

Prof. Dr. Dieter Kranzlmüller

Dr. Nils gentschen Felde

Dr. Karl Furlinger

Stephan Reiter

Christian Straube

Einführung in die Bedienung von Linux

Workshop im Rahmen des Informatik-Probestudiums 2012

Ziele des Informatik-Studiums:

Fundierter Überblick über die Kernkonzepte und (technischen) Grundlagen der Informatik aus den Bereichen

Rechnerarchitektur

Betriebssysteme

System- und Anwendungsprogrammierung, Software Engineering

Rechnernetze, verteilte und mobile Systeme

(Web-)Informationssysteme, Datenbanken

Theoretische Informatik, Algorithmik, Komplexität

Spezielle Vertiefungen, z.B. IT-Sicherheit, IT-Management

Anwendungsbereiche für Lösungen aus der Informatik kennen und selbständig

Lösungen anforderungsgemäß entwickeln und umsetzen können

Ein Informatiker ist kein Programmierer! Programmierung/Softwaretechnik ist nur eine der Schlüsselkompetenzen eines Informatikers. Programmiersprachen zu beherrschen (und die Fähigkeit zu besitzen, sich neue Programmiersprachen schnell anzueignen) ist ein wichtiges Werkzeug jedes Informatikers.

Ziele dieses Workshops:

Einblicke in die Informatik als Wissenschaft und das Informatik-Studium vermitteln

Eine Entscheidungsgrundlage schaffen für die Beantwortung folgender Fragen:

Entspricht meine Vorstellung vom Informatik-Studium der Realität?

Interessieren mich die Inhalte des Informatik-Studiums?

Genügen meine Vorkenntnisse, um das Studium mit Erfolg durchzuziehen?

Benötige ich noch weiteres Material/Informationen über das Informatik-Studium,
bevor ich mich dafür/dagegen entscheiden kann?

Tipps für das Eignungsfeststellungsverfahren (EFV) im Informatik-Studiengang:
In den Bewerbungsunterlagen angeben, dass das Probestudium/dieser Workshop
besucht wurde!

- **Tag 1 (Fr):**
 - Grundlagen der technischen Informatik mit Übungen
 - Einführung in die Bedienung von Linux
 - Übungen
- **Tag 2 (Mo):**
 - Übungen
 - HPC und paralleles Rechnen
- **Tag 3 (Di): Grundlagen der Computergrafik**
 - POV-Ray
 - Ausblick und Demo
- **Tag 4 (Mi): Grundlagen der IT-Sicherheit**
 - Grundlagen der IT-Sicherheit mit Übungen
 - Demonstration "Passwörter unter Windows knacken"

Allgemeines zum Betriebssystem Unix/Linux

Was ist ein Betriebssystem (OS)?

Ein Bündel von Programmen, die im Prinzip zwei Aufgaben erledigen:

- saubere Abstraktionen der Betriebsmittel
- koordiniert den Zugriff auf gemeinsame Betriebsmittel

(A. Tanenbaum)



Unix ist ein Multi-User-/Multi-Tasking-Betriebssystem

- Viele „Derivate“: **Linux (Debian, Ubuntu, ...)**, Solaris, AIX, HP-UX, xBSD, etc.

(Minimale) Bestandteile eines Betriebssystems:

- Systemkern (Kernel)
- Kommandozeileninterpret (Shell)
- (Weitere) Dienstprogramme (z.B. Compiler, Editor)

Anforderungen an ein Betriebssystem:

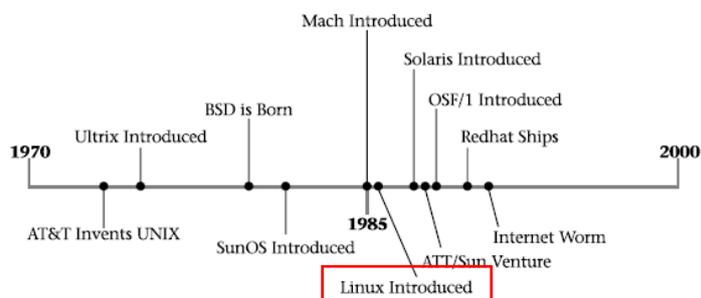
- Aufgaben des Betriebssystem-Kerns:
 - Prozessverwaltung: Prozesserzeugung, Scheduling
 - Speicherverwaltung: Partitionierung, Virtueller Speicher/Paging
 - Dateiverwaltung: Dateisysteme, Dateitabellen
 - Benutzerverwaltung

Sonstige Anforderungen:

- Fehlerfreie Ausführung: Synchronisation, Deadlock-Behebung
- Sicherheit: z.B. Authentifizierung & Autorisierung

Timeline:

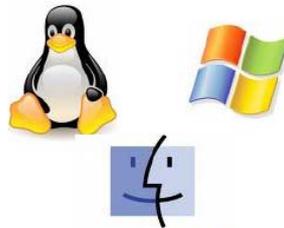
- Early 1970s: UNIX is born to be wild!
- Wann und wo kommt Linux ins Spiel?



