Abschnitt 6.1.3) können die Zugriffsrechte angezeigt werden. Dabei zeigen die Spalten 2-4 die Zugriffsrechte für den Besitzer der Datei an, die Spalten 5-7 für die Gruppe und die Spalten 8-10 für alle anderen Benutzer:

```
-rwxr-xr-- 1 otto  gast      278 Jul 24 12:01 doit
drwxr-xr-x 2 otto  gast    2048 Jul 17 13:27 Pics
-rw----- 1 otto  gast     216 Jul 16 12:35 test.dvi
-rw----- 1 otto  gast      62 Jul 16 15:01 test.tex
```

Das hier gezeigte Verzeichnis beinhaltet ein Unterverzeichnis (Pics) und drei Dateien. Alle gehören dem Benutzer otto und der Gruppe gast.

doit ist eine für otto und alle Mitglieder der Gruppe gast lesbare und ausführbare Datei, wobei nur otto die Datei verändern darf und alle, die nicht der Gruppe gast angehören, die Datei lesen dürfen.

Pics ist ein Verzeichnis (gekennzeichnet durch das „d“ am Anfang der Zeile), dessen Inhalt jeder lesen kann, in das jeder wechseln darf, in dem aber nur otto, der Eigentümer, Dateien erstellen und löschen darf.

Die anderen Dateien (test.dvi und test.tex) darf nur otto lesen und ändern; ausführbar sind sie auch für ihn nicht.

Die Ausführbarkeit einer Datei als Programm hängt nicht vom Namen ab (wie z. B. bei Windows durch Dateisuffixe wie .exe oder .com), sondern nur vom Zugriffsrecht, das nicht für alle Benutzer dasselbe sein muss.

1.5 Prozesse

Die oben beschriebene Multi-Tasking-Fähigkeit von UNIX-Systemen, die Möglichkeit, dass mehrere gestartete Programme quasi gleichzeitig aktiv sein können, bedeutet auch, dass es Mechanismen geben muss, auf irgendwelche gestarteten Programme später Einfluss zu nehmen.

Jedes gestartete Programm ist ein Prozess, über den u. a. folgende Informationen vorliegen:

- Die Benutzernummer (UID) des Benutzers, der den Prozess gestartet hat.
- Eine Prozess-Nummer (PID), anhand der er identifiziert werden kann.
- Die Prozess-Nummer des übergeordneten Prozesses (PPID), von dem aus der Prozess gestartet wurde.
- Das Terminal (TTY), von dem aus der Prozess gestartet wurde.
- Der Befehl (CMD), mit dem der Prozess gestartet wurde.

Mit diesen Informationen ist es möglich, einen Prozess ausfindig zu machen, um ihn nachträglich zu beeinflussen, beispielsweise zu löschen (vgl. Abschnitt 8.3).

1.6 Ansprechpartner

Ansprechpartner für sämtliche Ausbildungsrechner der FG Mathematik und Informatik sind

1. Dr. Feuerstein, Raum G 14.18, Durchwahl 2818, E-Mail: feuerstein@math.uni-wuppertal.de
2. Dr. Galgon, Raum G 14.05, Durchwahl 3777, E-Mail: galgon@math.uni-wuppertal.de
3. Dr. Arndt, Raum G 14.16, Durchwahl 2033, E-Mail: arndt@math.uni-wuppertal.de

Bei Problemen kann auch immer eine E-Mail an den Benutzer root (das ist der Systemadministrator-Account) auf der betroffenen Maschine oder an root@math.uni-wuppertal.de gesandt werden (vgl. Abschnitt 10.3). Die Mail wird an die passenden Ansprechpartner weitergeleitet. Vor einer Problemmeldung sollte aber stets das Hilfesystem (vgl. Abschnitt 7) benutzt werden.

1.7 Accountvergabe

Die Accountvergabe (Vergabe von Zugangsberechtigungen) für die Ausbildungsrechner regelt die Systemadministration der Fachgruppe.

Voraussetzung für die Beantragung eines Rechnerzugangs ist die Teilnahme an der Veranstaltung *Einführung in die Benutzung der Ausbildungsrechner*, die jeweils zu Beginn eines jeden Semesters gehalten wird (und zu der dieses Skript gehört).

Tutor*innen der Informatik-Übungen in den Ausbildungsrechnerräumen können zusätzlich zu ihrem Account auf den Rechner-Pools auch eine Zugangskarte für die Schließanlagen an den Computerräumen erhalten. Für diese Karten ist ein Pfand von 10,- Euro zu hinterlegen.

2 Einige Verhaltens-Regeln

Neben der Verwaltungs- und Benutzungsordnung des ZIM ist die Benutzerordnung für die Ausbildungsrechner der FG Mathematik und Informatik (siehe Anhang) zu beachten. In aller Kürze hier einige wichtige Regeln:

- **Die Rechner dürfen auf keinen Fall ausgeschaltet werden!**
- **Geben Sie Ihr Passwort niemandem weiter! Sie haften für sämtliche unter Ihrem Account vorgenommene Aktionen.**
- **Lesen Sie regelmäßig Ihre E-Mail! Die Systemadministration kommuniziert mit Ihnen auf diesem Weg.**
- **Verwenden Sie niemals bereits bedrucktes Papier oder Umweltschutzpapier in den Druckern.**
- **Löschen Sie Ihre Druckaufträge aus der Warteschlange, bevor Sie den Raum verlassen.**
- **Beachten Sie die Netiquette (vgl. Abschnitt 10.2).**
- **Überlassen Sie die Einstellung der Kühlung (die normalerweise eingeschaltet sein sollte) dem Fachgruppenpersonal. Bei laufender Kühlung sollten Fenster und Türen stets geschlossen bleiben.**
- **Falls Sie als letzter den Raum verlassen, schließen Sie Fenster und Tür und löschen Sie das Licht.**
- **In den Computer-Räumen darf die Heizung nicht eingeschaltet werden.**
- **Das Verzehren von Speisen und Getränken ist in den Rechnerräumen grundsätzlich verboten.**
- **Die Verwendung eigener Rechner (Notebooks etc.) ist in den Computer-Räumen nur an den dafür vorgesehenen Plätzen gestattet. Insbesondere ist es untersagt, bestehende Installationen zu manipulieren, sowie eigene Rechner an die Datensteckdosen in den Fachgruppenräumen anzuschließen. (Verwenden Sie anstelle dessen das WLAN-Netz der Universität.)**

2.1 Vernünftiger Umgang mit Ressourcen

Die verfügbaren Ressourcen an Plattenplatz, Rechenkapazität und Verbrauchsgütern sind begrenzt, deshalb halten Sie sich an folgende Hinweise:

- Rechenintensive Produktions-Rechenläufe sollen nach Absprache mit den unter 1.6 genannten Ansprechpartnern nachts und am Wochenende laufen. Solche Prozesse machen unter Umständen weiteres sinnvolles Arbeiten an den Rechnern unmöglich. Wie Prozesse zu einem späteren Zeitpunkt gestartet werden, ist in Abschnitt 5.3 beschrieben.
- Drucken Sie nicht mehr als nötig. Besprechen Sie notwendige größere Ausdrücke vorher mit der Systemadministration. **Der Ausdruck von privaten Daten sowie von Skripten und Online-Handbüchern ist untersagt!**
- Halten Sie Ihren Datenbestand klein und räumen Sie Ihr HOME-Verzeichnis regelmäßig auf. Nicht mehr benötigte Daten sollen gelöscht oder archiviert werden. Welche Daten regelmäßig gelöscht werden sollen und wie dies geschieht, ist in Abschnitt 11.2 beschrieben.
- Belegen Sie nicht mehr als einen Arbeitsplatz. Andere Benutzer möchten auch an den Rechnern arbeiten.
- Sperren Sie Ihre Konsole (siehe Abschnitt 4.4) nur, wenn Sie den Raum nur für wenige Minuten verlassen wollen. Loggen Sie sich ansonsten aus (vgl. Abschnitt 3.5).
- **Bitte beachten Sie, dass Aktionen der Benutzer durch das System protokolliert werden.**

2.2 Was man nicht tut, auch wenn es geht ...

UNIX bietet eine Reihe von Sicherheitsmechanismen, um den unbefugten Zugriff auf Daten abzuwehren. Es kommt allerdings immer wieder vor, dass Benutzer absichtlich oder unbedacht Zugriff auf ihre Daten zulassen. Deshalb gibt es einige Verhaltensregeln, deren Missachtung dem eigenen Ansehen schadet:

- Blicken Sie anderen Benutzern nicht bei der Passwort-Eingabe „über die Schulter“.
- Lesen Sie keine Daten anderer Benutzer, auch wenn diese nicht explizit für Sie gesperrt sind.
- Sollten Sie feststellen, dass ein Benutzer, ohne den Bildschirm gesperrt zu haben, eingeloggt ist, aber sich nicht im Raum befindet, warten Sie einen Moment, ob er vielleicht zurückkommt und loggen Sie ihn dann ggf. aus. Senden Sie ihm eine E-Mail, um ihn auf seine Vergesslichkeit hinzuweisen.

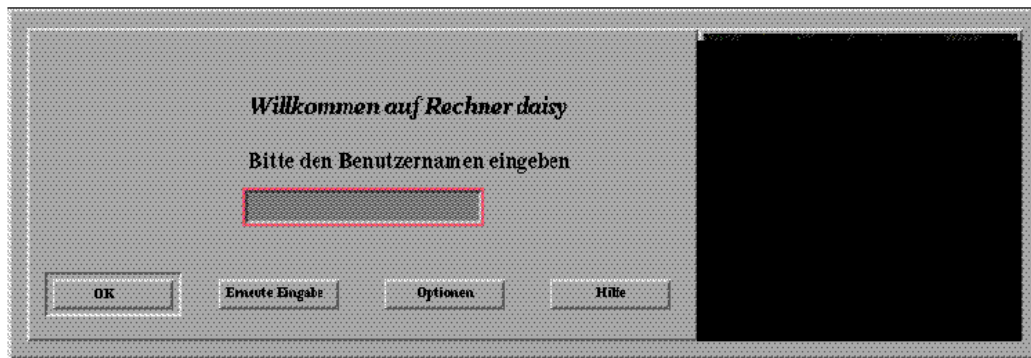
3 Login - Anmelden im System

Um an einem der Rechner zu arbeiten, muss man sich zunächst anmelden. Dieser Vorgang heißt „Login“ und besteht darin, seinen Benutzernamen und dann nach Aufforderung sein Passwort einzugeben. Ist beides richtig eingegeben worden, kann man mit dem System arbeiten.

3.1 Anmeldung an der Konsole

Wurde ein Rechner eine bestimmte Zeit nicht benutzt, ist ein Bildschirmschoner aktiv, der den Monitor dunkel schaltet. Will man sich an einem Rechner anmelden, dessen Bildschirm nichts anzeigt, genügt ein Druck auf die Taste <Return> oder eine Bewegung mit der Maus, damit innerhalb weniger Sekunden der Monitor wieder etwas anzeigt.

Auf allen Rechner-Pools ist ein graphisches Anmeldeprogramm aktiv. Der normale Anmeldevorgang erfolgt dadurch, dass man Benutzernamen und Passwort in einer Anmeldemaske eingibt.



Sind der Benutzername oder das Passwort nicht richtig eingegeben worden, erscheint eine Fehlermeldung und man kann einen erneuten Versuch starten.

Je nachdem, an welchem Rechnersystem man sich einloggen will, stehen verschiedenen weitere Optionen zur Auswahl. So hat man bei einem Linux-Rechner die Möglichkeit, unter **Sitzungsart** eine graphische Oberfläche auszuwählen. Einige Beispiele:

- KDE / Plasma – Das K Desktop Environment
- GNOME – Das GNOME Desktop Environment des GNU Projekts
- Xfce – Ein einfacher Fenstermanager.

Üblicherweise wird stets der zuletzt gewählten Desktop gestartet.

Ansonsten gibt es noch die Möglichkeit einer Befehlszeilen-Anmeldung in einer textorientierte Login-Shell ohne graphische Oberfläche. Diese Anmeldeart wird auch bei entferntem (remote) Login benutzt.

Der Wechsel zu einer (von insgesamt sechs) Textkonsolen, an denen eine Befehlszeilen-Anmeldung möglich ist, geschieht durch die Tastenkombinationen <Strg>+<Alt>+<F1> (<F2> ... <F6>). Zurück zur graphischen Oberfläche gelangt man durch <Strg>+<Alt>+<F7> oder <Strg>+<Alt>+<F8>.

Hinweis:

Auf den Ausbildungsrechnern sind die regulären Studierenden-Zugangsdaten gültig, die Sie mit der Einschreibung erhalten haben. Nutzen Sie zur Änderung Ihres Passworts den entsprechenden Dienst, den Sie auf den Webseiten des ZIM unter **Mein Zugang** finden.

Um die Rechner nutzen zu können, ist die einmalige lokale Anmeldung in einem der Ausbildungsrechnerräume erforderlich. Bei dieser ersten Anmeldung wird Ihr HOME-Verzeichnis erstellt. Sobald dieses Verzeichnis existiert, ist auch eine entfernte (remote-)Nutzung der Rechner möglich.

3.2 Kommandozeilen-Login

Wählt man die Befehlszeilen-Anmeldung, erhält man zunächst die Aufforderung, sich anzumelden:

```
Welcome to openSUSE Leap 15.4 - Kernel 5.14.21-150400.24.81-default (tty1).
l112 login:
```

Nun gibt man seinen Benutzernamen ein und drückt die <Return> -Taste:

```
l112 login: otto
```

Dann folgt die Aufforderung, das Passwort einzugeben:

```
Password:
```

Hier gibt man nun das Passwort ein, gefolgt von der <Return> -Taste. Zu beachten ist, dass das eingegebene Passwort nicht angezeigt wird. Wurden Benutzername und Passwort richtig eingegeben, erscheint eine Meldung der Art:

```
Have a lot of fun...
otto@l112:~>
```

Waren der eingegebene Benutzername oder das Passwort falsch (z. B. weil Groß- und Kleinschreibung nicht beachtet wurden), erscheint eine Fehlermeldung und man kann einen weiteren Versuch starten.

3.3 Passwort

Bei den Ausbildungsrechnern werden die regulären Studierenden-Zugangsdaten genutzt, daher kann das Passwort nur Uni-weit geändert werden. (Beachte Hinweis in Abschnitt 3.1.)

Um das eigene Passwort auf normalen UNIX-Systemen per Kommando zu ändern, gibt man (an einem Shell-Prompt) das Kommando **passwd** ein. Nach Eingabe des alten Passwortes muss man zweimal das neue Passwort eingeben (um zu vermeiden, dass durch Tippfehler ein falsches Passwort gilt und man sich nicht mehr anmelden kann):

```
otto@l112:~> passwd
Changing password for otto.
Enter login(LDAP) password: .....
Neues Passwort: .....
Geben Sie das neue Passwort erneut ein: .....
LDAP password information changed for otto
otto@l112:~>
```

Es gibt gewisse Mindestanforderungen an die Komplexität gültigen Passwörter, etwa, dass neben Buchstaben auch Zahlen oder Sonderzeichen enthalten sein müssen. Wenn das eingegebene neue Passwort diesen Ansprüchen nicht genügt, erhält man eine entsprechende Fehlermeldung.

3.4 Login auf einem anderen Rechner übers Netz

Um sich über das Netz auf einem anderen Rechner einzuloggen, verwendet man den Secure Shell Client mit dem Kommando **ssh**:

```
ssh [-l username] [ hostname ] oder ssh username@hostname
```

Wird kein Benutzername durch **-l *username*** angegeben, so wird derselbe Benutzername verwendet, unter dem man auf der lokalen Maschine eingeloggt ist.

Das sichere remote login Programm **ssh** baut eine verschlüsselte Verbindung zu dem angegebenen Rechner auf:

```
otto@l113:~> ssh l112 -l otto
Password: .....
Last login: Fri Apr 21 09:34:05 2006 from wmail5.math.uni-wuppertal.de
Have a lot of fun...
otto@l112:~>
```

Unsicherer remote-Zugang

Neben dem sicheren remote-Zugang via **ssh** gibt es zwei weitere Kommandos: **rlogin** und **telnet**.

Bitte beachten Sie aber, dass sicheren Verbindungen grundsätzlich immer der Vorzug gegenüber nicht sicheren remote-Zugängen zu geben ist!

rlogin unterscheidet sich von **telnet** im wesentlichen durch höheren Komfort und die Möglichkeit, die Passwortabfrage zu umgehen. Dafür steht **telnet** auch auf vielen Nicht-UNIX-Plattformen zur Verfügung. Zwischen UNIX-Rechnern ist meist **rlogin** vorzuziehen. Der Aufruf der Programme erfolgt durch

```
telnet [-l username] [ hostname ]
```

```
rlogin [-l username] hostname
```

Wird **telnet** ohne Parameter gestartet, wird in den interaktiven Modus von **telnet** geschaltet, von dem aus **telnet**-Kommandos eingegeben werden können. Dies ist am Prompt **telnet>** erkennbar. Gibt man **help** am **telnet**-Prompt ein, erhält man eine Liste der **telnet**-Kommandos.

Wird der Name eines Rechners als Parameter angegeben, erfolgt eine Verbindung zu dem Rechner:

```
otto@wmail15:~> telnet wmpi06
Trying 132.195.95.38...
Connected to wmpi06.
Escape character is '^]'.

SunOS 5.7

login: otto
Password: .....
Last login: Fri Jul 22 17:31:07 from wmail15
Sun Microsystems Inc.   SunOS 5.7           Generic October 1998
wmpi06%
```

rlogin hat keinen interaktiven Modus, sondern baut immer eine Verbindung zu dem angegebenen Rechner auf. Ist kein Benutzername mit **-l username** angegeben worden, wird ein Login mit demselben Benutzernamen wie auf der lokalen Maschine versucht:

```
otto@wmail15:~> rlogin wmpi06
Password: .....
Last login: Fri Apr 21 10:02:44 from wmail15
Sun Microsystems Inc.   SunOS 5.7           Generic October 1998
wmpi06%
```

Hinweis:

Aus Sicherheitsgründen sind die Rechner der Fachgruppe Mathematik und Informatik mit den unsicheren Kommandos **telnet** und **rlogin** (sowie die Kommandos **rsh**, **rcp**, **ftp**, ...) nicht erreichbar. Wenn Sie einen Rechner von außerhalb erreichen wollen, verwenden Sie stattdessen den Secure Shell Client **ssh** oder entsprechende Programme, die eine sichere Verbindung ermöglichen.

Da desweiteren der remote-Zugang per **ssh** nur innerhalb des Universitätsnetzes möglich ist, ist es für ein Login von außerhalb der Bergischen Universität erforderlich, zunächst eine VPN-Verbindung herzustellen, durch die der entsprechende Rechner ein (virtueller) Uni-Rechner wird. Aktuelle VPN-Clients und weitergehende Informationen zu den Außenzugängen zum Uni-Netz sind stets Sie auf den Webseiten des ZIM verfügbar.

3.5 Ausloggen

Bei einem Kommandozeilen-Login oder einem Login über's Netz gibt man einfach **exit** oder **logout** (oder, je nachdem welche Kommando-Shell man benutzt, <Strg>+<D> auf deutschen bzw. <Control>+<D> auf amerikanischen Tastaturen) ein und wird ausgeloggt:

```
otto@l112:~> exit
logout
l112 login:
```

Erfolgt durch **exit** kein Logout oder erscheint bei Eingabe von **logout** eine Meldung der Art

```
/home/otto> logout
Nicht in einer Login-Shell.
```

oder

```
otto@l112:~> logout
bash: logout: not login shell: use 'exit'
```

so wurde von der Shell aus eine weitere gestartet. Dann muss solange **exit** (oder <Strg>+<D>) eingegeben werden, bis alle Sub-Shells beendet sind und die Login-Shell beendet werden kann.

```
/home/otto> exit
exit
otto@l112:~> exit
exit
otto@l112:~> exit
logout
l112 login:
```

Unter den graphischen Oberflächen gibt es einen Menü-Button, um die Sitzung zu beenden. Wurde die Sitzung manuell von der Kommandozeile gestartet, muss noch ein Logout wie oben beschrieben durchgeführt werden. Ansonsten erfolgt das Ausloggen automatisch.

Beim KDE unter Linux findet man den Punkt zum Abmelden im Menü „Programme starten“ (unten links auf dem Bildschirm) oder im Hintergrund-Menü, das man durch einen Rechtsklick mit der Maus auf dem Bildschirmhintergrund erhält.

