

1	GRUNDLAGEN	3
1.1	DIE STUDIERENDEN-RECHNERPOOLS	3
1.1.1	<i>Allgemeine Studierenden-Rechnerpools</i>	3
1.1.2	<i>Die FIM-Studierenden-Rechnerpools</i>	3
1.1.3	<i>Die Hausordnung der FIM-Studierenden-Rechnerpools</i>	5
1.2	EIN- UND AUSLOGGEN IN DEN FIM-RECHNERPOOLS	5
1.2.1	<i>Ein- und Ausloggen unter der grafischen Oberfläche</i>	5
1.2.2	<i>Passwörter</i>	6
1.3	ARBEITEN UNTER GRAPHISCHEN BENUTZEROBERFLÄCHEN	6
1.3.1	<i>Grafische Arbeitsumgebungen unter Linux</i>	7
1.3.2	<i>Das Arbeiten mit der LINUX Desktop-Umgebung Xfce</i>	7
1.3.3	<i>Bildschirmsperre</i>	9
1.3.4	<i>Programme beenden</i>	10
1.4	GRUNDBEGRIFFE ZUM KOMMANDO-ORIENTIERTEN ARBEITEN IN DER LINUX-SHELL	10
1.4.1	<i>Die Shell als Grundlage für die Eingabe von Linux-Befehlen</i>	10
1.4.2	<i>Linux-Befehle eingeben und ausführen lassen</i>	11
1.5	DER UMGANG MIT DATEIEN UND DIRECTORIES	13
1.5.1	<i>Dateien und Directories: Begriffe und Grundlagen</i>	13
1.5.2	<i>Arbeiten mit Dateien und Directories</i>	16
1.5.3	<i>Zugriffsrechte und Datenschutz</i>	17
1.5.4	<i>Löschen und Rückspeichern von Daten</i>	18
1.5.5	<i>Quota-Regelung zur Speicherplatzbeschränkung</i>	18
1.6	EDITIEREN / TEXTVERARBEITUNG	18
1.7	DRUCKEN, UMGANG MIT STANDARD-DATEITYPEN	19
1.7.1	<i>Öffnen/Anzeigen gängiger Dateitypen unter Linux</i>	19
1.7.2	<i>Ausdrucken mit den Campusdrucker-Geräten</i>	19
1.8	BENUTZUNG VON SCANNERN UND BRENNERN	20
1.9	NUTZUNG VON NETZWERK-DIENSTEN	21
1.9.1	<i>FIM-Rechnernamen und -adressen</i>	21
1.9.2	<i>Adressierung von FIM-Usern via Internet</i>	21
1.9.3	<i>Electronic Mail</i>	21
1.9.4	<i>Das World Wide Web</i>	24
1.9.5	<i>VPN und Eduroam: Mit dem eigenen Rechner in Uni-Netzen arbeiten</i>	26
1.9.6	<i>Kostenloser Online-Speicher mit LRZ Sync+Share</i>	27
1.10	SOFTWAREANGEBOTE FÜR STUDIERENDE	27
1.10.1	<i>Softwareangebote der Universität</i>	27
1.10.2	<i>Spezielle Softwareangebote für Teilnehmer am FIM-Lehrbetrieb</i>	27
2	VERTIEFTES ARBEITEN MIT DER LINUX-SHELL	28
2.1	DATEN UND DIRECTORIES BEARBEITEN MIT DER SHELL	28
2.1.1	<i>Pfade und Working Directories</i>	28
2.1.2	<i>Directory-Inhalte ansehen mit Kommando ls</i>	29
2.1.3	<i>Gängige Shell-Befehle für den Umgang mit Dateien und Directories</i>	30
2.1.4	<i>Zugriffsrechte setzen und entfernen</i>	31
2.1.5	<i>Speicherbedarf von Benutzerdaten optimieren</i>	32
2.2	PROZESSE BEOBACHTEN UND MANIPULIEREN	32

2.2.1	<i>Der Zusammenhang Programm-Prozess</i>	32
2.2.2	<i>Details zum Programmaufruf über die Shell</i>	33
2.2.3	<i>Orientierung über laufende Prozesse mit "ps" und „top“</i>	34
2.2.4	<i>Beenden bzw. Abbrechen von Programmen bzw. Prozessen</i>	34
2.2.5	<i>Hintergrundprozesse</i>	35
2.2.6	<i>Sequentielle Ausführung von Prozessen</i>	35
2.2.7	<i>Ausführung von lang laufenden Prozessen mit <i>cp-run</i></i>	35
2.2.8	<i>Systemsicherheit: Logging von Benutzeraktionen</i>	36
2.3	REMOTE-AUSFÜHRUNG VON BEFEHLEN, REMOTE-ZUGRIFF AUF DATEN	36
2.3.1	<i>Secure Shell: Befehle auf anderen Rechner im Netz ausführen lassen per <i>ssh</i></i>	36
2.3.2	<i>Grafikorientiertes Arbeiten auf anderen Rechnern</i>	37
2.3.3	<i>Dateien im Netz übertragen: <i>Secure Copy (scp)</i> oder per <i>FIM-Webspace</i></i>	37
2.4	SHELL-NUTZUNG FÜR FORTGESCHRITTENE	38
2.4.1	<i>Verwendung von Platzhaltern und Wildcards</i>	38
2.4.2	<i>Umlenken von Ein- und Ausgaben, Befehle verbinden über Pipes</i>	38
2.4.3	<i>Das Feintuning der Arbeitsumgebung über die Shell</i>	39
3	LINUX - HINTERGRUNDWISSEN	40
3.1.1	<i>Die UNIX/Linux-Story</i>	40
3.1.2	<i>Bestandteile eines UNIX-Arbeitsplatzes</i>	40
4	ANHANG: LINUX-KURZINFO	41

ANLAGEN:

ÜBUNGSAUFGABEN ZUR RECHNEREINFÜHRUNG



1 Grundlagen

1.1 Die Studierenden-Rechnerpools

1.1.1 Allgemeine Studierenden-Rechnerpools

Den Studierenden der **Universität Passau** steht für Studienarbeiten sowie freies Üben eine Reihe von Rechnerpools zur Verfügung. Die meisten dieser Pools werden i. d. R. im Rahmen des **Computer-Investitions-Programms (CIP)** eingerichtet, man spricht daher auch von **CIP-Pools**.

Die **allgemeinen vom ZIM betriebenen CIP-Pools** bieten als Arbeitsumgebung Windows sowie schwerpunktmäßig Office-Software. Sie finden solche Pools z. B. im Erdgeschoß der FIM (Räume IM 044-047) sowie im benachbarten Gebäude Juridicum. Die Betreuung der **allgemeinen Studenten-rechnerpools** obliegt dem **ZIM** (Zentrum für Informationstechnologie und Medienmanagement), dem allgemeinen IT-Dienstleister der Uni.

Diese Pools können mit Ihrer allgemeinen **Uni-Kennung** genutzt werden. Mit dieser **Kennung (oft auch als „ZIM-Kennung“ oder „Stud-IP-Kennung“ bezeichnet)** können Sie außerdem auf Ihren offiziellen Uni-Mailaccount zugreifen, Services wie das Studieninformationssystem **Stud-IP**, das System **HISQIS** zur Prüfungsanmeldung oder eigene Geräte via **Eduroam** bzw. **VPN-Zugang** in das **Uni-WLAN** einbinden. Bei Problemen mit der allgemeinen Uni-Kennung wenden Sie sich bitte an den **ZIM-Support** (Raum 119 IM).

Das ZIM bietet im Rahmen der O-Wochen vor dem Studium auch regelmäßig eine **Einführung in die allgemeinen IT-Dienste der Universität** an. Die **Folien**, die auf dieser Veranstaltung gezeigt werden, sowie Infos zu weiteren Angeboten des Rechenzentrums finden Sie **unter <http://www.zim.uni-passau.de/erstsemesterinfo>**.

1.1.2 Die FIM-Studierenden-Rechnerpools

Die **FIM** (Fakultät für Informatik und Mathematik) betreibt **eigene Studierenden-Rechnerpools**, um Ihren Studierenden die benötigte **fachspezifische Umgebung** bereit stellen zu können. Für die FIM-Pools benötigen Sie eine **spezielle FIM-Rechnerkennung**.

FIM-Kennung einrichten und verwalten per AMS

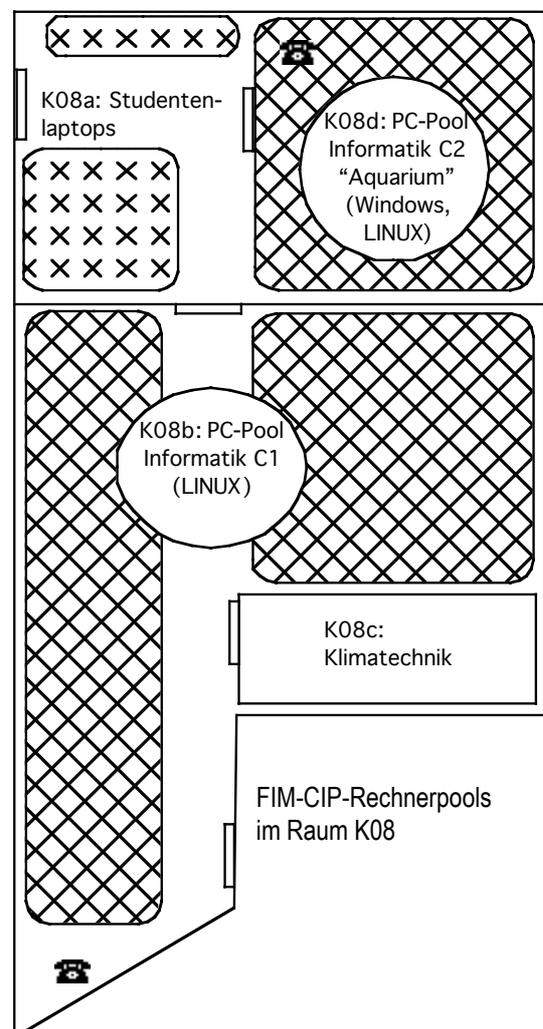
Das Neuanlegen sowie die Verwaltung Ihrer **FIM-Kennung** erfolgt über das **FIM-Account-Management-System AMS**:

ams.fim.uni-passau.de

Diese Website ist nur innerhalb des Uninetzes zugänglich. Vom Rechner daheim aus oder Ihrem Mobilgerät aus benötigen für den Intranet-Zugang **VPN oder Eduroam** (siehe 1.9.5.).

Einrichten Ihrer FIM-Kennung

- Loggen Sie Sich mit Ihrer ZIM-Kennung in einem der **allgemeinen Uni-Rechnerpools** unter Windows ein
- **Alternativ können Sie diesen Prozess auch auf Ihrem Smartphone** oder Heimrechner durchführen, wenn diese mit Eduroam bzw. OpenVPN (siehe später im Skript) ins Uni-Netz eingebunden wurden.



- Öffnen Sie in einem **Browser** die AMS-Website und wählen Sie „New FIM Account“ aus
- Das AMS leitet Sie dann auf Ihren offiziellen Uni-Mail-Account, wo Sie eine für 15 Minuten gültige **PIN** finden, um **die Einrichtung Ihrer FIM-Kennung abzuschließen**.
- **Am Ende** des Vorgangs erhalten Sie eine **zweite Mail mit Ihrem FIM-Kennungsnamen**, andernfalls einen Fehlerbericht.

FIM-Kennung verlängern: FIM-Kennungen müssen in bestimmten Abständen verlängert werden. Sie erhalten dann eine entsprechende Warnmail, so dass Sie über das FIM-AMS rechtzeitig verlängern können. Mit dem Auslaufen Ihrer allgemeinen Unikennung erlischt auch Ihre FIM-Kennung.

Wenn Sie das FIM-AMS nicht nutzen können, da Sie z. B. über keine allgemeine Uni-Kennung verfügen, auf der das System aufbaut, oder es ein Problem mit dem AMS gibt, erhalten Sie **Hilfe beim FIM-Support (persönlich in Zi. 014 IM oder per Mail)**. Bei allen Support-Anfragen, die die Verwaltung einer Kennung betreffen (neu einrichten, Passwortänderung...) ist entweder eine CampusCard (mit Foto) oder ein Lichtbildausweis vorzulegen. So können wir feststellen, ob Sie auch berechtigt sind, für eine bestimmte FIM-Kennung Änderungen zu veranlassen.

Räume und Öffnungszeiten der FIM-Pools

Im **Untergeschoß IM** finden Sie die FIM-Pools für individuelles, **freies** sowie Vorlesungsbegleitendes **Üben (hausintern per Tel erreichbar über die letzten vier Ziffern)**:

- **Raum K08b (PC-Pool Informatik C1):** Telefon erreichbar über 0851/509-3015
- **Raum K08d (PC-Pool Informatik C2 „Aquarium“):** Telefon 0851/509-3136

Zugang und Öffnungszeiten: Beide Poolräume sind **durchgehend zugänglich**. Nachts oder am Wochenende kommt man in das Gebäude IM (und damit in die Rechnerpools) über den Kartenleser am Gebäudeeingang. Der Zugang erfolgt mit Ihrer **CampusCard** (Zugangsrecht ist automatisch frei geschaltet bei Studierenden in FIM-Bachelor und Masterstudiengängen; bei FIM-Lehrmatsstudiengängen bitte Zugang bei Bedarf im FIM-Dekanat (Raum IM 216) frei schalten lassen).

Daneben betreibt die FIM noch den **Raum 028 IM (Tutorienrechnerpool)**, der allerdings nur im Rahmen **betreuter Rechnerübungen** zugänglich ist.

Für die Erstellung von Studienarbeiten und Praktika gibt es weitere Rechner an den Lehrstühlen.

Betriebssysteme in den FIM-Studentenrechnerpools

- **Raum K08b: Ausschließlich Linux verfügbar.** Ein **Ausschalten bzw. ein Neustart dieser Rechner ist verboten**, da auf diesen Rechnern regelmäßig Berechnungen laufen, die Tage oder sogar Wochen dauern. Ein Ausschalten dieser Rechner würde also evtl. viel Arbeit kaputt machen.
- **Raum K08d sowie im Raum 028 IM: Linux sowie Windows.** Beim Bootvorgang, d. h. beim Rechnerstart, treffen Sie die Wahl des gerade benötigten Betriebssystems. Bitte schalten Sie einen Rechner, falls er nicht unter dem passenden Betriebssystem läuft, nicht einfach mit dem Ausschaltknopf aus, sondern führen Sie einen Neustart über das Bildschirmmenü aus.

Support / Ansprechpartner für die FIM- Studentenrechnerpools

Hauptamtliche Mitarbeiter: Klaus Schießl (Dipl.-Inf.) Innstraße 33, Raum 017 IM, Tel. 0851/509-3012 E-Mail: schiessl@fim.uni-passau.de Franz Rautmann (Dipl.-Ing.) Innstraße 33, Raum 014 IM, Tel. 0851/509-3013 E-Mail: rautmann@fim.uni-passau.de	Studentische Mitarbeiter: <i>Sven Gebauer</i> <i>Jakob Jüger</i> <i>David Klinger</i>
---	--

Problemmeldungen im Bereich der FIM-Rechner **persönlich bei uns im Büro**, per **Rechnerraum-Telefon** Nummer 3012 bzw. 3013 oder **per E-Mail** an support@fim.uni-passau.de.

1.1.3 Die Hausordnung der FIM-Studierenden-Rechnerpools

Achtung: Verstöße gegen die Hausordnung können mit Entzug der Rechnerkennung geahndet werden. Delikte im Umgang mit Daten, Software und Rechnern werden straf- und zivilrechtlich verfolgt!

- Beachten Sie die [Benutzerrichtlinien](https://www.fim.uni-passau.de/it-dienste/fim-kennungen/benutzungsordnung/) !
(<https://www.fim.uni-passau.de/it-dienste/fim-kennungen/benutzungsordnung/>)
- **Essen, Trinken, Computerspielen und Rauchen** in den Rechnerräumen sind **untersagt**
- Die Fenster in den Rechnerräumen sind geschlossen zu halten, da sonst die Klimaanlage nicht mehr richtig arbeitet. Wenn es zu generell zu heiß oder zu kalt in den Pools ist, geben Sie uns bitte Bescheid.
- **Personenbezogene bzw. private Daten** (z. B. MP3-Audiodateien, Filme) dürfen nicht gespeichert werden. Insbesondere ist die widerrechtliche Verbreitung urheberrechtlich geschützter Daten untersagt.
- **Nur Sie selbst dürfen unter Ihrer Kennung arbeiten.** Sie sind auch dafür verantwortlich, was auf Ihrer Kennung passiert. Viele Aktionen werden unter Linux und Windows automatisch temporär mitprotokolliert und können Ihnen im Ernstfall nachgewiesen werden.
- Vergewissern Sie sich, dass Sie sich an allen Geräten, an denen Sie sich einloggen, beim Verlassen des Arbeitsplatzes auch wieder **ausloggen**, da sonst Unbefugte auf Ihrer Kennung Schaden anrichten können. In diesem Fall sind Sie für Schäden haftbar.
- Die **Weitergabe von FIM-Passwörtern ist nicht gestattet.**
- Sofern nicht explizit erlaubt (Beispiel: erlaubt sind z. B. USB-Datenträger) ist es verboten, eigene **Geräte** an die Poolrechner anzuschließen. Fragen Sie im Zweifelsfall vorher bei der FIM-Systembetreuung an!
- An unseren Rechnern können Sie unter Linux eine **Lock-Funktion** benutzen, um Ihren Arbeitsplatz für eine kleinere Pause zu reservieren. Bedenken Sie aber bitte, dass auch andere Studenten Zugang zu den Rechnern brauchen. Bei Missbrauch der Lock-Funktion droht Entzug der Rechnerkennung.
- Sie können auf den Pool-Rechnern der FIM **eigene Web-Homepages** anlegen (siehe Ende Kapitel 1). Wir weisen in diesem Zusammenhang darauf hin, dass die Rechner laut Benutzerantrag nur für Studienzwecke benutzt werden dürfen und sich auch der Inhalt Ihrer FIM-Webseiten in diesem Rahmen bewegen muss.

1.2 Ein- und Ausloggen in den FIM-Rechnerpools

Mit Ihrem **FIM-Kennungsnamen** können Sie sich in den FIM-Pools **sowohl unter Linux als auch – sofern auf dem jeweiligen Rechner verfügbar – Windows** einloggen. Achten Sie bitte bei der Eingabe von Kennung und Passwort auf **Groß- und Kleinschreibung!**

1.2.1 Ein- und Ausloggen unter der grafischen Oberfläche

Da die FIM-Rechner für den Mehrbenutzerbetrieb ausgelegt sind, müssen Sie sich zu Beginn der Sitzung am Rechner authentifizieren. Hierzu gehört die Eingabe des Benutzernamens sowie eines Passworts. Dieser Vorgang wird als **Einloggen** oder auch **Login** bezeichnet.

- Beim grafischen Login empfängt Sie ein Fenster, wo Sie Ihre **Benutzerkennung** und dann Ihr **Password** eingeben.
- **Besonderheiten vor dem ersten Linux-Login**
 - In der Rechnereinführung nutzen wir die grafische Oberfläche **Xfce mit dem Window-Manager xfwm** (siehe auch nächstes Kapitel), die Sie **als Default einstellen** sollten.
 - Wenn Sie beim Login **im FIM-Pool** sitzen: Wählen Sie links unterhalb des Login-Felds XFCE als **grafische Benutzeroberfläche aus**. Wenn Sie Remote per X2Go arbeiten, ist auf unserer [Website](#) erklärt, wie man XFCE einstellt.
 - **Verlassen der grafischen Oberfläche in XFCE: Maus nach rechts oben** bewegen, auf Benutzernamen klicken und „**Abmelden**“ auswählen.
- **Besonderheiten beim Windows-Login (nur im Pool, nicht aber remote möglich)**
 - Achten Sie darauf, dass die Anmeldung an der Domain „**FIMSMB**“ eingestellt ist

Das **Ausloggen** aus der grafischen Oberfläche erfolgt allgemein über Menüs, die sich je nach verwendeter grafischer Oberfläche und Betriebssystem am oberen oder unteren Bildrand befinden.

1.2.2 Passwörter

1.2.2.1 Zulässige Passwörter

Für die Wahl eines neuen Passwortes sollten Sie unbedingt die folgenden **Sicherheitsregeln** berücksichtigen:

1. Die gewählte Zeichenfolge muss aus **8, sollte besser aber aus mindestens 10 Zeichen** bestehen
2. Im Passwort müssen mindestens **drei der folgenden Zeichenklassen** enthalten sein: **Kleinbuchstaben, Großbuchstaben, Ziffern, Sonderzeichen**
3. Ihr Passwort darf **in keinem Wörterbuch** der Welt zu finden sein
4. **Persönliche**, jedermann zugängliche **Daten** wie Ihren Benutzernamen, Namen oder Vornamen vermeiden
5. **Keine Eigennamen** oder **geographischen Begriffe** verwenden
6. Begriffe aus den Kategorien 3-5 einfach nur in umgekehrter Reihenfolge, d. h. **von hinten nach vorn buchstabiert** sind verboten
7. Sie sollten sich das Passwort merken können, ohne es aufzuschreiben.
8. Das FIM-AMS verhindert, dass Sie das zuletzt genutzte Passwort als neues Passwort wieder verwenden können.

