



Grundlagen der Linux-Kommandozeile

Informatica Feminale, Bremen 2008

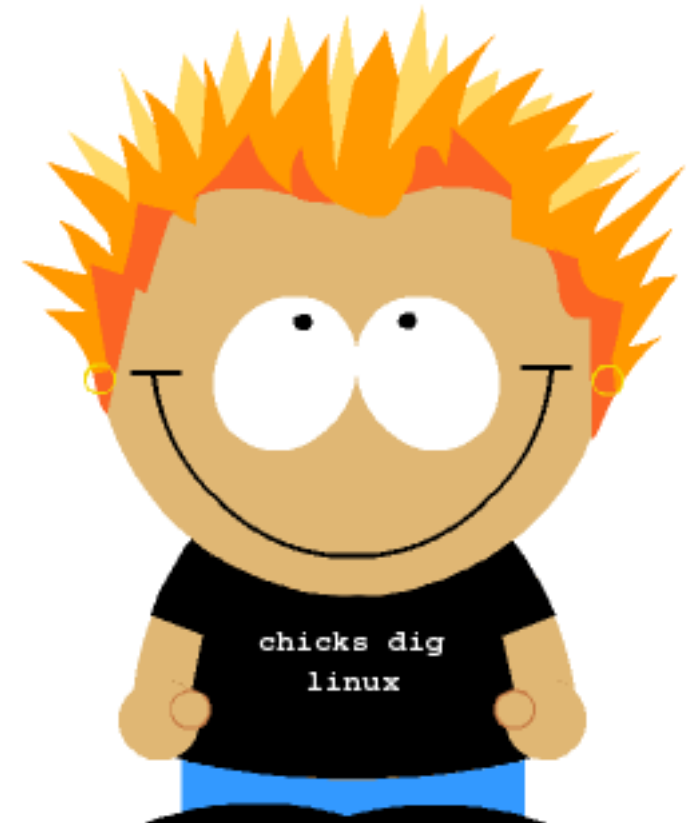
Jutta Horstmann

data in transit

I T - B E R A T U N G

Jutta Horstmann (Dipl. Inform., Dipl. Pol.)
IT-Beraterin

- Schwerpunkt: Freie Software
- zuerst: EDV-Support
- dann: Systemadministration,
Datenbankadministration
- dann: Datenbank-Entwicklung
- dann: selbständig



data in transit

data in transit
I T - B E R A T U N G

- gegründet 2005 in Berlin
- 2006 Umzug nach Bonn
- Schwerpunkte:
Web-Programmierung, CMS, Datenbanken
Konzeption, Systemarchitektur, Requirement Engineering
- <http://www.dataintransit.com>

Vorstellungsrunde: Warum will ich auf die Kommandozeile...


- Linux-System
 - Auf dem Desktop?
 - Auf dem Server?
- DOS-Kommandozeile?
- Schon Erfahrungen gesammelt?
- Vorteile / Nachteile zur grafischen Oberfläche?
- Programme, die es nur auf der Kommandozeile gibt?



Die Themen



Grundlagen



Erste Schritte – Auf dem System umsehen



Verzeichnisse und Dateien



Nützliche Helferlein



Wildcards



Pipes



Eingabeumleitung Ausgabeumleitung

Grundlagen

- Benutzerschnittstelle zum Linux-Betriebssystem:
Shell oder **Kommandozeile** (CLI: *Command Line Interface* und *Command Line Interpreter*)
- Nutzerin kann Kommandos eintippen, welche der Computer direkt ausführt:
 - von der Shell zur Verfügung gestellte Kommandos (z.B. “echo”)
 - Externe Programme (z.B. “grep” oder “firefox”) ausführen oder starten

Grundlagen

- Shell verfügt über eigene Scriptsprache, mit der die Nutzerin selbst kleine Programme schreiben kann (=> Kurs Shell-Programmierung!)
- Es gibt eine große Zahl verschiedener Shells:
 - sh: Bourne Shell (Stephen Bourne, 1977)
 - bash: Bourne Again Shell (Teil von GNU, 1980er)
 - z-shell, korn-shell, c-shell, tc-shell, rc-shell ...
- Im Kurs: bash (Linux, UNIX, Mac OS X, Windows: CygWin)

Jetzt gehts los!