4 Graphische Oberflächen

Die graphischen Oberflächen unter UNIX basieren auf **X-Window**, einem Netzwerk-orientierten Fenstersystem. Damit ist es möglich, Programm-Ausführung und -Ausgabe auf verschiedenen Rechnern durchzuführen. Das Layout der Oberfläche (**Graphical User Interface**), also: des Hintergrundes, der Fenster, der Menüs und Symbole sowie Position und Art der Fenster-Steuerelemente (**Widgets**) wird durch den auf **X-Window** basierenden Fenstermanager, die graphische Oberfläche, bestimmt.

Als graphische Oberflächen stehen auf den Linux-PCs unter anderem das **K Desktop Environment KDE** (ursprünglich: **Kool Desktop Environment**) und das **GNOME Desktop Environment (GNU Network Object Model Environment)** zur Auswahl.

4.1 Umgang mit Fenstern, Maus und Tastatur

Meist sind unter einer graphischen Oberfläche auf dem Bildschirm (**Desktop**) mehrere Anwendungs-Fenster sichtbar. Um Eingaben in einem dieser Fenster zu machen, muss sich dieses von den anderen hervorheben. Man spricht vom **FOCUS**, der sich auf dieses Fenster bezieht. Das bedeutet, dass sich Eingaben über die Tastatur auf dieses Fenster beziehen. Um einem Fenster den Focus zuzuweisen, wird dieses mit der Maus angeklickt. Die graphischen Oberflächen können auch so konfiguriert werden, dass ein Bewegen des Mauszeigers auf das Fenster genügt, um ihm den Focus zu geben.

Unter graphischen Oberflächen gestartete Programme können als normales Fenster oder als Symbol (**Icon**) auf dem Bildschirm sichtbar sein. Zu einem Icon verkleinerte Programme sind noch aktiv, arbeiten also weiter, falls das Programm überhaupt etwas bearbeitet. Eingaben an Programme können allerdings in der Regel nur an Programme mit geöffnetem Fenster gemacht werden (eine Ausnahme ist dabei das Icon für den Drucker, vgl. Abschnitt 9).

Je nachdem, mit welcher Oberfläche man arbeitet, werden die iconifizierten (aktiven) Applikationen entweder direkt auf dem Desktop abgelegt oder in einer Leiste am Bildschirmrand gesammelt. Icons, die auf dem Desktop abgelegt sind, sind unter einiger Oberflächen also aktive Programme; unter KDE werden alle aktiven Programme in der Task-Leiste am unteren Bildschirmrand angezeigt wogegen Symbole auf dem Desktop zum Starten der betreffenden Programme verwendet werden.

Der Wechsel vom Icon zum Fenster geschieht, je nachdem, wo die Icons abgelegt werden, durch Einfach- oder Doppelklick mit der linken Maustaste auf das Icon. Um ein Fenster zu einem Icon zu verkleinern, dienen, je nach graphischer Oberfläche, die Auswahl von **Schließen** oder **Minimieren** im System-Menü des Anwendungs-Fensters oder ein Iconifikationsknopf am Fensterrand.

4.1.1 Maus

Die Bedienung graphischer Oberflächen, die auf **X-Window** basieren, erfolgt überwiegend mit einer Maus mit drei Tasten. Handelt es sich um eine Maus mit Scrollrad, so dient dieses als der mittleren Maustaste. Die Zuordnung der Maustasten zu den Ereignissen, die dadurch ausgelöst werden sollen, lässt sich konfigurieren; die Bezeichnungen "linke Maustaste", "mittlere Maustaste" und "rechte Maustaste" im folgenden entspricht der Standardeinstellung. Mit der Maus werden System- und Anwendungs-Menüs bedient, Text markiert sowie die Fensterverwaltung gesteuert.

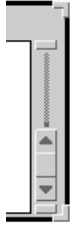
Die Markierung von Text geschieht normalerweise mit der linken Maustaste. Text wird markiert, indem die Maus bei gedrückter linker Taste darüber gezogen wird; die mittlere Taste dient dann zum Einfügen von markiertem Text. Mehrfach-Klicken mit der linken Maustaste markiert im Allgemeinen logische Einheiten: Ein Doppel-Klick das Wort unter dem Mauszeiger, ein Dreifach-Klick die Zeile unter dem Mauszeiger, ein Vierfach-Klick den ganzen Text im Fenster bzw. Editor. Die genaue Funktion der Mehrfach-Klicks unterscheidet sich je nach graphischer Oberfläche und Programm, zu dem das entsprechende Fenster gehört.

Klickt man mit der rechten Maustaste auf den Inhalt eines Programm-Fensters, werden programmspezifische Menüs angezeigt.

Durch Drücken und Festhalten der linken Maustaste, während sich der Mauszeiger auf dem Rand eines Fensters befindet, kann dieses auf dem Desktop verschoben werden. Das Fenster macht jede Bewegung der Maus solange mit, bis die linke Taste losgelassen wird, und bleibt dann an dieser Position stehen.

Befindet sich der Mauszeiger auf der Ecke oder dem Rand eines Fensters, ändert sich das Symbol des Mauszeiger.

Drückt und hält man dann die linke Taste, kann man das Fenster durch Bewegung der Maus vergrößern oder verkleinern.



In einigen Programm-Fenstern besteht die Möglichkeit, sich mit Scrolling (Rollen) nicht mehr im Fenster sichtbare Bereiche anzusehen. Solche Fenster haben an der Seite einen vertikalen (wie im linken Bild), manchmal auch oben oder unten einen horizontalen Rollbalken. Durch Klicken mit der Maus auf die dort angezeigten Pfeile oder durch Verschieben des Balkens durch Drücken und Festhalten der linken Maustastekann man den Inhalt innerhalb des Fensters rollen lassen, falls der Rollbalken nicht wie im rechten Bild deaktiviert ist.



4.1.2 Tastatur

Auf Linux-Rechnern gibt es einige Tastenkombinationen, die man kennen sollte:

Tastenkombination	Aktion
<Strg>+<Alt>+<F1> ...	Wechsel in die erste Textkonsole ...
<Strg>+<Alt>+<F6>	Wechsel in die sechste Textkonsole
<Strg>+<Alt>+<F7> <Strg>+<Alt>+<F8> ...	Wechsel in die erste Graphikkonsole Wechsel in die zweite Graphikkonsole ...
<Strg>+<F1> <Strg>+<F2> ...	Umschaltung zwischen virtuellen Desktops
<Strg>+<Alt>+<L>	Bildschirm sperren
<Alt>+<F2> <Strg>+<Esc> <Strg>+<Alt>+<Esc>	Eingabefenster für Kommandos zum Starten von Programmen Anzeige der Prozessliste (ähnlich dem Windows-Taskmanager) Mauszeiger wird zum Totenkopf, mit dem man Prozesse beenden kann, indem man auf das entsprechende Fenster klickt
<Strg>+<Alt>+<Backspace>	gewaltsamer Neustart der graphischen Oberfläche (Achtung: Beendigung aller eigenen Prozesse!)

4.2 KDE – Ein Linux Desktop

Auf den Ausbildungsrechnern ist standardmäßig der Desktop KDE ("K Desktop Environment") installiert.

KDE ist so konzipiert, dass Benutzern der Umstieg von Microsoft Windows zu Linux leicht fällt. So sind die Symbole auf dem Bildschirm keine laufenden Programme, sondern Verknüpfungen (Short Cuts), durch deren Anklicken man Programme starten kann. Am unteren Bildschirmrand befindet sich, ähnlich wie unter Microsoft Windows, die Menüleiste mit (konfigurierbaren) Schaltflächen. Wenn sich der Mauszeiger einen Moment über diesen Schaltflächen in Ruhe befindet, wird ein Hilfetext angezeigt.

Der KDE besteht (in der Standardkonfiguration) aus vier virtuellen Desktops, von denen immer einer auf dem Bildschirm sichtbar ist und zwischen denen man durch die mit den Nummern **1**, **2**, **3** und **4** versehenen Schaltflächen auf der Menüleiste oder den Tastenkombinationen <Strg>+<F1> ... <Strg>+<F4> wechseln kann.

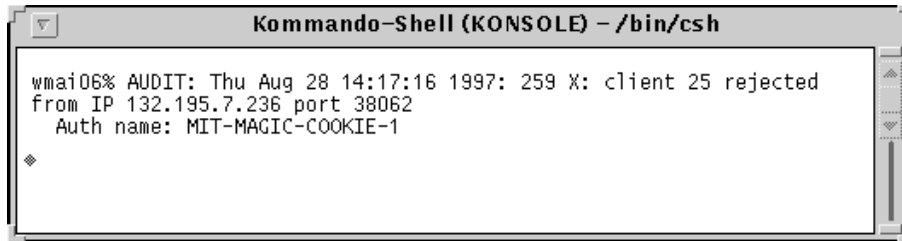
4.2.1 KDE konfigurieren und beenden

KDE bietet verschiedene Möglichkeiten, das Erscheinungsbild zu konfigurieren. Dazu verwendet man unter anderem die Einträge **Hintergrund einrichten...** und **Arbeitsfläche einrichten...** im Hintergrund-Menü des Desktops, das man durch einen Rechtsklick mit der Maus auf dem Bildschirmhintergrund erhält. Desweiteren steht zur Einrichtung der Arbeitsumgebung das KDE-Kontrollzentrum zur Verfügung, das über die Menüs in der unteren Menüleiste gestartet wird.

Im Menü "Programme starten" (unten links auf dem Bildschirm) oder im Hintergrund-Menü, findet man desweiteren den Punkt **Abmelden...** zum Beenden des KDE.

4.3 Konsole-Fenster

Unter allen graphischen Oberflächen gibt es eine spezielle Variante der Kommando-Eingabe-Fenster, die als **Konsole** bezeichnet werden und deren Funktion es ist, (meist Fehler-)Meldungen auszugeben, die keinem speziellen Kommando-Fenster zuzuordnen sind.

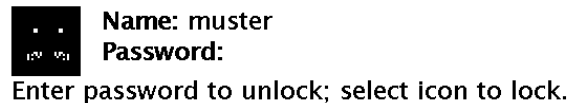


Ein solches Fenster sollte immer geöffnet sein; es kann dabei soweit verkleinert werden, dass gerade noch zu sehen ist, ob Fehlermeldungen auftreten. Allerdings sollte in diesem Fenster nicht normal gearbeitet werden, da sonst normale Ausgaben und Fehlermeldungen durcheinandergeraten.

4.4 Bildschirm sperren

Um kurz den Raum zu verlassen, ist es nicht nötig, sich abzumelden, sondern es kann der Bildschirm gesperrt werden, so dass für eine erneute Benutzung lediglich das Passwort des eingeloggten Benutzers eingegeben werden muss.

Um den Bildschirm zu sperren, klickt man unter KDE auf **Bildschirm sperren** oder **Sitzung sperren** im Hintergrund-Menü oder im Menü "Programme starten", oder man verwendet alternativ die Tastenkombination $\langle \text{Strg} \rangle + \langle \text{Alt} \rangle + \langle \text{L} \rangle$. Um die Sperre aufzuheben, drückt man eine Taste und gibt dann sein Passwort ein:



Dieses Sperren des Bildschirms darf nur benutzt werden, wenn man kurz den Raum verlässt. Wer länger abwesend ist, muss sich ausloggen.

4.5 Programme unter graphischen Oberflächen

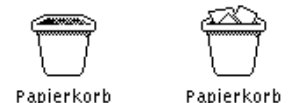
Unter KDE stehen eine Reihe von Anwendungen zur Verfügung, die sich ähnlich verhalten, wie vergleichbare Programme unter Microsoft Windows. Wie diese im einzelnen zu handhaben sind, kann mit Hilfe der Online-Hilfen, durch Ausprobieren und durch Nachsehen in den in Abschnitt 7 genannten Dokumenten in Erfahrung gebracht werden.

4.5.1 Dateimanager

Ein Dateimanager kann durch Auswählen des entsprechenden Eintrags im **K-Menü**-Menü oder durch Klicken auf das Haus-Symbol in der Menüleiste gestartet werden.

Der Name des aktuellen Verzeichnisses ist in der Zeile „Adresse“ aufgeführt, im Fenster darunter finden sich die Verzeichnisse und Dateien, die im aktuellen Verzeichnis gespeichert sind.

Zusätzlich befindet sich auf dem Desktop ein Symbol für einen Mülleimer oder einen Papierkorb. Dieser dient als Ziel für das Löschen von Dateien und Verzeichnissen (sie werden in den Papierkorb verschoben). Ob der Papierkorb leer oder voll ist, erkennt man am Icon.



Durch Anklicken des Papierkorbes öffnet sich ein Dateimanagerfenster, wonach durch Verschieben mit der Maus markierte Dateien wiederbelebt werden können. Zum endgültigen Löschen der Dateien und Verzeichnisse wählt man im Kontextmenü des Papierkorbs (durch Anklicken des Icons mit der rechten Maustaste) den Eintrag **Papierkorb leeren**.

Der Dateimanager bietet die Möglichkeit, Dateien und Verzeichnisse zu kopieren, zu löschen und zu verschieben, sowie durch Anklicken des Dateisymbols eine (vordefinierte) Aktion auszuführen. Bei Verzeichnissen ist die durch Mausklick mit der linken Maustaste eingeleitete Aktion der Wechsel in dieses Verzeichnis, bei Programmen werden diese gestartet, bei bestimmten Anwendungsdateien wird ein passendes Betrachtungsprogramm gestartet (beispielsweise bei Postscript- oder DVI-Dateien oder bei Graphiken in vielen Formaten). Bei allen Dateien, die mit keiner speziellen Anwendung verknüpft sind, öffnet sich ein Fenster, in dem man das zu startende Programm auswählen kann.

Der Dateimanager läßt sich nach eigenen Wünschen und Vorlieben konfigurieren. Die entsprechenden Einstellungsmöglichkeiten erreicht man über die Menüs am oberen Rand des Dateimanagerfensters.

Ist eine Diskette oder eine CD bzw. DVD in das entsprechende Laufwerk eingelegt worden, wird ein eigenes Dateimanager-Fenster für das Wechselmedium geöffnet, so dass zwischen beiden Fenster Dateien kopiert und verschoben werden können. Gleiches gilt für USB-Sticks. Es erscheint ggf. ein zusätzliches Icon auf dem Desktop, über welches das Gerät angesprochen werden kann. Da mobile Datenträger in den UNIX-Dateibaum eingehängt werden, muß vor der Entnahme des Datenträgers stets dessen Einbindung in das Dateisystem mit einem Rechtsklick auf das entsprechende Icon und **Laufwerk-Einbindung lösen** gelöst werden. Durch das Kommando *eject* oder Rechtsklick auf das CD-Icon und **Auswerfen** wird die CD bzw. DVD ausgeworfen.

Nähere Informationen über den Dateimanager bietet die Desktop-Hilfe (vgl. Abschnitt 7.2). Hinweise zur Verwendung der ZIP-Laufwerke finden sich in Abschnitt 11.3.

4.5.2 Texteditoren

Zur Eingabe und Bearbeitung von Texten dient unter der graphischen Oberfläche KDE der Texteditor **KWrite**. Man startet ihn durch den Eintrag im KDE-Menü (unten links auf dem Desktop) oder durch Eingabe von **kwrite [datei]** von der Kommandozeile.

Klickt man eine Textdatei im Dateimanager mit der rechten Maustaste an, so kann diese durch Auswahl von **Öffnen mit...** und **kwrite** sofort zur Bearbeitung geöffnet werden.

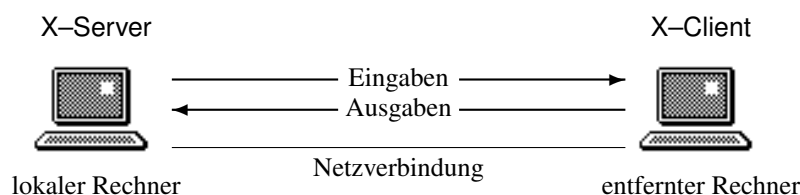
Die Texteditoren sind in der jeweiligen Desktop-Hilfe (vgl. Kapitel 7) genauer beschrieben.

Als ein weiterer, sehr komfortabler und daher zu empfehlender Editor, der sich insbesondere zur Bearbeitung von Programm-Quelltexten eignet, bietet sich **XEmacs** (die X-Window-Version des rein textbasierten Editors **Emacs**) an, der durch das Kommando **xemacs [datei]** gestartet wird.

4.6 Entfernte Ausgabe von X-Anwendungen

Eine der Stärken von X-Window ist die Möglichkeit, Anwendungen entfernt, also auf anderen Maschinen ablaufen, sich die (graphischen) Ausgaben aber lokal ausgeben zu lassen. Dabei spielt es keine Rolle, unter welchem Betriebssystem die entfernte Maschine arbeitet, solange dort nur die gewünschte X-Anwendung ablaufen kann. Man spricht dann vom **Client-Server-Prinzip**, wobei der **X-Server** das Programm (auf der lokalen Maschine) ist, das die Ausgaben auf den Bildschirm bringt und der **X-Client** jede Applikation ist, die ihre Ausgaben über den **X-Server** darstellt, unabhängig davon, ob die Applikation lokal oder entfernt läuft.

Für den Fall, dass **X-Server** und **X-Client** auf derselben Maschine (unter demselben Account) laufen, ist nichts weiter zu beachten; die Ausgaben erfolgen automatisch auf dem (lokalen) Bildschirm. Anders ist dies, wenn **X-Server** und **X-Client** auf verschiedenen Maschinen laufen:



Damit solch ein verteiltes Arbeiten möglich ist, müssen drei Voraussetzungen erfüllt sein:

1. Man benötigt einen Zugang auf der entfernten Maschine.
2. Der X-Client auf der entfernten Maschine muss wissen, wohin es seine Ausgaben senden soll.
3. Der X-Server auf der lokalen Maschine muss Programm-Ausgaben der entfernten Maschine erlauben.

Man beachte den dritten Punkt: Es ist natürlich unsinnig, jedem von irgendeinem Benutzer auf irgendeinem Rechner gestarteten Programm zu erlauben, auf den eigenen Desktop zu schreiben. Dies ist zunächst nur solchen Programmen erlaubt, die auf der Maschine ablaufen, auf der der X-Server läuft und demjenigen Benutzer gehören, dem der X-Server gehört (der ihn gestartet hat). Um Ausgaben von Programmen, die auf anderen Rechnern laufen und/oder anderen Benutzern gehören, zu erlauben, dient das Kommando **xhost**.

xhost + [rechner] erlaubt Ausgabe-Zugriffe (für *rechner*)
xhost - [rechner] verbietet Ausgabe-Zugriffe (für *rechner*)
xhost Gibt an, ob Zugriffskontrolle eingeschaltet ist und welche Rechner zugreifen dürfen

Die Anweisung auf dem entfernten Rechner, wohin Ausgaben eines X-Clients zu leiten sind, geschieht mit der DISPLAY-Variable:

setenv DISPLAY rechner:0.0 (C-Shell)
DISPLAY=rechner:0.0 ; export DISPLAY (Bourne-Shell / BASH)

Um beispielsweise auf dem Rechner l112 die Datei test.c mit dem Texteditor zu bearbeiten, während man selbst am Rechner l02 arbeitet, sind folgende Schritte notwendig:

```
otto@l02:~> xhost + l112
l112 being added to access control list
otto@l02:~> ssh l112
Password: .....
Last login: Wed Apr 26 16:20:14 2006 from l101.studs.math.uni-wuppertal.de
Have a lot of fun...
otto@l112:~> DISPLAY=l02:0.0; export DISPLAY
otto@l112:~> kwrite test.c
```

Dann wird auf dem X-Server des Rechners l02 ein kwrite-Fenster geöffnet, das auf dem Rechner l112 die Datei test.c bearbeitet. Der Name des entfernten Rechners wird im Titel in spitzen Klammern angezeigt: <@l112>.

Verwendet man den Secure Shell Client **ssh** mit aktiviertem X-Tunneling (was auf den Ausbildungsrechnern die Standardeinstellung ist), so ist die X-Ausgabe, also die entfernte DISPLAY-Variable und das lokale Zugriffsrecht, automatisch richtig konfiguriert.

Hinweis: Wird in einer X-Applikation auf einem entfernten Rechner z. B. ein Ausdruck angefordert, so wird dieser auf dem (default-)Drucker des entfernten Rechners ausgegeben, also im Allgemeinen dort, wo dieser Rechner steht. Das gilt ebenso für andere lokale Ressourcen beim Arbeiten auf entfernten Computersystemen, also (ZIP-)Disketten, CD-ROM-Laufwerke, Festplatten etc.!

4.7 Entfernte Anmeldung mit xdm/kdm

Falls alle Rechner in einem der Räume belegt sind oder dort eine Veranstaltung stattfindet, aber ein Rechner in einem anderen Rechnerraum frei ist, besteht durch xdm (bzw. kdm unter Linux) die Möglichkeit, einen Login-Bildschirm auf einem der nicht zugänglichen Rechner zu erhalten. Dazu ist in der Anmeldemaske unter **Optionen** zunächst **Entfernte Anmeldung**, dann **Host-Namen eingeben...** auszuwählen und der Name des entfernten Rechners einzugeben.

