Unix I – Einführung in das Unix Betriebssystem

Stephan Thesing

II. Vorlesung, 25.10.1999

- **Accounts**
- **Filesystem**

Unter Unix kann es mehrere Benutzer (gleichzeitig) geben.

Benutzer werden anhand ihrer User-ID unterschieden.

Zur besseren Lesbarkeit: User-Namen (Abbildung User-IDs $\leftrightarrow$ Namen).

Jeder Benutzer muß sich beim System anmelden (Einlogen).

Zur Authentifizierung muß der Benutzer dabei sein Passwort angeben.

Das Passwort ist der einzige Nachweis, daß der Mensch vor dem Rechner wirklich der Benutzer ist, der er vorgibt zu sein.

$\Rightarrow$ Passwörter sind **wichtig**.

Unter Unix gibt es auch das Konzept der Gruppe.

Eine Gruppe ist eine Sammlung von Benutzern.

Jeder Benutzer ist in mindestens einer Gruppe (`Student` auf den CIP-Pools).

Ein Benutzer kann aber auch in mehreren Gruppen sein.

Wichtig für die Zugriffsrechte des Benutzers:

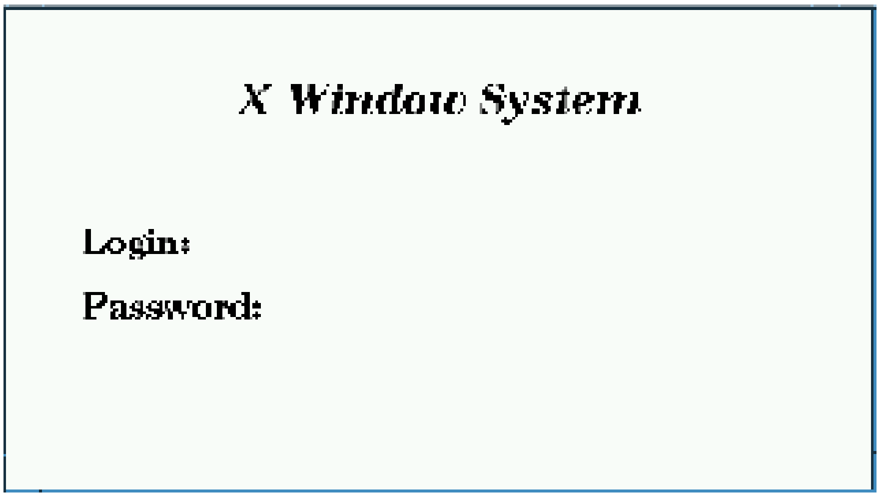
Unix bietet die Unterscheidung von Rechten nach Gruppenzugehörigkeit.

Etwa kann man ein bestimmtes File nur für Mitglieder der Gruppe `Helden` lesbar machen.

Passwörter

- Sollen nicht anderen Leuten bekannt gegeben werden.
- Sollen nicht aus leicht merkbaren Worten bestehen (Vorname der Freundin ist ganz schlecht:-)
- Sollen nicht eingegeben werden, wenn andere Leute zusehen.
- Sollen aus Buchstaben (Groß- & Kleinbuchstaben) und Ziffern, Sonderzeichen bestehen. (Vorsicht bei CTRL- Sequenzen)
- Sollen immer um die acht Zeichen lang sein.
- Sollen immer verschloßen verwahrt werden.

Einlogen



Graphische Benutzeroberfläche XWindow schon beim Einlogen aktiv.

Wird automatisch nach Authentifizierung für den User gestartet.

Was passiert beim Einlogen?

1. Via xdm (XWindow)
   - Für die ELCs: Einloggen auf den FreeBSD PCs
   - Für SS4: Einloggen auf dem Rechner selbst
   - Username und Passwort werden verifiziert (richtiges PW für diesen User?)
   - Es wird das XWindow Startup File des Benutzers ausgeführt (`.xsession`) im Home-Directory.
   - Dieses gibt i.a. ein `xterm` Fenster (Terminals) mit der Login-Shell des Benutzers.

2. An der Console oder via Netzwerk (`telnet`)

   - Username + PW werden überprüft.
   - Die Login-Shell des Benutzers wird gestartet.

In allen Fällen wird die User-ID und das aktuelle Directory auf die Werte gesetzt, die in der Datenbank für den Benutzernamen eingetragen sind.

In der Shell (xterm oder Console) kann man nun Programme aufrufen, Variablen setzen, z.B.

```
>ls

HiWis             vorlesung_1.tex
RCS               vorlesung_2.aux
UGruppen          vorlesung_2.dvi
Vorstellung       vorlesung_2.log
ueb.tex           vorlesung_2.ps
```


Am Ende einer Sitzung am Rechner muß man sich abmelden (auslogen).

Dies geschieht unter der graphischen XWindow Oberfläche durch Anwahl eines Menupunktes im Hintergrundfenster. (In etwa der Form `Quit TWM`)

An das Menu kommt man durch Drücken der linken Maustaste, wenn der Cursor über dem Hintergrund steht.

Via Netzwerk/Console: durch Eingeben des Kommandos `logout`.