1.2.2.2 Passwort ändern / Passwort vergessen

Ändern Sie bitte Ihre Passwörter in regelmäßigen Abständen. Bei neuen Kennungen sind die Passwörter anfangs für beide Systemwelten, d. h. die FIM-Windows-Umgebung sowie die FIM-Linux-Umgebung **gleich eingestellt**.

Wenn Sie Ihr Passwort ändern wollen oder Ihr Passwort vergessen haben, können Sie sich selbst ein neues Passwort mit dem **FIM Account-Management-System (ams.fim.uni-passau.de)** setzen. Dabei wird für alle FIM-Arbeitsumgebungen (Windows und LINUX) einheitlich ein neues Passwort gesetzt.

1.2.2.3 Verwendung eines Passwort-Managers

Mit dem Studiumbeginn werden Sie sich zusätzliche neue Passwörter merken müssen. Wir empfehlen den Einsatz eines Passwort-Managers (z. B. das kostenlose, in Varianten für viele Geräteklassen verfügbare **KeePass**), um diese Passwörter an einer Stelle zu sammeln und zu verwalten. Sie brauchen sich dann nur noch das Zugangspasswort zur KeePass-Datenbankdatei zu merken, um an die anderen Passwörter zu kommen. Bitte nicht auf einen Backup der KeePass-Datenbankdatei vergessen!

1.3 Arbeiten unter graphischen Benutzeroberflächen

Eine **grafische Benutzeroberfläche** ist eine **Softwarekomponente**, die dem Benutzer eines Computers die Interaktion mit der Maschine über grafische Symbole erlaubt. Da Sie mit hoher Wahrscheinlichkeit ausreichend Erfahrung beim Arbeiten unter der grafischen Benutzeroberfläche von Windows mitbringen, konzentriert sich diese Einführung im Folgenden auf das Arbeiten unter Linux.

Die **Basistechnologie** bei graphischen Benutzeroberflächen im Linux-Bereich heißt **X-Windows**. **X-Windows** stellt Bausteine und Regeln für den Bau einer grafischen Oberfläche zur Verfügung und besteht aus einer **Server-Komponente** sowie darauf aufsetzenden **Clients**.

- Der **X-Server** läuft auf dem Rechner am Arbeitsplatz und stellt seine (grafischen) Dienste den X-Clients zur Verfügung. Er enthält den Grafikkartentreiber sowie Treiber für Tastatur, Maus und andere Eingabegeräte und kommuniziert mit dem X-Client.
- Ein **X-Client** ist ein Anwendungsprogramm wie z. B. ein Webbrowser, das die grafischen Ein-/Ausgabe-Dienste des X-Servers benutzt. Er kann auf demselben, oder auch auf irgendeinem entfernten Rechner laufen (sofern eine Netzwerkverbindung zwischen beiden besteht). Der X-Client benutzt die Dienste des X-Servers, um eine grafische Darstellung zu erreichen, und empfängt von ihm die diversen Ereignisse wie Tastenanschläge, Mausbewegungen, Klicks usw.

1.3.1 Grafische Arbeitsumgebungen unter Linux

Unter Linux haben Sie die **Wahl zwischen mehreren auf X-Windows basierenden grafischen Arbeitsumgebungen**. Diese unterscheiden sich voneinander durch Ihre **Fenster-Manager (Window-Manager)**, **Toolkits** zur Programmierung der grafischen Oberfläche sowie eine Reihe von **Anwendungen** zum bequemen Arbeiten in der jeweiligen Umgebung. Der **Window-Manager** kümmert sich um **Aufbau und Aussehen der Objekte am Bildschirm**. Konkret verwaltet der Window-Manager z. B. Fenster, das Scrollen, Menüs sowie die Icons (kleine Symbole für Fenster).

In dieser Rechnereinführung stellen wir die Oberfläche **Xfce** vor. Xfce ist die am zweitmeisten verbreitete Desktop-Umgebung (nach KDE). Gründe hierfür sind die einfache Bedienbarkeit in Verbindung mit einem geringen Ressourcenverbrauch. Poweruser nutzen gern die Umgebung „i3“. Sie selbst entscheiden jeweils bei der LINUX-Anmeldung, welche Kombination von Arbeitsumgebung/Window-Manager Sie später verwenden wollen.

1.3.2 Das Arbeiten mit der LINUX Desktop-Umgebung Xfce

Xfce (<https://xfce.org/>) ist modular aufgebaut und besteht aus **Kernkomponenten** wie

- **der Arbeitsfläche (auch als Schreibtisch bezeichnet)** sowie
- konfigurierbaren **Leisten** (vgl. folgende Abbildung)
- ein **Anwendungsmenü**, über das Sie Anwendungen auswählen und Einstellungen vornehmen können.

Dazu kommen **Zusatzkomponenten** wie z. B.

- **der Anwendungsfinder**, um Programme zu suchen und auszuführen, eine
- **eine Terminal-Anwendung**, um befehlsorientiert zu arbeiten sowie der
- **der Dateimanager Thunar**.

Beim **ersten Start des Xfce** werden Sie evtl. gefragt, wie Sie die Xfce-Leisten initial konfiguriert haben möchten. Wählen Sie dann die **StandardEinstellung** aus. Anpassungen können Sie später noch vornehmen.

Auf der nächsten Seite finden Sie die Beschreibung der XFCE-Arbeitsfläche, wie Sie von uns für Sie voreingestellt wurde. Die weitere Anpassung obliegt Ihren eigenen Vorstellungen.

1a

1b

2

3

1c

1d

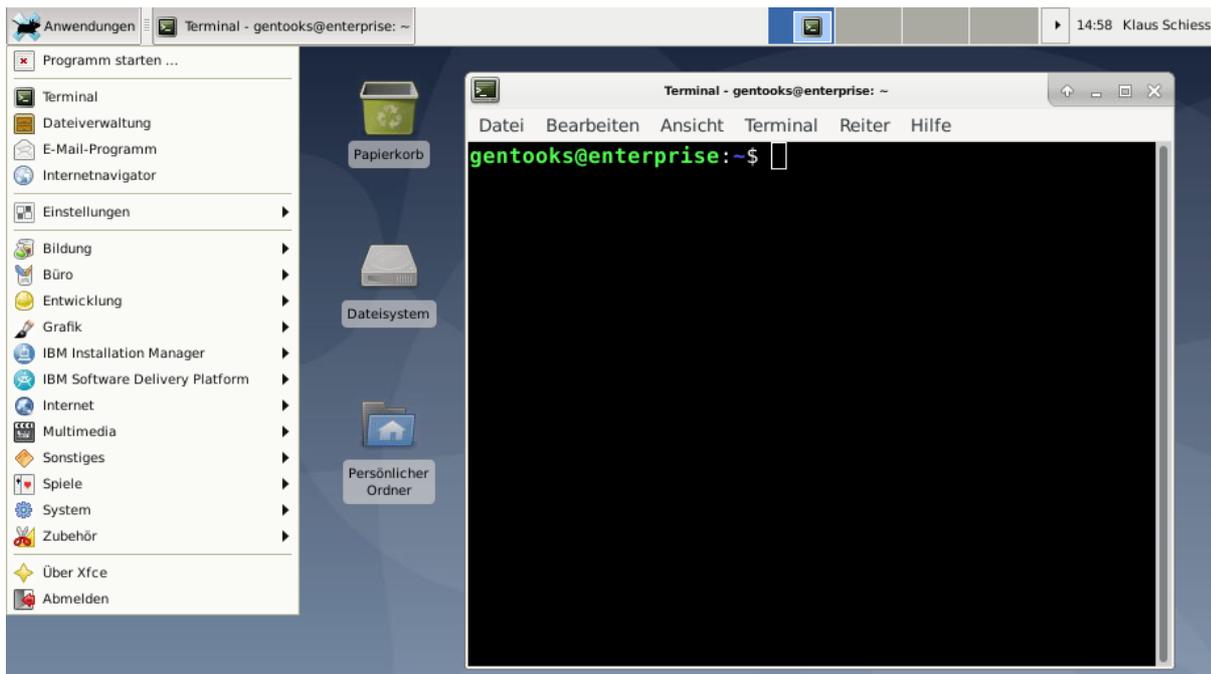


Abbildung: Bildschirmanzeige des Xfce nach Öffnen des Anwendungsmenüs sowie eines Terminals

Voreingestellte Standardelemente der Arbeitsoberfläche (Schreibtisch):

1. Standard-Elemente der oberen Leiste:

- a. **Anwendungsmenü:** Programme starten und Einstellungen für System und Arbeitsoberfläche vornehmen. Die wichtigsten Punkte sind ganz oben:
 - i. „**Programm starten**“ öffnet ein Fenster mit dem **Anwendungsfinder**, um Programme zu suchen und auszuführen. Eine Suche starten Sie, indem Sie einfach den Programmnamen eintippen. Wenn Sie auf den **grünen Pfeil** klicken, wird die Anzeige ausführlicher. Oft benötigte Programme können Sie als **Starter auf der Arbeitsoberfläche oder in der Leiste** platzieren, indem Sie das Programmicon mit gedrückter linker Maustaste an den gewünschten Ort ziehen.
 - ii. Darunter finden Sie **vier wichtige Anwendungen (Terminal, Dateiverwaltung Thunar, Standard-Email-Programm, Standard-Internetnavigator)**
- b. **Fensteranzeige:** Icons zu Fenstern der gerade laufenden Anwendungen. In der Abbildung sieht man den Titel des Fensters, das zur Anwendung Terminal gehört.
- c. **Virtuelle Arbeitsflächen (Arbeitsflächen-Umschalter):** Hier können 4 virtuelle Desktops nach Bedarf auf den Monitor geschaltet werden.
- d. **Uhr / Sitzungssteuerung:** Hier kann man den Bildschirm sperren oder sich abmelden.

2. **Desktop-Icons auf Schreibtisch:** In der Voreinstellung werden drei Icons auf den Desktop gelegt. Oben sehen Sie Ihren „Papierkorb“ für zu löschende Objekte. Ein Anklicken der Icons „Dateisystem“ bzw. „Persönlicher Ordner“ öffnet den Dateimanager **Thunar**, entweder mit allen Systemdaten oder Ihrem persönlichen Datenbereich.

3. **Fenster zur Anwendung Terminal:** Die Anwendung Terminal stellt ein Fenster zur Verfügung, in dem man textbasiert Kommandos eingeben kann. Man sagt auch **Shell-Fenster** nach dem darin laufenden Kommandointerpreter, der als Shell bezeichnet wird. Das Aussehen von Konsolen-Fenstern ist über die Menüleiste veränderbar. Sie können Terminal-Fenster z. B. über das Anwendungsmenü (1a) oder die untere Leiste (4b) starten.

Das Arbeiten mit Fenstern sowie der Maus unter Xfce

Mit der Maus können Sie Objekte manipulieren (z. B. Fenster), auswählen sowie Menüs bedienen. In Abhängigkeit davon, wo sich die Maus befindet, besitzen die Maustasten unterschiedliche Funktionen.

Fensterverhalten steuern mit der Maus

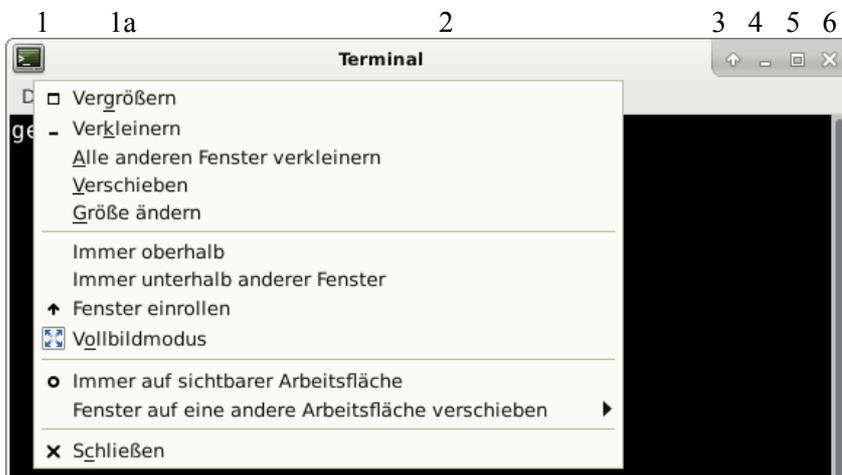


Abbildung: Fenster der Anwendung Terminal mit Xfce-Titelleiste

Aufbau der Xfce-Titelleiste von Anwendungsfenstern

- 1) **Anwendungssymbol:** Fensterverhalten steuern (siehe ausgeklapptes Menü 1a)
- 2) **Titel der zum Fenster gehörenden Anwendung**
- 3) **Fenster einrollen:** Symbol „Pfeil nach oben“; Fenster wird auf Titelleiste reduziert
- 4) **Minimieren:** das Fenster verschwindet von der Arbeitsfläche, der Fenstertitel wird aber in der Xfce-Leiste weiter angezeigt. Hierüber kann man das minimierte Fenster wieder auf der Arbeitsfläche anzeigen lassen.
- 5) **Maximieren:** Symbol „Pfeil nach oben“; das Fenster wird bildschirmfüllend angezeigt.
- 6) **Schließen:** Klickt man auf das X rechts in der Titelleiste, wird die Anwendung beendet
- 7) **Größe des Fensters ändern:** Klicken auf Rand des Fensters, dann Ziehen in gewünschte Größe
- 8) **Fenster bewegen:** In Titelleiste klicken und mit gedrückter Maustaste das Fenster zum Zielort ziehen.

Text zwischen Anwendungen kopieren

In grafischen Arbeitsumgebungen auf Linux-Systemen kopiert man Text von einem Fenster in ein anderes, indem man mit dem Mauszeiger zum Beginn der **Textsequenz** geht und **mit gedrückter linker Maustaste** den zu kopierenden **Text markiert**. Nun lässt man die Maustaste los und bewegt den Mauszeiger dahin, wo der Text eingefügt werden soll. Durch **Drücken der mittleren Maustaste bzw. des Scrollrads** wird der kopierte **Text an Position des Cursors eingefügt**. Über **Doppelklick** mit der linken Maustaste kann man einfach ein **einzelnes Wort** zum Kopieren markieren, über **Dreifachklick** eine ganze **Zeile**.

1.3.3 Bildschirmsperre

Über die **Funktion Bildschirmsperre** schützen Sie Ihren Rechnerarbeitsplatz **vor unbefugtem Zugriff**, wenn Sie beim Arbeiten eine kleine Pause einlegen wollen. Sie starten die Sperre über die Sitzungssteuerung rechts oben in der Xfce-Leiste. Durch Eingabe Ihres **Passwords** können Sie die Sperre wieder aufheben. Das **missbräuchliche Sperren** von Arbeitsplätzen **über einen längeren Zeitraum zu Reservierungszwecken ist verboten!**

1.3.4 Programme beenden

Zum **Beenden der Programme** verwenden Sie bitte unbedingt die **Befehle**, die Ihnen das jeweilige Programm in Form eines **Menüeintrags, Buttons oder einer Tastenkombination** zur Verfügung stellt und nicht das Kill-Symbol in der Titelleiste des Programmfensters. Programme wie **Firefox** oder **Thunderbird** hinterlassen sonst Datenmüll und lassen sich beim nächsten Aufruf dann evtl. nicht mehr sauber starten.

1.4 Grundbegriffe zum Kommando-orientierten Arbeiten in der Linux-Shell

1.4.1 Die Shell als Grundlage für die Eingabe von Linux-Befehlen

Im Folgenden werden die Begriffe „**Befehl**“ und „**Kommando**“ **synonym** verwendet. Um **Linux-Befehle ausführen lassen** zu können benötigen Sie die Hilfe eines Systemprogramms namens **Shell**, das Ihre **Eingaben als Befehle erkennt** und **für deren Ausführung sorgt**. Die **Aufgabe** der Shell ist also die eines **Kommando-Interpreters**. Auf den Linux-Rechnern haben wir für Sie eine Shell mit Namen **bash (Bourne again shell)** voreingestellt.

Unter der **grafischen Oberfläche** erhalten Sie eine **Shell**, indem Sie ein **Konsole-Terminal starten**.

Nach dem Start zeigt Ihnen die Shell durch eine besondere Zeichenfolge, den **Prompt**, dass der Computer bereit ist, Eingaben zu akzeptieren. Der **Prompt der bash** sieht standardmäßig auf FIM-Studierendenrechnern so aus:

```
mueller@hecht: ~ $
```

- Am Anfang steht der **Name des Benutzers**, der die Shell gestartet hat,
- Nach dem Trennzeichen “@” (gesprochen: at) folgt der **Name des Rechners** (hier „hecht“), auf dem die Shell läuft.
- Nach dem Doppelpunkt wird der **Ort im Dateisystem angezeigt**, wo Sie gerade arbeiten, entweder als **Pfadangabe** (siehe nächstes Kapitel) oder wie oben als **Symbol** (hier: Symbol **Tilde** „~“; dieses Symbol steht für Ihr **Home Directory**, d. h. das Directory, wo Ihre persönlichen Daten liegen).
- Das Zeichen „\$“ bildet den Abschluss des Prompts. Hiernach können Sie Ihre Befehle eingeben.

Wie dieser Prompt aussieht, ist über Systemeinstellungen konfigurierbar, kann also an die Vorlieben des Benutzers angepasst werden. Nach dem Prompt gibt der Benutzer Befehle ein.

Ein **Befehl** wird nach **Drücken der Return - Taste <Ret>** von der Shell zur **Ausführung** gebracht.

Zum **Beenden einer Shell-Sitzung** („**Logout**“) gibt es mehrere Alternativen:

- Kommando **logout** **eintippen** (dann Return-Taste drücken; geht nur in einer Login-Shell)
- Oder alternativ: Drücken der Tastenkombination **<Strg-d>**
- Auch das Kommando **exit** funktioniert in manchen Situationen zum Logout.

1.4.2 Linux-Befehle eingeben und ausführen lassen

1.4.2.1 Die Struktur von Linux-Befehlen

Befehlszeilen besitzen folgenden **Aufbau**:

Befehl -Option(en) Parameter

Optionen (auch **Flags** genannt), steuern die **Wirkungsweise** des Befehls

Optionen ist i.d.R. das Zeichen "-" vorangestellt, seltener ein "+", manchmal auch gar nichts. Bei Befehlen mit mehreren möglichen Optionen kann man diese in der Befehlszeile entweder hintereinander angeben:

Befehl -opt1 -opt2...

oder oft auch zusammenfassen:

Befehl -opt1opt2...

Parameter geben **ergänzende Informationen** zum Befehl (z. B. Dateiname einer zu löschenden Datei)

Beispiel 1: `uname`

Der Befehl `uname` für sich alleine gibt aus, unter welchem Betriebssystem Ihr Rechner läuft

Beispiel 2: `uname -a`

Gibt man als Option den Schalter "-a" mit an erhält man zusätzlich ausführliche Systeminfos angezeigt wie z. B. den Rechnernamen oder die Version des verwendeten Linux-Kernels.

Beispiel 3: `cp quelle.txt ziel.txt`

Der Befehl `cp` kopiert Objekte. Über zwei Parameter ist hier angegeben, was kopiert werden soll (Datei `quelle.txt`) und wie die Kopie heißen soll (Datei `ziel.txt`).

Wichtig: Für dieses Skript gilt, dass alle **kursiv gedruckten Begriffe in Linux-Befehlszeilen nicht buchstabengetreu abzutippen** sind, sondern von Ihnen **sinngemäß ersetzt** werden sollen (so ist z. B. *kennung* durch Ihre Benutzerkennung im Pool zu ersetzen).

Linux unterscheidet zwischen Groß- und Kleinschreibung: Dies gilt sowohl bei der Eingabe von Befehlen als auch später beim Umgang mit Datei- oder Directorynamen. Falls Sie im obigen Beispiel 1 den Befehl „`uname`“ nicht klein schreiben, sondern z. B. „`Uname`“, wird der Befehl nicht erkannt und das System bringt eine Fehlermeldung.

Korrektur von Tippfehlern/Eingabefehlern: Über die **Pfeiltasten** kann man in einer Befehlszeile die Einfügemarke zum Tippfehler bewegen, dann mit der **Delete-Taste** oder der **Backspace-Taste** den Fehler löschen und die Korrekturen einfügen.

Eingeben langer Befehlszeilen: Eine Befehlszeile wird normal nach Drücken der Return-Taste sofort ausgeführt. Wollen Sie einen Befehl eingeben, der sich über mehrere Zeilen zieht, geben Sie als **letztes Zeichen der Zeile** das Symbol "\" ein bevor Sie die Return-Taste betätigen. Dann wird die Zeilenschaltung von der Shell nicht als Signal zum sofortigen Ausführen interpretiert und Sie können in der nächsten Zeile den Befehl weiterschreiben.

1.4.2.2 Befehle für den Alltag

<code>mensa</code>	Zeigt den Mensa-Speiseplan an
<code>date</code>	liefert Informationen über die aktuelle Uhrzeit
<code>clear</code>	Bildschirminhalt des Konsolen-Fensters leeren
<code>whoami</code>	zeigt an, welchem Benutzer eine Shell zugeordnet ist, d.h. wer gerade mit Ihrer arbeitet
<code>w</code>	zeigt an, welche Benutzer am Rechner angemeldet sind und was sie ausführen

1.4.2.3 Bequemes Arbeiten mit der Bash (oder: Linux für Schreibfaule)

Eine Bequemlichkeit der Bash ist deren **Ergänzungsfunktion** („File Name Completion“, d. h. Vervollständigung von Datei- bzw. Kommandonamen). Wenn Sie den Namen eines Befehls oder einer Datei eingeben, hören Sie nach der Anzahl von Buchstaben auf, die den Befehl oder den Dateinamen eindeutig macht und betätigen die **Tab-Taste**. Die Bash ergänzt dann den fehlenden Text und Sie sparen sich das Eintippen der restlichen Zeichen.