Unter Linux erhält man durch das Kommando

X :1 -once -query Rechnername

eine (weitere) X-Konsole für ein entferntes graphisches Login. Um anschließend zwischen den verschiedenen graphischen Konsolen umzuschalten, verwendet man die Tastenkombinationen <Strg>+<Alt>+<F7> für die erste Graphikkonsole, <Strg>+<Alt>+<F8> für die Zweite, usw.

5 Eingaben an der Kommandozeile

Eine Kommandozeile wird von der Konsole (ohne Start einer graphischen Oberfläche) und jedem Kommandozeilen- oder Shell-Fenster zur Verfügung gestellt. Dort ist jeweils eine Shell (ein Kommandointerpreter) aktiv, es kann jeder UNIX-Befehl abgesetzt werden und jedes Programm gestartet werden. Es macht für die unter KDE laufenden Programme keinen wesentlichen Unterschied, ob sie per Menü oder von der Kommandozeile gestartet werden.

Eine Shell stellt einen Prompt (Eingabeaufforderung) zur Verfügung, nimmt Eingaben des Benutzers entgegen und wertet diese aus, sobald sie mit <Return> abgeschlossen wurden. Unter UNIX stehen verschiedene Shells zur Verfügung: Die Bourne-Shell (**/bin/sh**), die Bourne-Again-Shell (**/bin/bash**), die C-Shell (**/bin/csh**), die Korn-Shell (**/bin/ksh**) und evtl. noch weitere Shells.

Standardeinstellung für die Benutzer der Ausbildungsrechner, also diejenige Shell, die beim Öffnen eines Shell-Fensters automatisch gestartet wird, ist die Bourne-Again-Shell (kurz: **BASH**), es besteht aber jederzeit die Möglichkeit, in andere Shells zu wechseln. So startet man etwa durch Eingabe des Kommandos **csh** in einer **BASH** eine C-Shell.

Ist eine eingegebene Kommandozeile syntaktisch richtig, wird sie ausgeführt; andernfalls erscheint eine Fehlermeldung der Shell (nicht zu verwechseln mit Fehlermeldungen von Programmen). Der Grund für derartige Fehlermeldungen ist meist, dass einige Zeichen für die Shell eine besondere Bedeutung haben, die nicht bedacht wird. Beispiele für Fehlermeldungen der C-Shell, die eine Kommando-Eingabe nicht interpretieren konnte:

```
/home/otto> (  
Zu viele '('.  
/home/otto> ls >  
Fehlender Name für Umlenkung.
```

Solche für die Shell relevanten Sonderzeichen können mit `\` maskiert werden, sie verlieren dann ihre Bedeutung für die Shell. Außerdem können Worte, die Sonderzeichen enthalten, in einfache (`' ... '`) oder doppelte (`" ... "`) Anführungszeichen eingeschlossen werden. Für Shells haben üblicherweise folgende Zeichen eine besondere Bedeutung:

```
! ? * \ [ ] ( ) @ $ & > < | ~ ; ' " `
```

Voraussetzung für den erfolgreichen Start eines Programmes ist, dass es beim Aufruf gefunden wird. Dabei spielt der Suchpfad für Programme eine wichtige Rolle: Dieser enthält (in der Umgebungsvariablen **PATH**) eine Liste von Verzeichnissen, die nach aufgerufenen Programmen durchsucht werden (vgl. Abschnitt 12.1).

```
/home/otto> guten tag  
guten: Befehl nicht gefunden
```

Bei der Eingabe von Befehlen muss zwischen zwei Arten von Kommandos unterschieden werden:

- interne Shell-Kommandos
- externe ausführbare (vgl. Abschnitt 1.4) Dateien im Dateibaum

Name und Aufrufsyntax der internen Shell-Kommandos variieren zwischen den verschiedenen Shells. Ein Beispiel für ein in jeder Shell vorhandenes internes Kommando ist das **test**-Kommando. Wenn ein Programm erstellt wird, das ebenfalls den Namen **test** hat, hat das interne Kommando Vorrang und wird beim Aufruf von

```
otto@l11:~> test
```

gestartet. Will man das eigene Programm **test** starten, muss unbedingt der Pfadname mit angegeben werden:

```
otto@l11:~> ./test
```

Dann wird das Programm **test** im aktuellen Verzeichnis anstelle des internen Shell-Kommandos gestartet.

5.1 Umleitung und Verkettung von Kommandos

5.1.1 Umleitung von Ein- und Ausgabe

Normalerweise lesen Programme ihre Eingaben von der Tastatur und geben ihre Ausgaben und Fehlermeldungen in dem Shell-Fenster aus, von dem aus sie gestartet wurden. Man spricht von **Standardeingabe**, **Standardausgabe** und **Standardfehlerausgabe**. Alle diese Standardkanäle können auf Dateien umgeleitet werden.

Bei der C-Shell geschieht die Umleitung der Eingabe, Ausgabe und Fehlerausgabe durch:

<code>programm > datei</code>	Schreibt die Ausgabe von <i>programm</i> in die (nicht vorhandene) Datei <i>datei</i> .
<code>programm >> datei</code>	Hängt die Ausgabe von <i>programm</i> an die (vorhandene) Datei <i>datei</i> an.
<code>programm >& datei</code>	Schreibt die Ausgaben und Fehlermeldungen von <i>programm</i> in die (nicht vorhandene) Datei <i>datei</i> .
<code>programm >>& datei</code>	Hängt die Ausgaben und Fehlermeldungen von <i>programm</i> an die (vorhandene) Datei <i>datei</i> an.
<code>programm < datei</code>	Liest die Eingaben für <i>programm</i> aus der Datei <i>datei</i> .
<code>programm << text</code>	Liest Eingaben für <i>programm</i> solange von der Standardeingabe, bis eine nur aus <i>text</i> bestehende Zeile auftritt.

Dabei können die Umleitungen auch kombiniert werden:

```
/home/otto> gzip -f < file.in > file.out
```

Man beachte, dass die Umleitung des Fehlerkanals auch den Ausgabekanal einschließt. Will man nur die Fehlerausgabe umleiten, geht dies so:

```
(programm > terminal ) >& datei
```

Dabei ist *terminal* der Name des aktuellen Shell-Fensters oder der Konsole. Diesen erhält man durch das Kommando **tty**.

```
/home/otto> tty
dev/pts/6
/home/otto> ( ls test1 test2 > /dev/pts/6 ) >& test.err
test1
/home/otto> cat test.err
ls: test2: Datei oder Verzeichnis nicht gefunden
```

Das **tty**-Kommando kann auch direkt in den Befehl integriert werden:

```
/home/otto> ( ls test1 test2 > `tty` ) >& test.err
```

Hinweis: Andere Kommandointerpreter (**sh**, **bash**, ...) unterscheiden sich in der Syntax, vor allem für die Umleitung von Fehlermeldungen. Genaueres findet sich im Online-Handbuch der entsprechenden Shell (vgl. Abschnitt 7.4). Glücklicherweise ist aber bei den am häufigsten benötigten Umleitungen

myprg > ausgabe.txt	Ausgabe von myprg in ausgabe.txt umleiten
myprg >& ausgabe.txt	Ausgabe und Fehlermeldungen von myprg in ausgabe.txt umleiten
myprg >> ausgabe.txt	Ausgabe von myprg an ausgabe.txt anhängen
myprg < eingabe.txt	Eingaben für myprg aus eingabe.txt lesen

die Syntax aller Shells identisch.

5.1.2 Verkettung von Kommandos durch Filter

Filter (Pipes) dienen dazu, die Ausgabe eines Programmes an ein anderes weiterzuleiten. Damit kann man eine Reihe voneinander abhängiger Befehle aufbauen und sich nur das Endergebnis anzeigen lassen. Das zugehörige Zeichen ist das Pipe-Symbol |.

```
programm1 | programm2
```

Startet *programm1* und danach *programm2*, wobei *programm2* die Ausgabe von *programm1* als Eingabe liest.

Prinzipiell kann jedes Programm rechts vom Pipe-Symbol stehen; nimmt das zweite Programm keine Eingaben von der Standardeingabe entgegen, entfällt die Ausgabe des ersten Programms.

Im Beispiel wird die Ausgabe des Kommandos **ls** durch einen Filter an das Programm **more** übergeben, mit dem man sich die Ausgabe Seite für Seite ansehen kann:

```
otto@l01:~> ls | more
```

Die Fehlerausgabe ist von der Filterung nicht betroffen, Fehlermeldungen werden dem nachfolgenden Programm nicht übergeben, sondern direkt ausgegeben.

5.2 Hintergrundprozesse

Die Eingabe eines Kommandos an der Kommandozeile bewirkt, dass die aufrufende Shell bis zur Beendigung des Kommandos blockiert ist. Bei Programmen, die eine Ausgabe im Shell-Fenster bewirken, ist dies auch sinnvoll. Wird allerdings ein Programm gestartet, das keine Text-Ausgabe hat (beispielsweise eine X-Applikation), ist diese Blockierung der Shell unnötig und lästig. Dem kann dadurch begegnet werden, dass das entsprechende Programm im Hintergrund gestartet wird. Das bedeutet, dass die aufrufende Shell das Programm startet, aber nicht auf dessen Beendigung wartet, sondern sofort wieder für die Eingabe weiterer Kommandos bereit ist.

Das Starten im Hintergrund wird durch das Zeichen **&** am Ende des Befehls bewirkt. Als Beispiel wird hier der Texteditor im Hintergrund gestartet:

```
otto@l101:~> kwrite test.c &
[1] 1125
```

Der Editor erscheint auf dem Bildschirm und es kann sowohl im Editor als auch weiter im Shell-Fenster gearbeitet werden. Die Ausgabe bedeutet, dass dies der erste aktive Hintergrundprozess ist ([1]), dessen Prozessnummer 1125 ist. Nach Beendigung des Hintergrundprozesses erscheint im Fenster der aufrufenden Shell die Meldung

```
[1] Done kwrite test.c
```

Mit Hilfe des **ps**-Kommandos (vgl. Abschnitt 8.3) kann man sich ansehen, welche Prozesse noch aktiv sind.

5.3 Starten von Programmen zu einem späteren Zeitpunkt

Bei längeren Rechenjobs ist es meist sinnvoll, diese Nachts laufen zu lassen, während die Maschinen nicht genutzt werden und niemand durch lange Ausführungszeiten gestört wird. Zum Absetzen solcher Jobs dient das Kommando **at**. Der Aufruf von **at** ist:

```
at [-f at_datei] [-t startzeit]   startet at-Job
at [-f at_datei] startzeit       startet at-Job
at -l                               listet alle at-Jobs des Benutzers
at -r at_job_id                   entfernt at-Job at_job_id
```

Die beiden Aufrufe zum Starten eines **at**-Jobs unterscheiden sich in der Angabe der Startzeit:

Bei der ersten Möglichkeit (mit **-t startzeit**) wird die Startzeit im Format MMDDhhmm angegeben (jeweils zwei Stellen für Monat, Tag, Stunde und Minute).

Bei der zweiten Möglichkeit erfolgt die Angabe der Startzeit u. a. wahlweise als: h, hh, hhmm oder hh:mm, dazu ggf. die Angabe eines Tages als *today*, *tomorrow*, *+ n days*, *+ n weeks*, *Monday* ... *Sunday*. Weitere Möglichkeiten der Zeitangabe liefert das **man**-Kommando (vgl. Abschnitt 7.4).

Wird keine Datei mit **-f at_datei** angegeben, müssen die auszuführenden Befehle an der folgenden Eingabeaufforderung eingegeben werden. Jeder einzelne Befehl wird mit **<Return>** abgeschlossen, bis anstelle eines weiteren Befehls die Tastenkombination **<Strg>+<D>** gedrückt wird.

Das **at**-Kommando gibt eine Job-Nummer zurück, unter der die Anweisungen bearbeitet werden. Zu beachten ist, dass bei Angabe von einzulesenden Programm-Dateien diese danach nicht mehr für das eingegebene **at**-Kommando verändert werden können, da das **at**-Kommando diese vollständig einliest.

Als Beispiel soll heute um 23 Uhr das Programm **test1** gestartet und die Ergebnisse in die Datei **test.out** geleitet werden.

```
otto@l1201:~> at 23:00
at> cd /home/otto/programme
at> ./test1 > test.out
at> <EOT>
job 3 at 2006-05-16 23:00
```

Das **<EOT>** ist die Meldung der Shell auf das eingegebene **<Strg>+<D>**, mit dem die Eingabe abgeschlossen wurde.

Mit **at -l** kann der Job angezeigt, mit **at -r** kann er gelöscht werden:

```
otto@l1201:~> at -l
3      2006-05-16 23:00 a otto
otto@l1201:~> at -r 3
```

Hinweis: Für das Starten von **at**-Jobs auf einem Rechner ist ein stets im Hintergrund laufender Prozess, der **at**-Dämon **atd** zuständig. Um einen reibungslosen Übungsbetrieb in den Rechnerräumen zu gewährleisten, sind diese Dämonen nur auf den Praktikumsrechnern des PI-Pools in G.16.13 aktiviert.

5.4 Starten von Programmen mit nohup und nice

Wird ein Programm mit **nohup** (bedeutet „No HangUp“) gestartet, kann man sich ausloggen, ohne dass das Programm damit automatisch beendet wird:

```
nohup programm [ optionen ] &
```

Startet man ein Programm mit **nice**, kann dem Programm eine geringere Priorität gegeben werden, falls andere Benutzer an dem Rechner arbeiten wollen.

```
nice [ -level ] programm [ optionen ]
```

Möglich sind **nice**-Level von 1 bis 19, wobei ein höherer Wert eine größere Verringerung der Prozesspriorität bedeutet. Eine höhere Priorität darf von normalen Benutzern nicht vergeben werden. Arbeiten keine anderen Benutzer auf dem Rechner, wirkt sich eine geringere Prozesspriorität nicht aus.

Soll ein Programm tagsüber rechnen, sollte auf jeden Fall mit **nice -19** die Prozesspriorität herabgesetzt werden!

nohup und **nice** können auch kombiniert werden:

```
nohup nice -19 programm [ optionen ] &
```

Hinweis: Sprechen Sie bitte entsprechende Programmläufe stets mit den zuständigen unter 1.6 genannten Ansprechpartnern ab. Aktive Prozesse von Benutzern, die nicht mehr an den entsprechenden Rechner eingeloggt sind, werden ansonsten von der Systemadministration ohne Vorwarnung abgebrochen.

6 Grundlegende UNIX-Kommandos

6.1 Dateien und Verzeichnisse

6.1.1 Abkürzungen und Wildcards

Bei der Angabe von Verzeichnis- und Dateinamen sind folgende Platzhalter und Abkürzungen möglich:

- * Eine beliebige Zeichenkette
- ? Ein beliebiges Zeichen
- ~ Der Name des HOME-Verzeichnisses (nur C-Shell)
- . Das aktuelle Arbeitsverzeichnis
- .. Das übergeordnete Verzeichnis

6.1.2 Arbeitsverzeichnis bestimmen und wechseln

In jeder Shell ist ein Verzeichnis als Arbeitsverzeichnis (CWD - Current Work Directory) festgelegt. Beim Login oder in einem neuen Shell-Fenster ist dies zunächst das HOME-Verzeichnis. Alle Verzeichnis- und Dateinamen mit relativem Pfadnamen (also ohne "/" am Anfang) beziehen sich auf dieses Verzeichnis. Zum Anzeigen des Arbeitsverzeichnisses dient das Kommando **pwd** (**print working directory**), ein Wechsel des Arbeitsverzeichnisses erfolgt mit **cd** (**change directory**):

```
cd [ neues_verzeichnis ]
pwd
```

Dabei bedeutet ein Aufruf von **cd** ohne Argumente den Wechsel ins HOME-Verzeichnis. Bei der Angabe des neuen Verzeichnisses sind die oben aufgeführten Platzhalter erlaubt

```
otto@l14:~/daten/neu> pwd
/home/otto/daten/neu
otto@l14:~/daten/neu> cd ..
otto@l14:~/daten> pwd
/home/otto/daten
otto@l14:~/daten> cd
otto@l14:~> pwd
/home/otto
otto@l14:~> cd d*n/n*
otto@l14:~/daten/neu> pwd
/home/otto/daten/neu
```

6.1.3 Verzeichnisinhalt anzeigen

Den Inhalt eines Verzeichnisses kann man sich mit **ls** (**list directory**) anzeigen lassen. Die wichtigsten Optionen sind:

```
ls [ -adFIR ] [ name ... name ]
```

Dabei bedeuten die Optionen:

- a Versteckte Dateien (beginnen mit einem Punkt) auch anzeigen
- d Bei Verzeichnissen nur Namen, nicht Inhalt anzeigen
- F Kennzeichnung von Verzeichnissen mit "/", Verweisen mit "@" und Programmen mit "**"
- l Lange Anzeige: Zugriffsrechte, Größe, Eigentümer, Gruppe und Datum
- R Inhalt von allen Unterverzeichnissen rekursiv ausgeben

Ist das angegebene Datei-Argument eine Dateiname, wird dieser angezeigt, ist es ein Verzeichnisname, wird der Inhalt des Verzeichnisses angezeigt.

```
otto@l14:~> ls
bin          Documents  myprg          reg426.txt    test.txt      VP-UML Projects
daten        IT_home    OpenOffice.org1.1 reg473.txt    tmp           workspace
Desktop      Mail       public_html    rup           vdmprofile
```

```
otto@l14:~> ls -F
bin/          IT_home/          public_html/    test.txt        workspace/
daten/        Mail/             reg426.txt     tmp/
Desktop/      myprg*           reg473.txt     vdmprofile
Documents/    OpenOffice.org1.1/ rup/           VP-UML Projects/

otto@l14:~> ls tmp daten
daten:
neu test1

tmp:
Script.pdf
```

Versteckte Dateien (und Verzeichnisse) beginnen mit einem Punkt und werden normalerweise von `ls` nicht angezeigt. Diese Dateien befinden sich meist im HOME-Verzeichnis eines Benutzers und enthalten in der Regel benutzerspezifische Einstellungen für das System, die Shell, Oberflächen und Programme. Beispiele für solche Dateien sind `.cshrc` (Abschnitt 12) und `.forward` (Abschnitt 10.3.5).

6.1.4 Kopieren und Umbenennen von Dateien und Verzeichnissen

Kopieren von Dateien und Verzeichnissen geschieht mit dem Kommando `cp` (copy):

```
cp [ -i ] datei Neuer_Name
cp [ -i ] datei [ datei ... ] verzeichnis
cp -R verzeichnis verzeichnis
```

Im ersten Fall wird eine Kopie der Datei unter einem zweiten Namen erstellt. Im zweiten Fall werden Kopien aller angegebenen Dateien unter denselben Namen in einem anderen Verzeichnis erstellt. Im dritten Fall wird ein komplettes Verzeichnis rekursiv im angegebenen Verzeichnis dupliziert.

Umbenennen und Verschieben von Dateien/Verzeichnissen erfolgt mit dem Kommando `mv` (move):

```
mv [ -i ] datei Neuer_Name
mv [ -i ] datei [ datei ... ] verzeichnis
```

Mit der ersten Variante kann einer Datei oder einem Verzeichnis ein neuer Name gegeben werden, mit der zweiten Variante werden Dateien und Verzeichnisse an einen anderen Ort verschoben. Die Option `-i` bewirkt, dass nachgefragt wird, bevor existierende Dateien überschrieben werden.

6.1.5 Erstellen und Löschen von Dateien, Verzeichnissen

Neue Verzeichnisse werden mit `mkdir` (make directory) erstellt:

```
mkdir [ -p ] verzeichnis [ verzeichnis ... ]
```

Die Option `-p` bewirkt, dass ein ganzer Verzeichnisbaum rekursiv erstellt wird:

```
otto@l14:~> mkdir -p neu/neuer/am_neuesten
```

Mit `rmdir` (remove directory) werden Verzeichnisse gelöscht:

```
rmdir [ -p ] verzeichnis [ verzeichnis ... ]
```

Hier bewirkt `-p` analog, dass übergeordnete Verzeichnisse gelöscht werden, falls sie nach Entfernung der Unterverzeichnisse leer sind.

Zum Löschen von Dateien dient das Kommando `rm` (remove):

```
rm [ -i ] datei [ datei ... ]
rm -r [ -i ] verzeichnis [ verzeichnis ... ] [ datei ... ]
```

Bei Angabe der Option `-i` muss das Löschen jeder einzelnen Datei bestätigt werden. Die Option `-r` ermöglicht das rekursive Löschen ganzer Unterverzeichnisse (**Vorsicht!**).

