

Infoblatt Linux

Übungen zu Betriebssysteme

Hochschule Aalen
Fakultät Elektronik und Informatik
Prof. Dr. Rainer Werthebach

Inhaltsverzeichnis

1	Anmeldung.....	2
	Befehle eingeben.....	3
2	Hilfe.....	4
3	Befehle oder Kommandos.....	5
	3.1 Verzeichnisverwaltung.....	5
	3.2 Datei Kommandos.....	6
	3.3 Benutzer und Rechteverwaltung.....	8
	3.4 Prozessmanagement.....	9
4	Editor.....	9
5	Entwicklung.....	10

1 Anmeldung

Für die Übungen steht der virtuelle Server `in-bs.htw-aalen.de` bereit. Um sich auf diesen Server anzumelden, starten Sie im PC-Pool das Programm **PuTTY**:

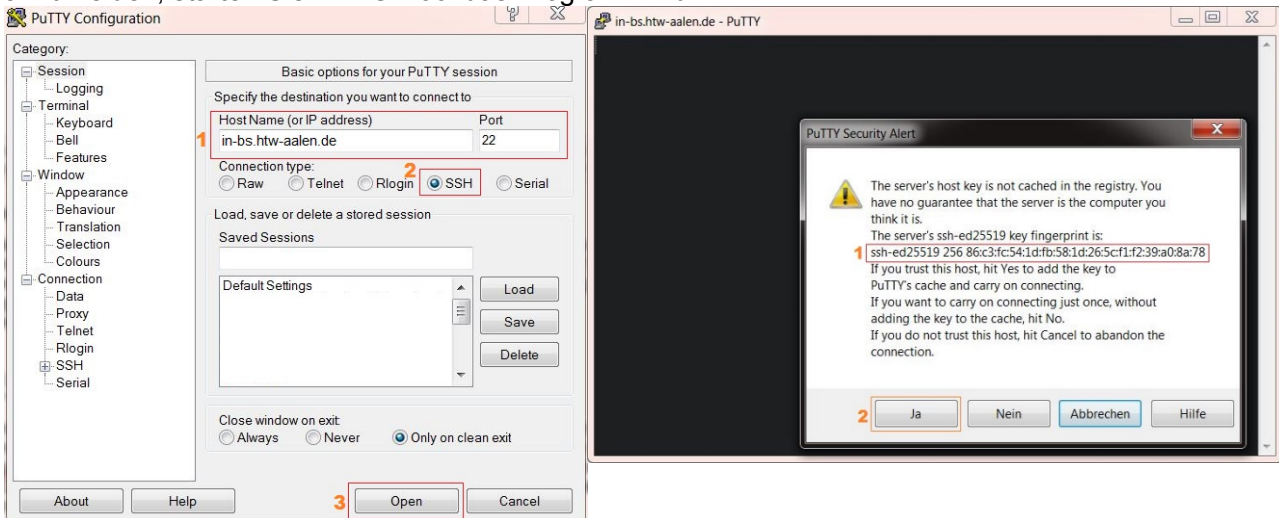


Abbildung 1: PuTTY Startbildschirm

Abbildung 2: PuTTY Security Alert

In Abbildung 1 sehen Sie den Startbildschirm von PuTTY. Geben Sie in das mit 1 markierte Feld die Adresse des virtuellen Servers ein. Achten Sie darauf, dass der Connection type auf SSH steht (2). Durch einen Klick auf Open (3) starten Sie die Verbindung.

Wenn Sie sich zum ersten Mal auf den Server einwählen, sehen sie zunächst das PuTTY Security Alert Fenster (Abb. 2). Es informiert Sie darüber, dass Sie sich auf einen Server anmelden, den das Programm PuTTY vorher noch nicht besucht hat und gibt Ihnen den Fingerprint des Servers (1) zurück, an dem man erkennen kann, ob es sich wirklich um den fraglichen Server handelt. Der Fingerprint des Servers `in-bs.htw-aalen.de` lautet:

```
ssh-ed25519 256 86:c3:fc:54:1d:fb:58:1d:26:5c:f1:f2:39:a0:8a:78
```