Alte Befehle wiederholen: Ein Service der Bash besteht darin, dass sie sich **Eingaben merkt**, so dass man schon eingegebene **Befehlszeilen sehr einfach nochmals ausführen** lassen kann. Dies ist besonders bei langen Kommandozeilen eine erhebliche Erleichterung. Auf alte Befehle kann man ganz einfach über die **Pfeiltasten** zugreifen (Pfeil nach oben bzw. unten drücken, bis man den Befehl gefunden hat, den man noch einmal ausführen möchte) und durch Drücken der Return-Taste **wieder ausführen lassen**.

Fortgeschrittene Aspekte: Alternativ zu den Pfeiltasten können Sie spezielle Befehle nutzen, um sich **frühere Befehle auflisten** zu lassen **und** diese noch einmal **auszuführen**:

<code>history</code>	Gibt eine Anzahl zuletzt eingegebener Kommandos auf dem Bildschirm aus (Befehls-History), wobei die Kommandos in der Reihenfolge ihrer Eingabe numeriert sind
<code>!nummer</code>	Sieht in der Befehls-History nach und wiederholt das Kommando, das zur angegebenen Nummer gehört
<code>!!</code>	Wiederholt die Ausführung des letzten eingegebenen Kommandos
<code>!!text</code>	Wiederholt das letzte eingegebene Kommando und hängt als Modifikation des Kommandos die Zeichenfolge "text" an
<code>!text</code>	Wiederholt das letzte Kommando, das mit der angegebenen Zeichenfolge "text" begann

1.4.2.4 Hilfen beim Umgang mit Befehlen

Niemand kann sich alle Shell-Befehle samt möglicher Optionen merken. Am einfachsten erhält man eine **Kurzübersicht zum genauen Gebrauch eines Befehls**, wenn man in der Shell den Befehlsnamen und dann die **Option "-- help"** am Bildschirm eintippt. Eine Kurzinfo zu `date` erhält man z. B. über `date --help`.

Genauere Informationen zu einem Kommando liefern die jeweiligen **Manual Pages**:

- **Aufruf einer Manual-Page:** Um genaueres über einen Linux-Befehl zu erfahren, gibt man in der Shell `man befehlsname` ein. Um z. B. mehr über das Kommando `date` zu erfahren, gibt der Benutzer `man date` ein.
- **Navigieren in der Anzeige einer Manual-Page:** Mit der **Leertaste (Space)** blättert man **Bildschirm-seitenweise**, mit der **Return-Taste** **zeilenweise** durch eine Manual-Page. Durch Eingabe von **"b"** (für "back") kann man wieder eine Bildschirmseite **zurückblättern**. Die **Ausgabe** einer Manual Page wird durch **"q"** (für "quit") **abgebrochen**
- **Suche nach Begriffen:** Innerhalb der angezeigten **Manual-Page** kann man **nach Begriffen suchen**, indem man zuerst `,` und dann den **Suchbegriff** eingibt (z. B. `/examples`) und schließlich die Returntaste drückt. Die erste Stelle der Manualpage, wo der Begriff auftaucht, wird hierauf am Bildschirm angezeigt. Drückt man anschließend **"n"**, springt man zum **nächsten Auftreten des Suchbegriffs**.

<p>W(1) Linux User's Manual W(1)</p> <p>NAME w - Show who is logged on and what they are doing.</p> <p>SYNOPSIS w - [husfV] [user]</p> <p>DESCRIPTION w displays information about the users currently on the machine, and their processes. The header shows, in this order, the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes....</p> <p>COMMAND-LINE OPTIONS -h Don't print the header. -u Ignores the username while figuring out the current process and cpu times. To demonstrate this, do a "su" and do a "w" and a "w -u"....</p> <p>FILES <u>/var/run/utmp</u> information about who is currently logged on <u>/proc</u> process information</p> <p>SEE ALSO free(1), ps(1), top(1), uptime(1), utmp(5), who(1)</p> <p>AUTHORS w was re-written almost entirely by Charles Blake, based on the version by Larry Greenfield <greenfie@gauss.rutgers.edu> and Michael K. Johnson <johnsonm@redhat.com>.</p>	<p>Links oben steht der Name des Befehls, zu dem die angezeigte Manual-Page gehört. Daneben steht in Klammern, aus welcher Unterabteilung (Section) des Manual-Systems die angezeigte Manual-Page stammt. Die Abbildung zeigt eine Manual-Page zum Kommandonamen "w", wobei die Manual-Page aus der Section 1 stammt.</p> <p>Danach finden Sie unter "NAME" den Namen des Befehls sowie eine Kurzbeschreibung von dessen Wirkung.</p> <p>Der Punkt "SYNOPSIS" sagt Ihnen, welche Syntax für den Gebrauch des Befehls genau zu benutzen ist. Die fünf Optionen "husfV" sind von eckigen Klammern umgeben, die beim tatsächlichen Gebrauch des Kommandos nicht mit eingetippt werden dürfen. Sie selbst entscheiden nach Bedarf, ob bzw. welche Optionen sie mit angeben.</p> <p>Der Punkt "DESCRIPTION" erläutert Ihnen ausführlich die Einsatzmöglichkeiten und den Gebrauch des Kommandos.</p> <p>Unter "COMMAND-LINE-OPTIONS" finden Sie alle Optionen noch mal im Einzelnen erklärt.</p> <p>Bei manchen Manpages finden Sie auch einen Punkt "FILES", der angibt, welche Dateien ein Befehl bei seinem Gebrauch benötigt oder beeinflusst.</p> <p>Manchmal finden Sie einen Punkt "EXAMPLES", der Beispiele für den Gebrauch des Befehls liefert.</p> <p>Schließlich finden Sie unter "SEE ALSO" noch die Namen anderer Befehle, die mit dem Gebrauch des Befehls zusammenhängen.</p>
---	--

Abbildung: Interpretation von Manpages am Beispiel der Manpage zum Kommando „w“

Fortgeschrittene Aspekt, wenn Sie mehr mit der Shell arbeiten wollen:

- **Schlagwortsuche über alle Manual Pages:** Um sich eine Liste der Manual-Pages ausgeben zu lassen, in denen ein bestimmter Begriff vorkommt, benutzt man die Option "-k". "man -k date" liefert z. B. eine Liste aller Manual Pages, in denen der Begriff „date“ vorkommt.
- **Aufruf von Manual-Pages, wenn ein Befehlsname im Linux-System mehrfach belegt ist:** Bedingt durch den Aufbau von Linux-Systemen kommt es vor, dass Befehle mit unterschiedlicher Funktion den gleichen Kommandonamen tragen, also bestimmte Kommandonamen wie z. B. „uname“ doppelt belegt sind. Zur Unterscheidung der Manual-Pages zu gleich lautenden Kommandonamen sind diese in unterschiedlichen Teilbereichen (**Sections**) des Manual-Systems abgelegt. Möchte man z. B. wissen, in welchen Sections Manual-Pages zu uname existieren, gibt man „**whatis** uname“ ein. „man uname“ liefert nämlich nur die erste gefundene Manual-Page zu diesem Kommandonamen. Das muss aber nicht unbedingt die von Ihnen gesuchte sein. Kennt man die Section, wo die gewünschte Manual-Page liegt, kann man über die Option „-s“ gezielt diese eine Manual-Page anzeigen lassen. „man -s 2 uname“ liefert z. B. die Manual-Page zu uname in Manual-Section 2. Um sich alle Manual-Pages zu einem Begriff anzeigen zu lassen, gibt man „man -a **begriff**“ ein, im Beispiel also „man -a uname“.

1.5 Der Umgang mit Dateien und Directories

1.5.1 Dateien und Directories: Begriffe und Grundlagen

1.5.1.1 Der Aufbau des Dateisystems

Wenn Sie auf dem Computer etwas abspeichern, werden Ihre Daten in **Dateien** abgelegt und verwaltet. Egal ob eine Datei einen Text oder ein JPG-Bild enthält, rein technisch ist eine **Datei** aus Sicht des Rechners immer nur eine **Abfolge von Zeichen** („sequentielle Folge von **Bytes**“). Wie kann man nun

auf einem Mehrbenutzer-System eine Vielzahl von Dateien, die noch dazu verschiedenen Benutzern gehören, geeignet sortieren und verwalten?

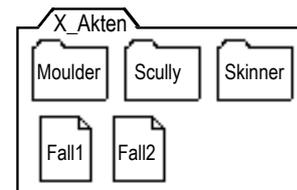
Für den Einstieg soll zunächst die Datenablage in einem Büro betrachtet werden: In jedem Büro geht man mit Daten um, die in Dokumenten erfasst werden. Dokumente zum selben Thema können in Ordnern zusammengefasst werden. Jeder Ordner kann selbst wiederum Bestandteil einer weiteren übergeordneten Ablageebene sein. Die Gesamtheit aller Dokumente und der ineinander verschachtelten Ordner bildet das Gesamtarchiv.

Auf Rechnern werden Daten analog verwaltet:

- Ein Dokument entspricht einer **Datei** (engl.: **File**).
- Dem Ordnersystem im Büro entsprechen auf dem Rechner **Verzeichnisse (Directories)** für die geordnete Ablage von Dateien. Wie in einem übergeordneten Ordner mehrere kleine Ordner untergebracht sein können, kann auch ein Directory andere Directories beinhalten. Ein Directory, das selbst andere Directories beinhaltet, wird als **Vaterverzeichnis (Parentdirectory)** dieser Directories bezeichnet, die darin liegenden Directories nennt man **Unterverzeichnisse (Subdirectories)** dieses Directories.
- Das Gesamtarchiv ist in der Linux -Philosophie auch nur ein Directory, das Dateien und Subdirectories beinhaltet. Dem Gesamtarchiv entspricht in Linux der Begriff des **Dateisystems** (engl. **Filesystem**). Dieses gleicht einem auf den Kopf gestellten Baum (vgl. Abb. nächste Seite), wenn man die Verschachtelung und Abhängigkeiten von Dokumenten und Ordnern aufmalt. Man spricht daher auch vom **Dateibaum**.

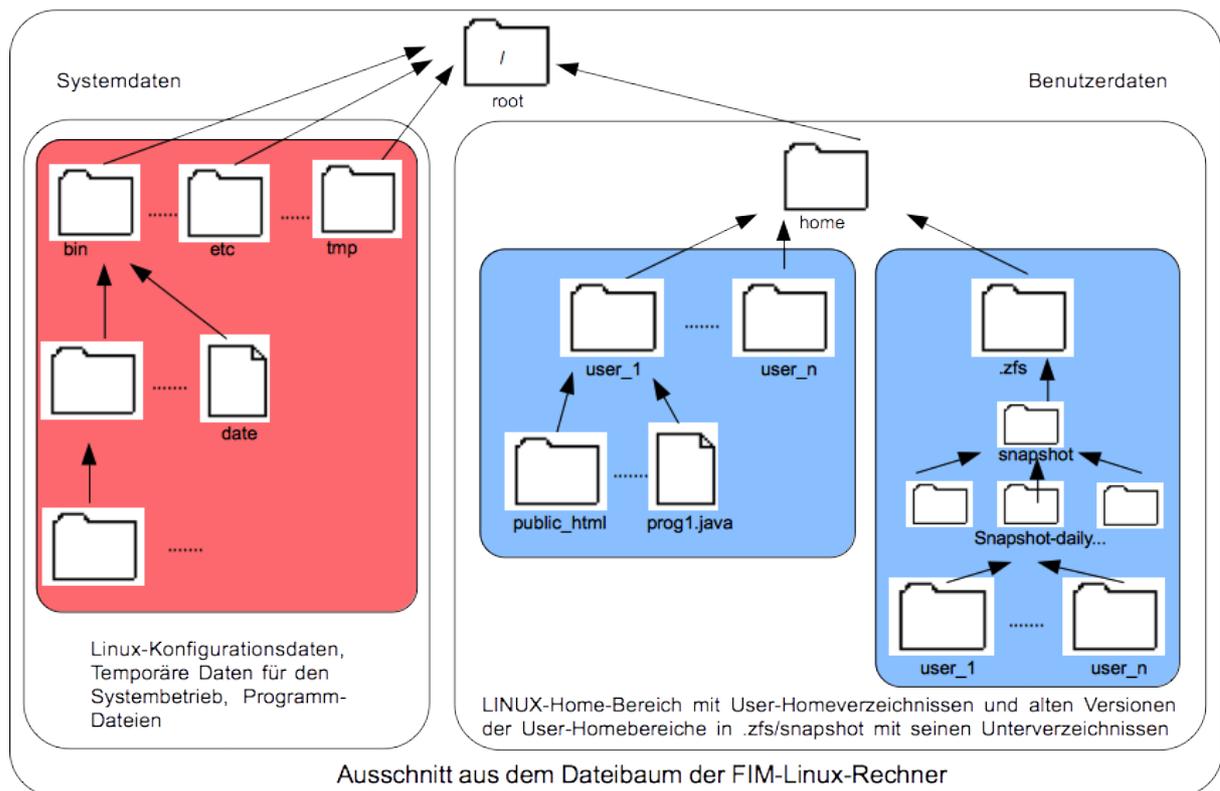
Beispiel (vgl. Abbildung):

Ein Directory namens X_Akten beinhaltet neben zwei Dateien die untergeordneten Directories Moulder, Scully und Skinner. X_Akten ist das Vaterdirectory dieser Directories. Die drei genannten Directories werden als Subdirectories des Directories X_Akten bezeichnet. Die Directories Moulder, Scully und Skinner könnten selbst wieder Subdirectories enthalten.



Wie das Dateisystem auf einem Rechner konkret aussieht, wird zum einen durch das verwendete Betriebssystem vorgegeben, ist in Teilbereichen aber auch vom jeweiligen Systemverwalter beeinflussbar. Exemplarisch wird im Folgenden der **Aufbau des Dateisystems auf den FIM-Rechnern im Linux-Betrieb** beschrieben (vgl. folgende Abbildung):

- Das **root-Directory**: Das Verzeichnis der **obersten Ebene** (entspricht im Bürobeispiel dem gesamten Archiv) wird als „**root**“ (deutsch: **Wurzel**) bezeichnet, da sich von hier aus der Dateibaum verzweigt. Im "root"-Directory finden Sie Systemdateien sowie Directories mit System- sowie Benutzerdaten.
- Die **Systemdaten**: Einige Subdirectories des „root“-Directories heißen z. B. „etc“, „usr“ oder „bin“ und beinhalten Systemdateien sowie Programme.
- Die **Benutzerdaten**: Generell liegen die Benutzerdaten im Directory „home“ mit seinen Unterverzeichnissen.
 - Für jeden Benutzer wird beim Einrichten der Kennung im Ordner home ein Subdirectory angelegt, das ihm für die **Aufbewahrung und Verwaltung seiner Dateien** zur Verfügung steht. Dieses Directory trägt den jeweiligen Benutzernamen und wird als „**Home Directory**“ dieses Benutzers bezeichnet.
 - Ebenfalls im Directory „home“ findet man den Ordner **.zfs**. Im Rahmen unserer **Datensicherungen** werden hierin im Ordner **snapshot** mit seinen Unterverzeichnissen **alte Versionen der User-Home Directories** gesichert. Das können Sie nutzen, um alte Stände Ihrer Daten oder gelöschte Dateien zu finden und in Ihr reguläres Home Directory rückzuspeichern. Das klappt natürlich nur, wenn die benötigte Datei von einer unserer Sicherungen erfasst wurde.
 - Wenn Sie sich unter **Windows** einloggen, wird Ihnen für Ihre Benutzerdaten ein eigener, von Ihren LINUX-Daten unabhängiger Datenbereich zugeteilt. Ihr **Linux - Home Directory** finden Sie bei Bedarf zusätzlich unter dem **Laufwerksbuchstaben Z**.



1.5.1.2 Linux-Dateitypen

- **Textdateien** (enthalten nur Informationen im ASCII-Code, d. h. einem Standardzeichensatz für Computer)
- **Binärdateien** (enthalten Nicht-Standardzeichen wie z. B. Code in Maschinsprache)
- **Directories** (Dateien, die Listen enthalten, wo Dateien physikalisch zu finden sind)
- **spezielle Dateien** (dienen u. a. zur Kommunikation mit Peripheriegeräten)

1.5.1.3 Benennung von Dateien und Directories

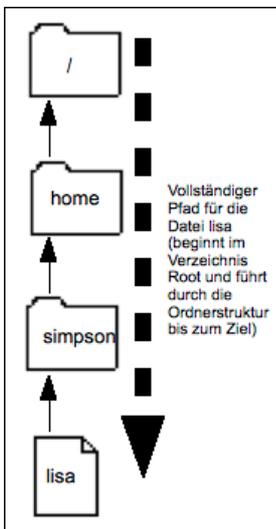
Zur Benennung von Linux-Dateien und -Directories dürfen Sie Namen mit bis zu 253 Zeichen bilden und prinzipiell fast alle Zeichen auf der Tastatur verwenden. Dabei sollten Sie einige Grundregeln beachten:

- Groß- und Kleinschreibung werden unterschieden
- Problematische Zeichen bei der Benennung von Dateien:
 - ? @ # \$ ^ & * () [] ' ` " | \ < > ; : ä ö ü Ä Ö Ü ß
 - Leerzeichen, Tabulatorzeichen, Zeilenumbruch und andere Steuerzeichen. Anstelle von Leerzeichen können Sie das Zeichen _ (Underscore) verwenden, das unproblematisch ist.
 - Vermeiden Sie auch + und - am Anfang eines Datei- oder Verzeichnisnamens. Wenn sie z. B. bei Shell-Kommandos Namen mit „-“ am Anfang als Parameter mit angeben, denkt die Shell, dass nach dem „-“ nun Optionen zum angegebenen Kommando folgen, statt das ganze als Dateinamen zu interpretieren.
 - Objektbezeichnungen können neben dem eigentlichen Objektamen noch eine **Pfadangabe**, d. h. eine Aufzählung von Directories, beinhalten (siehe unten). Zwischen den Directory-Angaben dient dabei der **Slash (/)** als **Trennzeichen**, weshalb ein Slash nicht für die eigentlichen Objektamen verwendet werden sollte.

1.5.1.4 Pfadangaben zur Adressierung von Dateien und Directories

Wenn Sie – egal ob unter Linux oder Windows - mit Directories und Dateien operieren, müssen Sie in der Lage sein, genau auszudrücken, welches **Objekt im Dateibaum sie meinen**. Wenn Sie z. B. jemand nur mitteilen, er soll sich eine Datei namens „lisa“ kopieren, geht daraus nicht hervor, wo im Dateibaum sich diese Datei befindet. Außerdem könnte es mehrere Dateien namens lisa im Dateibaum geben. **Eindeutigkeit erzielt man, wenn zusätzlich zum Objektnamen die Information liefert, wo im Dateibaum ein Objekt genau zu finden ist.** Dazu benutzt man **Pfadangaben**. Generell ist eine Pfadangabe (**oder auch kurz: Pfad**) die **Beschreibung eines Wegs durch den Dateibaum zu einem Objekt**. Man unterscheidet hierbei **vollständige** und **relative** Pfadangaben. Für den Anfang ist es ausreichend zu wissen, wie vollständige Pfade aufgebaut sind. Relative Pfadangaben braucht man vor allem beim Arbeiten mit der Shell. Deshalb werden diese auch erst weiter hinten im Script beschrieben.

Der **Weg von der Wurzel des Dateibaums zu einem Zielobjekt** (Datei oder Directory) wird als **vollständiger** (Zugriffs-)Pfad zu diesem Objekt bezeichnet. Man sagt auch **absoluter Pfad** oder einfach nur **Pfad** dazu. Ein Objekt im Dateibaum ist **eindeutig** bezeichnet, wenn man seine Lage im Dateibaum in Form des vollständigen Pfads angibt. Der **vollständige Pfad** für ein Objekt setzt sich unter Linux so zusammen:



- Er enthält die **Namen der Directories**, die **beginnend mit dem Wurzelverzeichnis root zum Zielobjekt durchlaufen werden müssen**. Das Wurzelverzeichnis **root** wird in Pfadangaben nicht ausgeschrieben, sondern durch das **Symbol „/“** (sprich: „Slash“) dargestellt
- Als **Trennzeichen** zwischen den Directory-Namen wird jeweils ein **„Slash“** gesetzt
- Am **Ende** steht der **eigentliche Namen des Zielobjekts**

Beispiele: Die Angabe des vollständigen Pfads für...

- die oberste Ebene des Dateibaums, die Wurzel, lautet einfach: `/`
- das Directory „simpson“ in der Abbildung lautet: `/home/simpson`
- die Datei „lisa“ in der Abbildung lautet: `/home/simpson/lisa`

Linux für Schreibfaule: Bei Pfadangaben für Linux-Home Directories kann man zur Abkürzung das **Tildezeichen** verwenden. Statt der Pfadangabe für das eigene Home Directory schreibt man dann: `~`

Man kann generell die vollständige **Pfadangabe für ein Linux – Home Directory eines Benutzers** durch das **Tildezeichen** gefolgt von dessen **Benutzerkennung**, also „**~kennung**“ ersetzen.