Mit dem Kommando `touch` kann eine neue (leere) Datei erstellt werden:

touch *datei* [*datei* ...]

Existiert eine angegebene Datei bereits, bekommt sie die aktuelle Zeit als Zeitpunkt der letzten Änderung. Weitere Optionen zeigt die Online-Hilfe zu **touch**.

6.1.6 Links

Ein Link ist ein Verweis auf eine Datei oder einen Datei- bzw. Verzeichnisnamen. Man unterscheidet sogenannte Hard- und Soft-Links. Ein Hard-Link verweist auf eine existierende Datei (genauer: Auf deren Position im Dateisystem) und ist ein weiterer Name für diese Datei. Ein Soft-Link ist ein Verweis auf einen Datei- oder Verzeichnisnamen, wobei das Ziel des Verweises nicht notwendigerweise existieren muss. Gespeichert wird hier nur der Name des Zieles, auf das verwiesen wird.

Das Erzeugen eines Links geschieht mit **ln** (link):

ln *Existierende_datei Neuer_Name*

ln -s *datei Neuer_Name*

ln -s *verzeichnis Neuer_Name*

Bei Angabe der Option **-s** wird ein Soft-Link erzeugt. Das Löschen von Links geschieht mit dem Kommando **rm** wie bei einer „normalen“ Datei. Die Zugriffsrechte der Datei (des Verzeichnisses), auf die der Link verweist, werden auf den Link übertragen. Ist die Datei ausführbar, kann sie auch über den Link als Kommando gestartet werden.

Beim Anlegen eines Soft-Links wird als Verweis genau der Name eingetragen, der bei **ln** angegeben wurde. Dies führt zu einem beliebigen Missverständnis bei der Nutzung von **ln**:

```
otto@l10:~> cd /usr/bin
otto@l10:~> ln -s mailx ~/bin/MeinMail
otto@l10:~> cd ~/bin
otto@l10:~/bin> ls -l MeinMail
lrwxrwxrwx  1 otto  users    5 2006-05-17 12:18 MeinMail -> mailx
otto@l10:~/bin> ./MeinMail
-bash: ./MeinMail: Datei oder Verzeichnis nicht gefunden
```

Bei der Ausgabe von **ls -l** zeigt das **l** am Anfang der Zeile an, dass es sich um einen Soft-Link handelt. Dieser Link zeigt auf die Datei **mailx**, die nun im selben Verzeichnis wie **MeinMail** gesucht wird. Dort kann sie aber nicht gefunden werden. Richtig ist das Anlegen eines solchen Links mit:

```
otto@l10:~> ln -s /usr/bin/mailx ~/bin/MeinMail
otto@l10:~> cd ~/bin
otto@l10:~/bin> ls -l MeinMail
lrwxrwxrwx  1 otto  users   14 2006-05-17 12:22 MeinMail -> /usr/bin/mailx
```

Das Löschen eines Links erfolgt wie gehabt mit **rm**.

6.1.7 Dateiinhalte anzeigen

Mit **cat** (concatenate) wird der Inhalt von Dateien angezeigt; mit **more** und **less** wird ebenfalls der Inhalt von Dateien ausgegeben, wobei die Ausgabe (abhängig von der Fenstergröße) seitenweise dargestellt wird, so dass im Text geblättert werden kann.

cat *datei* [*datei* ...]

more *datei* [*datei* ...]

less *datei* [*datei* ...]

Die Steuerung der Ausgabe bei **more** und **less** erfolgt durch folgende Tasten:

<Leertaste>	eine Bildschirmseite weiter
<Return>	Eine Zeile weiter
b	Eine Bildschirmseite zurück
q	Beenden
:n	Nächste Datei anzeigen
<i>/text</i>	Nach <i>text</i> suchen

6.1.8 Text in Dateien suchen

Das Kommando **grep** sucht in Dateien nach Text:

```
grep [ -cilv ] text datei [ datei ... ]
```

Besteht der Suchtext aus mehreren Worten, ist er in doppelte Anführungszeichen einzuschließen. Die Zeilen, die den Suchtext enthalten, werden ausgegeben; falls in mehreren Dateien gesucht wurde, werden die Dateinamen mit ausgegeben. Die Optionen bedeuten:

- c** Nur Anzahl der Zeilen, die *text* enthalten, ausgeben
- i** Groß-/Kleinschreibung ignorieren
- l** Nur Namen der Dateien ausgeben, die *text* enthalten
- v** Nur Zeilen ausgeben, die *text* nicht enthalten

Es gibt Varianten von **grep**, die erweiterte Suchmöglichkeiten bieten (**egrep**, **fgrep**). Mehr Informationen dazu bietet die Online-Hilfe.

6.1.9 Zugriffsrechte ändern

Um die Zugriffsrechte an einer Datei oder einem Verzeichnis zu ändern, wird das Kommando **chmod** (**change mode**) benutzt:

```
chmod modus datei [ datei ... ] [ verzeichnis ... ]  
chmod [ -R ] modus verzeichnis [ verzeichnis ... ]
```

Die Angabe von *modus* erfolgt ohne Leerzeichen und hat dabei folgendes Format:

```
[ wer ] operator rechte [ , wer operator rechte ... ]
```

wobei die einzelnen Komponenten folgende Werte annehmen können:

<i>wer</i>		<i>operator</i>		<i>rechte</i>
u	Inhaber (User)	+	Recht setzen	r Lesen
g	Gruppe (Group)	-	Recht löschen	w Schreiben
o	Der Rest der Welt (Other)			x Ausführen
a	Alle (All)			

Es gibt noch eine zweite Möglichkeit, die Zugriffsrechte anzugeben, die im Online-Handbuch zu **chmod** (vgl. Abschnitt 7.4) beschrieben ist.

Um also die Datei **doit** für sich selbst und die Gruppe ausführbar zu machen sowie für alle, die nicht der Gruppe angehören, unlesbar zu machen, gibt man ein:

```
otto@l1104:~> chmod ug+x,o-r doit
```

Mit **-R** können rekursiv einem Verzeichnis mit allen Dateien und Unterverzeichnissen neue Rechte vergeben werden.

6.1.10 Dateien suchen

Mit Hilfe des **find**-Kommandos können Verzeichniseinträge (Dateien und Verzeichnisse) nach verschiedensten Kriterien gesucht werden:

```
find Pfad-Liste Bedingungs-Liste
```

Pfad-Liste ist eine Liste von Verzeichnisnamen, die nicht leer sein darf. Die angegebenen Verzeichnisse dienen als Startpunkte einer rekursiven Suche.

Bedingungs-Liste ist eine nicht-leere Liste von Bedingungen, die durch logisches UND verknüpft werden. **find** führt die angegebene Aktion für alle Dateien und Verzeichnisse aus, auf die alle angegebenen Bedingungen zutreffen. Ein Auszug mit den wichtigsten der erlaubten Bedingungen:

-name *Name* Wahr für alle Einträge mit Namen *Name*
-type f Wahr für alle einfachen Dateien
-type d Wahr für alle Verzeichnisse
-user *Name* Wahr für alle Einträge, die Benutzer *Name* gehören
-perm [-]*modus* Wahr für alle Einträge, die die angegebenen Zugriffsrechte haben
-print Gibt alle gefundenen Einträge aus (Standard, falls keine Aktion angegeben wird)
-exec *Kommando* ; Führt *Kommando* für gefundene Einträge aus
-ok *Kommando* ; Führt *Kommando* für gefundene Einträge mit Bestätigung aus

Im folgenden Beispiel wird ausgehend vom Verzeichnis `/usr/local/sw` nach allen Verzeichnissen mit Namen, die mit `font` beginnen, gesucht:

```
otto@l110:~> find /usr/local/sw -type d -name font* -print
/usr/local/sw/matlab6/sys/ghostscript/fonts
/usr/local/sw/Mathematica-4.2/SystemFiles/IncludeFiles/TeX/texmf/fonts
/usr/local/sw/Mathematica-4.2/SystemFiles/Java/Linux/lib/fonts
/usr/local/sw/Together6.2/jdk/jre/lib/fonts
/usr/local/sw/Together6.2/lib/font
/usr/local/sw/VP_Suite1.1/jre/lib/fonts
/usr/local/sw/JBuilderX/jdk1.4/jre/lib/fonts
/usr/local/sw/VP_Suite2.2/jre/lib/fonts
```

6.1.11 Informationen über Plattenbelegung

Der Quota-Mechanismus ermöglicht es, die Anzahl und Größe der Dateien eines Benutzers zu beschränken. Dabei gibt es für belegten Plattenplatz und Dateianzahl jeweils zwei Werte: Das **Softlimit** ist eine Schranke, die kurzfristig überschritten werden kann, das **Hardlimit** kann keinesfalls (auch nicht kurzfristig) überschritten werden. Falls das Softlimit überschritten wird, hat der Benutzer eine Woche Zeit, den zuviel belegten Platz freizugeben. Ist diese Frist überschritten oder das Hardlimit erreicht, kann der Benutzer keine Dateien mehr anlegen.

Hinweis: Der Quota-Mechanismus ist auf den Ausbildungsrechnern im Moment nicht aktiviert. Alle Benutzer sind daher verpflichtet, ihren jeweils genutzten Plattenspeicher selbst zu kontrollieren!

Auf den Ausbildungsrechnern bedeutet das Erreichen des Hardlimits bzw. Überschreiten des Softlimits um mehr als eine Woche, dass man sich nicht mehr normal einloggen kann, da das Desktop-Login sofort Status-Dateien anlegen will, was dann nicht mehr möglich ist. Das Login schlägt fehl, wobei der Benutzer ggf. keine Fehlermeldung zu sehen bekommt. Nun ist es erforderlich, sich per Kommandozeilen-Login (vgl. Abschnitt 3.2) anzumelden und durch Löschen die Anzahl der Dateien zu verringern bzw. den verwendeten Plattenplatz zu reduzieren.

Tip: Die Cache-Verzeichnisse der Webbrowser (Firefox, Opera, ...) sind Kandidaten für viele Dateien, die bedenkenlos gelöscht werden können! Die maximale Größe dieser Zwischenspeicher ist konfigurierbar.

Informationen über den belegten Plattenplatz und die eigene Quota bieten die Kommandos **quota** und **du** (**d**isk **u**sage):

quota Gibt Meldungen über erreichte Limits aus
quota -v Gibt Statistik über belegten Platz aus

```
otto@l201:~> quota -v otto
Disk quotas for user otto (uid 3008):
  Filesystem blocks quota limit grace files quota limit grace
  /home/otto 1916428 1800000 2000000 3days 31031 90000 100000
```

otto darf also 1,8 Gigabyte (kurzfristig 2 Gigabyte) Plattenplatz benutzen und 90000 (kurzfristig 100000) Dateien bzw. Verzeichnisse anlegen. Das Softlimit für den Plattenplatz hat er um 116428 Kilobytes überschritten, die er innerhalb von etwa drei Tagen löschen muss. Diesen Platz belegt er mit 31031 Dateien.

du [**-sk**] [*verzeichnis*] ... [*verzeichnis*]

du gibt rekursiv den durch ein Verzeichnis und durch alle Unterverzeichnis belegten Plattenplatz aus. Die Option **-k** bewirkt die Ausgabe in Kilobyte statt 512-Byte-Blöcken, **-h** erzeugt eine gut lesbare Ausgabe mit Angabe der Einheiten, **-s** unterbindet die Ausgabe der Unterverzeichnis-Größen. Bei Angabe von *verzeichnis* wird dieses anstelle des aktuellen Verzeichnisses durchsucht.

```
graupe@l1101:~> du -h
```

```

8,0K    ./xemacs
8,0K    ./Documents
8,0K    ./public_html
5,2M    ./Programme/C
4,0K    ./Programme/C++
4,0K    ./Programme/Pascal
5,2M    ./Programme
5,4M    .

graupe@l1101:~> du -ks Programme/
5312    Programme/

```

6.2 Dateitransfer zwischen verschiedenen Rechner übers Netz

Um Dateien von einem Rechner auf einen anderen zu kopieren, verwendet man das Secure Copy Kommando **scp**:

```
scp [username@hostname1:]datei1 [...] [username@hostname2:]ziel
```

Wird kein Benutzername *username@* angegeben, so wird derselbe Benutzername verwendet, unter dem man auf der lokalen Maschine eingeloggt ist. Läßt man zusätzlich den Hostnamen weg, so wird die lokale Maschine selbst angesprochen. Falls es sich beim Kopieren mit **scp** bei *ziel* um ein bereits existierendes Verzeichnis handelt, so werden die zu kopierenden Dateien in dieses kopiert, ansonsten wird *ziel* als neuer Dateiname verwendet.

Durch das Secure Copy Kommando **scp** werden die angegebenen Dateien über einen sicheren (verschlüsselten) Kanal vom Rechner *hostname1* auf den Rechner *hostname2* kopiert:

– **scp Ergebnis user@l201.studs.math.uni-wuppertal.de:Daten**

Kopiert die Datei *Ergebnis* aus dem aktuellen Verzeichnis (CWD) des lokalen Rechner unter dem neuen Dateinamen *Daten* in das HOME-Verzeichnis des Benutzers *user* auf dem PI-Rechner l201 (und damit auf dem gesamten PI-Cluster) bzw. unter dem Namen *Ergebnis* in das bereits bestehende Unterverzeichnis *Daten* des HOME-Verzeichnisses.

– **scp user@l101.studs.math.uni-wuppertal.de:Daten/Ergebnis Ergebnis2**

Kopiert die Datei *Ergebnis* im Unterverzeichnis *Daten* des HOME-Verzeichnisses des Benutzers *user* auf dem IT-Rechner l101 (und damit auf dem gesamten IT-Cluster) unter dem neuen Namen *Ergebnis2* in das lokale aktuelle Verzeichnis oder unter dem Namen *Ergebnis* in das bereits existente Unterverzeichnis *Ergebnis2*.

Hinweis:

Um per Secure Copy oder mit dem Secure Shell Client von einem der Ausbildungsrechner auf ältere Systeme, z. B. Sun-Workstations zuzugreifen, ist es erforderlich, die Verwendung des SSH-Protokolls Version 1 durch **scp -1** bzw. **ssh -1** zu erzwingen.

Für einen remote-Zugriff z. B. vom eigenen Heim-PC auf einen der Ausbildungsrechner muss der ssh-Client des zugreifenden Rechners das SSH-Protokoll Ver. 2 sowie die Authentifikation gegenüber dem LDAP-Server („response authentication“) beherrschen. Verwenden Sie (z. B. bei Verwendung der SSH Secure Shell unter Microsoft Windows) als Authentifikationsmethode „keyboard interactive“ und geben Sie Ihr Passwort als „authentication response“ ein.

Bitte beachten Sie:

Beim entfernten Zugriff auf Rechner im Internet (remote-Login, Dateitransfer, ...) ist sicheren Verbindungen stets der Vorzug gegenüber unsicheren Datenübertragungen zu geben. Verwenden Sie also nach Möglichkeit

- den Secure Shell Client **ssh** anstelle von **telnet**, **rlogin**, **rsh**, ...
- das Secure Copy Kommando **scp** (bzw. **scp1** für größere Datenmengen) anstelle von **rcp**
- das Secure File Transfer Program **sftp** anstelle von **ftp**
- sichere E-Mail-Zugänge (SSL/TLS) für das Abrufen (POP3/IMAP) und Versenden (SMTP)
- verschlüsselte WWW-Zugänge (https, shtml) für den Zugriff auf Webseiten mit Formularen, in denen Sie schutzwürdige Daten wie Passwörter oder Kreditkartennummern eintragen wollen.

6.3 Dateiformate und passende Programme

Zwischen Dateien und Programmen besteht eine Abhängigkeit, die besagt, welche Daten ein Programm benötigt und mit welchen Programmen eine Datei bearbeitet werden kann. Um diese Abhängigkeit zu nutzen, gibt es Konventionen, wie der Name einer Daten-Datei aussieht. Das übliche Kriterium für den Typ (also das Format) einer Datei ist das Datei-Suffix, die Endung des Dateinamens nach einem Punkt.

Programm-Quelltexte werden von einem Compiler (Übersetzer) gelesen, der versucht, aus den darin beschriebenen Anweisungen ein ausführbares Programm zu erstellen.

Suffix	Datei-Typ	Programm	Beschreibung
.c	C-Quelltext	cc, gcc	C-Compiler
.cc	C++-Quelltext	CC, g++	C++-Compiler
.f	Fortran-77-Quelltext	f77, g77	Fortran-77-Compiler
.f90	Fortran-90-Quelltext	f90	Fortran-90-Compiler
.p	Pascal-Quelltext	pc, gpc, pxsc	Pascal-Compiler
.java	Java-Quelltext	javac	Java-Compiler

T_EX ist ein Satzsystem zur Erstellung von Dokumenten; **Postscript** und **PDF** sind Seitenbeschreibungssprachen.

Suffix	Datei-Typ	Programm	Beschreibung
.tex	T _E X-Eingabedatei	tex, latex, pdflatex	T _E X-Compiler
.dvi	T _E X-Ausgabe	xdvi dvips dvipdf	T _E X-Ausgabe ansehen T _E X-Ausgabe in Postscript wandeln T _E X-Ausgabe in PDF wandeln
.ps	Postscript-Dokument	ghostview, pageview, gv	Postscript-Betrachter
.eps	Postscript-Bild		wie .ps
.pdf	PDF-Dokument	okular, epiphany	PDF-Betrachter

Archive dienen der Datensicherung und der Reduzierung der Größe selten benötigter Dateien. Dabei werden mehrere Dateien oder ganze Verzeichnissebäume zusammengefasst und/oder mit Hilfe geeigneter Programme so komprimiert, dass die Dateigröße ohne Informationsverlust deutlich verringert werden kann. Mit den entsprechenden Programmen können die Originaldaten wiederhergestellt werden.

Suffix	Datei-Typ	Programm
.tar	Tape-Archiv	tar
.gz	mit gzip komprimierte Datei	gunzip
.tgz	Kurzform für .tar.gz	tar/gunzip
.Z	mit compress komprimierte Datei	uncompress
.zip	zip -/pkzip-Archiv	unzip

Graphiken und Bilder werden in vielen unterschiedlichen Formaten (teilweise (verlustfrei oder verlustbehaftet) komprimiert) gespeichert. Gebräuchliche Graphik-Formate sind z. B. .gif, .tiff, .jpg, .bmp, .tga und .png. Auch Postscript-Dateien (.ps und .eps) können als Graphiken betrachtet werden. Zur Betrachtung und Bearbeitung solcher Bilder sind Programme wie **xv**, **gimp** und **KView** installiert.

Hinweis:

Viele Programme erwarten, dass die Namen der ihnen zugeordneten Dateien über das korrekte Suffix verfügen. Andererseits gibt es auch Applikationen, etwa der Dateimanager unter KDE, die den Typ einer Datei nicht nur am Datei-Suffix erkennen, sondern auch am konkreten Datei-Inhalt. Bei Shell-Scripts, also Textdateien mit Befehlsfolgen für spezielle Kommandointerprete, wird die entsprechende zu startende Shell durch die erste Zeile festgelegt. Beispielsweise beginnt ein C-Shell-Script mit der Zeile:

```
#!/bin/csh
```

Ein BASH-Script erkennt man an folgender ersten Zeile:

```
#!/bin/bash
```

7 Hilfen

7.1 CIP-Hilfesystem

Auf dem CIP-Pool gibt es ein in der Fachgruppe Mathematik und Informatik entwickeltes Hilfesystem im HTML-Format, in dem die wichtigsten Informationen zum Umgang mit den Rechnern zusammengetragen sind. Das System wird laufend aktualisiert und ergänzt. Durch das Kommando

firefox <https://www.math.uni-wuppertal.de/de/cip-hilfesystem/>

wird der Web-Browser Firefox gestartet, und man gelangt auf die Startseite des Hilfesystems, von wo aus weitere Links zu den einzelnen Themen verweisen. Das CIP-Hilfesystem ist natürlich auch von den anderen Rechnerpools aus erreichbar, allerdings muss eventuell ein anderer Internet-Browser gestartet werden.

7.2 SuSE-Hilfe / KDE-Hilfe

Auf den Ausbildungsrechnern steht mit SuSE-Hilfe ein umfassendes und komfortables Hilfesystem auf HTML-Basis zur Verfügung. Durch einen Klick auf das Rettungsring-Symbol in der Menü-Leiste am unteren Bildschirmrand oder durch Auswahl des entsprechenden Eintrags im Menü „Programme starten“ wird der Web-Browser Konqueror gestartet und die Startseite des Hilfesystems angezeigt. Neben einer Suchfunktion gibt es hier eine große Anzahl von Links zu weiteren Informationsseiten, darunter eine Tour, bei der die wichtigsten Programme vorgestellt werden, sowie eine Support-Datenbank mit Antworten auf viele Fragen zum System.