Quelle: UNIX System Administration: A Beginner's Guide, Steve Maxwell, McGraw-Hill/Osborne, 2002

Linux:

- **Unix-ähnliches** Betriebssystem
- Erste Version: ca. 1991
- Open Source
- Linux-Distributionen(Auszug):
 - Debian
 - Fedora
 - Red Hat
 - SuSe



Microsoft Windows:

- Erste Version: 1985
- Closed Source
- Versionen (Auszug):
 - Vorläufer: MS DOS
 - Windows 10. bis 3.11
 - Windows 95, 98, ME
 - Windows NT, 2000
 - Windows XP, Vista, 7
 - Server 2003, 2008

Apple Mac OS:

- Erste Version: 1984
- Closed Source
- Aktuelle Version: Max OS X



Nutzerschnittstellen:

- Früher: Kommandozeileninterpreter (Eingabeaufforderung, Shell)
- Heute: (Auch) Graphische Benutzeroberfläche

Eingabemethoden:

- Tastatur
- Maus
- Tablet
- Touchpad
- Touchscreen
- Mimik-/Gestikerkennung
- ...

Unix: Ursprünglich rein kommandozeilenorientiert

- Nach dem Start wartet eine Anmelde-Aufforderung
 - Textkonsole

Linux:

- Graphisch X-Window-Manager (Beispiel: KDE oder GNOME)
- X-Window: Eine schöne bunte Welt mit Fenstern, Knöpfen, etc.

UNIX®

Celebrating 40 years uptime



Microsoft Disc Operation System (DOS) 5.0:

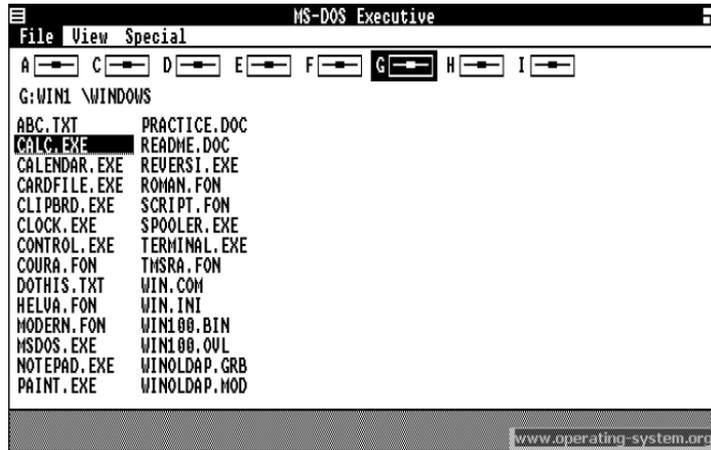
```

GRAFTABL COM      11237 03.02.90  13:00
GRAPHICS COM      19758 03.02.90  13:00
GRAPHICS PRO      21232 03.02.90  13:00
EXE2BIN  EXE       8584 03.02.90  13:00
EXPAND  EXE      14835 03.02.90  13:00
JOIN    EXE      17934 03.02.90  13:00
LCD     CPI      10771 03.02.90  13:00
LOADFIX COM       1273 03.02.90  13:00
INFO    TXT      15690 03.02.90  13:00
MANINFO TXT      10313 03.02.90  13:00
PRINTER SYS      18852 03.02.90  13:00
REPLACE EXE      20194 03.02.90  13:00
SUBST   EXE      18574 03.02.90  13:00
TREE    COM       6974 03.02.90  13:00
COMMAND COM      50031 03.02.90  13:00
DOSHELL INI      17830 28.05.06  21:18
      83 Datei(en)  2147161 Byte
      26421248 Byte frei

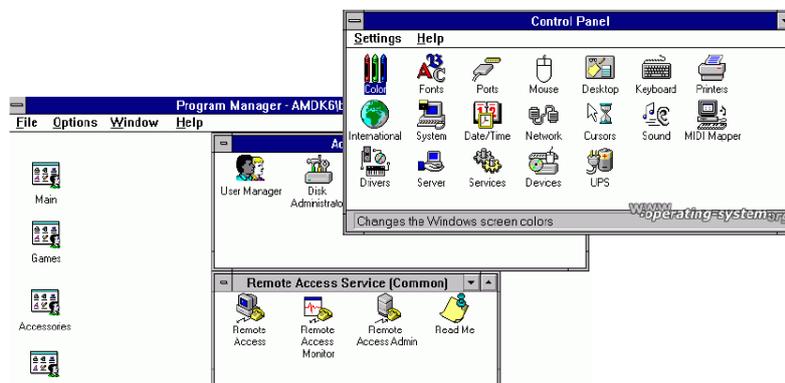
C:\DOS>ver
MS-DOS Version 5.00

C:\DOS>
  