Mit einem Klick auf Ja (2) vertrauen Sie diesem Server. Anschließend werden Sie zur Anmeldekonsole (Abb. 3) weitergeleitet:

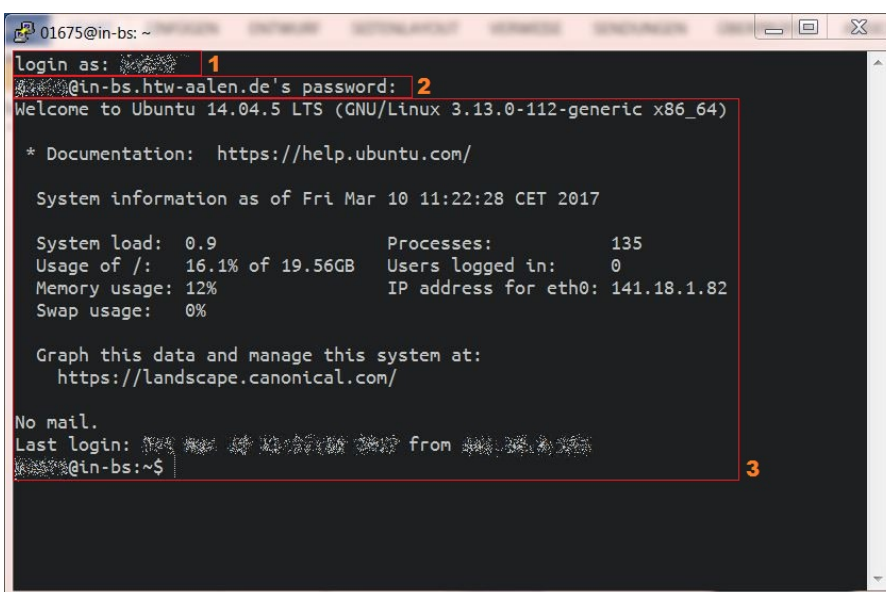


Abbildung 3: PuTTY Anmeldekonsole

Zunächst sehen Sie nur die Zeile 1, in der Sie Ihre Matrikelnummer als Login eingeben. Anschließend werden Sie nach Ihrem Passwort gefragt. Hierbei handelt es sich um dasselbe Passwort, welches Sie auch unter Windows verwenden.

Wenn das Kennwort für den gegebenen Benutzer-Login korrekt ist, meldet sich der Befehlsinterpreter, auch Shell genannt, sofort mit dem Eingabeprompt (3). Stimmen die eingegebenen Daten nicht, so wird eine Fehlermeldung ausgegeben. Das Passwort erscheint bei der Eingabe nicht auf dem Bildschirm. Standardmäßig wird als Eingabeprompt ein \$ - Zeichen verwendet.

Allgemein: `user@computer:Pfad Eingabeaufforderung`

Beispiel: `12345@in-bs:~$`

Hinweise:

~	Pfad zum Home-Verzeichnis
\$ oder % oder #	Eingabeaufforderung (Prompt)

Eine Sitzung beendet man entweder mit `logout`, `exit` oder `<Strg>+<D>`.

Befehle eingeben

Mit einem Befehl wird dem Betriebssystem mitgeteilt, welche Funktionen es ausführen soll. Die eingegebenen Befehle werden von dem Befehlsinterpreter gelesen und anschließend ausgeführt. Jeder Befehl verfügt über eine eigene Syntax (Befehl -Optionen Parameter), in der die erforderlichen und wahlfreien Optionen, Dateien und Parameter festgelegt sind. Eine Option besteht aus einem Minuszeichen, gefolgt von den Einzelbuchstaben aller gewünschten Optionen ohne weitere Zwischenräume. Optionen sind durch Leerzeichen vom übrigen Befehl getrennt. Einige Programme, z.B. Compiler, verlangen jedoch Zwischenräume bei getrennter Angabe aller Optionen, die jeweils mit einem eigenen Minuszeichen eingeleitet werden.