Wenn man im Hintergrundmenü unter **Hilfe** auf den Eintrag **Handbuch zu KDesktop** klickt, wird das KHelpCenter gestartet, in dem eine Einführung in KDE und die wichtigsten Informationen zum Umgang mit der graphischen Oberfläche zu finden sind. Man gelangt auf diesem Weg auch zu Online-Handbüchern der verschiedenen Programme sowie zu Unix-Handbuchseiten mit einer Liste der Kommandozeilenbefehle.

7.3 Hilfe-Optionen der UNIX-Kommandos

Die meisten UNIX-Kommandos liefern bei Eingabe von *command --help* eine Reihe von Hilfen zu ihrer Anwendung und den weiteren Optionen des Kommandos:

```
/home/fpf> pwd --help
Aufruf: pwd [OPTION]
Den vollständigen Dateinamen des aktuellen Verzeichnisses ausgeben.

--help      diese Hilfe anzeigen und beenden.
--version   Versionsinformation anzeigen und beenden.
```

7.4 Online-Handbuch

Mit Hilfe des Kommandos **man** kann man sich Informationen zur Syntax und zum Verhalten von UNIX-Kommandos anzeigen lassen:

man [*-s* *sektion*] [*-M* *pfad*] *name* Zeigt die Onlinehilfe zum Kommando oder zur Funktion *name*
whatis *name* Gibt eine einzeilige Beschreibung von *name*

Die einzeiligen Kurzbeschreibungen, die mit dem Kommando **whatis** abrufbar sind, lassen sich wiederum mit dem Kommando **apropos** nach Schlüsselwörtern durchsuchen:

```
otto@l110:~> whatis g77
g77 (1)                    - GNU project Fortran 77 compiler

otto@l110:~> apropos Fortran
f2c (1)                    - Convert Fortran 77 to C or C++ . . .
g77 (1)                    - GNU project Fortran 77 compiler
libfpvm (3)                - PVM C and Fortran programming libraries
libpvm (3)                - PVM C and Fortran programming libraries
```

Um sich beispielsweise die Syntax des **ls**-Kommandos anzeigen zu lassen, gibt man **man ls** ein und bekommt die Onlinehilfe zu **ls**, wobei mit der Leertaste seitenweise durch die Ausgabe geblättert werden kann.

Die Manual-Seiten des Online-Handbuchs sind in Sektionen unterteilt, wobei bei Vorhandensein mehrerer Manual-Seiten zu einem Namen in verschiedenen Sektionen mit der Option **-s** die gewünschte Sektion angegeben werden kann. Die wichtigsten Sektionen sind:

1	Benutzer-Kommandos
1m	System-Kommandos
2	System-Aufrufe
3	C-Bibliotheks-Funktionen
3f	Fortran-Bibliotheks-Funktionen
4	Dateiformate

Durch Eingabe der Kommandos **man man**, **man whatis** und **man apropos** kann man sich schließlich die Manual-Seiten der drei Kommandos mit der Erklärung der möglichen Optionen anzeigen lassen. Oder die entsprechende einzeilige Kurzbeschreibung durch **whatis**:

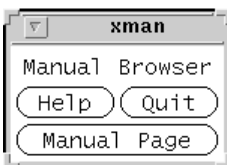
```
otto@l110:~> whatis whatis
whatis (1) - durchsucht die Indexdatenbank nach Kurzbeschreibungen
```

Beim Aufruf des **man**-, **apropos**- und **whatis**-Kommandos werden die in der Umgebungsvariablen **MANPATH** enthaltenen Verzeichnisse durchsucht. Bei der Angabe der Option **-M** wird stattdessen das angegebene Verzeichnis durchsucht. Auf den Ausbildungsrechnern ist die Umgebungsvariable **MANPATH** so voreingestellt, dass die Online-Hilfen zu den wichtigsten Kommandos automatisch gefunden werden.

Zusätzlich zur rein textbasierten Version gibt es mit **xman** ein graphisches Frontend für das **man**-Kommando, das auf allen Ausbildungsrechnern unter den graphischen Oberflächen zur Verfügung steht.

Das Programm wird durch Eingabe von **xman** in einem Kommando-Fenster oder über das Hilfe-Menü gestartet.

Zunächst sieht man den **xman**-Browser:



Durch Klicken auf **Manual Page** erscheint ein weiteres Fenster, in dem man die Sektionen (Kapitel) der Online-Hilfe und dann konkrete Manual-Seiten auswählt.

```
Manual Page
Options Sections The current manual page is: cc.

User Commands cc(1)

NAME
cc - C compiler

SYNOPSIS
cc [ -# ] [ -### ] [ -Aname [(tokens) ] ]
[ -B [static|dynamic] ] [ -C ] [ -c ] [ -Dname [=tokens] ]
[ -d [yln] ] [ -dalign ] [ -E ] [ -erroff=t ] [ -errtags=a ]
[ -fast ] [ -fd ] [ -flags ] [ -fnonstd ] [ -fns=yes|no ]
[ -fprecision=p ] [ -frounder ] [ -fsimple=h ] [ -fsingle
]
[ -fstore ] [ -ftrap=t ] [ -G ] [ -g ] [ -H ]
[ -h name ] [ -Idir ] [ -i ] [ -keepmp ] [ -KPIC ]
[ -kpic ] [ -ldir ] [ -lname ] [ -mc ] [ -misalign ]
[ -misalign2 ] [ -mr ] [ -mr,string ] [ -mt ] [ -native ]
[ -nofstore ] [ -noqueue ] [ -O ] [ -o filename ] [ -P ]
[ -p ] [ -Q [yln] ] [ -qp ] [ -Rdir[:dir] ] [ -S ] [ -s ]
[ -uname ] [ -v ] [ -v ] [ -Wc,arg ] [ -w ]
[ -X [a lc is lt ] ] [ -x386 ] [ -x486 ] [ -xa ] [ -xarch=a ]
[ -xautopar ] [ -xcache=c ]
[ -xCC ] [ -xcg89 ] [ -xcg92 ] [ -xchip=c ] [ -xcode=v ]
[ -xcrossfile=[n] ] [ -xdepend ]
[ -xe ] [ -xexplicitpar ] [ -xF ] [ -xhelp=f ]
[ -xildoff ] [ -xildon ] [ -xinline=[f1,...,fn] ]
[ -xlibmfee ] [ -xlibmil ] [ -xlic_lib=1 ] [ -xlicinfo ]
[ -xloopinfo ]
[ -xM ] [ -xM1 ] [ -xMerge ] [ -xnolib ] [ -xnolibmil ]
[ -xO[1|2|3|4|5] ] [ -xP ] [ -xparallel ] [ -xpentium ]
[ -xpg ] [ -xprefetch=yesno ] [ -xprofile=p ]
[ -xreduction ] [ -xregs=r ] [ -xrestrict=F ] [ -xs ]
[ -xsafe=mem ] [ -xsb ] [ -xsbfast ] [ -xsfpconst ]
[ -xspace ] [ -xstrconst ] [ -xtarget=t ] [ -xtemp=dir ]
[ -xtime ] [ -xtransition ] [ -xunroll=n ] [ -xvpara ]
[ -Yc,dir ] [ -YA,dir ] [ -YI,dir ] [ -YP,dir ] [ -YS,dir ]
[ -Z1 ] [ -Z1p ]

DESCRIPTION
```

7.5 Sonstige Dokumentationen

Ein großer Teil der installierten Software ist zusätzlich in Form von **HTML**-, **DVI**-, **PS**- und **PDF**-Dateien dokumentiert. Auf den Ausbildungsrechnern befindet sich die Dokumentation in Unterverzeichnissen von **/usr/share/doc**. Als Betrachtungsprogramme verwendet man üblicherweise **xdvi** für **DVI**-, **ghostview** oder **gv** für **Postscript**- und **acroread** für **PDF**-Dateien, für Dokumentationen im **HTML**-Format einen der installierten Webbrowser.

8 Was tun bei Problemen

8.1 Login schlägt fehl

Wenn auf den Ausbildungsrechnern das normale Desktop-Login fehlschlägt, kann es sein, dass sich die Konfigurationsdateien für die ausgewählte graphische Oberfläche, die im HOME-Verzeichnis des Benutzers gespeichert sind, in einem pathologischen Zustand befinden. Bei eingeschaltetem Quota-Mechanismus (vgl. Abschnitt 6.1.11) kann es auch sein, dass der Benutzer seine Quota überschritten hat. In diesem Fall hilft nur der Wechsel auf eine der Textkonsolen (vgl. Abschnitt 3.1), sich dort per Befehlszeile einzuloggen (das ist dann noch möglich) und erst einmal, wie in Abschnitt 11.2 beschrieben, aufzuräumen, oder die Konfigurationsdateien der graphischen Oberfläche zu reparieren bzw. zu löschen.

8.2 Probleme beim Programm-Aufruf

Programm wird nicht gefunden

Wenn beim Programmstart das Programm selbst nicht gefunden wird, erscheint eine Fehlermeldung der Art:

```
otto@l10:~>xdvi
-bash: xdvi: command not found
```

Vorausgesetzt, der Programmname wurde ohne „Tippfehler“ eingegeben und das gewünschte Programm existiert, wird nicht dort nach dem Programm gesucht, wo es sich befindet. Das Programm kann mit Hilfe des Kommandos **find** lokalisiert werden (vgl. Abschnitt 6.1.10). Abhilfe schafft entweder das Aufnehmen des entsprechenden Verzeichnisses in den Suchpfad (vgl. Abschnitt 12.1) oder ein Aufruf mit absolutem Pfad:

```
otto@l10:~> /usr/bin/xdvi
```

Auch eine ausführbare Datei im aktuellen Verzeichnis muß nicht automatisch gefunden werden, wenn man sie durch Eingabe des Dateinamens zu starten versucht. Ist das aktuellen Verzeichnis (siehe 6.1.1) nicht im Suchpfad enthalten, so muß der Pfad spezifiziert werden, um das Programm zu starten. Verschiedene Varianten sind z. B.:

```
otto@l10:~/bin> ./myprg
otto@l10:~/bin> /home/otto/bin/myprg
otto@l10:~/bin> ~/bin/myprg
```

Wichtig ist auch, dass der Suchpfad eine Reihenfolge für die Suche nach ausführbaren Dateien festlegt. Höchste Priorität haben stets die internen Kommandos der entsprechenden Kommando-Shell, anschließend werden dann die Verzeichnisse in der Reihenfolge durchsucht, die im Suchpfad festgelegt ist. Steht dabei das aktuelle Verzeichnis (•) an letzter Stelle, werden eigene Programme, die denselben Namen haben, wie andere, vorher im Suchpfad befindliche Dateien, nicht ausgeführt, wenn man zum Starten nur den Dateinamen eintippt. Auch in diesem Fall muß der Pfadname mit angegeben werden.

Eine Fehlermeldung der Art

```
otto@l10:~> myBASHscript
-bash: ./myBASHscript: Keine Berechtigung
```

bedeutet, dass die entsprechende Datei kein (für diesen Benutzer) ausführbares Programm ist. Bei eigenen Shell-Scripts wurde vermutlich vergessen, mit **chmod u+x** Ausführungsrechte für das Script zu vergeben.

Bibliotheken werden nicht gefunden

Fehlt beim Programmstart eine benötigte Bibliothek, wird eine Fehlermeldung folgender Art ausgegeben:

```
/home/otto> xdvi
ld.so.1: xdvi: fatal: libXaw.so.5: can't open file: errno=2
Killed (Abgebrochen)
```

Dann muss der Suchpfad für Bibliotheken angepasst werden, wie in Abschnitt 12.1 beschrieben.

8.3 Prozesse killen

Will man ein laufendes Kommando abbrechen, kann man zunächst in der Shell, in der das Kommando gestartet wurde, die Tastenkombination `<Strg>+<C>` eingeben. Bei Programmen, die nicht im Hintergrund gestartet wurden, klappt dies zumeist.

Ansonsten muss man, um laufende Prozesse abbrechen, diese zunächst identifizieren. Informationen über laufende Prozesse erhält man mit dem Kommando `ps`:

`ps [-ef]`

Ohne Argumente gibt `ps` die Prozesse aus, die von der Shell aus gestartet wurden, von der auch `ps` gestartet wurde. Die Option `-e` bewirkt die Ausgabe aller laufenden Prozesse, `-f` eine umfassende Ausgabe der Prozessinformationen.

```
otto@l110:~> ps
  PID TTY          TIME CMD
 31798 pts/0    00:00:00 bash
   1609 pts/0    00:00:00 ps

otto@l110:~> ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
otto         31798 31794  0 May17 pts/0    00:00:00 -bash
otto         1612 31798  0 13:12 pts/0    00:00:00 ps -f
```

Bei der Ausgabe von `ps` bedeutet

UID	der Name des Prozessinhabers	TTY	das den Prozess kontrollierende Terminal
PID	Die Nummer des ausgegebenen Prozesses	TIME	die vom Prozess benötigte CPU-Zeit
PPID	die Nummer des aufrufenden Prozesses	CMD	das gestartete Kommando
STIME	die Startzeit des Prozesses		

Um einen Prozess zu beenden, benutzt man das Kommando `kill`:

`kill [-signal] pid`

Dabei ist `pid` die Prozessnummer des abzubrechenden Prozesses, die man durch `ps` erhalten hat. `signal` ist das Signal, das dem Prozess übergeben werden soll. Dies kann mit Namen oder einer Signalnummer angegeben werden. Wird kein Signal angegeben, wird das Signal 15 bzw. TERM (Terminate) übergeben, das den Prozess auffordert, sich normal zu beenden. Schlägt dies fehl, sollte das Signal 9 bzw. KILL den Prozess entfernen. Zu beachten ist, dass durch `kill -9` ein Prozess beendet wird, ohne dass normalerweise durchgeführte Aufräumarbeiten (z. B. Dateien schließen) erfolgen. Deshalb soll `kill` zunächst mit dem Standardsignal versucht werden.

```
otto@l110:~> ps
  PID TTY          TIME CMD
 31798 pts/0    00:00:00 bash
   1716 pts/0    00:00:00 kwrite
   1739 pts/0    00:00:00 ps
otto@l110:~> kill 1716
[1]+  Beendet                kwrite
otto@l110:~> ps
  PID TTY          TIME CMD
 31798 pts/0    00:00:00 bash
   1744 pts/0    00:00:00 ps
```

Nach Beendigung eines Prozesses mittels `kill` erhält man die Mitteilung, ob der Prozess sich normal terminiert hat:

```
[1]+  Beendet                kwrite
```

oder per KILL-Signal entfernt wurde:

```
[1]+  Getötet                 kwrite
```

Selbstverständlich kann ein Benutzer nur solche Prozesse mit `kill` abbrechen, die ihm selbst gehören.

Eine komfortablere Methode, Prozesse unter Angabe des Namens (des gestarteten Kommandos) und nicht der Prozessnummer abbrechen, besteht in den (weitgehend äquivalenten) Kommandos


```
pkill [ -signal ] cmdname
killall [ -signal ] cmdname
```

Zur Beendigung des Web-Browsers Firefox kann man beispielsweise **pkill firefox** verwenden.

Diese Methode ist zwar einfacher, aber auch riskanter! Es besteht nämlich die Gefahr, weitere, parallel gestartete Prozesse zu beenden, die keine Probleme bereiten, etwa dann, wenn mehrere Texteditoren gestartet wurden, um Textdateien zu bearbeiten:

```
otto@l11:~> ps -f
UID          PID  PPID  C  STIME TTY          TIME CMD
otto         21805 21802  0 14:17 pts/7        00:00:00 -bash
otto         22254 21805  0 14:32 pts/7        00:00:00 kwrite test.txt
otto         22270 21805  0 14:33 pts/7        00:00:00 kwrite myprg.c
otto         22305 21805  0 14:35 pts/7        00:00:00 ps -f
otto@l11:~> pkill kwrite
[1]- Beendet          kwrite test.txt
[2]+ Beendet          kwrite myprg.c
otto@l11:~> ps -f
UID          PID  PPID  C  STIME TTY          TIME CMD
otto         21805 21802  0 14:17 pts/7        00:00:00 -bash
otto         22316 21805  0 14:35 pts/7        00:00:00 ps -f
```

8.4 remote-kill

Blockiert ein Prozess das Fenstersystem derart, dass kein Shell-Fenster mehr aktiviert werden kann, um ihn abzubrechen, muss er von einer anderen Maschine aus beendet werden. Ein Kandidat dafür ist der Web-Browser Firefox, der gelegentlich das Fenstersystem vollständig einfrieren lässt. Auf den Linux-PCs hat man nun die Möglichkeit, auf eine der virtuellen Textkonsolen zu wechseln, um sich dort einzuloggen. Der Wechsel zu den sechs Textkonsolen geschieht durch die Tastenkombinationen `<Strg>+<Alt>+<F1>`, ..., `<Strg>+<Alt>+<F6>`, zurück zur graphischen Oberfläche gelangt man durch `<Strg>+<Alt>+<F7>`. Ansonsten loggt man sich kurzfristig auf einem anderen Rechner ein. Ist kein Rechner frei, bittet man einen der anderen Benutzer, ein Shell-Fenster zu öffnen, von dem aus man sich remote auf der blockierten Maschine anmelden kann. Mit

```
ssh rechnername -l benutzername
```

loggt man sich dann auf dem Rechner ein, auf dem der Prozess gestoppt werden soll. Mit **ps -ef** und ggf. einer per Filter anschließenden Textsuche mittels **grep** (vgl. Abschnitte 5.1.2 und 6.1.8) sucht man dann die Prozessnummer des blockierenden Prozesses und bricht ihn mit **kill -9 PID** ab.

Dann beendet man das remote-Login mit **exit** und loggt sich ggf. auf dem zweiten Rechner aus.

```
otto@l10:~> ssh l11 -l otto
Password: .....
Last login: Tue May 23 14:21:54 2006 from l01.studs.math.uni-wuppertal.de
Have a lot of fun...
otto@l11:~> ps -ef | grep firefox
otto         22012 22008  2 14:21 pts/7        00:00:02 /opt/MozillaFirebird/lib/firefox-bin
otto         22014 22012  0 14:21 pts/7        00:00:00 /opt/MozillaFirebird/lib/firefox-bin
otto         22015 22014  0 14:21 pts/7        00:00:00 /opt/MozillaFirebird/lib/firefox-bin
otto         22019 22014  0 14:21 pts/7        00:00:00 /opt/MozillaFirebird/lib/firefox-bin
otto         22073 22026  0 14:23 pts/6        00:00:00 grep firefox
otto@l11:~> kill -9 22012
otto@l11:~> ps -ef | grep firefox
otto         22098 22026  0 14:25 pts/6        00:00:00 grep firefox
otto@l11:~> exit
logout
Connection to l11 closed.
otto@l10:~>
```

Wenn, wie in diesem Fall, das gesuchte Kommando mehrmals auftaucht, wobei die Prozesse mit höherer Prozessnummer (22014, 22015, ...) von früheren Prozessen (22012, ...) gestartet wurde, ist es in der Regel günstiger, zunächst den späteren Prozess zu killen. Wenn zum Beispiel der erste Prozess nur ein Script zum Starten weiterer Prozesse ist, beendet sich dieser anschließend selbst.

Natürlich kann man auch beim remote-kill die Kommandos **pkill** und **killall** benutzen, jedoch mit der oben beschriebenen Gefahr des versehentlichen Beendens weiterer Prozesse.

Klappt diese Vorgehensweise nicht, kann versucht werden, den Window-Manager zu killen (dies aber nicht mit **kill -9**). Der Window-Manager hat unter KDE den Namen **kwin**, man gibt also das Kommando **killall kwin** ein.

Auf Linux-Rechnern lässt sich die aktuelle X-Window-Sitzung auch durch einen Neustart des X-Servers beenden. Dazu verwendet man die Tastenkombination `<Strg>+<Alt>+<Backspace>`.

Sollte alles nicht funktionieren, ist die Systemadministration zu benachrichtigen (siehe Abschnitt 1.6).

Schalten Sie niemals die Rechner aus!

8.5 Wenn nichts mehr hilft

Falls Probleme auftreten, die mit den oben beschriebenen Mitteln nicht zu beheben sind, scheuen Sie sich nicht, andere im Raum arbeitende Benutzer um Rat zu fragen.

Wenn Sie nicht mehr weiter wissen, senden Sie eine Mail an root@math.uni-wuppertal.de (siehe Abschnitt 10.3), in der Sie Ihr Problem möglichst genau beschreiben. Falls Sie Fehlermeldungen in einem Shell-Fenster sehen, die mit dem Problem in Zusammenhang stehen, kopieren Sie diese (vgl. Kapitel 4) in Ihre Mail.