Beispiel: Als Abkürzung für den vollständigen Pfad `/home/vkboss` darf man kürzer auch `~vkboss` schreiben.

1.5.2 Arbeiten mit Dateien und Directories

Beim Umgang mit Dateien und Directories haben Sie die Wahl, ob Sie dazu ein grafisches Tool wie z. B. in Xfce den Dateimanager `thunar` verwenden oder mit der Shell arbeiten wollen. Beide Methoden haben Ihre Vor- und Nachteile. Der Dateimanager `thunar` ermöglicht Ihnen beispielsweise durch einfaches Anklicken das Öffnen von Dokumenten mit voreingestellten Standardanwendungen. Sind beim Zugriff auf Objekte Pfadangaben oder Wildcards (siehe später) im Spiel, erledigen Sie Ihre Arbeit oft schneller mit der Shell.

1.5.2.1 Der Xfce-Standard-Dateimanager `thunar`

`thunar` ist der **Standard-Dateimanager** von Xfce. Andere Dateimanager im System sind z. B. `nautibus` oder der etwas komplexere `dolphin`. `Thunar` ist einerseits sehr ressourcenschonend (und damit schnell), bietet aber dem Einsteiger alles, was man braucht.

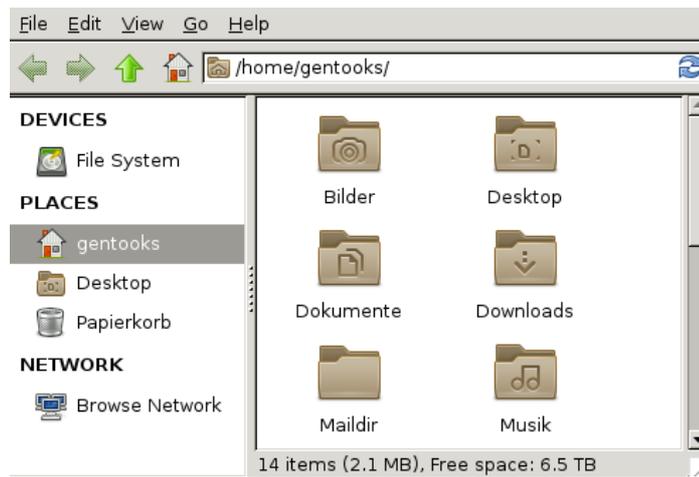


Abbildung: thunar-Fenster in der Standard-Voreinstellung (Anzeige des Home Directories des Benutzers gentooks)

Thunar starten: Über **Aktenschrank-Symbol in der Xfce-Leiste** oder aus der **Shell mit Kommando thunar**.

Ausgewählte Elemente des thunar -Fensters (von oben nach unten, vgl. Abbildung):

- **Menüleiste:** Aktionen und Einstellungsmöglichkeiten für thunar. Hat man diese versehentlich ausgeblendet, kann man Sie mit **<Strg-m>** wieder einblenden lassen.
- **Adressleiste (Location selector):** Hier steht, welches **Directory gerade angezeigt wird**. Unter „Ansicht“ können Sie wählen, ob dieser Bereich im „**Pfadleistenstil**“ oder im „**Werkzengleistenstil**“ (siehe Abb. oben) dargestellt wird. Beim Pfadleistenstil werden Directories mit Ihren Unterverzeichnissen als Folge von Schaltknöpfen dargestellt, über die Sie das anzuzeigende Directory einstellen. Um eine **Pfadangabe zum Anzeigen eines Directories per Hand einzugeben, wählen Sie den Werkzengleistenstil**. Damit wird die **Adressleiste editierbar**.
- Unterhalb der Adressleiste auf der linken Seite bietet die **Seitenleiste** von thunar schnellen Zugriff auf Ordner und Geräte, kurz „**Orte (Places)**“ genannt. Bei der Darstellung haben Sie wieder die Wahl zwischen „Lesezeichen“ (Standardeinstellung, siehe oben) sowie einer „Baumansicht“. In beiden Ansichten steht dabei
 - der Ort mit dem Haussymbol für Ihr **Linux-Home Directory**
 - „**Dateisystem**“ für das gesamte Filesystem des Rechners ab dem **Root-Directory**.
 - Um in der Lesezeichen-Darstellung einen **Ordner** für den **Schnellzugriff** hinzuzufügen, **ziehen** Sie diesen einfach mit der Maus **in die Seitenleiste**.
- Rechts neben der Seitenliste befindet sich die **Verzeichnisanzeige**: Hier wird der **Inhalt** des in der Adressleiste ausgewählten **Ordners** angezeigt. In der Menüleiste können Sie unter „Ansicht“ einstellen, ob wie die Verzeichnisinhalte dargestellt werden (**Symbol, Detail-** oder die informationsreiche **Listenansicht**).
- Am unteren Fensterrand findet sich die **Statusleiste**, die einen Überblick über das angezeigte Verzeichnis gibt.

Mehr zu thunar finden Sie z. B. unter <https://wiki.ubuntuusers.de/Thunar/>

1.5.3 Zugriffsrechte und Datenschutz

Wie schon angesprochen sind Linux-Rechner **Multiuser-Systeme**, d. h. man teilt sich insbesondere das Dateisystem mit vielen anderen Benutzern. Für den Anfang genügt es zu wissen, dass Ihre Daten standardmäßig für andere Nutzer (außer den Systemadministratoren) nicht lesbar oder veränderbar sind. **Daten-Zugriffsrechte** ansehen und einstellen kann man mit dem Dateimanager oder über die Shell (siehe Kapitel über „Fortgeschrittene Aspekte“).

1.5.4 Löschen und Rückspeichern von Daten

Dateien und Directories löscht man entweder mit Hilfe eines Dateimanagers wie thunar oder über den Shell-Befehl „**rm**“ (bei Directories Option „-r“ nicht vergessen). Standardeinstellung für „rm“ ist ein **Löschen ohne Rückfrage**.

Wird eine Datei gelöscht oder beim Editieren geändert, ist der alte Stand nicht mehr rekonstruierbar, sofern keine Sicherung existiert. Vor Änderungen ist es also ratsam, selbst eine **Sicherungskopie** anzulegen.

Automatisch erstellte Backups (Snapshots) Ihrer Daten finden Sie im Order /home/.zfs/snapshot

Hier liegen **Sicherungen** Ihrer Daten vom **aktuellen Tag (snap-hourly)**, den **vergangenen Tagen (snap-daily)**, **Wochen (snap-weekly)** und **Monaten (snap-monthly)**. An der Datumsangabe im Ordner erkennen Sie, von wann die Backup-Kopie genau stammt, können so den benötigten früheren Stand Ihrer Daten finden und bei Bedarf in Ihr reguläres Home Directory zurückkopieren. Wir fertigen auch noch längerfristige Backups Ihrer Daten an, als Sie sehen können. Bei Bedarf kommen Sie bitte auf die FIM-Systembetreuung zu.

Wichtig: **Wir heben nicht jeden Backupstand auf**. Z. B. heben wir von den hourly-Snapshots eines Tages nur einen einzigen Snapshot länger auf, der den Tag dann als daily-snapshot repräsentiert. Von 7 daily-Snapshots wird wiederum nur einer als Repräsentant für die entsprechende Woche aufgehoben usw..

1.5.5 Quota-Regelung zur Speicherplatzbeschränkung

Damit unsere Festplatten nicht überlaufen haben wir für die Home Directories unserer Benutzer eine **Quota**-Regelung eingeführt. Jeder Benutzer darf dabei bis zu **8 GB Daten** in seinem Home Directory ablegen. Ein Überziehen (auch kurzfristiger Art) ist nicht möglich. Gerät man nahe an diese Grenze, erhält man eine automatische Warnmail, die erklärt, wie man herausfindet, welche Daten den Speicherplatz auffressen, so dass man nicht benötigte Daten löschen kann. Sie können sich den verbrauchten Speicherplatz in der Shell (siehe später) oder über den Dateimanager thunar anzeigen lassen. Bei Quota-Mehrbedarf wenden Sie sich bitte an die FIM-Systembetreuung!

1.6 Editieren / Textverarbeitung

Bei der Programmierung besteht die Notwendigkeit, Text in Dateien abzuspeichern. Gefragt sind deshalb Programme, die es möglichst komfortabel erlauben, Text in Dateien abzulegen und bei Bedarf zu modifizieren. Solche Programme nennt man **Editor**. Unter Linux gibt es viele **Editorprogramme**:

kate, kwrite	Einfach zu bedienen und optimiert für KDE; wir empfehlen kate als Standardeditor
xemacs, emacs	Dieser Editor bietet speziell für Programmierer gute Unterstützung; Empfehlung als Standardeditor für Power-User
vi, vim	Der "vi" ist ein Editor, den es seit Anbeginn auf jedem Linux-System gibt. Eine bequeme Variante des vi heisst „vim“. Empfehlung für Systemadministratoren.

Daneben gibt es für spezielle Anforderungen weitere Software für die Texteingabe bzw. Programmierung:

- **Textverarbeitung**: Kostenlose Alternative zu Microsoft-Office: **LibreOffice** (Aufruf: "**libre-office**").
- **Erstellen mathematischer Texte**: Die besten Ergebnisse (vor allem beim mathematischen Formelsatz) erhalten Sie mit dem Textsatzsystem **TeX** (grafisches Frontend für TeX unter KDE: **ki-le**).
- **Programmieren**: Geht es um Programmentwicklung ist die Programmierumgebung **Eclipse** erste Wahl, die neben einem integrierten Editor Zusatzfunktionen für die Programmierung bietet (Aufruf: „**eclipse**“)

1.7 Drucken, Umgang mit Standard-Dateitypen

1.7.1 Öffnen/Anzeigen gängiger Dateitypen unter Linux

Bevor das Drucken erläutert wird, geben wir hier zunächst einen generellen Überblick über häufig benötigte Dateitypen und mit welchen Anwendungen man Sie am besten öffnet, um sich den Inhalt anzuschauen. Damit sind Sie dann in Lage, vor dem Drucken den Inhalt der Datei kontrollieren und zu sehen, wie das Druckergebnis sein wird.

Dateitypen erkennen

In der grafischen Oberfläche: Wenn Sie mit einem Dateimanager wie **thunar** eine Datei anklicken, wird unten in der Statusleiste der Typ der Datei angezeigt.

In der Shell: Kommando "**file dateiname**" eingeben, um den Typ der Datei „dateiname“ festzustellen.

Kommandos für das Öffnen verschiedener Dateitypen:

In Dateimanagern wie **thunar** genügt ein Doppelklick auf das Dateisymbol, um eine Datei zu öffnen und den Inhalt anzeigen zu lassen. **thunar ordnet nämlich Dateien je nach Typ automatisch bestimmten Anwendungen zum Öffnen zu.**

Manchmal ist diese Zuordnung aber nicht optimal voreingestellt und **thunar** schlägt Ihnen nicht die gewünschte Anwendung vor, um ein Dokument zu öffnen. Das können Sie aber korrigieren. Wählen Sie die Datei über Anklicken mit rechter Maustaste aus und führen dann „mit anderer Anwendung öffnen“ aus. Über eine Kontrollbox können Sie die von Ihnen gewählte Anwendung auch permanent zur Standardanwendung für alles Dokumente eines Typs machen.

Übersicht Standardanwendungen für oft benötigte Dateitypen:

- **Textdateien:** Textdokumente kann man mit dem Editor **kate** anschauen und ggfs. modifizieren. Für eine schnelle Anzeige des Inhalts einer Textdatei im Shellfenster nutzt man das Kommandos **less**.
- Für die meisten anderen Dateitypen (PDF, Postscript, JPG-Bilder...) können Sie den KDE-Dokumentviewer **okular** nutzen oder Anwendungen, die für den jeweiligen Dateityp noch speziellere Funktionen anbieten:
 1. Für Bilder verwenden Sie Programme wie **xv** oder **gimp** zur Bearbeitung und Vorschau am Bildschirm
 2. Anzeige von PDF-Dokumenten am Bildschirm am besten mit **okular**, alternativ **evince** (unter Gnome).
 3. **Postscript-Dokumente:** Postscript ist eine Seitenbeschreibungssprache für Drucker. Über Postscript-Befehle erhält der Drucker Anweisungen, was er wohin zu zeichnen hat. Eine Postscript-Datei erkennt man daran, dass in der ersten Zeile am Anfang „%!“ steht. Den Inhalt einer Postscript-Datei können Sie mit dem Kommando **gv dateiname** anzeigen lassen (Alternativen zu gv: **okular** oder **evince**).

1.7.2 Ausdrucken mit den Campusdrucker-Geräten

Auf dem Universitätscampus finden Sie in allen Gebäuden Multifunktionsgeräte (Campusdrucker), mit denen Sie kopieren, scannen und von allen Poolrechnern aus auch ausdrucken können. Die Druckersoftware ist so konfiguriert, dass diese praktisch mit allen Dateitypen zurecht kommt. Die Abrechnung der Ausdrücke erfolgt über ein Druck-Guthaben, das Sie an entsprechenden Automaten auf Ihre Mensacard buchen. Auf Campusdruckern mit Farbdisplay ist auch Farbdruk möglich, wobei der Preis pro Seite entsprechend steigt. Alle Campusdrucker auf dem Unigelände hängen an einem zentralen Druckerserver, wo Ihr Druckjob zunächst zwischengespeichert wird. Ihr Ausdruck erfolgt auf demjenigen Campusdrucker, an dem Sie Ihre Mensacard einstecken und so den Druckjob abrufen. Sie sind also nicht auf die Nutzung des Campusdruckers in den FIM-Studentenrechnerpools eingeschränkt, sondern können Ihren Druckjob auf jedem Campusdrucker ausgeben lassen.

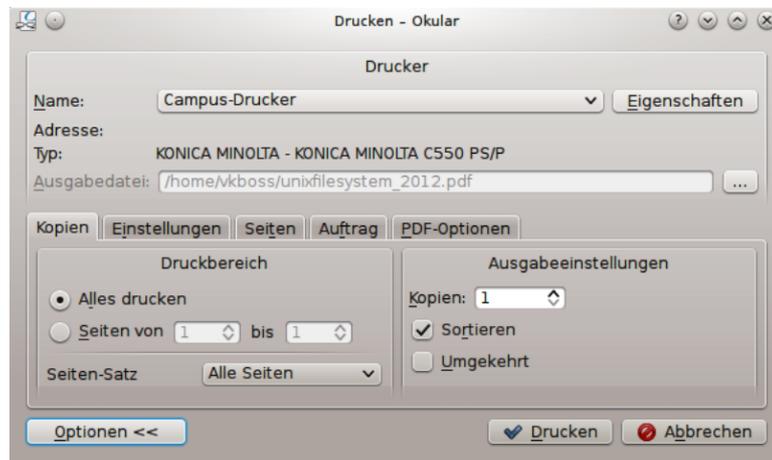
Was tun bei Problemen beim Ausdrucken auf den Campusdruckern: Die Campusdrucker werden von einer **externen Firma** auf dem Unigelände betrieben. Wenden Sie sich bei Problemen bitte an die Betreiberfirma im Untergeschoß des Gebäudes Wirtschaftswissenschaften. Alternativ können Sie die Betreiberfirma auch kostenlos über unsere Haustelevone erreichen (Telefonnummer hängt jeweils neben dem Campusdrucker).

Drucken auf den Campusdruckern

Wählen Sie beim Ausdrucken (egal ob unter Linux oder unter Windows) im Printmenu des jeweiligen Programms „Campus-Drucker“ aus.

Tipps zu den Campusdruckern:

- Man kann über **Optionen** einstellen, **welche Seiten** man genau drucken möchte und kann beim Drucken z. B. auch **verkleinern**, so dass 2 oder 4 Seiten des Originaldokuments beim Drucken auf eine Seite passen. Das spart Geld beim Ausdrucken.
- Man kann auch einstellen, ob man **Graustufen- oder Farbdruck** haben möchte (Farbdrucke sind teurer!).
- Wenn Sie als Drucker **Campusdrucker** wählen, müssen Sie nach dem Drücken auf den Print-Button anschließend noch in einem weiteren Fenster Ihre **Mensakartenummer (nur die Nummer ohne den Buchstaben davor!) eingeben**.



Bei älteren Linux-Programmen wie z. B. dem xemacs oder gv ist nicht erkennbar oder konfigurierbar, welcher Drucker angesteuert wird, wenn man den Print-Button anklickt. Wenn man nicht sicher sein kann, auf welchem Drucker der Ausdruck letztendlich landet, sollte man zum Ausdrucken einen anderen Weg gehen. Speichern Sie den Druckinhalt zunächst in einer Datei. **Öffnen Sie zum Drucken diese Datei dann mit kate (bei Texten) oder okular, wo Sie problemlos die Campusdruck-Systeme als Ausgabegeräte einstellen können.**

1.8 Benutzung von Scannern und Brennern

Sehr **schnelle, kostenlos benutzbare Scanner** finden Sie in den **Bibliotheksbereichen**. Im Gegensatz zu Standard-Scannern benötigen diese Scanner pro Scanvorgang nur eine Sekunde und sind somit hervorragend für das Scannen von Skripten oder Büchern geeignet. Abgespeichert wird jeweils auf dem von Ihnen mitzubringen **USB-Stick**.

Auch die **Campusdrucker** sind als Scanner verwendbar, wobei hier allerdings **jeder Scan etwas kostet**.

In manchen Poolrechnern der FIM finden Sie **DVD-Brenner**. **Standardprogramm zum Brennen** unter Linux ist das Programm **k3b**.

Genauere Hinweise zur Benutzung unserer Scanner, Brenner oder USB-Sticks entnehmen Sie bitte unseren Informationen im WWW: <http://www.fim.uni-passau.de/it-dienste/>

1.9 Nutzung von Netzwerk-Diensten

In diesem Kapitel erhalten Sie einen Überblick, welche wesentlichen Internetdienste die Universität bzw. die FIM anbieten und welche persönlichen Einstellungen dabei für Sie gelten (E-Mail-Adresse, FIM-Homepage...).

1.9.1 FIM-Rechnernamen und -adressen

Der **Aufbau der Rechnernamen** der FIM-Studentenrechner entspricht folgendem Schema:

enterprise.fim.uni-passau.de

Am Anfang steht der **eigentliche Rechnername**. Die weiteren Bestandteile der Adresse (getrennt durch Punkte) beschreiben Zugehörigkeiten zu **Domains (Bereichen)**. „enterprise“ gehört zur Domain „de“, steht also in Deutschland, gehört des weiteren zur Domain „uni-passau“ und wird in der Domain „fim“ genutzt. Die Poolrechner sind aktuell nach Flussfischen, Raumfahrzeugen, Robotern aus Film und Literatur sowie nach Winden oder Windsystemen benannt.

Innerhalb der FIM-Studentenrechnerpools genügt als Adressangabe „enterprise“, um diesen Rechner eindeutig zu identifizieren, da es innerhalb unserer eigenen Domain nur einen Rechner namens enterprise gibt. Wollen Sie den Rechner hingegen von außerhalb der Universität Passau aus ansprechen, so müssen Sie bei der Adresse die Domainzugehörigkeiten mit angeben, also „enterprise.fim.uni-passau.de“ schreiben, da es mit Sicherheit viele Rechner mit Namen enterprise im Internet gibt.

1.9.2 Adressierung von FIM-Usern via Internet

Um mit einem Kommilitonen, der an einem FIM-Poolrechner arbeitet, per Internet Kontakt aufzunehmen, müssen Sie dessen **Internet-Adresse** kennen. Diese besteht aus zwei Teilen, nämlich

- dem Benutzernamen (entspricht der FIM-Kennung, die beim Login benutzt wird)
- der Adresse des Rechners, von dem aus er das Internet benutzt (rechneradresse)

Beides zusammen bildet dann verbunden durch das Symbol „@“ die Internet-Adresse des Benutzers:

benutzername@rechneradresse

Beispiel: vkboss@enterprise.fim.uni-passau.de

Diese Adressform ist vor allem wichtig, wenn Sie Chat-Dienste benutzen, die auf Direktverbindungen setzen.

1.9.3 Electronic Mail

Alle offiziellen Email-Informationen der Universität erhalten Sie über Ihren Standard-Uni-Email-Account. Stellen Sie deshalb unbedingt sicher, dass Sie diesen Standard-Uni-Mailaccount auch regelmäßig lesen. Alles nötige hierzu erfahren Sie auf den Webseiten des ZIM.

Für die Unterstützung des FIM-Forschungs- und Lehrbetriebs betreibt die FIM ergänzend **eigene Mailservices, um FIM-spezifische Informationen auszutauschen** (z. B. Veranstaltungshinweise, Stellenausschreibungen für stud. Hilfskräfte, Informationen im Rahmen des Übungsbetriebs). Mit Ihrer FIM-Kennung haben Sie automatisch auch Zugang zu den FIM-Mailservices.

Um keine wichtigen Informationen zu versäumen, sollten Sie unbedingt regelmäßig alle Ihre Uni-Mailaccounts lesen. Sie können dazu entweder direkt eigene **Mailaccounts in Ihrem Mail-Programm konfigurieren** (Infos hierzu im folgenden Abschnitt) **und/oder** für die Uni-Mailaccounts eine **Weiterleitung an Ihren privaten Standard-Mailaccount** einrichten (Infos weiter unten für FIM-Mailserver bzw. auf den ZIM-Webseiten).