Beispiel: `gcc -s -o p.p.c -lm`

Hinweise:

- Durch Verwendung eines Semikolons (;) können zwei Befehle in derselben Befehlszeile eingegeben werden.
- Bei Befehlen ist die Groß-/Kleinschreibung zu beachten.
- Ein sehr langer Befehl kann über mehrere Zeilen eingegeben werden, indem am Ende der Zeile ein Backslash eingegeben und die `<Return>`-Taste gedrückt wird.
- Mit `<Strg>+<C>` werden die meisten Programme abgebrochen.

2 Hilfe

Jedes UNIX-Kommando kann im Online-Handbuch man (für manual page, Handbuchseite) nachgeschlagen werden. Idealerweise liegt für jedes Kommando, jeden Aufruf der Programmierschnittstelle (Systemfunktionen, C-Bibliothek) und jede wichtige Konfigurationsdatei eine eigene man page vor.

Die Kommandos des Manpage-Systems:

Kommando	Beschreibung
man	Handbuchseiten (MANual page). Ruft die Manpage (liefert eine ausführliche Beschreibung) für ein bestimmtes Programm, eine bestimmte Funktion usw. auf.
whatis	Gibt eine einzeilige Kurzbeschreibung des Programms aus.
help	Einfache Hilfestellung für Anfänger.
apropos	Sucht in den Kurzbeschreibungen nach einer Zeichenfolge und gibt die passenden Einträge samt Kurzbeschreibung zurück. Nützlich wenn man sich z.B. an ein Kommando nicht mehr erinnern kann oder wenn man das Kommando für eine bestimmte Operation nicht kennt.

Für eine schnelle Hilfe zu den Manpages hilft auch ein `man man` weiter.

Abschnitte in Man:

Abschnitt 1	Nutzerbefehle und Kommandos
Abschnitt 2	Systemaufrufe
Abschnitt 3	Subroutinen und Bibliotheksaufrufe
Abschnitt 4	Geräte und Spezialdateien
Abschnitt 5	Dateiformate
Abschnitt 6	Spiele
Abschnitt 7	Verschiedenes, Makropakete und Konventionen
Abschnitt 8	Systemverwaltung
Abschnitt 9	Kernelverwaltung
Abschnitt n	Neu

Zu den jeweiligen auf dem System installierten Paketen findet man unter `/usr/share/doc/-PAKETNAME` Informationen.

3 Befehle oder Kommandos

3.1 Verzeichnisverwaltung

Dateien sind unter Unix (wie eigentlich auf allen modernen Betriebssystemen) in Verzeichnissen angeordnet. Das Unix-Dateisystem ist hierarchisch strukturiert und hat, grafisch dargestellt, die Form eines umgedrehten Baumes. Unter Unix gibt es nur einen einzigen Verzeichnisbaum. Der Ausgangspunkt des Baumes ist die Wurzel (root), die mit / (slash), dargestellt wird und auch root-Verzeichnis genannt wird.

Den Weg durch die Verzeichnisse zu der gewünschten Datei nennt man Zugriffspfad. Zu jeder Zeit des Dialogs verfügt man über ein aktuelles Verzeichnis (directory), dessen Namen man mit pwd erfährt. Nach dem login ist dies das Home-Verzeichnis des Benutzers.

Die Angabe des Zugriffspfades bezieht sich entweder auf das Verzeichnis, in dem man sich gerade befindet oder auf das root-Verzeichnis; entscheidend dafür ist das erste Zeichen der Pfadangabe (s. cd).

In dem Verzeichnis /bin liegen die am häufigsten benutzten Dienstprogramme. Die seltener benutzten Dienstprogramme befinden sich unter /usr/bin, und die selbst installierten unter /usr/local/bin.

Mit folgenden Kommandos (s. nächste Seite) kann man sich innerhalb von Verzeichnissen bewegen und Operationen an diesen ausführen.