Beachten Sie bitte, dass die Rechner nicht ausgeschaltet werden dürfen, da das Schreiben von gepufferten Dateioperationen dann nicht mehr gewährleistet ist. Schlimmstenfalls kann dies zu Datenverlust und irreparablen Schäden des Dateisystems führen! Denken Sie daran, daß bei Ausschalten eines Multi-User-Systems im Allgemeinen auch Prozesse anderer Benutzer in Mitleidenschaft gezogen werden!

9 Drucken

In allen Ausbildungsrechnerräumen sind Netzwerk-Laserdrucker i. d. R. mit Duplexeinheit für den zweiseitigen Ausdruck verfügbar. Üblicherweise verkleinern alle Drucker die Ausgabe automatisch, so dass jeweils zwei Seiten auf ein Blatt gedruckt werden.

Auf den CIP-Rechnern und den IT-Rechnern besteht jeweils eine Beschränkung der pro Tag, d. h. innerhalb von 24 Stunden druckbaren Seiten von maximal 20 Seiten. Ist dieses Limit überschritten, so kann man bis zum Ablauf dieser Frist keine weiteren Ausdrücke vornehmen.

Bitte beachten Sie, dass die für alle Benutzer der Ausbildungsrechner verbindliche Benutzungsordnung der Fachgruppe Mathematik und Informatik sparsamen Umgang mit Ressourcen, also insbesondere auch mit Papier und Toner, vorschreibt.

Die an die Ausbildungsrechner angeschlossenen Drucker sind primär für den Ausdruck von Übungsaufgaben und dergleichen gedacht. Drucken Sie nicht mehr, als für die Bearbeitung der Übungsaufgaben notwendig ist! Insbesondere dürfen Skripts nicht auf den Ausbildungsdruckern gedruckt werden!

Für den Fall, dass ein größerer Ausdruck benötigt wird (in der Regel nur für die finale Version einer Abschlussarbeit), werden die Ausdrücke von den Betreuern der Cluster auf anderen Druckern in der Fachgruppe durchgeführt. In jedem Fall sind größere Ausdrücke vorher mit der Systemadministration abzusprechen. Eine weitere Möglichkeit ist schließlich noch der (kostenpflichtige) Ausdruck über die Rechner des Internet-Cafes auf den damit vernetzten Kopierern.

9.1 Druckbare Dateien

Die Drucker sind so konfiguriert, dass sie nur Postscript-Dateien drucken. Wird von einem Programm (wie dem Dateimanager) aus gedruckt, werden Textdateien automatisch in Postscript gewandelt. Will man eine Textdatei mit **lp** drucken, wird diese ebenfalls in Postscript gewandelt, allerdings werden zu lange Zeilen dabei einfach abgeschnitten. Um dies zu vermeiden, sollten Textdateien vor dem Drucken mit **fold** behandelt werden, um den Text auf eine vorgegebene Breite (in Zeichen pro Zeile) zu formatieren:

```
fold [ -w breite | -breite ] textdatei | lp
```

Eine weitere Möglichkeit, einen formatierten Ausdruck von Textdateien zu erhalten, besteht darin, sie mit einem Texteditor, z. B. mit KWrite unter KDE, zu öffnen und dann die Druckfunktion des Editors zu verwenden.

Alle Dateien, die weder Textdateien sind noch im Postscript-Format vorliegen, können nur mit Hilfe entsprechender Programme gedruckt werden. So werden DVI-Dateien mit **dvips** gedruckt, für Bilddateien muss ein Betrachtungsprogramm (vgl. Abschnitt 6.3) gestartet und von diesem aus gedruckt werden. Zum Betrachten von Postscript-Dateien verwendet man auf den Linux-PCs die Previewer **gv** oder **ghostview**. Dateien im Portable Document Format (PDF) müssen mit dem Acrobat Reader (**acroread**) geöffnet werden und können dann mit der eingebauten Druckfunktion ausgegeben werden.

Bitte beachten Sie, dass Binärdateien (Programme) nicht gedruckt werden können!

Wenn Sie einen Quelltext zu einem Programm drucken wollen, ziehen Sie nicht das Symbol für das übersetzte Programm vom Dateimanager auf den Drucker!

9.2 Absetzen von Druckaufträgen

Druckaufträge können sowohl von der Kommandozeile aus als auch mit Hilfe eines Druckprogrammes unter den graphischen Oberflächen abgesetzt werden.

Die Kommandozeilenbefehle sind

```
lpr datei [ datei ... ]           druckt angegebene Datei(en) auf Standarddrucker
lpr -P drucker datei [ datei ... ] druckt angegebene Datei(en) auf Drucker drucker
```

Bei einem erfolgreichen Aufruf von **lpr** wird die Nummer des abgesetzten Druckjobs ausgegeben.

Unter KDE befindet sich auf dem Desktop ein Drucker-Symbol, auf das man eine zu druckende Datei mit festgehaltener linker Maustaste ziehen und dort loslassen kann. Desweiteren stehen mehrere Programme zur Verfügung, um Druckaufträge komfortabel zu verwalten. Man findet die **Drucker-Verwaltung** unter **Dienstprogramme** in den

Schaltflächenmenüs der Leiste am unteren Bildschirmrand.

Darüber hinaus bieten viele Programme die Möglichkeit, direkt vom Programm aus zu drucken. In der Regel wählt man dazu im Kopf des Programms den Menüpunkt Datei bzw. File und darin Drucken bzw. Print oder ein Drucker-Icon in der Symbolleiste.

9.3 Abbrechen von Druckaufträgen

Um einen Druckjob abzubrechen, benötigt man die Nummer des Druckjobs (die von **lp** oder **lpr** ausgegeben wurde). Diese erhält man entweder über ein graphisches Drucker-Programm (z. B. KJobViewer, den man durch Klick auf das Drucker-Symbol auf dem Desktop startet) oder mit den Kommandos **lpq** und **lpstat**:

lpstat Gibt alle aktiven eigenen Druckjobs aus
lpq Gibt alle aktiven Druckjobs aus

Hat man die Nummer erhalten, kann man den Druckjob mit **cancel** oder **lprm** abbrechen:

cancel job-id (**cancel -a** zum Löschen aller Druckjobs)
lprm job-id (**lprm -** zum Löschen aller Druckjobs)

In der Übersicht über die Druckaufträge, die man durch Klick auf das Drucker-Symbol erhält, hat man ebenfalls die Möglichkeit, Druckaufträge auszuwählen und (durch das Mülleimer-Symbol) aus der Warteschlange zu entfernen. Natürlich hat jeder Benutzer nur das Recht, die eigenen Druckjobs zu manipulieren und zu löschen.

9.4 Übersicht über die Druckkommandos

	Linux-Kommandos	
Postscriptdatei drucken	lpr ps-datei	lp ps-datei
Textdatei drucken	a2ps text-datei	lpr ASCII-datei
Druckerwarteschlange abfragen	lpq	lpstat
Druckjob löschen	lprm jobnummer	cancel jobnummer

9.5 Was tun bei Druckerproblemen

Falls der Drucker einen Fehler anzeigt (z. B. einen Papierstau) oder kein Papier mehr enthält, senden Sie eine kurze E-Mail an root@math.uni-wuppertal.de oder sagen Sie einem der unter 1.6 genannten Administratoren Bescheid. Öffnen Sie den Drucker nicht selbst und nehmen Sie keinerlei Einstellungen über die Konfigurationsmenüs vor. Legen Sie insbesondere nach Möglichkeit nicht selbst Papier nach. Verwenden Sie **niemals** bereits bedrucktes Papier oder Umweltschutzpapier in den Druckern (Verstaubung bzw. Verschmutzung der Mechanik). Falls das Schriftbild sehr hell oder unvollständig ist, wenden Sie sich an die entsprechenden Ansprechpartner (siehe Abschnitt 1.6), um die Toner-Kartusche überprüfen oder auswechseln zu lassen.

Sollte versehentlich eine Binärdatei gedruckt werden, so ist Folgendes zu tun:

1. Drucker offline schalten (Taste Start am Drucker drücken, so daß die Bereit-Anzeige erlischt).
2. Druckauftrag abbrechen (in einer Shell oder mit dem Druckertool).
3. Taste Job abbrechen am Drucker betätigen (um bereits an den Drucker übertragene Daten zu löschen).

Hinweis:

Druckaufträge, die nicht bearbeitet werden können, blockieren die Druckerwarteschlange und verhindern damit, dass nachfolgende Druckjobs abgearbeitet werden können. Dies kann z. B. wegen falscher Papieranforderungen (etwa US-Letter anstelle von DIN-A4) passieren. Falls also Ihre Druckaufträge aufgrund Papiermangels, Papierstau oder anderer Probleme nicht bearbeitet werden können, löschen Sie diese aus der Warteschlange (vgl. 9.3).

Löschen Sie stets Ihre Druckaufträge aus der Warteschlange, bevor Sie den Raum verlassen!

10 Internet und WWW

10.1 Grundsätzliches

Das Internet ist ein weltweiter Zusammenschluss miteinander vernetzter Rechner. Neben der Möglichkeit, über das Netz auf entfernten Maschinen arbeiten zu können (einen Zugang vorausgesetzt), stehen im Internet eine Reihe von Diensten zur Verfügung, die mit Hilfe geeigneter Programme allgemein genutzt werden können. Zum Beispiel:

E-Mail	Versenden und Empfangen elektronischer Post
WWW	Zugriff auf Angebote im World Wide Web
News	Austausch von Nachrichten, Fragen und Antworten auf Nachrichtentrettern
Anonymes FTP	Übertragen von frei verfügbaren Dateien und Programmen

10.2 Netiquette

Beim Umgang mit Netzressourcen gibt es einige Verhaltensregeln, die beachtet werden sollen. Diese Regeln nennen sich die *Netiquette* und beschreiben, wie man das Internet nutzt, ohne andere Nutzer dabei zu stören oder zu belästigen. Dabei gibt es keinen einheitlichen Netiquettetext, sondern eine Vielzahl von Dokumenten, die sich inhaltlich überschneiden.

Die *Netiquette* kann mit einem WWW-Browser z. B. unter folgenden Adressen geladen werden:

<https://de.wikipedia.org/wiki/Netiquette>
<https://tools.ietf.org/html/rfc1855>

10.3 E-Mail

10.3.1 E-Mail-Adressen

Um elektronische Post (E-Mail) versenden zu können, muss jeder potentielle Empfänger eine Adresse haben, unter der er zu erreichen ist. Jede E-Mail-Adresse hat die Form *benutzer@adresse*. Dabei ist *benutzer* meist der Benutzername des Empfängers unter der entsprechenden Adresse und *adresse* ein Domänenname (MX-Domain) oder der Name eines Rechners (FQDN) im Internet.

Als Studierende der Bergischen Universität Wuppertal haben Sie eine offizielle E-Mail-Adresse, die Sie für alle Ihre E-Mail-Kommunikation als Studierende nutzen müssen. Sie sind durch die Einschreibungsordnung verpflichtet, Ihr offizielles E-Mail-Konto beim ZIM regelmäßig abzurufen.

Die E-Mail-Adressen des Benutzers mit der Matrikelnummer 2310815 in der Uni lauten:

2310815@uni-wuppertal.de
 Vorname.Nachname@uni-wuppertal.de (als Aliasname)

Aliasnamen können Sie selbst über die Webseiten den ZIM unter **Mein Zugang** einstellen.

10.3.2 Thunderbird / Seamonkey Mail

Zur Bearbeitung elektronischer Post können auch die Programme des mozilla.org-Projektes (Seamonkey, Thunderbird, vgl. Abschnitt 10.4) benutzt werden.

Bei Seamonkey findet man den Menüeintrag **Mail & Newsgroups** zum Starten des Mail-Clients im **Window**-Menü. Alternativ kann man auf das Brief-Icon am unteren Fensterrand klicken. Der Mail-Client von Seamonkey lässt sich auch direkt per Kommando starten:

seamonkey -mail

Die Bedienung von Thunderbird und des Mail-Clients von Seamonkey ist mit der anderer Mail-Programme vergleichbar.

Konfiguration eines ZIM-Accounts

Um Ihre E-Mail an die offizielle Uni-Adresse mit einem E-Mail-Programm wie Thunderbird zu bearbeiten, müssen Sie zunächst einige Einstellungen vornehmen. Aktuelle E-Mail-Programme erkennen die notwendigen Einstellungen für Posteingangs- und Postausgangs-Server bereits an der E-Mail-Adresse, die bei der Konfiguration eines Kontos neben dem eigenen Namen zunächst einzugeben ist. Für die manuelle Konfiguration gehen Sie folgendermaßen vor:

Nach dem Starten des E-Mail-Programms wählen Sie dazu im Menü einen Eintrag wie **Edit - Preferences**. Es erscheint ein Fenster, in dem Optionen gesetzt werden können:

- Tragen Sie zunächst unter **Identity** Ihren Namen und Ihre E-Mail-Adresse ein.
- In der Kategorie **Mail & Groups** wählen Sie **Mail Server**.
- Dort aktivieren Sie den Eintrag **IMAP** unter **Mail server type**.
- Geben Sie Ihren ZIM-Benutzernamen unter **Mail server user name** ein.
- Tragen Sie mail@uni-wuppertal.de als **Outgoing mail (SMTP) server** und **Incoming mail server** ein.

Bei Seamonkey Mail und Thunderbird werden die notwendigen Einstellungen beim ersten Start abgefragt. Später finden Sie die entsprechenden Konfigurationsmöglichkeiten im **Edit**-Menü unter dem Eintrag **Mail & Newsgroups Account Settings...** (bei Seamonkey) bzw. **Account Settings...** (bei Thunderbird). Tragen Sie dort die oben genannten Daten ein, um auf Ihr E-Mail-Konto beim ZIM zuzugreifen. Weitere Hinweise finden sich auf den WWW-Seiten des ZIM unter der URL

<https://zim.uni-wuppertal.de/de/unsere-dienste/e-mail/>

Hinweis:

Beachten Sie bitte, dass die Verwaltung der elektronischen Post mit mehreren verschiedenen E-Mail-Clients in der Regel problematisch ist. Wenn Sie etwa Ihre E-Mail mittels Netscape per POP3-Zugriff abholen, wird sie in ein Netscape-eigenes lokales Verzeichnis kopiert, weshalb anderen Mail-Programme anschließend nicht oder nur eingeschränkt auf diese Post zugreifen können. Daher empfiehlt sich immer der Zugriff mittels IMAP, da die E-Mail dann auf dem Server verbleibt und direkt dort verwaltet werden kann.

Wenn Sie Ihre Mail über Mail-Server empfangen und verschicken (z. B. mail@uni-wuppertal.de), werden zum Teil persönliche Daten zwischen Ihrem Mail-Client und dem Mail-Server über das Internet ausgetauscht. Daher bieten moderne Mail-Server die Möglichkeit an, verschlüsselt (abhörsicher) über das Netz zu kommunizieren. Es kann sogar sein, dass Server ausschließlich diese Möglichkeit zur Verfügung stellen. Verwenden Sie daher, soweit möglich, stets sichere Verbindungen (SSL/TLS)! Normalerweise lässt sich dies bei den **Server Settings** durch Setzen eines Häkchens bei „Use secure connection (SSL)“ sowohl für den Postausgangs-Server (SMTP) als auch für den Posteingangs-Server (IMAP/POP3) konfigurieren.

10.3.3 KMail unter KDE

Als E-Mail-Programme steht neben den Kommandozeilen-Versionen und den Programmen der Mozilla-Familie auch das zur graphischen Oberfläche KDE gehörige KMail zur Verfügung. Letzteres wird per Icon in der Menüleiste, über die Start-Menüs oder durch das Kommando **kmail** gestartet.

Bei Verwendung des Mail-Programms KMail sind beim ersten Aufruf folgende Einstellungen vorzunehmen:

Menü **Einstellungen**, Menüpunkt **KMail einrichten...**

- **Identitäten** anklicken und persönliche Daten eintragen
- Icon **Netzwerk** anklicken
- Unter **Empfang** den IMAP-Server mail.uni-wuppertal.de eintragen.
- Unter **Sicherheit** als Verschlüsselung „SSL“ oder „TLS“ und als Anmelde-Methode „Einfacher Text“ einstellen.
- Unter **Versand** den SMTP-Server mail.uni-wuppertal.de eintragen
- Unter **Sicherheit** als Verschlüsselung „TLS“ einstellen.

Hinweis: Wird bei der Erstellung einer neuen Nachricht in KMail eine falsche Mail-Adresse angegeben, so kann die Mail nicht versandt werden, verbleibt im **Postausgang** und blockiert das Versenden jeder weiteren Mail. Die fehlerhafte Nachricht muss daher im **Postausgang** gelöscht werden.

10.3.4 Kommandozeilen-Programme

Die Bearbeitung von E-Mail an der Kommandozeile ist nicht sehr komfortabel, ist aber bei einem remote-Login recht nützlich. Die gebräuchlichen Programme dafür sind **mail** und **mailx**. Dabei bietet **mailx** einige zusätzliche Möglichkeiten, die über die Online-Hilfe in Erfahrung gebracht werden können.

mail Startet das Kommandozeilen-Mail-Programm interaktiv
mail empfaenger sendet den danach bis zu einem <Strg>+<D> eingegebenen Text an *empfaenger*

Beim interaktiven Start von **mail** erfolgt die Steuerung durch die Eingabe von Kommandokürzeln:

<Return>	Nächste Nachricht anzeigen
–	vorhergehende Nachricht anzeigen
d	Aktuelle Nachricht löschen und nächste anzeigen
s	Aktuelle Nachricht sichern
n	Nachricht Nummer <i>n</i> anzeigen
?	mail -Kommandos anzeigen
q	mail verlassen

Neben den oben genannten Kommandozeilen-Programmen steht noch das deutlich komfortablere, Textfenster-orientierte Programm **pine** zur Verfügung. Eine Beschreibung des Programms bietet die Online-Hilfe.

10.3.5 Weiterleiten von Mail

Die automatische Weiterleitung von E-Mail erfolgt durch Konfiguration Ihres Kontos auf dem Posteingangs-Server. Für Ihr offizielles Uni-E-Mail-Konto nutzen Sie dafür die Accountverwaltung auf den Webseiten des ZIM.

Wollen Sie auf herkömmlichen UNIX-Systemen Ihre eingehende Post automatisch an eine andere Adresse weiterleiten lassen, benötigen Sie den traditionellen **forward**-Mechanismus. Dazu erstellen und editieren Sie die Datei **.forward** in Ihrem HOME-Verzeichnis und tragen dort die Adresse(n) ein, an die Ihre Mail weitergeleitet werden soll. Wollen Sie nur eine Kopie jeder Mail weiterleiten, tragen Sie als Adresse dort auch Ihre lokale Adresse mit einem führenden Backslash ("****") ein.

Will Benutzer **otto** auf dem Rechner **myhost** seine E-Mail sowohl lokal behalten als auch auf seinen Rechenzentrums-Account **otto13@uni-wuppertal.de** weiterleiten, so sieht die Datei **.forward** folgendermaßen aus:

```
\otto@myhost
otto13@uni-wuppertal.de
```

Der Eintrag **\otto@myhost** bewirkt, dass die Mail auch weiterhin lokal ankommt. Vergessen Sie in diesem Fall nicht den führenden Backslash vor der lokalen Adresse! Sonst erkennt das Mail-Programm eine Endlosschleife bei der Zustellung; die Mail kommt gar nicht mehr an.

10.3.6 Spam-E-Mail

Unter Spam-E-Mail (**Unsolicited Bulk Email**) versteht man unverlangt zugesandte Massen-E-Mail. Im Allgemeinen handelt es sich um Werbe-Mails, die oftmals zusätzlich mit Schadsoftware (**Malware**: Viren, Würmer, Trojaner, Spyware) als E-Mail-Anhang versehen ist.

Das ZIM betreibt eine automatische Spam-Erkennung, bei der die Filterung unter der vollen Kontrolle des Benutzers verbleibt. Dazu werden allen eingehenden E-Mails Kopfzeilen zugefügt, die Auskunft darüber geben, ob es sich (wahrscheinlich) um Werbung handelt. Zusätzlich werden viele Spams durch sogenanntes Graylisting abgeblockt und alle eingehende elektronische Post durchläuft einen Scanner für angehängte Malware.

Seamonkey und Thunderbird bieten die Möglichkeit, unerwünschte Werbepost auszufiltern. Zum einen haben die Programme lernfähige Junk-Filter, mit denen sie selbst versuchen, Spam zu erkennen. Andererseits können und sollten die durch die Spam-Erkennung des Rechenzentrums zugefügten Kopfzeilen verwendet werden, um markierte Mails durch eigene Filterfunktionen in entsprechende Ordner zu verschieben oder direkt zu löschen. Auf den WWW-Seiten des Rechenzentrums und im CIP-Hilfesystem finden sich dazu genaue Informationen.