```

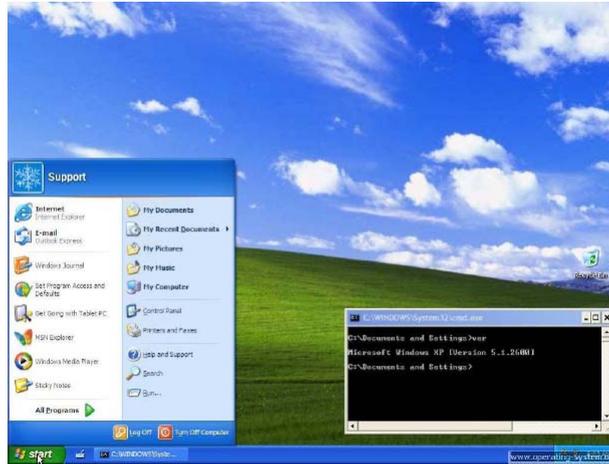
Microsoft Windows 1.0:



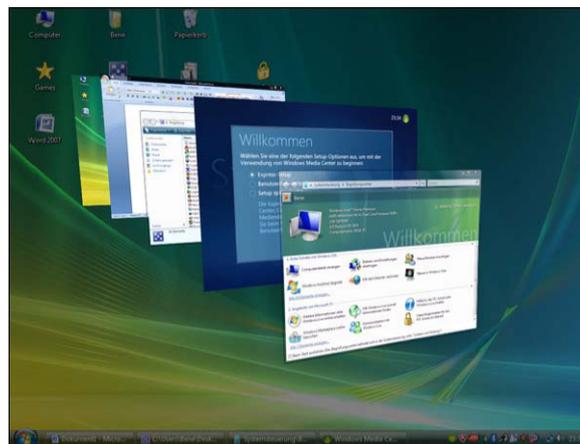
Microsoft Windows 3.1:



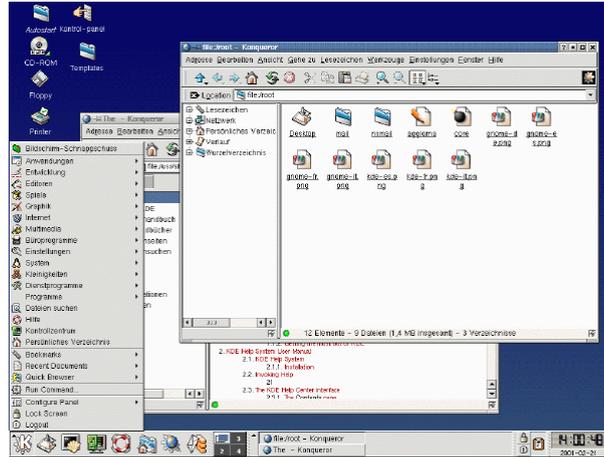
Microsoft Windows XP:



Microsoft Windows Vista:



SuSe Linux mit KDE:



Apple Mac OS X:



Unix/Linux vs. „die Anderen“:

- Ausgereifte Umgebung zur Programmentwicklung – Modular
- Sehr gute Dokumentation
- Zahlreiche leistungsfähige Kommandos – Kurz und flexibel
- Leistungsfähige graphische Benutzeroberfläche
- Weit verbreitet im wissenschaftlich-technischen Bereich
- **It's Free and Open Source!**



Try it out!