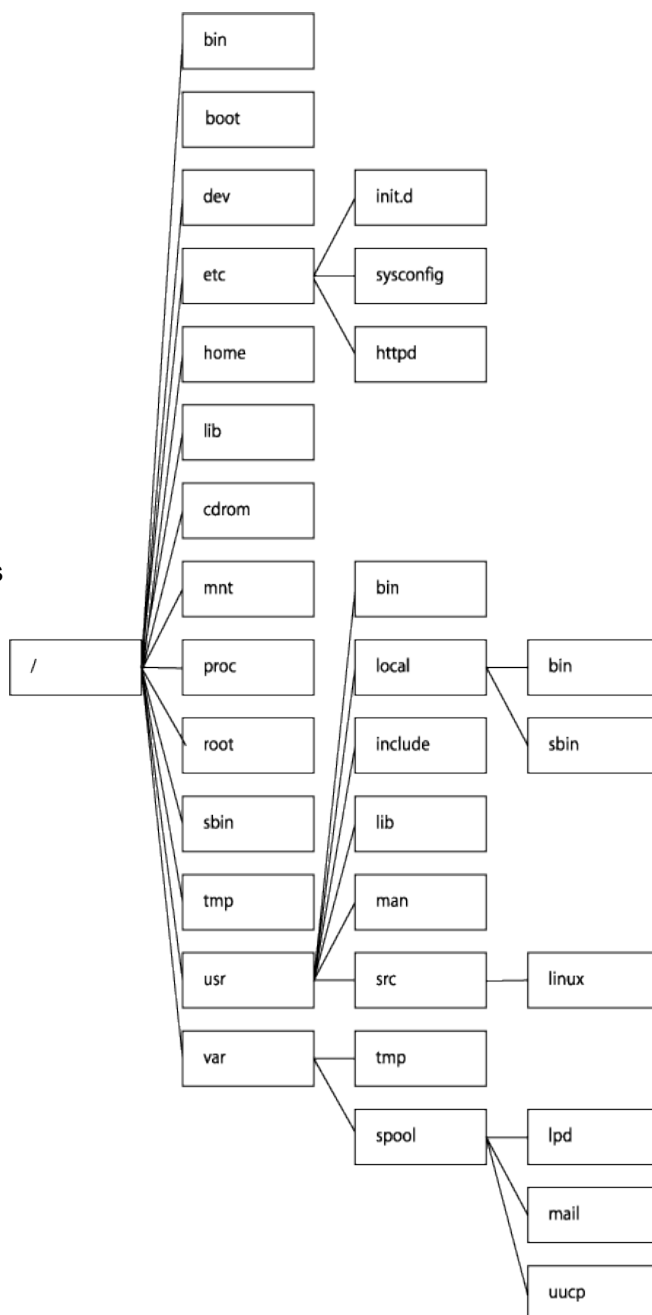


Abbildung 1: Verzeichnisstruktur

>	Umleitung der Ausgabe in Datei
>>	Umleitung der Ausgabe in Datei (Anhängen statt überschreiben der Datei)
<	Umleitung der Eingabe (aus Datei lesen)
	direkte Ein-Ausgabeumleitung (Pipe)

Kommando	Beschreibung
<code>pwd</code>	Pfad des aktuellen Verzeichnisses ausgeben (print working directory)
<code>cd</code>	wechsel in ein anderes Verzeichnis (change directory)
<code>mkdir</code>	erzeugt ein neues Verzeichnis (make directory)
<code>rmdir</code>	entfernt ein Verzeichnis (remove directory)
<code>ls</code>	Auflisten des Verzeichnisinhalts (list directory)
<code>ls -l</code>	Auflisten des Verzeichnisinhalts mit Details
<code>ls -a</code>	Auflisten des Verzeichnisinhalts mit versteckten Dateien

3.2 Datei Kommandos

Kommando	Beschreibung
<code>cat</code>	ausgeben einer (oder mehrerer) Datei(en) auf stdout (concatenate files)
<code>cp</code>	Datei kopieren (copy)
<code>mv</code>	Datei verschieben oder umbenennen (move)
<code>rm</code>	Datei löschen (remove)
<code>touch</code>	Anlegen einer Datei bzw. Dateizugriffsdatum setzen

Filter Programme:

Diese Programme eignen sich dazu, den Inhalt von Dateien zu Parsen und das Ergebnis direkt mit einem anderen Programm weiter zu verarbeiten.