Sie können die Markierungen von E-Mails durch Kopfzeilen auch für eine Filterung von Spam-E-Mail direkt auf einem IMAP-Server nutzen. Der Vorteil besteht darin, dass unerwünschte E-Mails dann nicht erst vom E-Mail-Client abgeholt werden muss, um als Spam erkannt und gelöscht zu werden.

10.4 Internet-Browser

Es gibt eine Reihe von Browsern für WWW-Dokumente im HTML-Format. Neben Seamonkey, Opera und dem Internet Explorer auf Windows-Rechnern ist Firefox derzeit einer der gebräuchlichen Web-Browser an dem hier exemplarisch der Umgang erläutert wird.

Der Firefox-Browser wird mit **firefox** oder über die Menüs gestartet. Ohne Angabe einer WWW-Adresse, einer sogenannten URL (Uniform Resource Locator) wird eine Startseite des Mozilla-Projektes (oder die selbst eingestellte Startseite) geöffnet. Um Firefox direkt mit einer Web-Seite aufzurufen, startet man das Programm mit

firefox *www-adresse* &

Eine solche *www-adresse* hat normalerweise die Form

`http[s]://rechneradresse[/verzeichnis][/datei]`

Beispielsweise zeigt

<code>https://www.uni-wuppertal.de</code>	die Startseite der Universität
<code>https://www.math.uni-wuppertal.de</code>	die Startseite der Fachgruppe Mathematik und Informatik
<code>https://www.math.uni-wuppertal.de/de/studium/</code>	Informationen für unsere Studierenden
<code>https://www.MathePrisma.de</code>	die Startseite des MathePrisma-Projektes

und unter der URL

`https://www.math.uni-wuppertal.de/de/studium/lehrveranstaltungen/rechnereinfuehrung.html`

finden sich Informationen zum Blockkurs „Einführung in die Benutzung der Ausbildungsrechner“ der Fachgruppe Mathematik und Informatik, zu dem dieses Skript gehört.

10.4.1 Navigation im WWW (Web-Surfen)

Im Firefox-Fenster wird das angeforderte Dokument angezeigt. Die unterstrichenen Textstellen sind (Hyper)Links, also Verweise auf andere Dokumente. Durch Klick auf einen solchen Link wird das Dokument geladen, auf das der Link verweist. Bewegt man nur den Mauszeiger auf den Text mit dem Link, wird am unteren Fensterrand die Adresse, auf die verwiesen wird, angezeigt. Ist der Verweis ein HTML-Dokument, wird dieses anstelle des aktuellen Dokumentes geladen. Andernfalls gibt es eine Reihe von Dateiformaten, für die eine Anwendung gestartet wird (beispielsweise xdvi für eine DVI-Datei). Ist das angegebene Dokument keiner Applikation zuzuordnen, wird dem Benutzer die Möglichkeit gegeben, ein Programm anzugeben oder das Dokument in seinem Verzeichnisbaum zu speichern.

Klickt man mit der Maus in das Textfeld mit der URL am oberen Fensterrand, kann man dort die Adresse eines Web-Dokumentes eingeben, das nach Eingabe von <Return> geladen wird.

Durch einen Klick auf das **Home**-Icon wird zu dem Dokument zurückgekehrt, mit dem Firefox gestartet wurde, bzw. das Dokument geladen, das als Startseite konfiguriert ist. Welche Seite dies ist, kann man im **Edit**-Menü unter **Preferences...** einstellen.

Ein Klick auf das **Stop**-Icon bewirkt, dass das Laden des angeforderten Dokumentes abgebrochen wird, das **Back**-Icon kehrt zum vorherigen Dokument zurück, das **Forward**-Icon geht nach Rückwärtsschritten wieder einen Schritt vorwärts, und mit dem **Reload**-Icon kann man das aktuelle Dokument neu laden, um ggf. Veränderungen wirksam werden zu lassen. Hält man während des Neuladens die <Shift>-Taste oder die <Shift>- und die <Strg>-Taste gedrückt, so werden Änderungen auf der Seite (z. B. in einem Formular) rückgängig gemacht.

Im **File**-Menü befinden sich Einträge, um das Dokument zu speichern oder zu drucken, ein weiteres Browser-Fenster oder ein neues Unterfenster zu öffnen sowie Firefox zu beenden. Von der Möglichkeit weiterer Fenster oder Unterfenster sollten Sie stets Gebrauch machen! Auf diesem Weg werden gemeinsam nutzbare Komponenten des Programms nur einmal in den Speicher geladen und Sie sparen Rechnerressourcen (Speicher, CPU-Zeit). **Vermeiden Sie es, mehrere Browser-Prozesse parallel zu starten!**

Im **View**-Menü besteht die Möglichkeit, das Erscheinungsbild des Browsers und der dargestellten Webseite den eigenen Bedürfnissen anzupassen. Man kann z. B. Schriftart und -größe wählen, sowie die Textkodierung anpassen, falls auf der Seite einige Zeichen falsch dargestellt werden.

Im **Bookmarks**-Menü können Verweise auf aktuelle Dokumente gesichert und gesicherte Dokumente geladen werden; so kann man sich eine Sammlung von Lesezeichen für oft besuchte Seiten anlegen.

10.4.2 Speichern von Dokumenten

Angezeigte Dokumente können im **File**-Menü gespeichert werden, wobei das Speichern der Dokumente als Text- oder HTML-Datei möglich ist.

Bei Dokumenten, für die eine Applikation gestartet wird, kann das Speichern von der gestarteten Applikation aus erfolgen. Dies geschieht in der Regel über das **File**-Menü der Applikation. Allerdings bieten nicht alle Anwendungen diese Möglichkeit.

Ein alternativer Weg ist das Speichern des Dokumentes unter Umgehung der Applikation. Dazu wird mit der rechten Taste auf den Link geklickt, der auf das Dokument verweist. Im dann erscheinenden Menü wird der Eintrag **Save Link Target As...** ausgewählt.

10.5 News

Usenet-News sind eine Form des Informationsaustausches im Internet. Dabei gibt es eine Unmenge an Newsgroups zu den verschiedensten Themen, in denen Nachrichten (Fragen, Antworten, Kommentare etc.) abgelegt werden. Jede dieser Gruppen kann als abonniert gekennzeichnet werden; damit werden alle aktuellen Nachrichten in dieser Gruppe beim Start eines News-Readers angezeigt.

Der Umgang mit News geschieht am einfachsten mit Seamonkey oder Thunderbird. Dazu startet man z. B. Seamonkey und wählt im Menü **Window** einen Eintrag **Mail & Newsgroups**. Alternativ kann Seamonkey mit **seamonkey-news** auch direkt als News-Reader gestartet werden.

Der für die Hochschule zuständige News-Server heißt news.cis.dfn.de und ist ggf. in den Einstellungen anzugeben. Durch klicken auf den Namen des News-Servers erhält man die Möglichkeit, die zu abonnierenden Gruppen auszuwählen.

Bitte lesen Sie, Bevor Sie sich aktiv an Diskussionen in News-Foren beteiligen, unbedingt die entsprechenden Kapitel der Netiquette (vgl. Abschnitt 10.2)! Ausserdem gibt es eine Reihe von Newsgroups, die speziell für News-Neulinge Informationen bereithalten (de.newusers.infos, de.newusers.questions, ...) und die man daher zunächst einmal abonnieren sollte.

10.6 FTP

FTP (File Transfer Protocol) ist ein Internet-Dienst zum Übertragen von Dateien zwischen Rechnern. Um Dateien von einem lokalen zu einem entfernten Rechner mit FTP zu übertragen, benötigt man neben dem Zugang auf dem lokalen auch einen Account auf dem entfernten Rechner.

Viele Programme werden per FTP allgemein zur Verfügung gestellt. Damit jeder auf diese Daten zugreifen kann, gibt es die Möglichkeit eines anonymen FTP-Zuganges. Ein solcher anonymer Zugang besteht aus dem Benutzernamen **anonymous** und der eigenen E-Mail-Adresse als Passwort.

Nach dem FTP-Login als anonymer Benutzer hat man einen eingeschränkten Zugang zu einem Teil des Verzeichnisbaumes des entfernten Rechners, nämlich den Teil, der allgemein verfügbare Daten enthält.

Die Rechnernamen anonymer FTP-Server beginnen meist mit **ftp**. Diese Adresse ist dem FTP-Programm anzugeben.

Auf den Ausbildungsrechnern ist neben dem Kommandozeilen-FTP auch das graphische FileZilla installiert; außerdem können auch Seamonkey und Firefox (mit entsprechenden AddOns) als FTP-Programm mit anonymem Zugang genutzt werden. Für eine verschlüsselte (nicht-anonyme) Datenübertragung verwendet man SFTP, das Secure File Transfer Program.

Beim Übertragen von Daten per FTP gibt es zwei Modi: ASCII- und binäre Übertragung. Bei der ASCII-Übertragung werden Textdateien zwischen Plattformen konvertiert (bei MS-DOS und UNIX wird beispielsweise das Zeilenende unterschiedlich gekennzeichnet, vgl. Abschnitt 11.3). Binäre Daten (beispielsweise Software-Archive) werden dabei allerdings zerstört. Bei der binären Übertragung werden die Daten genau so übertragen, wie sie auf dem Server vorliegen. Im Zweifelsfall ist dies die bessere Übertragungsart.

10.6.1 Kommandozeilen-FTP

ftp ist ein Kommandozeilen-FTP-Programm, das mit

ftp *ftp-server-adresse*

gestartet wird. Nachdem eine Verbindung aufgebaut ist, wird nach dem Benutzernamen (z. B. *anonymous*) und dem Passwort gefragt. Die Bedienung von **ftp** geschieht durch die Eingabe von Kommandos.

Die gebräuchlichsten Kommandos sind:

help	Gibt eine Liste aller ftp -Befehle aus
help <i>befehl</i>	Gibt eine Beschreibung von <i>befehl</i>
cd <i>verzeichnis</i>	Wechselt in das angegebene Verzeichnis auf dem entfernten Rechner
lcd <i>verzeichnis</i>	Wechselt in das angegebene Verzeichnis auf dem lokalen Rechner
dir	Gibt den Verzeichnisisinhalt aus
ascii	Schaltet auf ASCII-Übertragung um
bin	Schaltet auf binäre Übertragung um
get <i>datei</i>	Lädt <i>datei</i> vom FTP-Server herunter (download)
put <i>datei</i>	Lädt <i>datei</i> auf den FTP-Server herauf (upload)
mget <i>Dateiliste</i>	Lädt mehrere Dateien vom FTP-Server herunter (download)
mput <i>Dateiliste</i>	Lädt mehrere Dateien auf den FTP-Server herauf (upload)
bye	Schließt die Verbindung und beendet ftp

10.6.2 Secure File Transfer Program

Das Secure File Transfer Program (vgl. Abschnitt 6.2) startet man mit dem Kommando

sftp *user@sftp-server-adresse*

Die Bedienung geschieht auch hier über Kommandos, die im wesentlichen denen des **ftp**-Programms entsprechen. Dabei ist zu beachten, dass nicht jeder FTP-Server einen Zugang per **sftp** bietet und auch nicht jeder SFTP-Server einen unverschlüsselten Zugang erlaubt.

Da bei der Datenübertragung per FTP sämtliche Daten unverschlüsselt zwischen den Rechnern im Internet transportiert werden, sollten Sie immer dann, wenn es um schutzwürdige Daten geht, sichere Verbindungen (**sftp**) nutzen!

10.6.3 Graphische FTP-Programme

KBear und FileZilla sind graphische FTP-Programme und werden durch Eingabe von **kbear** bzw. **filezilla** oder über das Start-Menü gestartet. Diese Programme unterstützen neben FTP und SFTP noch eine Reihe weiterer Protokolle für die Datenübertragung. Ausserdem können sie es eine Liste von FTP-Servern verwalten, so dass ein Login darauf durch Auswahl des Servers aus der Liste möglich ist.

Beim ersten Start wird man von einem Assistenten durch die Konfiguration geleitet, wobei die wichtigsten Einstellungen vorgenommen werden können.

Anschließend kann man Verbindungen zu verschiedenen Servern herstellen und erhält neben einem Fenster, das den Zugriff auf die lokalen Daten ermöglicht, jeweils ein neues Fenster, in dem der Verzeichnisbaum des entsprechenden Servers dargestellt wird. Man hat damit eine komfortable Oberfläche zur Verfügung, unter der Dateien per Maus zwischen Verzeichnisfenstern hin und her kopiert werden können.

10.6.4 File Transfer mit Seamonkey und Firefox

Um z. B. Seamonkey als FTP-Programm für anonyme Zugänge zu nutzen, startet man Seamonkey mit

seamonkey *ftp://ftp-server-adresse*

Man erhält eine Liste der Dateien und Verzeichnisse auf dem entfernten Rechner und kann durch Klicken auf Verzeichnisse in diese wechseln bzw. durch Klicken auf Dateien diese herunterladen.

10.7 Informationssuche

10.7.1 Bibliotheks-Recherche

Unter der Adresse

<https://www.bib.uni-wuppertal.de>

kann auf den Server der Universitätsbibliothek zugegriffen werden. Der Link „Katalog“ führt auf eine Seite, auf der nach Anmeldung mit Benutzernummer und Passwort Bibliotheks-Recherche, Vormerkungen und Wiederausleihen vorgenommen werden können.

10.7.2 WWW-Suchmaschinen

Suchmaschinen im World-Wide-Web sind spezielle WWW-Seiten, die Eingabefelder bereitstellen, in denen die gesuchten Begriffe einzutragen sind. Jede dieser Suchmaschinen hat eine WWW-Adresse, anhand der sie in einem Browser wie Firefox angefordert werden kann. Einige gebräuchliche Suchmaschinen sind:

https://www.google.de	(WWW-Dokumente)
http://www.yasni.de/	(Personensuche)
https://duckduckgo.com/	(Suchmaschine ohne Tracking)
https://metager.de/	(Meta-Suchmaschine über deutschsprachige Suchmaschinen)
https://www.wolframalpha.com/	(Suchmaschine für wissenschaftliche Suchanfragen)

Das Ergebnis einer Suche ist meist eine Liste von Dokumenten oder Web-Adressen, in denen der gesuchte Begriff bzw. Name gefunden wurde.

11 Archivierung und Dateitransfer

Um den durch eigene Daten belegten Plattenspeicher gering zu halten und zum Transfer von Daten nach Hause ist es sinnvoll, Daten so zu archivieren, dass sie weniger Platz benötigen.

Darüber hinaus sammeln sich im Laufe der Zeit eine Menge nicht mehr benötigter Informationen an, die gelegentlich überprüft und ggf. gelöscht werden sollten.

11.1 Zugriff auf HOME-Verzeichnisse

Jedem Benutzer steht mit seinem HOME-Verzeichnis ein Teil des Verzeichnisbaumes zur Speicherung eigener Daten und Konfigurationsdateien zur Verfügung. In einer Kommando-Shell wechselt man durch Eingabe von `cd` (ohne Argumente) in sein Verzeichnis, öffnet man einen Dateimanager (z. B. Dolphin) durch Klicken auf das Symbol im Start-Menü, so befindet man sich ebenfalls im eigenen HOME-Verzeichnis. Diese Verzeichnisse befinden sich (physisch) jeweils auf demselben (Network File System-)Server des ZIM.

Für den Dateitransfer zwischen den Ausbildungsrechnern und einem Heim-PC (vgl. Abschnitt 6.2) ist es oft notwendig, den korrekten absoluten Pfad des HOME-Verzeichnisses im Dateibaum zu kennen. Dieser ist auf allen Rechnern identisch:

Pool	Raum	Verzeichnis	Pfad
CIP	G.14.11	HOME-Verzeichnis	<code>/home/user</code>
IT	G.16.15	HOME-Verzeichnis	<code>/home/user</code>
PI	G.16.13	HOME-Verzeichnis	<code>/home/user</code>

Da die Benutzerverzeichnisse aller drei Rechner-Pools auf demselben NFS-Server gespeichert sind, kann man dort direkt auf alle Daten im eigenen HOME-Verzeichnis zugreifen, unabhängig davon, in welchem Raum man sich befindet.

Um von anderen Linux-Rechnern auf die Daten der Ausbildungsrechner zuzugreifen oder umgekehrt, müssen diese zunächst über das Netz transferiert werden (vgl. Abschnitt 6.2).

11.2 Aufräumen im HOME-Verzeichnis

Sicherheitskopien und Logdateien

Verschiedene Programme erstellen Sicherheitskopien und Logdateien, die anhand des Namens bzw. des Suffixes zu erkennen sind. Diese können und sollten regelmäßig gelöscht werden.

Dateiname	Bedeutung
<code>core</code>	Speicherabzug bei Programmfehlern
<code>*%</code>	Sicherheitskopien des Texteditors
<code>*~</code>	Sicherheitskopien des Texteditors
<code>*.log</code>	Logdateien einer Übersetzung mit \TeX
<code>*.aux</code>	Logdateien einer Übersetzung mit \TeX
<code>*.o</code>	Objektdateien bei der Programmübersetzung
<code>*.tmp</code>	diverse temporäre Dateien

Dateien im Mülleimer und weitere Datensinken

Die mit dem Dateimanager in den Mülleimer verschobenen Dateien belegen weiterhin den vollen Speicherplatz. Deshalb sollte der Mülleimer regelmäßig durch Auswahl von **Mülleimer leeren** im Kontextmenü ausgeleert werden.

Selbst dann, wenn Sie nicht explizit neue Dateien erstellen oder sich Dateien aus dem Netz herunterladen und in Ihrem HOME-Verzeichnis speichern, werden durch die Nutzung unterschiedlicher Programme, wie z. B. das Surfen im WWW mit einem Webbrowser wie Firefox, Opera oder Seamonkey Daten erzeugt und gespeichert, die Speicherplatz in Ihrem Benutzerverzeichnis belegen. Bitte überprüfen Sie daher regelmäßig die „klassischen“ Datensinken und löschen Sie bei Bedarf überflüssige Dateien in Ihrem HOME-Verzeichnis:

Daten	Pfad
Webbrowser-Caches	.mozilla/firefox/kryptischer-Pfad/Cache/ .cache/mozilla/firefox/kryptischer-Pfad/Cache/ .opera/cache/
X-Windows-Fehlermeldungen	.xsession-errors*
diverse Zwischenspeicher	.cache
Mülleimer	.local/share/Trash/files

11.3 Disketten, CDs/DVDs, USB-Speicher, ...

Um Dateien per Datenträger auf einen DOS/Windows-Rechner zu übertragen, benötigt man eine (am besten DOS-formatierte) Diskette oder einen USB-Speicherstick.

Die Ausbildungsrechner haben an der Vorderseite USB-Schnittstellen. Als FAT-Wechselmedien formatierte USB-Massenspeicher werden automatisch vom Betriebssystem erkannt und als Icon auf dem Desktop oder im Dateimanager dargestellt. Ein Klick auf dieses Symbol öffnet dann ein Dateimanagerfenster.

Disketten werden in das Diskettenlaufwerk des Rechners eingeschoben, CDs und DVDs gehören in das per Knopf zu öffnende DVD-Laufwerk.

Disketten, CDs, DVDs und USB-Massenspeicher werden durch Klick auf das entsprechende Icon auf dem Desktop unter dem Pfad **/run/media/user/device** in das Dateisystem eingebunden. Der Zugriff auf Disketten und USB-Sticks erfolgt entweder mit Hilfe des Dateimanagers oder in einem Shell-Fenster per Kommando.

Bitte beachten Sie, dass bei UNIX, MS-DOS und Windows unterschiedlichen Konventionen für Dateinamen existieren: UNIX-Dateinamen dürfen 256 Zeichen lang sein und beliebig viele Punkte enthalten; MS-DOS-Dateinamen bestehen aus maximal acht Zeichen sowie ggf. einer Erweiterung (Suffix), die aus einem Punkt und bis zu drei weiteren Zeichen besteht. Außerdem sind unter MS-DOS nicht alle Zeichen in Dateinamen erlaubt. Unter Windows können Leerzeichen Teil der Dateinamen sein, was unter UNIX zumeist Schwierigkeiten bereitet, da ein Dateiname mit Leerzeichen als mehrere Dateinamen interpretiert wird.

Werden Dateien auf eine DOS-formatierte Diskette kopiert, werden auch die Dateinamen konvertiert. Dabei werden die ersten drei Zeichen hinter dem letzten Punkt zum Suffix und davor maximal acht Zeichen zum Dateinamen, wobei alle unter DOS nicht erlaubten Zeichen durch ein x ersetzt werden. Dies bewirkt, dass unter UNIX verschiedene Namen auf denselben DOS-Namen abgebildet werden können.