- Auf Rechner mit GUI:
 - Starten eines “Terminals” oder “Konsole”
 - Umschalten auf Nutzersitzung ohne GUI: STRG +ALT+ F1, zurück STRG+ALT+ F7
- Auf Rechner ohne GUI (oft bei Remote Server-Administration per ssh)
- Eingabeaufforderung (“Prompt”):
jh@neko:~

Erste Schritte: Auf dem System umsehen

- **whoami** – wer bin ich?
- **who** – wer ist sonst noch eingeloggt?
- **w** – wer ist sonst noch eingeloggt und macht was?
- **finger** - wer ist sonst noch eingeloggt (mit Zusatz-Informationen, z.B. voller Name)

Verzeichnisse und Dateien

Verzeichnisse:

- das oberste ist das “Root-Verzeichnis”: /
- wichtige weitere Verzeichnisse:
/bin, /etc, /home, /sbin, /tmp, /usr, /var
- das eigene Home-Verzeichnis:
/home/<loginname>/

In Verzeichnissen bewegen

- **pwd** (print working directory) – wo stehe ich?
- **cd** (change directory)
 - **cd** **..** ein Verzeichnis nach oben
 - **cd** ins eigene Home-Verzeichnis
 - **cd** **~** ins eigene Home-Verzeichnis
 - **cd** **-** ins vorherige Verzeichnis
 - das aktuelle Verzeichnis
 - **..** ein Verzeichnis darüber

Verzeichnisinhalte ansehen

- **ls** (list directory contents)
- **ls -la**

```
jh@neko:~> ls -la
insgesamt 5280
drwxr-xr-x 44 jh users 4096 2008-08-24 19:10 .
drwxr-xr-x  5 root root 4096 2008-08-13 12:47 ..
-rw-----  1 jh users 14310 2008-08-24 16:28 .bash_history
-rw-r--r--  1 jh users 2116 2008-08-18 13:23 .bashrc
drwxr-xr-x  2 jh users 4096 2008-08-13 16:57 bin
drwxr-xr-x  6 jh users 4096 2008-08-24 16:26 bkp
drwxr-xr-x 13 jh users 4096 2008-08-14 08:33 dataintransit
```

Erläuterung zum ls - Output

```
drwxr-xr-x  6 jh  users      4096 2008-08-24 16:26 bkp
```

d es handelt sich um ein Verzeichnis (directory)

rwxr-xr-x Rechte (read, write, execute; User, Group, All)

6 Anzahl der harten Links

jh users Name der Besitzerin und der Gruppe

4096 Größe in Bytes (Verzeichnis, nicht Inhalt!)

2008-08-24 16:26 Zeitpunkt der letzten Änderung

bkp Name des Verzeichnisses, der Datei ...

ls mit Argumenten

- `ls -la`
 - l long (langes Ausgabeformat)
 - a all (auch versteckte Dateien/Verzeichnisse)
- Weitere interessante Argumente:
 - h human (lesbare Dateigrößen)
 - l, -m (formatiere zeilenweise oder komma-getrennt)
 - X, -S, -t (sortiere nach Extension, Größe, letzter Änderung)

Verzeichnisse und Dateien anlegen

- **mkdir** <dirname> (make directory)
mkdir verzeichnis1
- **cd** <pfad>
cd verzeichnis1, cd ~/verzeichnis1
- **touch** <dateiname> - leere Dateien anlegen:
touch neuedatei
touch bla.txt blub boah.txt blossom
- **ls**

Löschen und umbenennen

- **rm** <datei> remove: Datei löschen
- **rmdir** <dir> Verzeichnis löschen
- **mv** <datei> <dateineu>
move: Datei umbenennen
- **cp** <datei> <datei.bkp>
copy: Kopie der Datei anlegen

Nützliche Argumente

- R / -r recursive: Führe Befehl rekursiv aus
 - cp -R kopiere Verzeichnis und alle Inhalte
- f force: Frage nie nach
 - rm -rf lösche Verzeichnis mit allen Unterordnern und Inhalten ohne nachzufragen (Achtung! Gefahr!!)
- i interactive: Frage immer nach
 - mv -i Verschiebe a nach b und frage nach, wenn dabei vorhandenes b überschrieben würde

Aufgabe

- Lege `verzeichnis2` in `verzeichnis1` an
- Lege `verzeichnis3` in `verzeichnis2` an
- Verschiebe oder kopiere Dateien von `verzeichnis1` in die neuen Verzeichnisse
- Lege in `verzeichnis3` eine Datei namens `informatica08` an
- Bewege dich in dein Home-Verzeichnis
- Sieh dir jetzt mit das Dateisystem mit allen neuen Verzeichnissen und Inhalten an

Lösung

- `cd verzeichnis1`
- `mkdir verzeichnis2`
- `mkdir verzeichnis2/verzeichnis3`
- `cp bla.txt verzeichnis2`
- `mv blubb verzeichnis2/verzeichnis3`
- `touch
verzeichnis2/verzeichnis3/informatica08`
- `cd`
- `ls -R`