- Lernen und probieren Sie einige Befehle aus!
- Die Befehle sind in den **roten Zeilen** zu sehen!
- **Hands-On:** Es gibt kleine Aufgaben zum Ausprobieren. Machen Sie mit!



Linux: Multi-User-Betriebssystem

- An- und Abmelden: Benutzerkennung + Passwort
- Persönliche Umgebung (Home-Verzeichnis)

Username:

Password:



Im wesentlichen zwei Typen von Benutzern:

- Normale Benutzer: Eingeschränkte Rechte
- Systemadministrator: **root** mit allen Privilegien (!)
(Unter Windows: „Administrator“)

Hands-On:

- Anmeldung auf dem Rechner: probstudium.lab.ifi.lmu.de
- Anmelde Daten: siehe Zettel!
- Passwort: siehe Zettel!
- Vorsicht! Passwörter sollen regelmäßig geändert werden...
...und nicht vergessen werden!
passwd ändert Ihr Passwort



Die Konsole/Terminal

- In einer Textkonsole wird normalerweise eine Shell gestartet
- Die Shell interpretiert die Befehle
- Befehle können mit **CTRL+C** abgebrochen werden
- Es gibt eine **history** Funktion
- **printenv** zeigt alle definierten Umgebungsvariablen an

Default-Variablen

- Beim Start der Shell gesetzt
HOME: Persönliches Verzeichnis des Benutzers
PATH: Suchpfad für Programme

```
MYVAR=TEST
export
echo $MYVAR
```

man [section] command

man -k expression

- Man-Pages: Hilfeseiten zu Programmen und Bibliotheksaufrufen
- Fast jeder Konsolenbefehl hat ein Manual
- Man-Pages: Band (**[section]**) meistens von 1 bis 8 untergliedert
- Manual-Pages: Mehrere Abschnitte **NAME, SYNOPSIS, DESCRIPTION, OPTIONS, COMMANDS, RETURN VALUES, ERRORS, etc.**

q beendet aktuelle man-page
/ sucht innerhalb man-page

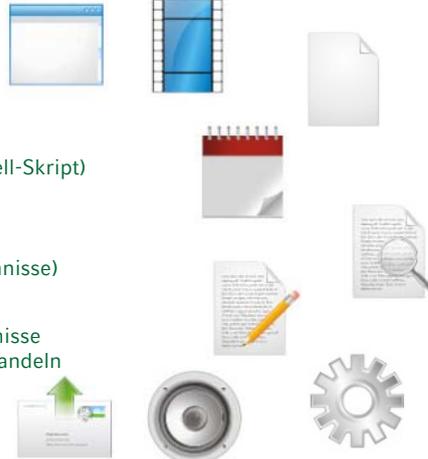
```
Terminal (heger@lmu.de) - User's Konsole
whoami(1)                                User Commands                                whoami(1)
NAME
  whoami - print effective userid
SYNOPSIS
  whoami [OPTION]...
DESCRIPTION
  Print the user name associated with the current effective user ID. Same as id -wn.
  --help display this help and exit
  --version
  output version information and exit
AUTHOR
  written by Richard Mlynarik.
REPORTING BUGS
  Report bugs to -bug-coreutils@gnu.org-.
COPYRIGHT
  Copyright © 2008 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3
  or later -http://gnu.org/licenses/gpl.html.
  This is free software; you are free to change and redistribute it. There is NO
  WARRANTY, to the extent permitted by law.
SEE ALSO
  The full documentation for whoami is maintained as a Texinfo manual. If the
  info and whoami programs are properly installed at your site, the command
  info whoami
  should give you access to the complete manual.
GNU coreutils 6.9.10.4-f886.dirty January 2008                                whoami(1)
Manual page whoami(1) line 3/41 [END]
```

Hands-On:

1. Lassen Sie sich Informationen zu dem Kommandos **ls** und **who** anzeigen

In Unix ist alles eine Datei!

- Dateien
- Textdateien, Bilddateien
- Verzeichnisse
- Ausführbare Dateien (Binärdatei oder Shell-Skript)
- Gerätedateien
- Pipes*
- Links (Verweise auf Dateien und Verzeichnisse)
- **Prinzip:** Anwendungen können Verzeichnisse und andere Dateien in gleicher Form behandeln (open, close, read, write, etc.)



* Pipe: Man kann Kommandos mit dem Pipezeichen | verbinden. Die Ausgabe des vorherigen Befehls wird als Eingabe des nächsten Befehls interpretiert. Pipes oder Pipelines dienen zur Inter-Prozesskommunikation.