1.9.3.1 Konfigurationseinstellungen für die Nutzung der Universitäts-Mailserver

Im Folgenden erhalten Sie alle Informationen, um ein Mail-Programm Ihrer Wahl (z. B. thunderbird, siehe weiter unten) so zu konfigurieren, dass Sie über die Uni-Mailservices Mails versenden und lesen können. **Mit den aufgeführten Einstellungen können Sie nicht nur auf den Poolrechnern Mail-Konten einrichten, sondern auch auf Ihrem Rechner zuhause.** Beachten Sie bitte, dass die Punkte „Verbindungssicherheit“ oder „Authentifizierungsmethode“ je nach verwendetem Mailprogramm auch anders benannt sein können.

1.9.3.1.1 Einstellungen für den FIM-Mailserver

Mail-Adressen an der FIM sind nach folgendem Schema aufgebaut:

benutzername@fim.uni-passau.de

Beispiel für den User vkboss: vkboss@fim.uni-passau.de

Einstellungen zum Versenden von Mail (SMTP) über den FIM-Mailserver

Servername "mail.fim.uni-passau.de" Zugangsdaten: Ihre FIM-Login-Kennung (LINUX-Login) Port 587 Verbindungssicherheit: "STARTTLS" (bei Mailclient Thunderbird) bzw. "SSL verwenden" (Apple Mail) Authentifizierungsmethode: "Passwort, normal" (bei Thunderbird) bzw. "Passwort" (Apple Mail)
--

Einstellungen zum Mailabruf vom FIM-Mailserver über IMAP

Mails, die auf Ihrer FIM-Kennung eingehen, landen in Ihrem Linux - Home Directory im Verzeichnis maildir (zumindest sofern keine Mailweiterleitung eingerichtet ist, siehe weiter unten). Für den Mailzugriff sollten Sie **IMAP** verwenden. Dabei verbleiben die Originale der Mails auf dem Server und können von verschiedenen Clients (Poolrechner, Smartphone usw.) aus bearbeitet werden.

Servername "imap.fim.uni-passau.de" Zugangsdaten: Ihre FIM-Login-Kennung (LINUX-Login) Port 993 Verbindungssicherheit: "SSL/TLS" (Mailclient Thunderbird) bzw. "SSL verwenden" (Apple Mail) Authentifizierungsmethode: "Passwort, normal" (Mailclient Thunderbird) bzw. "Passwort" (Apple Mail)
--

1.9.3.1.2 Einstellungen für Ihren allgemeinen Uni-Mailaccount (Outlook-Account)

Richten Sie in **Thunderbird** ein zusätzliches Konto für Ihren **allgemeinen Uni-Mailaccount** ein. Wählen Sie hierzu unter „Bearbeiten“ „Konten-Einstellungen“ aus, links unten dann „Konten-Aktionen“ und hier „E-Mail-Konto manuell hinzufügen“. Die allgemeine Uni-Mailadresse für z. B. den User meier13 lautet dann meier13@ads.uni-passau.de.

Beispielseinstellungen für den allgemeinen Uni-Mail-Account:

IMAP-Server:	email.uni-passau.de (Port 993)
Verbindungssicherheit:	SSL/TLS-Verschlüsselung
Kennung:	Ihre ZIM-Kennung im Kurzformat (z. B. „meier13“)
Passwort:	Das Passwort Ihrer ZIM-Kennung
SMTP-Server:	email.uni-passau.de (Port 587)
Verbindungssicherheit:	STARTTLS (bei Mailclient Thunderbird) oder SSL/TLS
Kennung und Passwort:	ZIM-Kennungsdaten wie bei IMAP

1.9.3.2 Thunderbird: Arbeiten mit E-Mail unter Linux

Wir empfehlen das Programm **Thunderbird** zum Lesen und Versenden von E-Mail in den FIM-Pools. Thunderbird finden Sie sowohl unter Windows als auch unter Linux. Der Start erfolgt wahlweise über Menüeinträge, Icons oder das Shell-Kommando „**thunderbird &**“. Zu Beginn müssen Sie zunächst **Ihren Mail-Account** (d. h. ein Benutzerkonto) einrichten (siehe nächster Abschnitt). **Weitere Einstellungen** für den Umgang mit Mail können Sie unter „**Bearbeiten**“ in den **Einstellungen** vornehmen. Hier können Sie z. B. unter dem Punkt **Allgemein** einstellen, ob und welche **Startseite** Thunderbird beim Starten lädt.

1.9.3.2.1 Einrichten des FIM-Mailaccounts mit Thunderbird

Zum **Anlegen eines E-Mail Accounts** gibt es eine **Einrichtungshilfe**, die beim ersten Start von Thunderbird automatisch startet. Beschrieben wird im folgenden das Anlegen eines Accounts für Ihre **FIM-Mail-Kennung** (cipkennung ist Platzhalter für Ihre FIM-Kennung) mit dieser Einrichtungshilfe (Konfigurationsdaten siehe auch oben):

Zu Beginn erscheint ein Feld „Konto einrichten (Mail-Account-Setup)“
 Beim ersten Start von Thunderbird werden Sie gefragt, ob Sie eine neue E-Mail-Adresse haben wollen. Da wir für Sie bereits eine E-Mail-Adresse vorbereitet haben, klicken Sie darunter an, dass Sie eine bereits existierende Adresse nutzen wollen. Danach kommt ein Fenster, in dem Sie nun Ihren FIM-E-Mail-Account konfigurieren können:
 Tragen Sie neben „Ihr Name (Your Name)“ Vor- und Familiennamen ein, als „Email Address“ cipkennung@fim.uni-passau.de und darunter das entsprechende Passwort (dann weiter)
 Thunderbird holt sich nun von unserem Mailserver alle weiteren Informationen und schlägt schon die richtigen Einstellungen für den Postausgang (SMTP) sowie den Mailabruf via IMAP vor (bei anderen Mailprogrammen müssen Sie diese Einstellungen ggfs. per Hand vornehmen)
 Richten sie mit „Fertig“ Ihren Account ein.

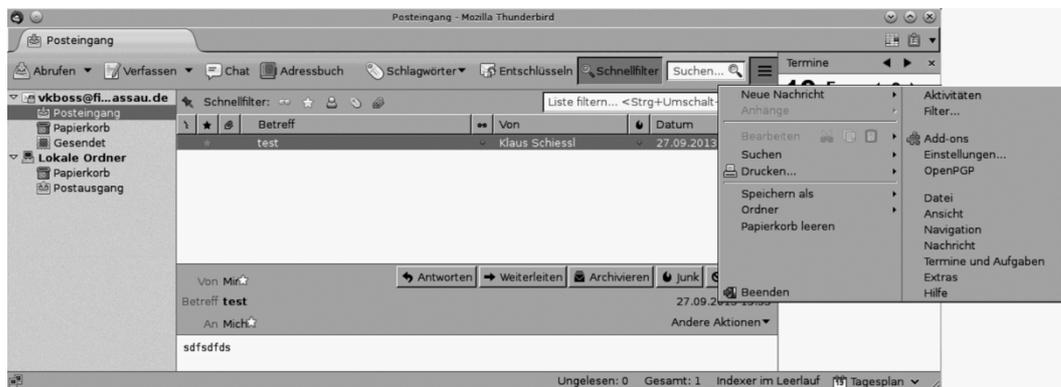
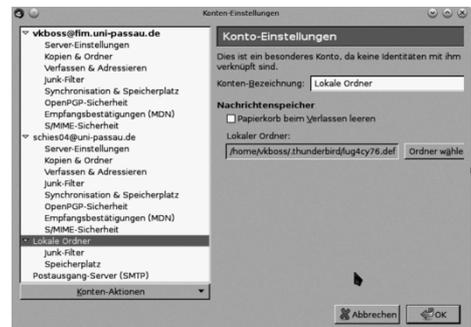


Abbildung: Thunderbird mit geöffnetem Auswahl-Menü zum Zugriff auf Einstellungen usw.

Thunderbird-Menü: Zur Menüanzeige führt die Schaltfläche  rechts oben in der Symbolleiste

Nehmen Sie nun **folgende „Konten-Einstellungen vor:**

1. Stellen Sie ein, dass Ihre **Mails als normaler Text** statt im HTML-Format verschickt werden: Unter „**Verfassen & Adressieren/Compositon & Adressing**“ entfernen Sie im rechten Bereich des Fensters das **Häkchen** neben „**Nachrichten verfassen im HTML-Format**“. Im Forschungsumfeld sind HTML-formatierte Mails eher ungebrauchlich.
2. Sorgen Sie nun dafür, dass **Mails im Papierkorb (Trash) automatisch gelöscht** werden, um Ihre Quota nicht zu belasten. Klicken Sie dazu auf „**Lokale Ordner/Local Folders**“ und setzen ein Häkchen bei „**Papierkorb beim Verlassen leeren/Empty Trash on Exit**“
3. Bestätigen Sie durch Klicken auf „**OK**“ Ihre Anpassungen und schließen dadurch das Einstellungs-Fenster.



Sie können nun auf den Account zugreifen, indem Sie links den gewünschten Mail-Account auswählen.

1.9.3.2 Account-Pflege: Papierkorb entleeren und Ordner komprimieren

Beim Arbeiten mit Thunderbird wird häufig vergessen, Ordner wie Trash/Papierkorb/Sent/Junk zu leeren: Beachten Sie bitte, dass sich hierin im Lauf der Zeit jede Menge E-Mails ansammeln. Wenn diese Ordner nicht regelmäßig geleert werden, führt dies zu einer **Quota-Belastung**. **Vorsicht:** Auch wenn Sie fleißig Mails mit „Delete“ löschen und dann den Mail-Mülleimer mit „Empty Trash“ entleeren, wird die gelöschte Mail erst auf der Festplatte entfernt, wenn Sie „**Alle Ordner des Kontos komprimieren**“ auswählen. Auch E-Mail belastet Ihre Quota.

1.9.3.3 Zugriff auf den FIM-Mailaccount über Webbrowser

Webmail-Interface: Die einfachste Möglichkeit, von außerhalb der FIM, z. B. im Urlaub, auf E-Mails auf Ihren FIM-Mailaccount zuzugreifen, ist das **Webmail-Interface** des FIM-Mailservers (zu erreichen über <https://webmail.fim.uni-passau.de/>). Beim Zugriff über Smartphone wird automatisch eine hierfür optimierte Darstellung angezeigt.

1.9.3.4 Weiterleiten des FIM-Mailaccounts an einen anderen Mailaccount

Wir empfehlen, Ihre FIM-Mails an den standardmäßig von Ihnen benutzten Mailaccount weiter zu leiten. Legen Sie -z. B. mit dem Editor kate- hierzu eine Datei **.forward** in Ihrem Home Directory an. Der Inhalt dieser Datei für den Benutzer vkboss könnte z. B. so aussehen:

```
vkboss@gmx.de, schiessl@web.de
\vkboss
```

Die Einträge in Zeile 1 leiten eintreffende Mails an die aufgeführte(n) Kennung(en) weiter, die zweite Zeile sorgt dafür, dass auch eine Kopie jeder Mail unter Ihrem Mailaccount bei der FIM abgelegt wird. **Vorsicht:** Wenn Sie den Backslash in der zweiten Zeile vergessen, bauen Sie eine ewige Mail-schleife! **Prüfen Sie unbedingt sofort, ob Ihre Weiterleitung auch wie gewünscht funktioniert!**

1.9.3.5 Zum Umgang mit Werbemails und virenverseuchten E-Mails

Wir lehnen Mails, die eindeutig als **Werbung (SPAM)** erkannt werden, schon **während der Annahme** ab. Durch diesen Kniff bleiben Sie von den meisten Werbemails verschont. Den Rest können Sie über den verwendeten Mail-Reader filtern. Die meisten Mail-Reader sind nämlich heute “intelligente” Mail-Reader, die mit Hilfe des Benutzers lernen, SPAM-Mail zu erkennen (z. B. Thunderbird mit eingebautem Junk-Filter und einstellbaren Filterregeln). Wichtig: **Geben Sie Absendern von Werbemails nie eine Rückantwort !** Damit wird Ihre Mail-Adresse nur noch wertvoller, da der Absender dann merkt, dass Sie auf Werbemail reagieren und Ihre Mail-Adresse in Benutzung ist.

Unser Mailserver arbeitet weiterhin mit einem **Virens scanner** zusammen, der **verhindert, dass virenverseuchte E-Mails** an Sie **zugestellt** werden bzw. dass von Ihnen losgeschickte, mit Viren verseuchte E-Mails Ihr Ziel erreichen.

1.9.4 Das World Wide Web

Das World Wide Web steht für die **Integration von Internet-Diensten unter einer leicht zu bedienenden Oberfläche**.

1.9.4.1 URLs - Das Adressierungsschema im WWW

Eine **URL (Uniform Resource Locator)** ist eine Art Adresse, die alle Angaben enthält, die für das Auffinden einer bestimmten Information im World Wide Web notwendig sind. Eine typische URL hat den folgenden **Aufbau**:

```
protokoll://hostname/pfadangabe
```

- Das **Protokoll** gibt die Art des Zugriffs und damit die Art der Information an. Für Informationen auf WWW-Servern ist "http" anzugeben, für andere Dienste das entsprechende Protokoll (ftp, mail ...).
- Der **Hostname** gibt die Internet-Adresse des Servers an, auf dem die Information gespeichert ist (voller Hostname mit "Domain").
- Die **Pfadangabe** gibt an, welche Datei vom Server auf den Client übertragen werden soll.

Beispiel für eine URL:

<http://www.fim.uni-passau.de/it-dienste/index.html>

Diese URL bezeichnet die Datei "index.html" im Directory "it-dienste" auf dem WWW-Server "www.fim.uni-passau.de".

1.9.4.2 Browser und Tools zum Zugriff auf das Web

Die Client-Software, die man benötigt, um das WWW zu nutzen, wird als **Browser** bezeichnet:

- Es gibt Browser, wie z. B. das Programm „lynx“, die in **alphanumerischen** Umgebungen wie der Shell rein Text-basierend laufen.
- In den FIM-Pools empfehlen wir als **grafikbasierten** Browser **Firefox**, der sowohl unter Windows als auch Linux verfügbar ist. Im nächsten Abschnitt finden Sie eine kurze Beschreibung zur Nutzung von Firefox.
- Wenn man möglichst schnell den Inhalt einer Webpage auf dem eigenen Rechner abspeichern will, geht das am schnellsten mit dem Linux-Kommando **wget** (Aufruf: wget mit Angabe der URL der gewünschten Website).

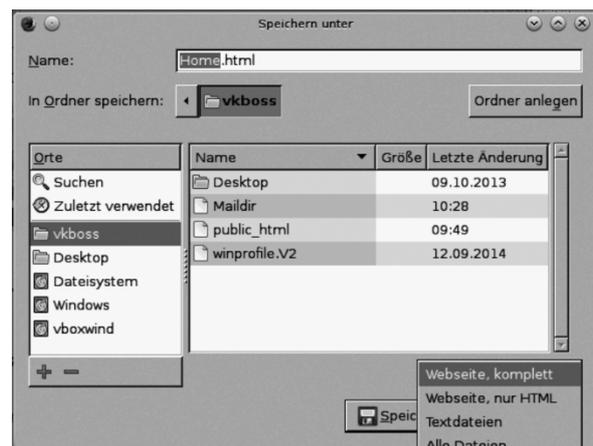
1.9.4.3 Arbeiten mit dem Webbrowser Firefox

Firefox finden Sie sowohl auf unseren Windows- als auch Linux-Rechnern. Von der Shell aus startet man Firefox am besten als **Hintergrundprozess** (Aufruf: **firefox &**), da er ein eigenes Fenster öffnet.

Firefox-Menü: Zur Menüanzeige führt die Schaltfläche  rechts oben in der Symbolleiste. Hier erreichen Sie grundlegende Funktionen wie „Drucken“ oder die Firefox-Einstellungen.

Webpages abspeichern und drucken:

- Die im Browser-Fenster angezeigte **Webpage** können Sie per Rechtsklick mit der Maus in dieses Fenster abspeichern (Option „**Seite speichern unter**“).
- Sie können die Webpage nun wahlweise (siehe Button rechts unten) in mehreren **Varianten speichern**. Wo die Webpage abgespeichert werden soll, legen Sie über die Auswahl links oben fest.
- Wenn Sie „**Webseite komplett**“ auswählen, werden auch alle auf der Webpage enthaltenen Bilder als separate **Bilddateien** mit abgespeichert.
- Um eine angezeigte **Webpage für den Ausdruck vorzubereiten** nutzen Sie die **Drucken-Funktion im Firefox-Menü**. Wenn Sie die Seite nicht gleich auf den Campusdrucksystemen drucken wollen, können Sie die Webpage auch für den späteren Ausdruck im **PDF-Format** abspeichern.



Firefox beenden:

Firefox beenden Sie, indem Sie im **Firefox-Menü** unter „Datei“ den Unterpunkt „**Beenden**“ auswählen. Dann werden alle gemachten Einstellungen auch gespeichert und alle temporären Dateien gelöscht, die Firefox beim Arbeiten anlegt.

Falls **Firefox abgestürzt ist oder unsauber beendet** wurde, startet Firefox evtl. beim nächsten Mal mit der **Fehlermeldung** „...another Mozilla Firefox is running...“ oder „profile is in use“. Beenden Sie dann Firefox sofort wieder und starten unser **Systemscript „cleanup“**. Danach können Sie Firefox wieder ohne Probleme starten.

1.9.4.4 Hilfen bei der Informationssuche im WWW

Richtig nutzen können Sie das WWW sowie das ganze Internet nur, wenn Sie sich der vorhandenen Hilfen zum Suchen und Merken von Informationen und guter Fundstellen bedienen.

Sicher sind Sie mit **Suchmaschinen** ("Search Engines") wie Google (www.google.de) vertraut, so dass an dieser Stelle hierauf nicht näher eingegangen wird.

Wikis und Blogs: Hilfreich sind auch Informationssammlungen in Form von Wikis (Beispiel: Wikipedia) oder Blogs, in denen User Erfahrungen zu bestimmten Bereichen austauschen.

Bookmarks: Wenn Sie interessante Websites im Web gefunden haben, können Sie sich diese über das Anlegen von **Lesezeichen** (in der WWW-Terminologie heißen sie auch „**Bookmarks**“, „**Favoriten**“ oder „**Hotlists**“) merken. Über das Lesezeichenmenü können Sie auf diese Websites dann später einfach per Maus zugreifen.

1.9.4.5 Eigene Dateien über den FIM-Webserver ins Netz hängen

Legen Sie Daten, die Sie im Web veröffentlichen wollen, einfach in das Verzeichnis **public_html** in Ihrem Linux-Home Directory. Diese Daten sind dann über den **FIM-Students-Webserver** allgemein zugreifbar.

Beispiel 1: Ihre CIP-Kennung lautet meier und Sie haben eine Datei namens java.txt nach public_html gelegt. Zu erreichen ist die Datei dann unter der URL <http://students.fim.uni-passau.de/~meier/java.txt>

Sollte es mit dem Zugriff nicht klappen, müssen Sie den Objekten in public_html evtl. noch die richtigen **Zugriffsrechte** geben (vgl. später, 2.1.4; **Rechte setzt man** mit grafisch mit einem **Dateimanager** oder in der **Shell** mit dem Befehl: **chmod 0644 object**). Damit werden für das angegebene Objekt für alle Benutzergruppen die Rechte **Lesen** sowie **Ausführen** gestattet. Vorsicht: Alle Objekte in Ihrem Verzeichnis public_html sind – sofern Sie keine weiteren Schutzmaßnahmen ergreifen - grundsätzlich weltweit lesbar und von Suchmaschinen erreichbar. Das kann aber sehr schnell abgestellt werden:

Beispiel 2: Der CIP-Benutzer meier möchte nun seine Objekte in public_html **vor dem Zugriff durch Suchmaschinen schützen**. Ruft nämlich eine Suchmaschine oder ein anderer Internet-User die URL <http://students.fim.uni-passau.de/~meier/> auf, wird der komplette Inhalt des Verzeichnisses public_html gelistet und von Google indiziert, sofern man das nicht verhindert. Die einfachste Möglichkeit, dieses Listing zu unterbinden, besteht darin, einfach eine (leere) Datei namens index.html in public_html zu erzeugen (mit **kate** oder Shell-Kommando: „**touch index.html**“). Dadurch wird bei Eingabe der URL aus diesem Beispiel das Listing des Directory-Inhalts unterbunden und stattdessen der (leere) Inhalt von index.html angezeigt. Kennt man aber die URL aus Beispiel 1, kann man mit dieser nach wie vor auf die Datei java.txt zugreifen.

1.9.5 VPN und Eduroam: Mit dem eigenen Rechner in Uni-Netzen arbeiten

Manche Angebote der Universität sind nur für Rechner aus dem Netz der Uni Passau zugänglich. Beispiele:

- **Benutzung des Uni-WLANs**
- Zugriff auf bestimmte **Webangebote der Universität, u. a. auch bestimmte Bibliotheksdienste**
- **Erreichbarkeit von Pool- sowie Lehrstuhlrechnern für Remote-Login mit ssh und andere Dienste**

Wenn Sie so ein Angebot mit Ihrem PC daheim oder von Ihrem Mobilgerät aus nutzen wollen ist es nötig, das betreffende Gerät temporär in das Netz der Uni einzubinden. Dazu betreibt das ZIM die Dienste VPN und Eduroam. Installationshinweise und im Fall von VPN die benötigte Software finden Sie auf den ZIM-Webseiten. Bei Fragen und **Problemen mit VPN und Eduroam** wenden Sie sich bitte an die **ZIM-Benutzerberatung!**

Funktionsweise: Bei Nutzung von VPN oder Eduroam wird Ihr Gerät virtuell in das Uni-Netz eingebunden und erhält eine entsprechende Netzwerkadresse, die dann den Zugriff auf die zugangsbeschränkten Angebote erlaubt.