Kommando	Beschreibung
<code>more, less</code>	Dateien Seitenweise anzeigen
<code>head</code>	Anfang einer Datei ausgeben
<code>tail</code>	Ende einer Datei ausgeben
<code>sort</code>	Sortiert eine Datei
<code>find</code>	Sucht nach einer Datei
<code>grep</code>	Sucht Muster in einer Datei
<code>wc</code>	Zählen von Zeichen, Wörtern und Zeilen (word count)

Dateien vergleichen:

Kommando	Beschreibung
cmp	Anzeige der ersten Stelle an der ein Unterschied vorkommt (compare)
diff	Anzeigen der Unterschiede zwischen Dateien (difference)

Archivierung:

Mit den folgenden Programmen können Dateien und Verzeichnisse unter Linux zu einem Archiv verpackt und komprimiert werden. Neben diesen Unix spezifischen Tools stehen natürlich die im Windows Umfeld üblichen Packer wie rar und zip ebenfalls zur Verfügung.

Endung „tar“ Zusammenfassen ohne Komprimierung

<code>tar cfv [ARCHIVNAME].tar [VERZEICHNIS1] [VERZEICHNIS2] [DATEI1] [DATEI2]</code>	Zusammenfassen
<code>tar xfv [ARCHIVNAME].tar</code>	Entpacken

"x" steht für extract, also auspacken, "f" für file (aus einer Datei) und "v" für verbose (ausführliche Ausgabe).

Verwendet man statt "x" ein "c" (create), kann man Verzeichnis und/oder Dateien einpacken und ein eigenes Archiv erzeugen.

Endung „gz“ Zusammenfassen mit Komprimierung

<code>tar cfvz [ARCHIV].tar.gz [VERZEICHNIS1] [DATEI1]</code>	Zusammenfassen
<code>tar xfvz [ARCHIV].tar.gz</code>	Entpacken

Endung „bz2“ Zusammenfassen mit Komprimierung

<code>tar cfvj [ARCHIV].tar.bz2 [VERZEICHNIS1] [DATEI1]</code>	Zusammenfassen
<code>tar xfvj [ARCHIV].tar.bz2</code>	Entpacken

gzip Kompressionstool (Lempel-Ziv)

bzip2 Kompressionstool (Burrows-Wheeler)

Mit gzip oder bzip2 kann nur eine einzige Datei komprimiert werden. Wenn man mehrere Dateien komprimieren möchte, muss man aus diesen erst einmal ein Archiv erstellen.

Dateisystem:

Kommando	Beschreibung
du	Anzeige des benötigten Speicherplatzes (disk usage)
df	Anzeige des freien Speicherplatzes auf einem Datenträger (disk free)
mount	Dateisysteme einhängen in den Verzeichnisbaum
umount	Dateisysteme aushängen aus dem Verzeichnisbaum
fsck	Dateisystem überprüfen (file system check)
mkfs	Dateisystem erstellen (make file system)

3.3 Benutzer und Rechteverwaltung

Unix ist von Anfang an ein Multi-User-Betriebssystem. Das bedeutet, dass verschiedene Benutzer am Rechner arbeiten können und voreinander abgeschottet werden: man kann anderen Benutzern erlauben oder verbieten, auf bestimmte Dateien zuzugreifen. Zusätzlich kann man Benutzer auch Gruppen zuordnen, denen kollektiv bestimmte Zugriffsrechte gewährt werden können. Eine besondere Rolle spielt der Benutzer root (Systemadministrator), der als einziger Benutzer vollen Zugriff auf das System hat.

```
rene@ares ~/Dokumente % ls -l
-rw-r--r--  1 rene rene      37937 2011-03-15 23:16 Infoblatt.odt
drwxrwxr-x  9 rene rene      4096 2010-11-26 22:30 javadocs/
drwxr-xr-x  2 rene rene      4096 2011-03-17 09:56 openvpn-extern/
-rw-----  1 rene rene      1577 2011-03-15 12:12 txt
-rw-r--r--  1 root root    197507 2011-03-15 17:33 wohnheim\ intense\ scan\
15march2011.xml
```