Verzeichnisstruktur:

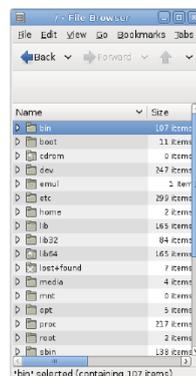
Windows:

Laufwerksbuchstaben



Linux:

Verzeichnisbaum



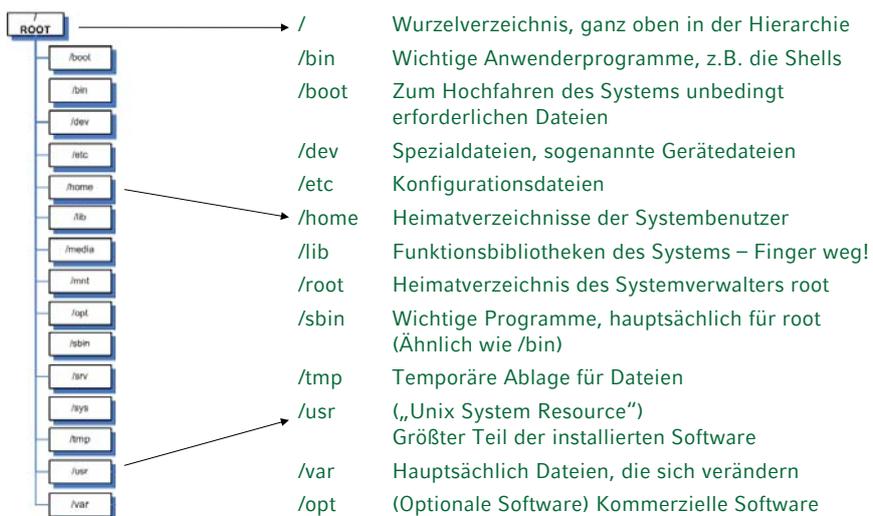
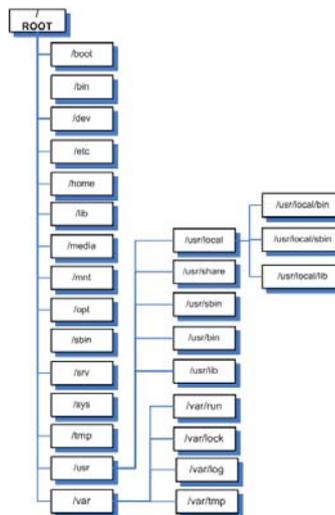
Root-Verzeichnis:

- Das oberste Verzeichnis
- Alle anderen Verzeichnisse liegen unterhalb dessen

Verzeichnisnavigation:

- FHS (File Hierarchy Standard): Angabe über was man wo im Dateisystem findet
- Verzeichnis: Wie Ordner (unter Windows auch so) genutzt
- Verzeichnisse: Können mehrere Dateien oder Verzeichnisse enthalten
- Datei: Hierarchisch sortiert im Verzeichnisbaum (root-Verzeichnis als Wurzel)
- Datei: Innerhalb des Verzeichnisbaums (absolute oder relative Pfadnamen)

Quelle: <http://www.debianadmin.com/linux-directory-structure-overview.html>
Stand: März, 2008



`pwd, ls [option]`

`cd`

`pwd` zeigt das aktuelle Verzeichnis an

`ls` zeigt Kenndaten von Dateien an, und listet den Inhalt des Verzeichnisses auf

`cd` wechselt in das angegebene Verzeichnis:

- `cd path` in den angegebenen Pfad
- `cd ..` in das darunter liegende Verzeichnis
- `cd /` zur Wurzel des Verzeichnisbaums
- `cd` in das eigene Home-Verzeichnis
- `cd -` in das letzte Verzeichnis

Hands-On:

1. Lassen Sie sich das aktuelle Verzeichnis anzeigen
2. Zeigen Sie den Inhalt des Verzeichnisses an
3. Wechseln Sie in das root Verzeichnis und dann zurück in ihr Home-Verzeichnis

Probstudium Informatik 2012

29

Suchmuster:

- Mehrere Dateinamen lassen sich gleichzeitig durch Suchmuster ansprechen

Sonderzeichen und ihre Bedeutung

`*` beliebige, auch leere Zeichenkette

`?` ein beliebiges einzelnes Zeichen

`[...]` ein Bereich von Zeichen, z.B. `[a-d]`

`[!...]` negierter Bereich von Zeichen, z.B. `[!a-d]`

`find` sucht im System nach Dateien

`grep` sucht in Dateien nach Zeichenkette(n)

Hands-On:

1. Suchen Sie nach/in beliebigen Dateien mit `find` und `grep`
`find [path] [expression]`
`grep [options] pattern [file list]`

Probstudium Informatik 2012

30

`mkdir [folder name]`

`rmdir [folder name]`

`mkdir` legt ein leeres Verzeichnis an

`rmdir` löscht ein leeres Verzeichnis

Hands-On:

1. Legen Sie die Verzeichnisse `probeStudium1`, `probeStudium2` und `probetest` an.
2. Versuchen Sie das Verzeichnis `probetest` zu löschen.
3. Überprüfen Sie, ob das Verzeichnis existiert bzw. gelöscht worden ist.
4. Kann man diese Verzeichnisse auf einmal anlegen?

`touch [filename]`

`rm [filename]`

`touch` legt eine neue leere Datei an

`rm` löscht Dateien und Verzeichnisse und ggf. deren Inhalt

- `-i` Es wird erst nach vorheriger Sicherheitsabfrage gelöscht
- `-r` Verzeichnisse werden rekursiv mit allen Unterverzeichnissen gelöscht (!)
- `-f` Unterdrückung aller Sicherheitsabfragen

Hands-On:

1. Legen sie eine leere Datei `testfile` im Verzeichnis `probeStudium1` an
2. Versuchen Sie `probeStudium1` mit `rmdir` zu entfernen
3. Versuchen Sie es jetzt mit dem Befehl `rm`

`cp [option] [source] [destination]`

`mv [source] [destination]`

`cp` kopiert `[source]` Datei/Verzeichnis nach `[destination]`
Die Originaldatei bzw. das Originalverzeichnis bleibt erhalten

- `cp -r` kopiert Verzeichnis inklusive Inhalt

`mv` zum Umbenennen oder Verschieben von Dateien oder Verzeichnissen
Originaldatei bzw. das Originalverzeichnis wird gelöscht

Hands-On:

1. Benennen Sie das Verzeichnis `probeStudium2` in `probeStudium` um
2. Legen Sie eine neue Datei `testfile` in ihrem Home-Verzeichnis an
3. Kopieren Sie diese Datei in das Verzeichnis `probeStudium`

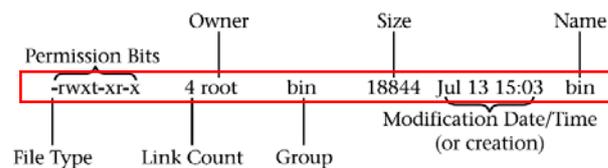
Zugriffsrechte:

- Drei verschiedene Zugriffsrechte:
 - `d` Directory
 - `r` Read – 4
 - `w` Write – 2
 - `x` Execute – 1
- Zu jeder Datei wird jeweils vergeben:
 - `u` User
 - `g` Group
 - `o` Others
 - `a` All

```

-rwxl-xr-x 4 root bin 18844 Jul 13 15:03 bin

```



Quelle: UNIX System Administration: A Beginner's Guide, Steve Maxwell, McGraw-Hill/Osborne, 2002

```
chmod [ugoa][+|=][rwx] [filename]
chown user:group [path]
```

chmod ändert der Zugriffsrechte von Dateien bzw. Verzeichnissen
chown ändert Besitzer bzw. Gruppe

Hands-On:

1. Was bedeuten die unteren Zeilen?

```
chmod 740 [path]
chmod u+rwx,g+r [path]
```
2. Legen Sie eine leere Datei an und ändern Sie die Zugriffsrechte so, dass alle im System Lese-Rechte besitzen und nur Sie Schreib-Rechte.
3. Nehmen Sie dieser Datei in ihrem Verzeichnis alle Rechte.
4. Wer kann diesen Zustand wieder rückgängig machen?

```
who [option]
whoami [option]
uname [option]
```

who zeigt alle Benutzer, die am System angemeldet sind: Nutzername, Terminal, Zeitpunkt der Anmeldung

whoami berichtet wer Sie sind im System

uname gibt Auskunft über das laufende System

Hands-On:

1. Ermitteln Sie alle Benutzer, die gerade auf dem Rechner arbeiten und finden Sie heraus von wo sie eingeloggt sind.
2. Wie heißt ihr Benutzer?
3. Was können Sie an Parametern für das laufende System sehen?