Niemals das Auslogen vergessen!

Anderer Benutzer könnten sonst ihren Account für unangenehme Dinge benutzen.

Die verfügbaren Shells werden in späteren Vorlesungen detailliert besprochen.

Fürs erste nur einige wenige Anmerkungen:

- In den `xterm` Fenstern wird immer eine Kopie der Login-Shell gestartet.
- Ebenso via Console, Netzwerk
- In der Shell gibt man Kommandos ein, die dann ausgeführt werden.
- Es gibt interne Shell-Kommandos und externe Kommandos.
- Externe Programme sind Dateien, die gesucht und ausgeführt werden.
- Ausgaben der Kommandos erscheinen im Fenster, Tatstatureingaben gehen in das (aktive) Fenster und an das Programm.

Kommandos können auch Argumente haben, die dann durch Leerzeichen getrennt dem Kommando folgen (Z.B. `ls /etc`)

Einige wichtige Kommandos:

- `man` gibt die Manualseiten zu Programmen u.ä. aus.
- `apropos` sucht nach Manualseiten, die ein Schlagwort enthalten
- `info` bietet eine Hypertextoberfläche für Programmdokumentationen (`texinfo`-Files)

Wenn man z.B. die Dokumentation zum `ls` Kommando haben möchte

```
>man ls

LS(1)          NetBSD Reference Manual          LS(1)

NAME
     ls - list directory contents

SYNOPSIS
     ls [-ACFLRSTWacdfgiklmnoqrstux1] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than
     directory, ls displays its name as well as any other
```


Für alle Seiten, die das Schlagwort `group` enthalten:

```
>apropos group

chgrp (1) - change group
chown (8) - change file owner and group
chown, lchown, fchown (2) - change owner and group of a file
getgid, getegid (2) - get group process identification
getgrent, getgrnam, getgrgid, setgroupent, setgrent, endgrent (3) -
            group database operations
getgrouplist (3) - calculate group access list
getgroups (2) - get group access list
```

```
>info
File: dir,      Node: Top,      This is the top of the INFO tree

  This (the Directory node) gives a menu of major topics.
  Typing "q" exits, "?" lists all Info commands, "d" returns here,
  "h" gives a primer for first-timers,
  "mTexinfo<Return>" visits the Texinfo manual, etc.

* Menu:

Texinfo documentation system
* Info: (info).                 Documentation browsing system.
* Texinfo: (texinfo).           The GNU documentation format.
* info program: (info-stnd).    Standalone Info-reading program.
* install-info: (texinfo)Invoking install-info. Updating
                              info/dir entries.
```


Manualseiten sind in acht Sektionen aufgeteilt, in Sektion 1 etwa sind die Kommandos.

Siehe hierzu (und zur Benutzung von `man`) allgemein: `man man`

Benutzen sie `man` und `info`!

Lesen sie die Beschreibungen, bevor sie Kollegen, Betreuer oder die RGB nerven.

Summary & Kommandos

- Jeder Benutzer hat eine User-Id, lesbare Form: User-Name
- Jeder Benutzer ist Mitglied einer oder mehrerer Gruppen
- Man muß sich beim System als User anmelden.
- Verifikation der Anmeldung durch ein geheimes Passwort.
- Nach Anmeldung wird XWindow oder Console-Shell gestartet.
- In der Shell kann man Kommandos eingeben.
- Dokumentation ist über `man`, `info` erreichbar.

- Die User-Id und Gruppen (auch anderer User) kann man sich mit dem Programm `id` anzeigen lassen.
- Die Login-Shell und zusätzliche Infos über den Account (normalerweise der volle Name des Users) kann man mit `chpass` ändern.
- Ein neues Passwort kann man mittels `passwd` eingeben. Das alte Passwort wird zur Verifikation verlangt, danach muß man zweimal das neue Passwort eingeben.
- Zum Auslogen gibt es das `logout` Kommando.

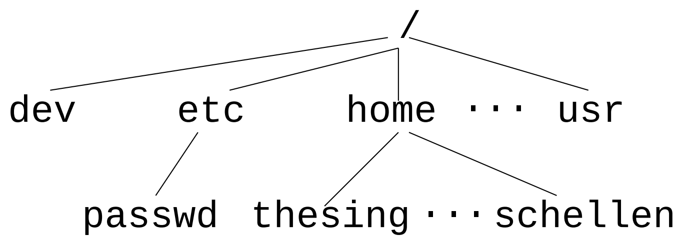
Filesystem

Wie alle BS kennt Unix Files und Directories.

Anders als z.B. unter M$-DOS gibt es in Unix aber keine Unterscheidung verschiedener Geräte, etwa Files auf verschiedenen Festplatten.

Unter Unix existiert nur ein Filebaum (-graph).

Der Baum hat genau eine Wurzel: `/`.



Files/Directories kann man über ihren Pfad ansprechen:

- `/etc/passwd`
- `/home/stud/thesing/Unix-Kurs`

Unix kennt verschiedene Typen von Einträgen im Filesystem:

- Reguläre Files
- Directories
- Links (hart, weich)
- Character-Special Devices (z.B. Terminals: `/dev/console`)
- Block-Devices (z.B. die Festplatten: `/dev/sd0a`)
- Sockets, FIFOs, ...