Hinweis:

Bitte lösen Sie unbedingt vor der Entnahme des Datenträgers dessen Einbindung in den Dateibaum mit einem Rechtsklick auf das entsprechende Icon und Auswahl von **Laufwerk-Einbindung lösen**. Erst anschließend darf der Auswurfknopf am Disketten- oder DVD-Laufwerk betätigt werden oder der USB-Stick entfernt werden. Durch das Kommando **eject** oder Rechtsklick auf das Icon und Auswahl von **Auswerfen** werden eine CD/DVD/BR oder ein USB-Speicher aus dem Dateibaum ausgehängt.

Wird ein USB-Datenträger trotz Laufwerks-Einbindung entfernt, kann es durch verzögerte Schreibvorgänge passieren, dass das Dateisystem zerstört wird! Wird eine Diskette einfach so ausgeworfen, bemerkt das System bei der nächsten eingelegten Diskette den Wechsel nicht und macht diese unbrauchbar.

11.4 Dateiformate zum Dateitransfer und zur Archivierung

Um den von einer Datei oder einem ganzen Verzeichnis belegten Platz zu reduzieren, besteht die Möglichkeit der Archivierung und Kompression. Bei der Archivierung werden mehrere Dateien und/oder Verzeichnisse in einer Datei zusammengefasst; die Kompression bewirkt eine Verringerung der Dateigröße. Damit wird auch der Dateitransfer (z. B. auf Diskette) vereinfacht, da die Anzahl und Größe der Dateien reduziert wird.

Archive und komprimierte Dateien erkennt man am Suffix im Dateinamen. Mit den entsprechenden Programmen können Archive erstellt und ausgepackt bzw. Dateien komprimiert und dekomprimiert werden.

Programm	Suffix	Funktion
tar	.tar	Archivierung (ohne Kompression)
gzip/gunzip	.gz	GNU Zip (Kompression einzelner Dateien)
tar z	.tgz	Archivierung (tar) mit Kompression (gzip) (.tar.gz)
bzip2	.bz2	(De-) Kompression einzelner Dateien
compress/uncompress	.Z	UNIX Compress
zip/unzip	.zip	Archivierung mit Kompression (entspricht DOS-pkzip/pkunzip)
unarj	.arj	Entpackt ARJ-Archive
lha	.lha	Erstellt/Entpackt LZH-Archive

Die .zip-, .arj- und .lha-Archive sind auf DOS/Windows-Systemen gebräuchlich, die Packer archivieren und komprimieren gleichzeitig. Die unter UNIX übliche Art der Archivierung trennt Archivierung (**tar**) und Kompression (**compress** bzw. **gzip**); die Kompressions-Programme können nur einzelne Dateien komprimieren.

Anleitung zu den Programmen gibt es in der Online-Hilfe (vgl. Abschnitt 7.4), bei **zip/unzip**, **unarj** und **lha** durch Aufruf ohne Parameter, bei **gzip/gunzip** und manchen **tar**-Versionen durch Aufruf mit dem Parameter **--help**.

Um große Dateien auf mehrere Disketten zu verteilen, dienen die Kommandos **bsplit** und **bmerge**. **bsplit** zerteilt eine große Datei in mehrere kleine, **bmerge** fasst die Teile wieder zusammen. Der Aufruf der Programme ohne Argumente gibt eine Aufruf-Hilfe aus.

Um beispielsweise das ganze Verzeichnis **Backups** mit **tar** und **gzip** in ein Archiv namens **Backup.tar.gz** zu archivieren und dann das Verzeichnis **Backups** zu löschen, gibt man ein:

```
otto@l1101:~> tar cf Backup.tar Backups
otto@l1101:~> gzip Backup.tar
otto@l1101:~> rm -rf Backups
```

oder kürzer, durch Kombination von **tar** und **gzip** durch die **tar z**-Option:

```
otto@l1101:~> tar cfz Backup.tgz Backups
otto@l1101:~> rm -rf Backups
```

Um das Archiv wieder herzustellen verwendet man:

```
otto@l1101:~> tar xfz Backup.tgz
```

12 Anpassen der Arbeitsumgebung

Die hier beschriebenen Möglichkeiten der Anpassung der Arbeitsumgebung beziehen sich auf die C-Shell und werden in der Datei `.cshrc` im HOME-Verzeichnis eingetragen. Änderungen in dieser Datei werden für neue Shells oder durch Eingabe von **source** `~/.cshrc` in einer laufenden Shell wirksam. Die meisten Einstellungen werden über **Umgebungsvariable** gesetzt, deren Wert man sich mit **env** (print **environment**) für alle bzw. **echo** `$variable` für nur eine Variable ausgeben lässt.

Andere Kommando-Shells wie die Bourne-Shell (Konfigurationsdatei `.profile`) oder die Bourne-Again-Shell (Konfigurationsdatei `.bashrc`) lassen sich in ähnlicher Weise durch (syntaktisch andere) Änderung der entsprechenden Konfigurationsdateien anpassen.

12.1 Pfade

Es gibt drei Umgebungsvariable für Suchpfade für das Arbeiten unter UNIX:

<code>PATH</code>	Suchpfad für Programme
<code>MANPATH</code>	Suchpfad für Manual-Seiten der Online-Hilfe
<code>LD_LIBRARY_PATH</code>	Suchpfad für dynamische Bibliotheken

Der Suchpfad für Programme wird benutzt, wenn ein Programm (ohne Pfadangabe) aufgerufen wird. Die in der Variable `PATH` enthaltenen Verzeichnisse werden in der aufgeführten Reihenfolge nach dem aufgerufenen Programm durchsucht. Um die `PATH`-Variable in der Datei `~/.cshrc` zu setzen, editiert man die Anweisung

```
set path = ( verzeichnis_1 verzeichnis_2 ... verzeichnis_n )
```

Die Variable `MANPATH` gibt an, in welchen Verzeichnissen beim Aufruf von **man** *befehl* nach der Manual-Seite für *befehl* gesucht wird.

In `LD_LIBRARY_PATH` wird festgehalten, in welchen Verzeichnissen nach dynamische Bibliotheken, die erst zur Laufzeit eines Programmes geladen werden, gesucht wird.

Die Variablen `MANPATH` und `LD_LIBRARY_PATH` werden (in `~/.cshrc`) wie folgt gesetzt:

```
setenv MANPATH verzeichnis_1:verzeichnis_2:...:verzeichnis_n
```

Beachten Sie bitte, dass die Variablen `PATH`, `MANPATH` und `LD_LIBRARY_PATH` bereits systemweit gesetzt sind. Wollen Sie Änderungen an diesen Variablen durchführen, ersetzen Sie den Inhalt nicht, sondern erweitern Sie ihn nur um die gewünschten Verzeichnisse:

```
set path = ( verzeichnis_1 verzeichnis_2 $path )
setenv MANPATH verzeichnis_1:verzeichnis_2:$MANPATH
```

12.2 Aliase

In manchen Shells (z. B. der C-Shell) besteht die Möglichkeit, mit Hilfe sogenannter **Aliase** Argumente fest an Kommandos zu binden oder Abkürzungen für Befehlsnamen einzurichten.

Um mit dem Befehl *name* das Kommando *kommando* zu starten, gibt man in der Datei `~/.cshrc` für die C-Shell oder in einem Shell-Fenster (dann wirkt es nur dort) ein:

```
alias name kommando
```

Falls *kommando* aus mehr als einem Wort besteht, muss es in doppelte Anführungszeichen (" ") eingeschlossen werden. Wenn beispielsweise dem Kommando **rm** zum Löschen von Dateien immer die Option **-i** übergeben werden soll, damit vor dem Löschen nachgefragt wird, kann dies mit

```
alias rm "rm -i"
```

geschehen. Der **Alias** kann durch Aufruf von

```
\rm datei.txt
```

umgangen werden. Um einen **Alias** zu löschen, gibt man ein:

```
unalias name
```

Die Einstellung der Standardoption `-i` für die Kommandos `rm`, `cp` und `mv` ist unbedingt zu empfehlen, auch wenn die Nachfrage beispielsweise beim Löschen von Dateien etwas lästig erscheint!

13 Installierte Software

Auf allen Ausbildungsrechnern der Fachgruppe Mathematik und Informatik ist die wichtigste Software für die Nutzung der Rechner im Rahmen des Studiums installiert. Dazu gehört Software zur Programmentwicklung (Editoren, Compiler, Debugger), für die Informatik-Ausbildung (z. B. Spezifikations-Tools, Datenbanksysteme, Graphikprogramme, MPI und OpenMP für die parallele Programmierung), zur Nutzung der Internetdienste (z. B. Web-Browser für die Internet-Recherche, E-Mail-Programme zur Kommunikation mit den Übungsgruppenleitern), mathematische und wissenschaftliche Software (z. B. Mathematica, Matlab), das wissenschaftliche Textsatz-System \LaTeX , Programme für die Didaktik der Mathematik und Informatik (z. B. Cinderella), usw.

Eine Auswahl der installierten Software ist hier kurz aufgeführt:

OpenOffice, LibreOffice	Office-Paket (Textverarbeitung, Tabellenkalkulation etc.)
\TeX , \LaTeX	Textsatz-Programm
Mathematica	Analysis-Paket
Matlab, Intlab	Lineare-Algebra Programm-Interpreter
Perl, Python, Ruby	GNU Script-Interpreter
Emacs, XEmacs	Textverarbeitung und mehr ...
Netscape, Seamonkey, Firefox, Opera	WWW-Browser, Mail-Client, News-Reader
GNUplot	Plotprogramm mit \TeX -Schnittstelle
xfig	Zeichenprogramm mit \TeX -Schnittstelle
xv, Gimp	Graphikbearbeitung
Rational, Together, Eclipse, Visual Paradigm, ...	Software-Entwicklungstools (OOP, UML, ...)
gcc	GNU Compiler (C, C++, Fortran77)
JDK	Sun Java Development Kit

Die installierte Software wird in Absprache mit den Dozentinnen und Dozenten der Mathematik, Informatik und Didaktik, sowie auf Anregung der Benutzerinnen und Benutzer der Ausbildungsrechner stets aktualisiert und ergänzt.

Hinweis: Zu \TeX und \LaTeX bietet das Rechenzentrum Einführungskurse an. Für einige Programmiersprachen, HTML, \LaTeX und Gimp ist im Rechenzentrum Literatur erhältlich.

A Benutzungsordnung für die Ausbildungsrechner

§1: Benutzungsberechtigung

1. Zur Benutzung der Ausbildungsrechner der FG² können zugelassen werden:
 - (1) Studenten der FG mit abgeschlossenem Vordiplom.
 - (2) Studenten der FG im Grundstudium zur Teilnahme an einer in der FG angebotenen Lehrveranstaltung.
 - (3) Sonstige Mitglieder der Hochschule zur Teilnahme an einer in der FG angebotenen Lehrveranstaltung.
2. Zulassung zur Benutzung erteilt die Systemadministration für Zugangsberechtigungen nach §1.1.1 bzw. der Veranstalter der Lehrveranstaltung bei Zugangsberechtigungen nach §1.1.2 und §1.1.3.
3. Die Bestimmungen dieser Benutzungsordnung und die Betriebsregelungen sowie die Netiquette sind Bestandteil des Bescheides über die Zulassung.

§2: Zulassungsverfahren

1. Die Zulassung zur Benutzung der Ausbildungsrechner der FG setzt die Teilnahme an einer von der FG angebotenen Einführung in die Nutzung der Ausbildungsrechner voraus.
2. Für Zugangsberechtigungen nach §1.1.2 oder §1.1.3 ist die Zulassung am Semesteranfang durch Unterschrift auf ein Formular gemäss Muster A zu beantragen. Account und Zugangskarte gelten bis Vorlesungsende.
3. Für Zugangsberechtigungen nach §1.1.1 ist die Zulassung nach abgelegtem Vordiplom durch Unterschrift auf ein Formular gemäss Muster B zu beantragen. Account und Zugangskarte gelten bis zur Exmatrikulation.
4. Die Zulassung erfolgt befristet im Rahmen der verfügbaren Kapazitäten (vgl. §3.1). Mit dem Zulassungsbescheid erfolgt die Zuweisung einer Benutzerkennung und die Ausgabe einer Zugangskarte zur Benutzung des Schliesssystems der DV-Räume. Die Benutzungserlaubnis ist personen- und kurs- bzw. projektbezogen und nicht auf andere Personen übertragbar.

Im Falle der Zulassung nach §1.1.2 gilt die Erlaubnis nur für die Dauer eines Semesters, kann aber bei Weiterbestehen der Voraussetzungen für die Benutzungsberechtigung verlängert werden. Bei der Zulassung nach §1.1.3 erlischt die Zulassung nach Ablauf der betreffenden Veranstaltung.
5. Die Nichterteilung einer Zulassung ist nur bei Vorliegen schwerwiegender Gründe beziehungsweise bei Kapazitätsengpässen möglich.
6. Die Benutzungserlaubnis von Hochschulmitgliedern und Hochschulangehörigen erlischt, sobald die Hochschulzugehörigkeit gemäss §11 UG nicht mehr gegeben ist, bei Zulassung nach §1.1.2 bzw. §1.1.3 sobald die Teilnahme an der relevanten Lehrveranstaltung nicht mehr regelmässig (in der Regel Nichtteilnahme an drei aufeinanderfolgenden Vorlesungs- oder übungsterminen) fortgeführt wird.
7. Die ausgegebenen Zugangskarten müssen spätestens zurückgegeben werden
 - (1) vor Aushändigung des Diplom-Zeugnisses bei Zulassung nach §1.1.1 ,
 - (2) bei Semesterende (vor Aushändigung der übungsscheine) bei Zulassung nach §1.1.2 bzw. §1.1.3.

²Fachgruppe Mathematik und Informatik im Fachbereich C – Mathematik und Naturwissenschaften der Bergischen Universität Wuppertal

§3: Rechte und Pflichten der Benutzer

1. Auf den Aushängen neben den DV-Räumen angekündigte Veranstaltungen in den DV-Räumen haben Nutzungsvorrang. Nicht durch Veranstaltungen benutzte Kapazität kann zur „freien“ Übung eingesetzt werden.
2. Die Benutzer haben das Recht, die für die Bearbeitung ihrer Aufgaben im Rahmen des Studiums notwendigen Einrichtungen und Betriebsmittel der Ausbildungsrechner der FG nach Massgabe der Zulassung im Rahmen der Benutzungsordnung und der Betriebsregelungen zu benutzen sowie die von der FG angebotenen Dienstleistungen in Anspruch zu nehmen.
3. Die Benutzer dürfen nur eigene Daten und Programme lesen oder verarbeiten. Sonstige Daten und Programme dürfen nur nach vorheriger Zustimmung des Verfügungsberechtigten gegenüber der FG gelesen oder verarbeitet werden.

Die Verantwortung für den fachlichen Inhalt der Programme und die fachliche Korrektheit der verwendeten Algorithmen bzw. der Verarbeitungsergebnisse liegt ausschliesslich beim Benutzer.

4. Die Benutzer sind verpflichtet:

- (1) Die Vorschriften der Benutzungsordnung und der Betriebsregelungen einzuhalten,
- (2) Geräte, Anlagen, Datenträger und sonstige Einrichtungen der FG sorgfältig und schonend zu behandeln und vor Verschmutzung zu bewahren,
- (3) Störungen, Beschädigungen sowie Fehler an DV-Anlagen und Geräten und Datenträgern nicht selbst zu beheben, sondern diese unverzüglich der FG zu melden,
- (4) die Rechner und Peripheriegeräte keinesfalls selbst auszuschalten oder zu rebooten,
- (5) in den Räumen der FG sowie bei Inanspruchnahme seiner Geräte, Datenträger und sonstiger Einrichtungen den Weisungen des Personals der FG Folge zu leisten,
- (6) die Benutzungsberechtigung auf Verlangen nachzuweisen,
- (7) die Benutzung auf die Bearbeitung der zum Studium nötigen Aufgaben zu beschränken,
- (8) Benutzerkennung und Passwörter vor Verwendung durch Dritte zu sichern,
- (9) jeglichen Verdacht der Mitbenutzung der Benutzerkennung durch Dritte sofort der FG zu melden,
- (10) der Leitung der FG auf Verlangen in begründeten Einzelfällen zu Kontrollzwecken Auskünfte über Programme und benutzte Methoden zu erteilen sowie Einsicht in die Programme zu gewähren,
- (11) ihre Daten und Programme so zu sichern, dass Schäden durch Verlust bei der Verarbeitung in der FG nicht entstehen können,
- (12) vor einer Verarbeitung personenbezogener Daten dies der FG mitzuteilen und – unbeschadet der eigenen Verpflichtung des Benutzers zum Datenschutz – die von der FG vorgeschlagenen Datenschutz- und Datensicherungsvorkehrungen zu beachten und zu nutzen,
- (13) bekannt gewordene Informationen über fremde Programme und Daten nicht ohne Genehmigung des Befugten weiterzugeben oder selbst zu nutzen,
- (14) das Erlöschen der Voraussetzungen für die Nutzungsberechtigung (z.B. durch Studienabbruch) unverzüglich zu melden und die Zugangskarte sofort zurückzugeben,
- (15) die Gepflogenheiten der Netiquette einzuhalten,
- (16) die Grundsätze der Wirtschaftlichkeit und Sparsamkeit zu beachten (Die Benutzung der Drucker der FG darf nur im Rahmen der Betriebsregelung — nachzufragen beim Zulassungserteilenden — erfolgen. Nichtbeachtung führt zur Sperrung der Druckerzugangserlaubnis),
- (17) in den DV-Räumen weder zu rauchen, noch zu trinken oder zu essen.

5. Die FG darf Programme der Benutzer mit deren Einverständnis zu Testzwecken einsetzen.

§4: Ausschluss von der Benutzung

1. Falls ein Benutzer gegen die Benutzungsordnung oder die Betriebsregelungen verstösst oder durch sein Verhalten der Betrieb der DV-Anlage der FG empfindlich gestört wird, kann die FG die Zulassung dieses Benutzers vorübergehend einschränken und in schwerwiegenden Fällen seine Benutzerkennung vorübergehend oder dauerhaft sperren.
2. Begründeter Verdacht der Ausspähung von Daten Dritter, des Erschleichens von Passwörtern Dritter, des Versuchs der Auffindung von Sicherheitslücken der DV-Anlagen der FG oder anderer über das Datennetz erreichbarer DV-Einrichtungen führt zur sofortigen Sperrung der Benutzerkennung bis zur Klärung.
Strafanzeige und endgültige Sperrung bleiben vorbehalten.

§5: Haftung

1. Die Benutzer haften für die von ihnen schuldhaft verursachten Schäden an DV-Anlagen und -Geräten, Datenträgern und sonstigen Einrichtungen der FG, für schuldhaft verursachte Verluste und Veränderungen der Daten und Programme der FG oder Dritter sowie für schuldhaft verursachte Schäden aus Verstössen gegen Rechtsvorschriften und die Bestimmungen dieser Benutzungsordnung.

Die Benutzer haben durch vorbeugende Massnahmen einen möglicherweise durch sie entstehenden Schaden so gering wie möglich zu halten.

2. Die FG übernimmt keine Gewähr für die inhaltliche Richtigkeit von Ergebnissen der Arbeit an seinen Einrichtungen.

Bestandteil dieser Benutzungsordnung sind die Betriebsregelungen des Hochschulrechenzentrums (HRZ) zum Thema Datennetze (vgl. <http://www.uni-wuppertal.de/hrz/netzinfo/netze.html>) und die Netiquette (vgl. <http://www.ping.at/guides/netmayer/>).

Bergische Universität Wuppertal

Fachbereich C – Mathematik und Naturwissenschaften

Fachgruppe Mathematik und Informatik

April 1997, April 2005

Literatur

- [1] Gilly, Daniel u.a.: *UNIX in a Nutshell*, dt. Ausgabe, O'Reilly/International Thomson Verlag 1995
- [2] Gulbins, Jürgen: *UNIX*, Springer 1995
- [3] Herold, Harald: *UNIX-Grundlagen*, Addison-Wesley 1995
- [4] Herold, Harald: *UNIX-Shells*, Addison-Wesley 1996
- [5] Krol, Ed: *Die Welt des Internet*, O'Reilly/International Thomson Verlag 1995
- [6] Levine, John R. und Levine Young, Margaret: *UNIX für Dummies*, International Thomson Verlag 1996
- [7] Mui, Linda: *Wenn der UNIX-Systemadministrator nicht zu finden ist*, O'Reilly/International Thomson Verlag 1996
- [8] Todina, Grace, u.a.: *UNIX – Ein praktischer Einstieg*, O'Reilly/International Thomson Verlag 1996
- [9] Wolfinger, Christine: *Keine Angst vor UNIX: Ein Lehrbuch für Einsteiger*, VDI-Verlag 1996