Einsatzgebiete von VPN und Eduroam:

Wenn Sie sich **auf dem Campus** befinden, nutzen Sie am besten **Eduroam**, um Ihre Geräte bequem mit dem Uni-WLAN zu verbinden. Eduroam wird einmalig eingerichtet und liefert dann automatisch beim Einklinken in das entsprechende WLAN die richtigen Zugangsdaten, ohne dass Sie aktiv etwas eingeben müssen. Über diese Technologie kommen Sie bzw. Ihr Gerät außerdem nicht nur ins **WLAN-Netz der Uni Passau**, sondern auch an anderen bayerischen Unis **kostenlos ins WLAN und damit an das Internet**.

Für die Nutzung von **VPN** muss eine entsprechende Software (kann nur von einem Rechner aus geladen werden, der schon im Uni-Netz hängt) installiert werden, die dann bei der Nutzung von **VPN** jedesmal gestartet werden muss. Das ist unbequemer als bei Eduroam, ist aber nötig, wenn Sie von **daheim** aus zugangsbeschränkte Angebote nutzen wollen. Beim Netzzugang von daheim aus ist Eduroam nämlich nicht verfügbar.

1.9.6 Kostenloser Online-Speicher mit LRZ Sync+Share

Das Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften (kurz LRZ) bietet für Beschäftigte und Studierende teilnehmender Hochschulen wie der Universität Passau **kostenlosen Onlinespeicherplatz** an. Der Zugriff erfolgt ähnlich wie bei Dropbox und ermöglicht aktuell die Ablage von bis zu **50 GB** an Daten.

Weiter Infos unter <http://www.zim-uni.passau.de/onlinespeicher>

1.10 Softwareangebote für Studierende

1.10.1 Softwareangebote der Universität

Details zu den folgenden Angeboten finden Sie unter:

<http://www.uni-passau.de/bereiche/beschaefigte/finanzen/it-einkauf/software/sonderkonditionen/>

(Googeln nach „uni passau microsoft“)

Microsoft-Software

Die Universität hat mit Microsoft einen Vertrag abgeschlossen, um allen Ihren Studierenden kostenlos die Nutzung von **Office 365** zu ermöglichen.

Sophos-Virens Scanner

Ebenfalls auf den Webseiten des IT-Einkaufs der Universität finden Studierende der Uni Passau den leistungsfähigen kommerziellen Virens Scanner Sophos für verschiedene Betriebssysteme zum **kostenlosen Download**.

1.10.2 Spezielle Softwareangebote für Teilnehmer am FIM-Lehrbetrieb

Details zu den folgenden Angeboten finden Sie unter:

<http://www.fim.uni-passau.de/it-dienste/lizenzprogramme/>

Für Studierende der FIM bieten wir folgende Software zum **kostenlosen Download** bzw. zur **kostenlosen Nutzung** im Rahmen einer Unilizenz an:

- Python: JetBrains Pycharm
- Softwareentwicklung: IBM Rational Architect
- XML: JetBrains IntelliJ Idea
- UML: Visual Paradigm
- UML: Yatta UML-Lab

2 Vertieftes Arbeiten mit der Linux-Shell

Im Bereich der Informatik und hier insbesondere der Programmierung können Sie viele Vorgänge auf dem Rechner über die Shell effizienter gestalten als mit grafischen Tools. Deshalb soll im zweiten Kapitel das Arbeiten mit der Shell vertieft werden.

2.1 Daten und Directories bearbeiten mit der Shell

2.1.1 Pfade und Working Directories

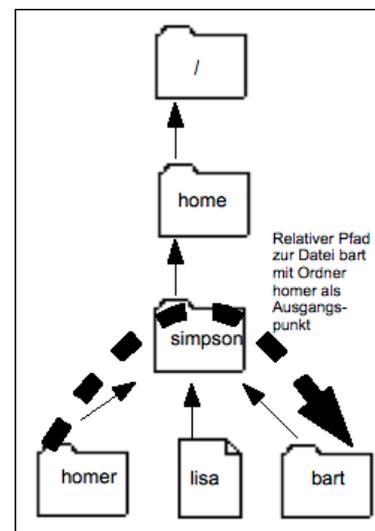
2.1.1.1 Relative Pfade

Pfade benutzt man, um Objekte im Filesystem eindeutig zu adressieren. Im ersten Kapitel haben Sie bereits gelernt, Objekte im Filesystem durch die Angabe des vollständigen (Zugriffs-) Pfads (d. h. dem Weg vom Root-Verzeichnis bis zum Objekt) eindeutig zu beschreiben.

Eine andere Möglichkeit, ein Zielobjekt im Dateibaum zu beschreiben, besteht darin, den **Pfad zum Zielobjekt nicht von der Root, sondern von einem Ausgangsdirectory her** anzugeben. Da diese Art der Pfadangabe nur eindeutig in Relation zum Ausgangspunkt ist, spricht man von einem **relativen Pfad**.

Regeln zum Formulieren relativer Pfadangaben:

- Immer wenn Sie im Dateibaum eine Ebene „nach oben“ gehen müssen, um das Zielobjekt zu erreichen, schreiben Sie zwei Punkte: „..“. Dies ist eindeutig, da jedes Objekt im Dateibaum nur eine Verbindung nach oben besitzt.
- Wenn Sie der Weg zum Zielobjekt „nach unten“ führt, gibt es i.d.R. mehrere Alternativen. Deshalb müssen Sie nun die **Namen** der Directories angeben, über die der Weg zum Ziel führt.
- Als **Trennzeichen** zwischen Directory-Angaben wird wie bei vollständigen Pfaden das „/“-Symbol, also ein „Slash“ verwendet
- Am **Ende** steht der **eigentliche Namen des Zielobjekts**
- **Liegt das Zielobjekt direkt im Ausgangsdirectory**, besteht der **relative Pfad einfach aus dem Objektname des Zielobjekts**, da keine Directories durchschritten werden müssen. Alle Objekte in Ihrem Working Directory können Sie also in Shell-Befehlen adressieren, indem Sie einfach den Objektname als Pfadangabe schreiben.



Relativer Pfad vom Directory homer zum Directory bart (siehe schraffierter Pfeil in der Abbildung):
../bart

Relativer Pfad vom Directory home zur Datei lisa: simpson/lisa

Relativer Pfad vom Directory simpson zur Datei lisa: lisa

2.1.1.2 Vollständige oder Relative Pfade verwenden ?

Beim Arbeiten am Rechner setzen Sie in der Regel die Pfadformulierung ein, die den wenigsten Schreibaufwand bedeutet. Das sind oft relative Pfadangaben, da man sein Ausgangsdirectory meist dahin verlagert, wo sich die zu bearbeitenden Objekte befinden – und dann genügt ja der Objektname als relative Pfadangabe. Möchte man dagegen einem anderen Benutzer z. B. per Mail mitteilen, wo er ein bestimmtes Objekt im Dateibaum findet, muß man vollständige Pfade zur Beschreibung verwenden.

2.1.1.3 Verlagern des Working Directories im Dateibaum

Nach dem Öffnen einer Shell ist als Ausgangsdirectory, d. h. als **Working Directory** Ihr Home Directory eingestellt. Über den Befehl „**cd**“ (**change directory**) kann man das **Working Directory im**

Dateibaum verlagern. Als Parameter gibt man über eine **Pfadangabe** das **Zieldirectory** an, welches das neue Working Directory werden soll.

Wenn Sie mit „cd“ üben und auf Wanderschaft im Dateibaum gehen, können Sie sich mit Hilfe des Kommandos „pwd“ (**p**rint **w**orking **d**irectory) jederzeit anzeigen lassen, wo im Dateibaum Sie sich gerade befinden.

Linux für Schreibfaule: Die Pfadangabe für Ihr **Working Directory** kann durch einen **Punkt** abgekürzt werden: `.`

Beispiele zum Gebrauch des Befehls cd (Bezug nehmend auf die Abbildungen auf der vorigen Seite):

Der Benutzer Simpson hat sich gerade eingeloggt und befindet sich nun in seinem Home Directory /home/simpson. Um sein Working Directory in das Parent Directory /home zu verlegen, hat er folgende Alternativen:

- * Angabe des Ziels über vollständigen Pfad: `cd /home`
- * Angabe des Ziels über relativen Pfad: `cd ..`

Um vom Directory /home aus das Directory /home/simpson/bart als Working Directory einzustellen, kommen folgende Alternativen in Frage:

- * Angabe des Ziels über vollständigen Pfads: `cd /home/simpson/bart`
- * Angabe des Ziels über relativen Pfad: `cd simpson/bart`

Linux für Schreibfaule: Durch **einfache Eingabe** von „cd“ (dann **Return ohne Angabe eines Pfades**), stellen Sie das **Working Directory zurück auf Ihr Home Directory**.

2.1.2 Directory-Inhalte ansehen mit Kommando ls

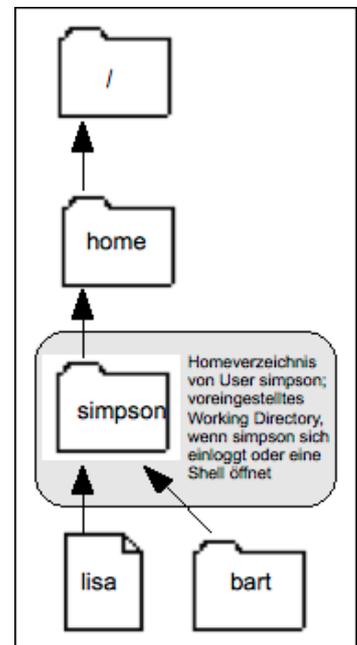
Zur **Auflistung des Inhalts eines Directories** dient das **Kommando**
ls

Beispiel: Der User simpson befindet sich nach dem Einloggen zunächst in seinem Home Directory (siehe Abbildung) und lässt sich darin vorhandene Objekte auflisten. Die Ausgabe besteht aus bart und lisa: `: ~ $`

```
simpson@hecht: ~ $ls
bart/      lisa
```

Durch Angabe von **Optionen** zu **ls** kann man das **Listing verfeinern**:

- Über die Option „l“, also „**ls -l**“ erzeugt man z. B. ein „**long listing**“, das u. a. anzeigt, wem das aufgelistete Objekt gehört.
- Die Option „a“ listet **alle Objekte** im Directory, auch die evtl. **verborgenen Objekte (Hidden Files)**, deren Name mit einem **Punkt beginnt**. **Hidden-Files** sind meist Dateien, die Kontrollinformation für Programme enthalten. Diese Hidden-Files sollte man ohne näheres Fachwissen nicht verändern, deshalb werden diese standardmäßig erst einmal vor dem Benutzer verborgen. In Ihrem Home Directory finden Sie z. B. eine Datei namens ".bashrc", die Kontrollinformation für das Verhalten Ihrer bash enthält.
- Optionen sind kombinierbar. Sie können z. B. schreiben „**ls -al**“



Tipp: Die Anzeige von Hidden-Files im Dateimanager **thunar** können Sie über den Menüpunkt „An-sicht“ aktivieren.

2.1.3 Gängige Shell-Befehle für den Umgang mit Dateien und Directories

Neue Dateien erzeugen Sie meist beim Arbeiten mit Programmen wie z. B. dem Texteditor kate oder der Programmierumgebung eclipse. Neue Directories erzeugen Sie über den Befehl „**mkdir**“.

Über den Befehl „**cp**“ kann man Objekte **kopieren** (bei Directories Option „-r“ nicht vergessen), mit „**mv**“ können Sie Dateien und Directories **umbenennen** oder – wenn als letzter Parameter ein **schon vorhandenes Directory** angegeben wird - Objekte in dieses Directory **verschieben**.

Beispiel: Als Working-Directory sei ihr Home Directory eingestellt, worin sich eine Datei namens datei1 befindet:

```
123 mueller@hecht: ~ $ cp datei1 datei2
    ...legt im Home Directory eine Datei namens datei2 mit gleichem Inhalt wie datei1 an
124 mueller@hecht: ~ $ mkdir verzeichnis1
    ...legt im Home Directory ein Directory namens verzeichnis 1 an
125 mueller@hecht: ~ $ mv verzeichnis1 verzeichnis2
    ...benennt dieses Directory in verzeichnis2 um; gäbe es schon ein Directory namens verzeich-
    nis2, würde verzeichnis1 nicht umbenannt, sondern in das Directory verzeichnis2 verschoben
126 mueller@hecht: ~ $ mv datei1 datei2 verzeichnis2
    ...verschiebt die Dateien datei1 und datei2 in das Directory verzeichnis2
127 mueller@hecht: ~ $ mv verzeichnis2/datei1 .
    ...verschiebt die Datei datei1 in das Working Directory, also hier unser Home Directory
128 mueller@hecht: ~ $ ls verzeichnis2
    ...listet den Inhalt des Directories verzeichnis2, der nun nur noch aus der Datei datei2 besteht
```

Übersicht: Shell-Befehle für den Umgang mit Dateien und Directories

Dateien	
<code>cp <i>quelldatei</i> <i>zieldatei</i></code>	Kopiert <i>quelldatei</i> nach <i>zieldatei</i> (falls <i>zieldatei</i> noch nicht existiert, wird diese angelegt; falls es <i>zieldatei</i> schon gibt, wird deren Inhalt überschrieben)
<code>mv <i>quellobjekt</i> <i>zielobjekt</i></code>	Umbenennen von <i>quellobjekt</i> (Datei oder Directory) nach <i>zielobjekt</i> ; über entsprechende Pfadangaben im Objektnamen kann man natürlich auch Quellen oder Ziele außerhalb des Working Directories festlegen und Objekte damit im Dateibaum verschieben (mv leitet sich von <i>engl. to move</i> ab)
<code>rm <i>datei</i></code>	Löscht die angegebene Datei
<code>less <i>datei</i> (auch: <i>more datei</i>)</code>	Inhalt von <i>datei</i> seitenweise am Bildschirm anzeigen
<code>head <i>datei</i></code>	Gibt die ersten Zeilen von <i>datei</i> auf dem Bildschirm aus
<code>tail <i>datei</i></code>	Gibt die letzten Zeilen von <i>datei</i> auf dem Bildschirm aus
<code>cat <i>datei</i></code>	Gibt den Inhalt von <i>datei</i> komplett auf dem Bildschirm aus
<code>cat <i>quelldatei1</i> <i>quelldatei2</i> > <i>ziel</i></code>	Schreibt den Inhalt der Dateien <i>quelldatei1</i> und <i>quelldatei2</i> in die Datei <i>ziel</i>
<code>diff <i>quelldatei1</i> <i>quelldatei2</i></code>	Vergleicht den Inhalt der Dateien <i>quelldatei1</i> und <i>quelldatei2</i>

Directories	
<code>cd <i>directory</i></code>	Verlagert das Working Directory nach <i>directory</i>
<code>pwd</code>	Gibt den Pfadnamen des aktuellen Working Directories aus
<code>ls [<i>directory</i>]</code>	Listet Dateien und Directories im Working Directory; optional kann man über den Parameter <i>directory</i> auch ein anderes Directory listen lassen
<code>mkdir <i>neuesdirectory</i></code>	Erzeugt das Directory <i>neuesdirectory</i>
<code>cp -r <i>quelldirectory</i> <i>zieldirectory</i></code>	Legt vom <i>quelldirectory</i> eine Kopie namens <i>zieldirectory</i> an (rekursiv, d.h. mit allen enthaltenen Dateien und Subdirectories)
<code>rmdir <i>directory</i></code>	Löscht <i>directory</i> (wenn leer)
<code>rm -r <i>directory</i></code>	Löscht <i>directory</i> rekursiv , d.h. mit allen Dateien und Unterverzeichnissen

2.1.4 Zugriffsrechte setzen und entfernen

Die FIM-Poolrechner sind Multiuser-Systeme, d. h. man teilt sich insbesondere das Dateisystem mit vielen anderen Benutzern. Standardmäßig sind Ihre Daten für andere Nutzer (außer den Systemadministratoren) nicht lesbar oder veränderbar. Sie erfahren nun, wie man die zu Grunde liegenden **Zugriffsrechte** ansieht und beeinflusst.

Zunächst muss dazu der Begriff der **Gruppe** (engl. group) eingeführt werden. Jeder Benutzer ist mindestens einer Benutzergruppe zugeordnet. Dies ist sinnvoll, wenn z. B. Benutzer gemeinsam ein Projekt bearbeiten. Der Systembetreuer kann für das Projekt dann eine Benutzergruppe einrichten, in die alle Projektmitarbeiter eingetragen werden. Die Mitglieder dieser Projektgruppe können für die im Projekt verwendeten Dateien dann besondere Zugriffsprivilegien für Gruppenmitglieder, aber Zugriffshürden für Benutzer außerhalb der Gruppe festlegen. Über den Befehl "**id**" kann man seine Gruppenzugehörigkeit(en) erfahren.

Jede Linux-Datei und jedes Directory wird beim Anlegen einem Eigentümer sowie einer Gruppe zugeordnet. Der Zugriff auf ein Objekt kann entsprechend für drei **Benutzerklassen** geregelt werden, nämlich

- * den **Eigentümer** (user, im Normalfall der Erzeuger des Objekts)
- * die **Gruppe** (group: Mitglieder der Gruppe, der ein Objekt zugeordnet ist)
- * die **restlichen Systembenutzer** (others)

Für diese drei Benutzerklassen können nun für alle Objekte im Dateisystem Zugriffsrechte vom jeweiligen **Objekt-Eigentümer** eingestellt werden. Unabhängig davon, wie der Eigentümer eines Objekts diese Rechte setzt, können jedoch die Systembetreuer (Benutzerkennung „root“) trotzdem auf alle Objekte im Dateibaum zugreifen.

Zugriffsrechte für Dateien:

*	r	Read , d. h. Lesezugriff auf eine Datei
*	w	Write , d. h. Schreibzugriff auf eine Datei (Ändern des Inhalts bis hin zum Löschen der Datei)
*	x	Execute , d. h. das Recht, eine Datei bzw. das hierin abgelegte Programm auszuführen

Zugriffsrechte für Directories:

*	r	Read , d. h. das Recht, den Inhalt des betreffenden Directories auflisten zu lassen
*	w	Write , d. h. das Recht auf Anlegen und Löschen von Objekten in einem Directory
*	x	Execute , d. h. das Recht, mit cd in das betreffende Directory zu wechseln oder durch eine Pfadangabe auf Objekte in diesem Directory zuzugreifen (z. B. bei Webseiten)

Beispiel: Über den Shell-Befehl „**ls -al**“ erzeugt man ein ausführliches Listing des aktuell eingestellten Working-Directories erzeugt. Im Kasten unten sehen Sie eine typische Ausgabe, wie man Sie nach „**ls -al**“ erhält. Jede Zeile beschreibt ein Objekt. In Spalte 1 jeder Zeile sieht man die Art des Objekts (bei Directories steht hier ein „d“), in Spalte 2-4 stehen die Zugriffsrechte, die für den Eigentümer des Objekts eingestellt sind, in Spalte 5-7 die Zugriffsrechte für die Gruppe und in Spalte 8-10 die Zugriffsrechte für die restlichen Systembenutzer. In der Abbildung darf die Datei .bashrc von jedermann gelesen, aber nur vom Eigentümer auch geändert werden. Je nach Voreinstellung für das Kommando ls finden Sie in der Ausgabe hinter **Directorynamen** einen Slash („/“), hinter **ausführbaren Dateien** ein Asterisk („*“). Das mit „/“ gelistete Directory steht für das Working-Directory, das mit „..“ aufgeführte Directory steht für das Parent Directory des Working Directories.

drwx-----+	47	simpson	general	4096	Feb	5	14:55	./
drwxr-xr-x+	12	root	wheel	512	Feb	1	16:17	../
-rw-r--r--+	1	simpson	general	41	Feb	1	13:59	.bashrc
-rwx-----+	1	simpson	general	869	Sep	5	18:13	javaprogl*
...								
<i>Dateiart und</i>	<i>Eigentümer</i>	<i>Gruppenzuordnung</i>	<i>Dateigröße</i>				<i>Name der Datei/des Directories</i>	
<i>Zugriffsrechte</i>	<i>Anzahl der Verweise auf Objekt</i>			<i>Zeitpunkt der letzten Änderung</i>				

Zugriffsrechte kann der **Eigentümer** eines Objekts entweder mit einem **grafischen** Dateimanager oder mit dem Shell-Befehl **chmod** ändern:

- Zugriffsrechte **verändern** mit Dateimanager thunar: **Objekt mit der rechten Maustaste anklicken** und hier **Eigenschaften** auswählen. Unter **Berechtigungen** können Sie über **verschiedene Berechtigungen** einstellen.
- In der **Shell** können sehr einfach über **Buchstabenkürzel** die **Rechte r, w, und x** für den **Eigentümer („u“)**, **Gruppenmitglieder („g“)**, **die restlichen Systembenutzer („o“)** und **alle zusammen („a“)** eingestellt werden. Beispiel „Leserecht für alle“ für eine Datei setzen:
`chmod a+r dateiname`

2.1.5 Speicherbedarf von Benutzerdaten optimieren

Damit unsere Festplatten nicht überlaufen haben wir für die Home Directories eine **Quota**-Regelung (Standard: 8 GB pro User) eingeführt. Sollte Ihr Speicherplatz im Home Directory nicht ausreichen, können Sie bei den Systembetreuern mit kurzer **Begründung** eine **Quotaerhöhung** beantragen (z. B. für Bachelorarbeit).