-rwxr--r--

0123456789

Position	Bedeutung
0	Dateityp: - = Datei, d = Verzeichnis, l = symbolischer Link, p = benannte Pipe, c = zeichenorientiertes Gerät, b = blockorientiertes Gerät, s = Socket
1-3	Lese,Schreib- und Zugriffsrechte für den Besitzer (user) r =read, w =write und x =execute
4-6	Lese,Schreib- und Zugriffsrechte für die Gruppe (group)
7-9	Lese,Schreib- und Zugriffsrechte für alle (other)

Kommando	Beschreibung
who	zeigt die aktiven Benutzer an
su	wechselt den Benutzer
passwd	Passwort setzen
chmod	Dateiberechtigungen ändern
chown	Datei Eigentümer oder Gruppe ändern
chgrp	Gruppenzugehörigkeit ändern
groups	Anzeige aller Gruppen, zu denen der Benutzer gehört

3.4 Prozessmanagement

Programme, welche unter Linux gestartet werden, stellen einen eigenständigen Prozess dar.

Jeder Prozess kann wiederum eigenständige Unterprozesse besitzen. Jeder Prozess besitzt eine eigenständige Prozess-ID (PID). Der Vater aller Prozesse ist init mit der PID 1. Jedem Prozess sind per default ein Eigentümer, eine PID, die PID des Elternprozesses (PPID), ein Scheduling-Parameter und eine TTY zugeordnet.

Für die Steuerung und Verwaltung der Prozesse stehen folgende Programme zur Verfügung:

Kommando	Beschreibung
ps	Statusanzeige der laufenden Prozesse des Benutzers
ps -e	Alle Prozesse des Systems anzeigen
kill	Beenden eines Prozesses
kill -9 1337	Unbedingtes Beenden des Prozesses mit der PID 1337
killall	alle Prozesse mit angegebenen Namen beenden
top	Interaktive Anzeige der laufenden Prozesse
free	Anzeige des freien Speicherplatzes (flüchtiger Speicher)
nice	setzen von Prozess Prioritätslevel
pstree	Anzeigen der Prozesse in Baumform
&	Starten eines Prozesses im Hintergrund (Postfix)
fg/bg	Prozess in den Hintergrund/Vordergrund
nohup	Prozess nach beenden des Terminal weiterlaufen lassen

4 Editor

Neben den grafischen Editoren unter dem X-Window System gibt es ein paar Standardeditoren, die in fast jedem unixartigen Betriebssystem verfügbar sind. Hierunter fallen unter anderem der nano, ed oder der vi bzw. vim.

Vim

Eine Datei kann mit folgendem Kommando bearbeitet werden:

```
vi DATEI
```

Falls diese Datei nicht existiert, wird sie beim Speichern angelegt. Der Editor unterscheidet drei grundlegende Arbeitsmodi:

command mode	Modus in dem vi-Kommandos eingegeben werden können
insert mode	Modus zur zeilenweise Eingabe von Text, wird mit ESC wieder verlassen
ex mode	Aufruf des Editors ex aus vi

Befehle im Vi:

:q	beendet den Editor
:q!	beendet ohne die Änderungen zu speichern
:w	schreibt die Datei
/MUSTER	Vorwärtssuche nach MUSTER
?MUSTER	Rückwärtssuche nach MUSTER
:%s/MUSTER/ERSETZUNG/g	ersetzt MUSTER durch ERSETZUNG global
:syn on	schaltet Syntax-Highlighting ein
:set bg=dark	schaltet in den Dark-Modus

5 Entwicklung

Tools zum Entwickeln unter Linux:

Kommando	Beschreibung
gcc	Standard GNU C Compiler
make	Tool zum automatisierten Compilieren von größeren Projekten

einfaches makefile:

Dateiname: makefile

Inhalt:

1: Ziel: Abhängigkeiten

2: ↵(TAB) Systembefehl

Beispiel:

1: all:
2: gcc meinQuellCode.c -o meinProgrammName

Weitere Infos z.B: <https://secure.wikimedia.org/wikibooks/de/wiki/Linux-Kompendium>

Gefundene Fehler oder Verbesserungsvorschläge bitte an:
rainer.werthebach@hs-aalen.de