Nützliche Helferlein

- Pfeiltaste ↑ - lässt durch die “history” aller eingegebenen Kommandos blättern
- Kommando + Bild ↑ - durch die “history” des Kommandos
- Tab-Completion – Tab-Taste ergänzt Kommandonamen
- ESC + '.' - holt das letzte Argument des letzten Bash-Befehls
- Copy: Markieren – Paste: Mittlere Maustaste oder beide
- Wildcards *, ?, [] (siehe nächste Folie)
- Unterbrechen eines laufenden Programms: STRG-c

Wildcards

- * Ein oder kein oder mehrere beliebige Zeichen:
`rm *.jpg`
- ? Ein beliebiges Zeichen: `ls datei_?.txt`
- [] Enthält erlaubte Zeichen: `ls bl[aui]*`,
`ls bla_[1-5]*`
- { Enthält erforderliche Zeichen:
`touch bla_{1,2,3,4,5,6,7}.txt`
- ! Verneinung: `ls bl[!ai]*`

Nützliche Programme

- `cat <dateiname>` gibt Dateiinhalte aus
- `more <dateiname>` gibt Dateiinhalte aus, mit Blätterfunktion
- `grep <muster> <datei(en)>`
liest Datei(en) und gibt alle Zeilen aus, die <muster> enthalten (z.B. `grep Informatica *`)
- `sort`: sortiert eine Ausgabe
- `echo`: gibt aus was eingegeben wurde

Ausgabeumleitung: Pipes

Die Ausgabe eines Kommandos wird als Eingabe eines weiteren Kommandos benutzt.

Beispiele:

```
who | grep <username>
```

```
ls -la | more
```

```
cat ~/.bash_history | sort
```

Ein- und Ausgabeumleitung

- `ls -t > verzeichnis.txt`
leitet die Ausgabe des Kommandos in eine Textdatei um, wenn `verzeichnis.txt` schon existiert: wird überschrieben
- `sort < verzeichnis.txt`
Leitet Dateiinhalt als Eingabe in Kommando
- `sort < verz.txt > verz_sortiert.txt`
Leitet Dateiinhalt als Eingabe in Kommando und dessen Ausgabe in neue Datei

An Dateiende anfügen: >>

```
echo "Inhalt" > verzeichnis.txt
```

Legt die Datei an und schreibt "Inhalt des Verzeichnisses" hinein. Existiert schon eine Datei dieses Namens, wird sie überschrieben.

```
ls >> verzeichnis.txt
```

Fügt die Ausgabe des Kommandos am Ende der Datei an (wenn sie noch nicht existiert, wird sie neu angelegt).

Umleitung mit Datei-Deskriptoren

- 0: Standardeingabe stdin (Tastatur)
- 1: Standardausgabe stdout (Terminal/Konsole)
- 2: Fehlerausgabe stderr (Terminal/Konsole)
- kommando `>&n`
- kommando `m>&n`
- `2>datei` (Umleitung von stderr in eine Datei)
Beispiel: `cp gibtsnich neuedatei 2>errfile`

Hilfe mit *man* oder *info*

Aufgabe: Jede nimmt ein Kommando und liest unter *man* oder *info* nach, wie es funktioniert, sucht eine oder mehrere sinnvolle Optionen raus und erklärt sie für die anderen.

Bsp: `find`, `info`, `uniq`, `tr`

- `man <befehl>`, `info <befehl>`

Auch möglich:

- `<befehl> --help`

Alias und Umgebungsvariablen

- Beispiel: `alias l='ls -laF'`
- Datei `~/ .bashrc` editieren
- `source .bashrc` oder `bash` (Änderungen laden)
- `alias` (Aliase anzeigen lassen)

- `env` (Umgebungsvariablen anzeigen lassen)
- Datei `/etc/profile`
- Datei `~/ .bash_profile`

Wo ist mein Zeug geblieben?

- **locate**

locate: warning: database '/var/lib/slocate/slocate.db' is more than 8 days old

- **updatedb** (als root auszuführen)

Literatur (online)

Einführung zur Unix/Linux-Shell:

<http://de.wikipedia.org/wiki/Unix-Shell>

Unix Power Tools (echt alt, aber online):

<http://docs.babitch.com/O'Reilly/unix/upt/index.htm>

By Jerry Peek, Tim O'Reilly, Mike Loukides
Second Edition August 1997

Wikibook über Shell-Scripting (aber auch Vergleich verschiedener Shells);

http://en.wikibooks.org/wiki/Bourne_Shell_Scripting

Fragen???



data in transit
Kontakt

<http://www.dataintransit.com>
jh@dataintransit.com