Unix kann verschiedene Filesystemtypen einhängen, etwa den Inhalt einer CD unter `/dev/cdrom`.

Ebenso können über das Netz Dateinen eines Rechners exportiert und von anderen importiert werden (NFS).

Durch Eingabe des Kommandos `mount` kann man sich einen Überblick verschaffen, etwa `>mount`

```
/dev/sd0a on / type ffs (local)
/dev/sd0g on /usr type ffs (local)
/dev/sd0e on /var type ffs (local)
auspex-qe6:/usr/local on /sunos/usr/local type nfs (nodev, read-only)
auspex-qe6:/solaris on /solaris type nfs (nodev, read-only)
mfs:192 on /tmp type mfs (asynchronous, local, nodev, nosuid)
auspex-qe6:/netbsd-sparc/local on /usr/local type nfs (nodev, read-only)
auspex-qe6:/home-b45 on /home type nfs (nodev, nosuid)
auspex-qe6:/var/spool/mail on /var/mail type nfs (nodev, nosuid)
```


Jede Datei und jedes Directory hat einen Besitzer (i.a. der Benutzer, der das Objekt erzeugt hat.)

Ebenfalls hat jedes Objekt eine Gruppe zugeordnet (dieselben Gruppen, wie bei Usern).

Jedes Objekt hat Zugriffsrechte. Diese legen fest, wer wie auf dieses Objekt zugreifen darf.

Dabei gibt es drei Klassen von Benutzern, für die man Rechte vergeben kann:

- Für den Besitzer des Objektes (Owner).
- Für alle Benutzer, die in der Gruppe sind, zu der das Objekt gehört (Group).
- Für die Welt (alle Benutzer) (World).

Für jede Klasse kann man drei Zugriffsrechte gewähren oder verbieten:

- Lesezugriff (**R**ead)
- Schreibzugriff (**W**rite)
- Ausführen (E**X**ecute)

Damit kommt man auf insgesamt 9 verschiedene Zugriffsrechtskombinationen.

$$\overbrace{\mathrm{RWX}}^{\text{Owner}}\ \overbrace{\mathrm{RWX}}^{\text{Group}}\ \overbrace{\mathrm{RWX}}^{\text{World}}$$

Man kann sich die Zugriffsrechte von Objekten mit dem `ls` Kommando ansehen:

```
>ls -l

drwxr-xr-x   2 thesing  RW-Flur    512 Oct 19 16:08 RCS
-rw-r--r--   1 thesing  RW-Flur 120991 Oct 18 13:28 vorlesung_1.ps
-rw-r--r--   1 thesing  RW-Flur   4884 Oct 18 13:27 vorlesung_1.tex
```

Das Listing gibt die Zugriffsrechte, den Link-Count, den Besitzer, die Gruppe, die Größe, das Datum des letzen Zugriffs und den Namen an.

Um bei Angaben von Filenamen nicht immer an der Wurzel anfangen zu müssen, gibt es für jeden Prozess das aktuelle Directory (Current Directory).

Dieses kann man in der Shell z.B. mit `cd /home/stud/schellen` ändern.

Ausgeben kann man es mit dem Kommando `pwd` (print working directory).

Alle Pfadangaben, die *nicht* mit `/` beginnen werden relativ zum aktuellen Directory interpretiert.

In jedem Directory existieren automatisch zwei Einträge namens '.' und '..'. Diese sind Verweise auf das Directory in dem sie stehen und das 'Vater-Directory'.

Andere Formen von Aliases kann man mit Links einführen:

- *Harte* Links erzeugen einen neuen Verweis auf das referenzierte Objekt, der hart auf das Objekt auf der Platte zeigt.
- *Weiche* Links enthalten nur den Namen des Objektes, auf das sie zeigen sollen. Dieses muß dazu nicht einmal existieren.

Beim Erzeugen eines Files/Directories:

- Wird der Besitzer auf den User gesetzt, der das Objekt kreiert
- Wird die Gruppe des Objektes auf die primäre Gruppe des Benutzers gesetzt
- Die Zugriffsrechte sind per default (Shell) `rw-rw-rw`, dies kann durch `umask` modifiziert werden: etwa `umask 022`
- Die Zugriffsrechte können explizit mittels `chmod` gesetzt werden.
- Die Gruppe eines Objektes kann mittels `chgrp` *`group file`* geändert werden.
- Dazu muß der Benutzer natürlich Mitglied jener Gruppe sein.

Kommandos

- Wechseln/Ausgabe des akt. Direct.: `cd`/`pwd`
- Anzeigen von Objekten: `ls` (man lese die Manpage dazu!)
- Ändern der Zugriffsrechte: `chmod`
- Erzeugen eines neuen Directories: `mkdir`
- Löschen von Files: `rm` (Vorsicht!)
- Umbennen/Verschieben von Files: `mv`
- Kopieren von Files: `cp`
- Ausgabe des Inhaltes: `cat`
- Pager: `less`