Editoren sind **sehr wichtig!**

Programm- und Systemdateien sind zumeist Textdateien

Terminal:

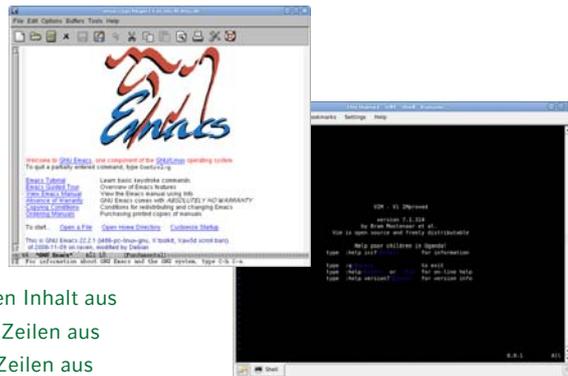
- vim/vi
- emacs

Graphisch:

- gedit
- kate

In Konsole Inhalt ausgeben:

- **cat [filename]** gibt den gesamten Inhalt aus
- **head [filename]** gibt die ersten Zeilen aus
- **tail [file name]** gibt die letzten Zeilen aus
- **less [filename]** gibt seitenweise Dateinhalt aus



Ausgabe

- **[command] > [filename]** leitet Ausgabe in eine neue Datei
- **[command] >> [filename]** hängt Ausgabe an eine bestehende Datei an

Eingabe

- **[command] < [filename]** der Dateinhalt ist Eingabe für ein Programm
- Warum Ein-/Ausgabe von Befehlen umleiten?
Manchmal ist es nützlich sich ein Ergebnis in eine Datei zu schreiben, anstatt auf den Bildschirm

Hands-On:

1. Schreiben Sie die Ausgabe vom **ls** aus ihrem Home-Verzeichnis in eine Datei
2. Hängen Sie die Ausgabe des Befehls **ls -l** an

Prozesse: Laufende Programminstanz

- Jedes laufende Programm hat mehrere Prozesse und jeder Prozess hat einen Status.
- Programmcode, Speicher (Ressourcen), etc.
- Jeder Prozess hat eine eindeutige Nummer (PID)
 - `ps [option]` gibt Übersicht über laufende Prozesse
 - `kill [pid]` terminiert Prozesse
- Die wichtigsten Prozess-Status:
 - R** running (läuft)
 - S** sleeping (schläft) und wartet auf Arbeit
 - Z** zombie (existiert kurz und verschwindet)
Kann entstehen, wenn der **Eltern-Prozess*** die Terminierung des Kind-Prozesses nicht (ordentlich) behandelt
 - W** belegt keine Seiten im Speicher

* Eltern-Prozesse können mehrere Kind-Prozesse haben, ein Kind hat immer genau einen Eltern-Prozess.

Hands-On: Einfaches bash-Skript:

- Machen Sie einen Editor ihrer Auswahl auf!
- Versuchen Sie den folgenden Code zu interpretieren (Tipp: man-pages)

```
#!/bin/bash
#Programm hallo.sh
echo "Wie heißt Du?"
read Name
echo "Hallo $Name"
```
- Damit das Skript ausführbar ist, müssen die Rechte angepasst werden:

```
chmod u+x hallo.sh
```
- Führen Sie das Skript aus!

```
./hallo.sh
```

Tipp: Es gibt einige Startskripte unter `/etc/init.d` (damit kann man auch viel lernen!)

ping

- Geben Sie den Befehl ein:
`ping 10.153.245.100`
- Ihr Rechner versucht den Rechner mit der IP-Adresse `10.153.245.100` im lokalen Netz zu erreichen
- **Was ist eine IP-Adresse?**
Jeder Rechner in einem Netz braucht eine eigene Adresse (hier: IP-Adresse)
- **Was macht der ping-Befehl?**
Verschickt eine Anfrage, um festzustellen, ob der Rechner erreichbar ist
Wartet auf Antwort von angefragtem Rechner
- Man kann auch direkt den Hostnamen eingeben: `ping www.google.com`

Hands-On:

1. Versuchen Sie andere Rechner von anderen Teilnehmern im Labor zu erreichen
2. Was können Sie aus den Ausgabezeilen des Befehls herauslesen?