Quotainformationen abrufen (Achtung: Eingabe des Passworts nötig, da damit ein spezieller Server abgefragt wird):

```
102 vkboss@hecht: ~ $ cipequota
Quota report for user "vkboss":
  quota: 15 GiB
  used:  6 GiB ( 40 %)
  free:  9 GiB ( 60 %)
```

Wenn Sie **genauer wissen wollen, wo die Daten liegen, die am meisten Speicherplatz belegen**, z. B. um aufzuräumen, können Sie mit dem Kommando **du** den aktuell belegten **Speicherplatz eines Verzeichnisses** (inkl. Unterverzeichnisse) anzeigen lassen. Ist die Ausgabe zu ausführlich, kann man sich über die Option „-s“ auch nur die Gesamtsumme des belegten Speicherplatzes ausgegeben lassen, ohne dass jedes Unterverzeichnis separat gelistet wird. Standardmäßig erfolgt die Speicherplatzausgabe in **Bytes**. Mit der Option „-h“ erfolgt die Ausgabe in Gigabyte, Megabytes usw. und ist besser lesbar.

Mit Kommandos wie **gzip** oder **bzip2** kann man Dateien komprimieren, so dass sie weniger Platz benötigen (Dekomprimieren mit **gunzip** bzw. **bunzip2**).

2.2 Prozesse beobachten und manipulieren

2.2.1 Der Zusammenhang Programm-Prozess

Wenn man ein Shell-Kommando oder ein anderes Programm startet, muss man unterscheiden zwischen

- * der **Programmdatei**, die die Repräsentation des Programms in Form eines Textes oder Bytecodes beinhaltet
- * der **Programmausführung**, bei der die Instruktionen, die in der Programmdatei enthalten sind, auch tatsächlich auf dem Prozessor des Computers zur Ausführung gebracht werden

Bei jeder Programmausführung wird auf Ebene des Betriebssystems ein so genannter **Rechen-Prozess** (auch: **Job** oder **Task** genannt) gestartet, der das Programm bei der Ausführung repräsentiert. Ein **Prozess** hat zwei Aufgaben:

- * Er sorgt dafür, dass die im Programmtext festgelegten Berechnungen auf dem Prozessor ausgeführt werden
- * Er verwaltet die Umgebung des Programms (Prozessumgebung) während dessen Ausführung im Rechner

Zur **Prozessumgebung** gehören z.B. Informationen darüber, auf welcher Programmdatei der Prozess basiert, welche Teile des Arbeitsspeichers belegt werden, wer den Prozess gestartet hat, oder wieviel Rechenzeit der Prozess schon verbraucht hat. Um systemintern Prozesse eindeutig unterscheiden zu können, erhält jeder Prozess vom Betriebssystem eine **Prozessnummer** (Jobnummer, Prozess-ID, oder kurz **PID**).

Auf Linux-Rechnern werden immer mehrere Prozesse gleichzeitig ausgeführt werden. Man spricht von **Multitasking**. In der Praxis sind neben Prozessen, die von Benutzern gestartet werden, immer auch Prozesse in Bearbeitung, die vom **Betriebssystem** gestartet werden, um den Computer zu **verwalten**.

Um zu verhindern, dass ein Prozess ewig lang den Prozessor für seine Berechnungen belegt und damit andere Prozesse blockiert, sorgt ein Programm des Betriebssystems dafür, dass jeder Prozess nach einer festgelegten Zeitspanne (man spricht von einer **Zeitscheibe**) den Prozessor wieder freigeben muss. Jeder Prozess besitzt eine **Priorität**, die entscheidet, wie oft er Zugriff auf den Prozessor erhält und wie lang die jeweilige Zeitscheibe dauert.

2.2.2 Details zum Programmaufruf über die Shell

Startet man ein Programm oder Linux-Kommando in der Shell, ist diese dafür zuständig, dass eine passende Datei im Dateisystem gefunden und ausgeführt wird.

Gibt man dabei nur den eigentlichen Kommando- oder Programmnamen ohne weitere Pfadinformationen an, durchsucht die Shell dabei standardmäßig eine Auswahl von **Directories**, die als **Suchpfad** in der **Shell-Variable "PATH"** voreingestellt sind. Über das Kommando "**echo \$PATH**" lässt man sich anzeigen, **welche Directories in welcher Reihenfolge im Rahmen dieses Standard-Suchpfads abgesucht werden**. Findet die Shell in einem dieser Directories eine ausführbare Datei mit Namen des eingegebenen Kommandos/Programms, wird das Absuchen der Directories in \$PATH abgebrochen und die gefundene Datei zur Ausführung gebracht, also ein entsprechender **Prozess** gestartet. Findet die Shell in keinem der Directories in \$PATH eine gleichnamige ausführbare Datei, erhält man die Fehlermeldung „**Kommando nicht gefunden**“.

Über "**type befehl**" kann man sich ausgeben lassen, ob überhaupt bzw. in welchem Directory in \$PATH die Shell eine ausführbare Datei zu einem bestimmten Befehl findet.

Wollen Sie **Dateien ausführen, die nicht in einem der Directories in \$PATH** liegen, müssen Sie der Shell den Ort der auszuführenden Datei mitteilen, indem Sie statt dem einfachen Dateinamen den **vollständigen oder relativen (Zugriffs-) Pfad** für die gewünschte Datei angeben.

Hinweis: Selbst erstellte ausführbare Dateien können Sie z. B. **im Verzeichnis ~/bin sammeln**. Alles was sich in Ihrem Home Directory im Subdirectory "bin" befindet, können Sie wie normale Linux-Befehle durch einfaches Eingeben des Dateinamens starten, ohne längere Pfade mit angeben zu müssen. Wir haben nämlich in der Shell-Variable PATH voreingestellt, dass bei der Suche nach ausführbaren Programmen auch Ihr eigenes bin-Directory berücksichtigt wird.

2.2.3 Orientierung über laufende Prozesse mit „ps“ und „top“

Einen Überblick über die aktuellen Prozesse auf dem Linux-System erhält man über das Kommando „ps“

Bei Eingabe von **ps ohne Optionen** in einer Shell bekommt man **seine eigenen Prozesse** aufgelistet, genauer, nur die eigenen Prozesse, **die an die Shell gebunden sind**, wo Sie ps eingegeben haben. Die **Ausgabe** von ps zeigt zu jedem angezeigten Prozess das zugehörige **Startkommando** sowie die hierfür vom System vergebene systemweit eindeutige **Prozessnummer (PID)** an. Durch Angabe von Optionen kann man z. B. steuern, ob man **alle eigenen Prozesse** "**ps -fu cippkennung**" oder die **Prozesse aller Systembenutzer** sehen möchte und welche Details „ps“ als Ausgabe liefert. Z. B. zeigt "**ps -elf**" die **Prozesse aller Benutzer im long format** auf Ihrem Rechner.

Beim Arbeiten im CIP-Pool werden Sie manchmal bemerken, dass Ihr Rechner in seiner Geschwindigkeit auf einmal nachlässt, obwohl Sie selbst keine rechenzeitintensiven Prozesse laufen lassen. Das liegt dann meist daran, dass auf Ihrem Rechner noch Prozesse von anderen Benutzern oder Systemprozesse laufen.

Welche Prozesse Ihren Rechner am meisten belasten, stellen Sie über das Kommando "**top**" fest.

2.2.4 Beenden bzw. Abbrechen von Programmen bzw. Prozessen

Bitte benutzen Sie im Normalfall die regulären Funktionen, die Ihnen das jeweilige Programm zum Beenden bietet.

Beenden von Programmen, wenn reguläres Beenden nicht funktioniert

- **Tastenkombinationen bei Programmen, die an eine Shell gebunden sind:** <Strg-c> sowie <Strg-\>
- "**kill**"-Befehl zum „Abschießen“ von Prozessen: Dazu braucht man die **Prozessnummer** des zu beendenden Prozesses, die man durch „ps“ oder „top“ in Erfahrung bringt. "kill 948" würde dann z. B. den Prozess mit der Nummer 948 beenden. In hartnäckigen Fällen kann zusätzlich die **Option „-9“** verwendet werden: "kill -9 948". Hierdurch sendet "kill" ein stärkeres Terminierungssignal, das dann allerdings dazu führt, dass der zu killende Prozesse keine Chance mehr hat, hinter sich sauber aufzuräumen (z. B. temporäre Dateien, die er für die Prozessverwaltung angelegt hat).

Praxisbeispiel: Was tun, wenn der **Linux-Rechner hängt und nicht mehr auf Eingaben** reagiert

Loggen Sie sich auf diesem Rechner alphanumerisch ein (lokal, siehe unten, bzw. falls nicht möglich, von einem anderen Poolrechner aus per ssh (2.3.1.)). Lassen Sie sich alle Ihre **Prozesse listen** (ps -fu cippkennung) und beenden Sie mit dem **kill – Kommando** die Prozesse, die den Rechner blockieren.

Start einer **Shell-Sitzung in einer alphanumerischen Umgebung** parallel zur Grafikumgebung:

- **Umschalten von der grafischen auf die alphanumerische Oberfläche:** Drücken Sie die **Tastenkombination „Strg-Alt-F1“** (statt F1 funktionieren alternativ manchmal auch F2 bis F6).
- Nach dem **Login in der alphanumerischen Oberfläche mit Kennung und Passwort** startet eine **Shell-Sitzung**, wo Sie kommandoorientiert arbeiten und z. B. Prozesse auflisten und abschießen können
- **Zurückschalten zur grafischen Benutzeroberfläche:** Über die **Tastenkombination „Strg-Alt-F8“** (statt F8 oft alternativ auch F7) kommen Sie jederzeit wieder zum **graphischen Modus zurück**.
- **Wichtig: Ausloggen aus der alphanumerischen Shell-Sitzung nicht vergessen, wenn Sie parallel eine grafische und eine alphanumerische Sitzung laufen haben!**

2.2.5 Hintergrundprozesse

Wenn Sie in einer Shell Programme wie z. B. „firefox“ starten, die ein **eigenes Fenster** aufmachen, ist die **Shell blockiert**, bis man so ein Programm beendet hat. Erst danach erscheint ärgerlicherweise der Eingabeprompt dieser Shell wieder. Das umgehen Sie, indem Sie **bei solchen Programmen beim Aufruf aus der Shell immer gleich das Symbol „&“ anhängen** (z. B. „firefox &“), damit das Programm als **Hintergrundprozess** startet und dadurch **von der Shell entkoppelt wird**. Grundsätzlich empfiehlt es sich, alle **Programme, die nach dem Start eigene Fenster öffnen, immer als Hintergrundprozess zu starten**.

Allerdings besitzen Sie dann von der Shell aus keine Einflussmöglichkeiten mehr auf die Ausführung des im Hintergrund befindlichen Prozesses. Z. B. können Sie einen Hintergrundprozess nicht mehr durch Drücken etwa der Tastenkombination <STRG - c> abbrechen. Um einen **Hintergrundprozess** wieder an eine **Shell zu binden**, müssen Sie ihn wieder in die interaktive Ebene, den „**Vordergrund**“ bringen. Dies geschieht, indem Sie in der Shell, von der aus der Prozess im Hintergrund gestartet wurde, das Kommando „fg“ (foreground) eingeben.

Haben Sie z. B. beim Start von firefox das „&“ **vergessen**, können Sie so einen Prozess **nachträglich in den Hintergrund schieben**. Drücken Sie zunächst die Tastenkombination <STRG-z>, um die Ausführung des Prozesses anzuhalten und starten dann über das Kommando „bg“ (background) dessen Fortsetzung im Hintergrund.

Vorsicht: Noch laufende Hintergrundprozesse verhindern das Ausloggen und müssen deshalb vor dem Ausloggen beendet werden. Hierzu bringt man diese zunächst über "fg" in den Vordergrund, um sie interaktiv zu beenden, oder man benutzt das oben erklärte Kommando kill.

2.2.6 Sequentielle Ausführung von Prozessen

Bisher haben Sie beim Arbeiten unter Linux in jeder Befehlszeile nur einen Befehl formuliert. Sie brauchen aber nicht jedes Kommando in eine neue Zeile zu schreiben, sondern können mehrere Kommandos nacheinander ausführen, indem Sie diese durch ein Semicolon „;“ trennen. Beispiel: date; id; date

2.2.7 Ausführung von lang laufenden Prozessen mit cip-run

Die aktuelle Voreinstellung in den FIM-Pools sieht vor, dass alle Prozesse eines Users beim Logout automatisch beendet werden. Manchmal ist es aber wichtig, Prozesse so starten zu können, dass diese das Logout des Users überleben. Solche Prozesse könnten z. B. lang laufende Berechnungen im Rahmen Ihrer Abschlussarbeit sein.

Für diesen Fall stellen wir Ihnen das Shell-Skript „**cip-run**“ zur Verfügung, mit dem Sie gezielt einzelne Prozesse über das Beenden Ihrer Session hinaus erhalten können.

Beispiel 1: Ausführung eines lang laufenden Compute-Prozesses mit cip-run

Sie haben im Rahmen Ihrer Bachelor-Arbeit ein Programm „intense_compute“ geschrieben, das Berechnungen ausführt, die mehrere Stunden oder sogar Tage dauern. Um zu verhindern, dass dieses Programm beim Logout beendet wird, starten Sie es einfach über die Option cip-run:

```
123 mueller@enterprise: ~ $ cip-run intense_compute
```

Ist der so gestartete **Prozess sehr rechenintensiv**, sollte man darauf Rücksicht nehmen, dass Rechner durch solche Prozesse sehr stark belastet und evtl. verlangsamt werden, wodurch ein normales Arbeiten für andere Benutzer so eines Rechners kaum mehr möglich ist. Dies vermeidet man, indem man über die Kommandos „**nice**“ bzw. „**renice**“ schon beim Start oder während der Ausführung des Programms die zugehörige **Prozess-Priorität** (steuert die Zuteilung von Prozessorzeit) deutlich verringert.

Evtl. **Ausgaben** eines solchen Prozesses können natürlich nicht auf den Bildschirm ausgegeben werden, sondern müssen **in eine Datei** umgeleitet werden (siehe später: Umlenken der Ausgabe...).

Beispiel 2: Bash-Sessions über den Logout-Vorgang hinaus erhalten

Normalerweise werden bash-Sessions beim Logout beendet. Im Rahmen z. B. von Bachelorarbeiten ist es zur Steuerung lang laufender Prozesse aber manchmal nötig, dass die zur Steuerung verwendeten bash-Sessions ebenfalls über das Logout hinaus erhalten bleiben. Hierzu nutzt man den **Terminal Multiplexer tmux**. tmux erlaubt es, mehrere bash-Sessions über ein einzelnes Fenster zu starten und zu verwalten. Wenn Sie **tmux über das cip-run-Kommando starten**, bleiben der tmux-Prozess sowie die hierzu gehörenden bash-Sessions über das Logout hinaus erhalten. Über tmux können Sie sich bei einem späteren Login auf dem entsprechenden Rechner wieder mit diesen bash-Sessions verbinden und so Ihre Prozesse weiter steuern. Die Verbindung klappt auch per ssh (siehe unten, 2.3.1):

```
ssh <user@hostname> -t tmux a
```

Wichtig: Melden Sie lang laufende Prozesse bitte im Vorfeld bei der FIM-Systembetreuung an. Dann werden solche Prozesse bei der Systemwartung nicht automatisch abgeschossen werden.

2.2.8 Systemsicherheit: Logging von Benutzeraktionen

Auf Linux-Systemen werden automatisch viele **Prozessaktivitäten mitprotokolliert** und über eine gewisse Zeit gespeichert. Dies geschieht zum einen, um bestimmte Systemparameter beobachten zu können, zum anderen aus Gründen der Systemsicherheit. Wir weisen Sie hiermit darauf hin, dass im Ernstfall auch viele Ihrer Benutzeraktionen nachvollzogen und nachgewiesen werden können.

2.3 Remote-Ausführung von Befehlen, Remote-Zugriff auf Daten

Im Folgenden geht es darum, über das Rechnernetz vom eigenen Arbeitsplatz aus Befehle auf anderen Rechnern im Netz ausführen zu lassen oder auf deren Datenbestand zuzugreifen. Dies ist z. B. sinnvoll, wenn man

- die Ressourcen (z. B. Rechenleistung) eines anderen Rechners im Netz nutzen will
- ein Programm nutzen möchte, das auf dem Rechner, an dem man sitzt, nicht installiert ist, wohl aber auf einem anderen Rechner im Netz
- ein selbst geschriebenes Programm auf einer anderen Rechnerarchitektur testen möchte
- von daheim aus auf Rechner und Ressourcen im Uni-Netz zugreifen möchte

Ausführliche Beschreibungen hierzu mit Hinweis auf benötigte Tools finden Sie auf der FIM-Website unter [IT-Dienste \(Studentenrechnerpools, Anleitungen\)](#).

2.3.1 Secure Shell: Befehle auf anderen Rechner im Netz ausführen lassen per ssh

Um Kommando-orientiert auf anderen Rechnern Befehle ausführen zu lassen nutzt man das Tool ssh.

```
ssh user@rechner
```

öffnet eine Shell-Sitzung auf dem angegebenen Rechner unter der angegebenen Kennung. Beispiel: „ssh vkboss@enterprise.fim.uni-passau.de“. Lassen Sie den Kennungsnamen weg, erfolgt das Login unter der Kennung, unter der Sie gerade arbeiten. Durch Eingabe von „exit“ oder Strg-D wird die ssh-Sitzung beendet.

Die Abkürzung ssh steht für "**Secure Shell**".

Bei Verwendung von ssh werden Informationen abhörsicher in verschlüsselter Form übertragen. Dazu verwendet jeder Rechner einen eigenen Schlüssel, den so genannten "**Host Key**". Wenn Sie das erste Mal eine ssh-Verbindung zu einem Rechner aufbauen, müssen Sie evtl. beim Verbindungsaufbau explizit mit „yes“ bestätigen, dass Sie den Keyangaben des Zielrechners trauen. Dabei speichert der lokale Rechner den Key des entfernten Rechners und prüft ihn bei jedem zukünftigen Verbindungsaufbau, um sicherzustellen, dass man mit dem "echten" Rechner verbunden ist, also niemand die über die Verbindung ausgetauschten Daten mitzuschneiden versucht.

Beispiel: Öffnen einer Shell für den User vkboss auf dem Rechner enterprise:

```
102 vkboss@hecht: ~ $ ssh vkboss@enterprise
```

```
The authenticity of host 'enterprise (132.231.1.122)' can't be established.
```

```
DSA key fingerprint is c6:8b:a9:5b:86:f2:d0:88:ee:e0:2c:4c:46:a7:b3:ab.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'enterprise,132.231.1.122' (DSA) to the list of known hosts
```

2.3.2 Grafikorientiertes Arbeiten auf anderen Rechnern

Wenn man grafikorientiert auf einem anderen, entfernten Rechner im Netz arbeiten möchte, kann man entweder ein komplettes Sitzungsfenster auf den eigenen Rechner übertragen oder mit Hilfe von ssh gezielt einzelne Programme mit grafischer Oberfläche auf dem entfernten Rechner starten und auf dem eigenen Rechner anzeigen lassen.

Für das **Übertragen einer kompletten Sitzung** empfiehlt sich das Programm X2Go (Ausführliche Beschreibung siehe FIM-Website unter [IT-Dienste \(Studentenrechnerpools, Anleitungen\)](#)). Dies ist die sinnvollste Alternative, wenn man z. B. vom Windows- oder MacOS-Rechner daheim aus grafikorientiert auf die FIM-Poolrechner im Linux-Betrieb zugreifen möchte.

Einrichtung einer Grafikweiterleitung per ssh

Hat man daheim auch LINUX laufen, kann man sich auch zunächst per ssh z. B. auf einem FIM-Poolrechner einloggen und dann gezielt die Grafikausgabe einzelner Programme auf dem Rechner daheim anzeigen lassen.

Die **Grafikumleitung (Display Forwarding)** zwischen zwei Rechnern **im CIP-Pool erfolgt automatisch**, wenn Sie eine Shell auf einem anderen Rechner im CIP-Pool öffnen.

Ansonsten besteht die Notwendigkeit, das **Display Forwarding vom entfernten Rechner auf den eigenen Rechner beim Aufbau der ssh-Verbindung explizit zu aktivieren**. Das funktioniert, indem man **beim Aufbau der ssh-Verbindung die Option „-X“** mit gibt.