tracert

- Untersucht und speichert den Weg eines Datenpakets bis zur Zielstation
- `tracert` funktioniert ähnlich wie `ping`
- Man bekommt mehr Informationen über die Netzverbindung
- **Warum `tracert`?**
Überprüft, ob Datenpakete auf dem Weg zum Ziel die richtige Route verwenden.
Testet die Laufzeit zwischen den einzelnen Stationen. Man kann z.B. Versuchen eine Station zu ermitteln, die ausgefallen ist
- Geben Sie folgenden Befehl ein:
`tracert 10.153.245.100`

Hands-On:

1. Versuchen Sie andere Rechner von anderen Teilnehmern im Labor zu erreichen
2. Was können Sie aus den Ausgabezeilen des Befehls herauslesen?

SSH (Secure Shell): entferntes (remote) Arbeiten

- Erlaubt das Anmelden/Arbeiten auf entfernten Rechnern
- Sie brauchen dort eine Benutzerkennung!
- Linux: ssh (Kommandozeile)
- Windows: z.B. PuTTY

- Rechnername oder die IP-Adresse müssen angegeben werden
 - ssh [option] [username@hostname](#)
 - [option]: Man-Pages!
 - [username]: Ihr Nutzernamen auf dem Fremdrechner
 - [hostname]: Name oder IP-Adresse des Fremdrechners

SCP (Secure Copy):

- Sicheres Kopieren von Dateien zwischen zwei Rechnern
- Linux: scp (Kommandozeile)
- Windows: WinSCP (graphisch)

- `scp [file1] username@hostname:[file2]`
Kopiert die lokale Datei [file1] auf einen Fremdrechner mit dem neuen Namen [file2]

- `scp username@hostname:[file1] [file2]`
Kopiert [file1] auf einen Fremdrechner zum lokalen Rechner als [file2]

- `scp -r` erlaubt das rekursive Kopieren ganzer Verzeichnisse

Aufgaben!

1. Melden Sie sich an der virtuellen Maschine an und erstellen Sie die Verzeichnisse `daten`, `src` und `source`
2. Mit welchem Parameter listet ls die versteckten Dateien im Verzeichnis? Zeigen Sie alle versteckten Dateien in ihrem Home-Verzeichnis an!
3. Kopieren Sie 5 beliebige Dateien aus verschiedenen Verzeichnissen ins Verzeichnis `src` und `source`
4. Das Verzeichnis `source` soll nun wieder komplett gelöscht werden
5. Schreiben Sie einen kurzen (beliebigen) Text mit dem Editor `vi` in die Datei `text` im Verzeichnis `Daten`
6. Kopieren Sie diesen Text auf den Nachbarrechner. Bitten den Nachbarn dafür einen neuen Nutzer anzulegen, dessen Passwort Sie bestimmen!
Tipp: Benutzen `scp` (`secure copy`) um Dateien auf Fremdrechner zu kopieren
7. Loggen Sie sich auf dem Nachbarrechners mit dem gerade für Sie neu eingerichtet Nutzernamen und Passwort per `ssh` (`secure shell`)
8. Da die vorhin kopiert Textdatei vertraulich ist, ändern Sie die Zugriffsrechte der Datei dahingehend ab, dass nur noch Sie sie lesen können

(Zusatz)aufgaben!

9. Schreiben Sie ein Shell-Skript, das die Zahlen 0 bis 9 ausgibt.
Tipp: Verwenden Sie dazu eine `while`-Schleife: `while [condition] ; do`
`<commands>`
`done`
10. Erweitern Sie das Skript so, dass der Beginn und das Ende der auszugebenden Zahlen als Kommandozeilenparameter übergeben werden
11. Fügen Sie noch eine Plausibilitätsüberprüfung in das Programm ein, welche überprüft, dass der übergebende Startwert kleiner oder gleich dem übergebenden Endwert ist. Im Fehlerfall soll das Skript terminieren und auf `/dev/stderr` eine Fehlerbeschreibung ausgeben
12. Nun soll bei jedem Schleifendurchlauf nach der Konsoleausgabe eine Sekunde gewartet werden
Tipp: `sleep`
13. Starten Sie ihr Programm mit dem Startwert 0, dem Endwert 1000. Schicken Sie es in den Hintergrund.
Identifizieren Sie nun die Prozessnummer (PID), die das Programm hat und terminieren den Prozess frühzeitig (`kill`)