Beispiel: Starten einer **Shell-Sitzung mit Grafik-Weiterleitung** auf dem **Rechner enterprise** von daheim aus:

- * Voraussetzungen: Läuft der Rechner daheim unter Linux, haben Sie bereits alle Software, die Sie brauchen (ssh sowie X-Server), an Board. Läuft der Rechner daheim unter Windows, benötigen Sie für eine ssh-Verbindung ein Hilfsprogramm wie z. B. **putty**, sowie für die Anzeige X-basierter Grafik unter Windows z. B. das kostenlose **Cygwin**.
- * Binden Sie nun zuerst Ihren **Ausgangsrechner daheim per OpenVPN in das Uninetz ein**, um die FIM-Poolrechner überhaupt erreichen zu können.
- * **Bauen Sie dann die ssh-Verbindung auf:**

Aufbau der ssh-Verbindung zu enterprise („-X“ sorgt für die Grafikweiterleitung):

```
ssh -X mueller@enterprise.fim.uni-passau.de
```

- * Starten Sie nun ein Programm auf "enterprise", z. B. durch Eingabe von „kate &“, wird das zu kate gehörige Fenster auf dem Bildschirm Ihres Rechners daheim angezeigt.

2.3.3 Dateien im Netz übertragen: Secure Copy (scp) oder per FIM-Webpace

Analog zum **Copy-Kommando cp, mit dem man Objekte im gleichen Dateisystem kopieren kann**, gibt es für das **Kopieren im Netzwerk** die Variante **Secure Copy** (Kommando: **scp**). Damit können Sie z. B. Daten via Internet vom Rechner daheim auf unsere FIM-Rechner kopieren bzw. umgekehrt. scp finden Sie auf jedem Linux-Rechner, Windows-User finden scp als Bestandteil des schon erwähnten putty (heißt hier dann pscp) oder nutzen „**winscp**“. Beachten Sie bitte wieder, dass Sie von daheim die FIM-Poolrechner nur in Verbindung mit OpenVPN erreichen können.

Beispiel: Kopieren des Directories java aus dem Home Directory des Users vkboss auf Rechner enterprise in das Working Directory des lokalen Rechners:

```
scp -r vkboss@enterprise.fim.uni-passau.de:~/java .
```

Wenn Ihnen das zu aufwändig ist, legen Sie die FIM-Daten, die Sie nach Hause kopieren wollen, einfach in das Verzeichnis **public_html** in Ihrem Home Directory. Diese Daten sind dann über den **FIM-Students-Webserver** weltweit (auch von anderen Benutzern!) zugreifbar (siehe 1.9.4.5.).

2.4 Shell-Nutzung für Fortgeschrittene

2.4.1 Verwendung von Platzhaltern und Wildcards

Beim Arbeiten mit Datei- und Directorynamen oder generell Zeichenfolgen ist es oft arbeitserleichternd, **Wildcards** zu nutzen. Wildcards sind Symbole, die als **Platzhalter für ein oder mehrere beliebige Zeichen** dienen. Durch die richtige Verwendung von Wildcards kann man zum einen Tipparbeit sparen, zum anderen Befehle auf sehr viele Zielobjekte (in der Praxis sind dies meist Dateien) gleichzeitig anwenden.

*	Diese Wildcard steht für eine beliebige Zeichenkette nicht festgelegter Länge
?	Diese Wildcard ersetzt ein beliebiges Einzelzeichen in einer Zeichenfolge
[xyz]	Dieser Ausdruck steht für eines der in der Auswahl zwischen den Klammern angegebenen Zeichen, hier also für ein „x“, „y“ oder „z“ (nicht aber z. B. für „xy“)

Beispiel zur Verwendung von Wildcards beim Arbeiten mit der Shell:

Voraussetzung: Man befindet sich in einem Directory, das die Dateien gard und garfield enthält sowie eine Unmenge anderer Dateien, deren Namen aber nicht mit "g" beginnen. Man möchte nun alle Dateien in diesem Directory listen lassen, deren Namen mit der Buchstabenfolge "gar" beginnen und aus genau 4 Buchstaben bestehen:

<code>ls</code>	würde als Ergebnis ein unübersichtliches Listing aller Dateien produzieren
<code>ls gar*</code>	würde als Ergebnis die Ausgaben "gard" und "garfield" liefern
<code>ls gar[efg]*</code>	würde als Ergebnis allein die Ausgabe "garfield" liefern
<code>ls gar?</code>	würde als Ergebnis die gewünschte Ausgabe, hier "gard", liefern

Vorsicht ist geboten, wenn man Wildcards in Zusammenhang mit Befehlen verwendet, bei denen Dateien überschrieben oder gelöscht werden können, also insbesondere beim Verwenden des Befehls „rm“.

Dateimanager und Wildcards: Während einfache Dateimanager wie thunar nicht mit Wildcards umgehen können, funktioniert das z. B. recht gut in der Suchfunktion von nautilus.

2.4.2 Umlenken von Ein- und Ausgaben, Befehle verbinden über Pipes

Auf Linux-Rechnern sind die **Tastatur** als **Standardeingabe** und der **Bildschirm** als **Standardausgabe** vordefiniert, da dies i. d. R. die bevorzugten Ein- bzw. Ausgabegeräte sind. Die meisten Systemprogramme in Linux (d. h. auch die Systembefehle) sind deshalb so konstruiert, dass sie ihre Information von der Standardeingabe lesen und ihre Ergebnisse auf die Standardausgabe schreiben. Da die Shell den Datenfluss zwischen Prozessen sowie der Standardein/ausgabe steuert, ist es möglich, das Standardverhalten zu ändern und z. B. **Dateien als Quelle bzw. Ziel für die Standardein- bzw. ausgabe festzulegen**:

<code>befehl > datei</code>	Lenkt die Ausgabe des Befehls in die genannte Datei um. Achtung: Falls die Zieldatei schon existiert, wird deren Inhalt ohne Nachfrage gnadenlos überschrieben!
<code>befehl >> datei</code>	Fügt die Ausgabe des Befehls an den bisherigen Inhalt der genannten Datei an
<code>befehl < datei</code>	Liest die für den Befehl benötigten Eingabedaten von der Datei "datei" ein
<code>befehl1 befehl2</code>	Die durch befehl1 produzierte Ausgabe wird unmittelbar (d. h. ohne Zwischenspeicherung in einer Datei) über eine sogenannte Pipe zur Eingabe von befehl2

Beispiel 1: Die Ausgabe des Kommandos ls soll in der Datei „listing“ gespeichert werden:

```
126 mueller@hecht: ~ $ ls > listing
```

Beispiel 2: Man möchte wissen, wie viele Wörter die Ausgabe des Kommandos `ls` enthält. Über eine Pipe macht man die Ausgabe von `ls` zur Eingabe des Kommandos `wc`, dessen Funktion es ist, Wörter zu zählen:

```
126 mueller@hecht: ~ $ ls | wc
```

2.4.3 Das Feintuning der Arbeitsumgebung über die Shell

Ihre **Shell** - in den FIM-Studentenrechnerpools standardmäßig die "**bash**" - fungiert als Benutzerschnittstelle und sorgt in Form von Voreinstellungen für Ihre **Benutzerumgebung (Environment)**. Diese Voreinstellungen sind in **Shell-Variablen** abgespeichert.

Übung: Lassen Sie sich durch Eingabe des Kommandos "`env`" anzeigen, welche globalen Shell-Variablen beim Einloggen mit welchen Einstellungen belegt wurden.

Beispiele für Shell-Variable:

<code>HOME</code>	Home Directory
<code>EDITOR,VISUAL</code>	Standardmäßiger Editor
<code>PATH</code>	Directories, worin die Shell bei der Kommandoausführung nach ausführbaren Dateien sucht

Befehle zum Belegen und Anschauen von Shell-Variablen:

<code>env</code>	Liefert die Liste der vorbelegten Shell-Variablen samt Wert
<code>export variable=wert</code>	Die Shell-Variable "variable" wird mit dem Wert "wert" belegt
<code>echo \$variable</code>	Der Wert der Shell-Variable "variable" wird ausgegeben

Ihre für Sie vordefinierte Umgebung erhält neben den Shell-Variablen noch andere Objekte, die man als **Alias** bezeichnet. Über **Alias-Definitionen** kann man **Kürzel** für längere Kommandos vereinbaren.

<code>alias</code>	Gibt eine Übersicht über alle gültigen Alias-Definitionen
<code>alias fire='firefox'</code>	Nach Eingabe dieser Alias-Definition können Sie Firefox durch Eingabe des Kürzels "fire" starten.
<code>unalias fire</code>	Löscht die Alias-Definition für das Kürzel "fire"
<code>alias x='chmod +x'</code>	Da zwischen <code>chmod</code> und <code>+</code> ein Leerzeichen vorkommt, müssen Sie die Zeichenfolge durch Setzen eines Apostrophs am Anfang und Ende als eine Einheit kennzeichnen. Sie können dann statt der Zeichenfolge " <code>chmod +x</code> " als Kürzel einfach nur " <code>x</code> " schreiben.

Der Alias-Mechanismus sowie die Verwaltung der Shell-Variablen sind Eigenheiten der jeweils verwendeten Shell. Die Kommandos zur Manipulation von Shell-Variablen und Alias-Definitionen, die in der von Ihnen bevorzugten Shell gelten, können anders heißen als hier aufgeführt und sind in der Manual Page zur `bash` bzw. der von Ihnen gewählten Shell aufgelistet (nicht aber unter den obigen Kommandonamen!).

Sie selbst besitzen die Möglichkeit, Shell-Variablen neu- bzw. um zu belegen. Ebenso dürfen Sie selbst Alias-Definitionen anlegen oder ändern. Soll eine Shell-Variable oder ein Alias nur für die aktuelle Shell-Sitzung belegt werden, geben Sie die Änderung einfach in der Shell ein. Wenn Sie Aliases oder Shell-Variablen-Änderungen **dauerhaft in Ihre Shell-Arbeitsumgebung** übernehmen wollen, können Sie diese in der Datei „`.bashrc`“ in Ihrem Home Directory eintragen.

Wichtig: Prüfen Sie die Funktionsfähigkeit eigener Einstellungen nach Änderung der Datei `.bashrc` sofort durch Öffnen einer neuen Shell!

Über das Kommando "`source .bashrc`" können Sie Änderungen in der laufenden Shell sofort einlesen und testen, ohne die `bash` neu starten zu müssen.

3 Linux - Hintergrundwissen

Auf den Rechnern in den FIM-CIP-Pools finden Sie als Betriebssystem Varianten des UNIX-Abkömmlings Linux. Deshalb soll in den folgenden Abschnitten ein Überblick über die Entstehung von UNIX gegeben werden.



3.1.1 Die UNIX/Linux-Story

Die Entwicklung von UNIX (siehe Abb.: Ritchie und Thompson in den Bells Labs bei AT&T) begann im Jahre 1969 mit folgenden Zielen:

- * Statt dem damals üblichen Batchbetrieb mit Lochkartenstapeln sollten mehrere Programmierer gleichzeitig im Team und **interaktiv** mit dem Rechner arbeiten können (**Mehrbenutzerbetrieb, Multiuser-Betrieb**)
- * Dies brachte die Anforderung mit sich, dass es möglich sein musste, die Kommandos der verschiedenen Benutzer gleichzeitig auf dem Rechner abzuarbeiten (**Mehrprogrammbetrieb**)
- * Weiterhin sollte der Vorgang der **Programmentwicklung** durch UNIX **erleichtert** und unterstützt werden.

Die Erfüllung dieser Zielvorgaben machte UNIX damals zu einem attraktiven Betriebssystem, das zunächst vor allem an Forschungseinrichtungen, später auch in Unternehmen rasche Verbreitung fand.

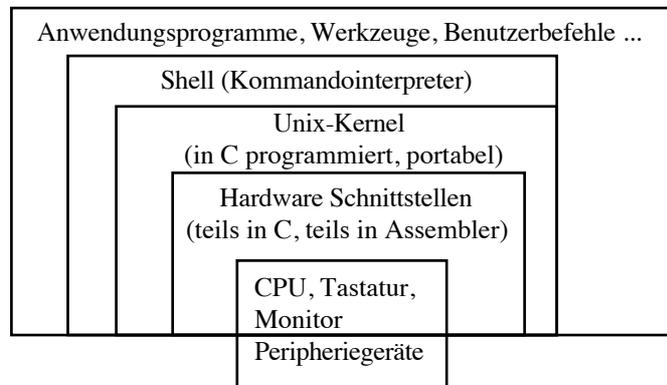
In den 90er Jahren wurde die UNIX-Szene neu belebt durch ein UNIX-ähnliches Betriebssystem namens **Linux** (Symbol: Pinguin). Dieses System wurde von **Linus Torvalds** seit 1991 zusammen mit vielen Internet-Helfern den professionellen UNIX-Systemen nachentwickelt und ist lizenzkostenfrei für viele Rechnerplattformen (PC, Mac) verfügbar. Anbieter wie Ubuntu, Debian, Redhat oder SUSE liefern **Linux** gleich **mit einer Grundausrüstung an Software** in Form so genannter **Distributionen** aus. In den FIM-Studentenrechnerpools finden Sie **Debian-Linux**.

3.1.2 Bestandteile eines UNIX-Arbeitsplatzes

Was der Benutzer von einem UNIX-System sieht, ist zum einen die Hardware, zum anderen eine Menge von Software, die benutzt werden kann:

- * Anwendungssoftware (vorgefertigt oder selbst programmiert)
- * Tools (d. h. zur Programmentwicklung benötigte Software-Tools (Editor, Compiler...))
- * Benutzerbefehle (UNIX-Kommandos)
- * Grafische Oberfläche

Ein UNIX-System umfasst aber noch mehr Komponenten, die dafür sorgen, dass die Software auf der Hardware überhaupt ausgeführt werden kann. Diese **Komponenten umgeben die Hardware in mehreren Schichten** wie eine Zwiebelschale (vgl. Abbildung).



Die Hardware für sich ist leblos, erst durch den **Kernel**, d. h. ein Stück Software wird ihr „Leben eingehaucht“. Der UNIX-Kernel stellt das Betriebssystem im eigentlichen Sinn dar. Der Kontakt zwischen Kernel und der Benutzerebene erfolgt über **Benutzerschnittstellen, d. h. entweder über eine grafische Oberfläche oder den UNIX-Kommandointerpreter**, den man als **Shell** bezeichnet.

4 Anhang: Linux-Kurzinfo



Fakultät für Informatik und Mathematik, Universität Passau © Dipl.-Inform. Klaus Schießl

Thema	Aktionen	Passende Befehle ¹
Ein- und Ausloggen	Einloggen unter grafischem Login	<i>Benutzernamen, Passwort und Session Type eingeben</i>
	Einloggen auf alphanumerischer Console (erreicht man unter Linux mit Ctrl-Alt-F1; mit Ctrl-Alt-F7 zurück zur grafischen Oberfläche)	<i>Nach „login.“ Benutzernamen eingeben u. Return drücken; nach „passwd.“ Passwort eingeben (wird nicht angezeigt)</i>
	Password ändern	<i>ams.fim.uni-passau.de</i>
	Ausloggen	<i>Ctrl-D, logout, exit, bye bzw. „Logout-Button“</i>
Online - Hilfe	Hilfe ohne graphische Benutzeroberfläche	<i>man oder Befehlsoption „ - - help“</i>
	Hilfe unter graphischer Benutzeroberfläche	<i>KDE-Hilfe, Webinfos</i>
Who is who, was ist los	Nachsehen, was im Pool los ist	<i>whoami, w, date</i>
	Was gibt es in der Mensa	<i>mensa</i>
Editieren	Editieren mit/ohne graph. Oberfläche	<i>kate, kwrite, xemacs / emacs</i>
Rechenprozesse	Anschauen, was läuft	<i>ps (nur eigene Prozesse) oder ps -elf</i>
	Abschießen	<i>kill (Prozessnummer angeben !)</i>
	Im Hintergrund starten	<i>& hinter Befehl anhängen bzw. nohup</i>
	Welche Prozesse belasten den Rechner	<i>top</i>
	Priorität verändern	<i>nice, renice</i>
	Prozess in den Vordergrund holen	<i>fg</i>
	Prozess in den Hintergrund schicken	<i>Strg-z (d.h. Prozessausführung zunächst stoppen) und dann bg</i>
	Prozess soll nach Logout weiterlaufen	<i>at, cron</i>
Programme beenden (regulär bzw. im Notfall)	Kein Patentrezept. Nebenstehend einige Möglichkeiten, die oft funktionieren:	<i>e, exit, q, quit, bye, Strg -d, Strg -x Strg -c (für emacs)</i>
	Programm abschießen	<i>Bei Bedarf zunächst Strg-z (d.h. Prozessausführung stoppen); mit ps die Prozessnummer rausfinden und mit kill den Prozess beenden</i>
	Wenn sonst nichts mehr hilft...	<i>Strg -c oder alternativ Strg -\ CIP-Hotline anrufen (3012 bzw. 3013)</i>
Grafische Oberfläche	Starten	<i>Erfolgt nach graphischem Login automatisch</i>
	Beenden	<i>Über KDE-Menü oder Button in Taskleiste</i>
	Neues Shell-Fenster erzeugen	<i>konsole</i>
	Rechner reservieren für kurze Pause	<i>Plasma Bildschirmsperre</i>
	Zeichnen	<i>xfig</i>
	Bilder anschauen u. bearbeiten	<i>okular, xv, gimp</i>
	Filme anschauen	<i>mplayer</i>
	Screendump erzeugen	<i>ksnapshot, gimp</i>
	Auf Rechner B gestartete grafische Anwendung auf dem Bildschirm von Rechner A anzeigen lassen (Beispiel: xemacs)	<i>Auf Rechner A: ssh -X username@B Auf Rechner B Grafikanwendung als Hintergrundprozess starten: xemacs &</i>
Shell (bash)	Letzte Befehle anzeigen	<i>history</i>
	Alte Befehle wiederholen u. modifizieren	<i>Pfeiltasten</i>
	Umgebungsvariablen ausgeben	<i>env</i>
	Alias erzeugen	<i>alias</i>
	Ergänzung von Datei- und Befehlsnamen	<i>TAB-Taste</i>
	Standardausgabe in Datei umlenken	<i>> bzw. >></i>
	Standardeingabe von Datei statt Tastatur	<i><</i>

Umgang mit Dateien und Directories	Wo bin ich im Dateibaum	pwd
	Inhalt eines Directories listen	ls <i>(ausführliches Listing: ls -alg)</i>
	Datei kopieren/Directory kopieren	cp / cp -r
	Datei bzw. Directory umbenennen	mv
	Datei löschen	rm
	Datei anschauen	more, less, head, tail
	In anderes Directory wechseln	cd
	Directory anlegen	mkdir
	Directory löschen	rmdir bzw. rm -r (falls Directory nicht leer)
	Zugriffsrechte ändern	chmod
	Dateien auf bestimmte Inhalte abchecken	grep
	Dateien mit bestimmten Eigenschaften suchen (z.B. mit bestimmten Namen)	find
	Dateiinhalte vergleichen	diff
Abkürzungen im Dateisystem	Vaterdirectory (d.h. eine Stufe höher)	..
	Aktuelle eingestelltes Working Directory	.
	Home Directory	~
Platzhalter, Wildcards	Beliebiges einzelnes Zeichen	?
	Beliebig lange Zeichenkette	*
Festplattenauslastung	Wieviel Festplattenplatz belege ich gesamt	df
	Speicherplatzbedarf eines Directories	du -k .
	Datei komprimieren	gzip, bzip2
	Komprimierte Datei wieder auspacken	gunzip, bunzip2
Bildschirmausgabe	Bildschirmausgabe anhalten	Strg-s
	Bildschirmausgabe fortsetzen	Strg -q
Programmieren	C++	g++
	Haskell	ghc
	Java	javac
	Programmierungsumgebung Eclipse	eclipse
Revisionskontrolle	CVS, Subversion	cvs, svn
UML	Rational Software Architect	rsa
Mathematikprogramme	Maple, Matlab, Statistiksoftware	xmaple, matlab, R
Texte eingeben	Texte editieren in der shell	vim, vi, emacs
	Texteditoren für grafische Umgebung	kate, kwrite, xemacs,
	Mathematischer Textsatz	tex, latex, kile (<i>LATEX-Frontend unter KDE</i>)
Bildbearbeitung, Zeichnen	Gimp, XV, Xfig	gimp, xv, xfig
Desktop-Publishing	Scribus	scribus
Office-Programme	LibreOffice	libreoffice
PDF-Dateien	Viewer	okular, evince
Postscript	Postscript-Viewer	okular, gv, evince
	Postscript-Dateien nachbearbeiten	psf, psnup, pssselect, psfilt, psbook, pstops
Documentviewer allgemein (u. a. für PDF und Postscript)	Okular (unter KDE, auch für Bilder geeignet)	okular
	Evince (unter Gnome)	evince
Netzwerk/Internet	Einloggen auf anderem Rechner	ssh
	E-Mail	thunderbird
	WWW	firefox, konqueror, opera
	Usenet, Newsgroups	thunderbird
	Chat	pidgin
	File Transfer	scp, sftp, ftp, ncftp
CD-Brennen	Siehe WWW-Infos	k3b
Scannen	Siehe WWW-Infos	gimp

CIP-Infos per WWW: <http://www.fim.uni-passau.de/it-dienste/>

Benutzerberatung: Gebäude IM, Zimmer 014 (Hr. Rautmann) bzw. Zimmer 017 (Hr. Schießl)

Wohin bei Problemen: Telefon: 3012 oder 3013 bzw. E-Mail an support@fim.uni-passau.de

Aus Gründen der Übersichtlichkeit wurde darauf verzichtet, bei den einzelnen Kommandos Optionen sowie benötigte Parameter aufzuführen. Genaueres finden Sie an entsprechender Stelle im Script zur Rechnereinführung bzw. in den Online-Manpages zu den einzelnen Befehlen.