

6.Übung

Systemsoftware (SYS)

Christian Baun
`cray@unix-ag.uni-kl.de`

Hochschule Mannheim – Fakultät für Informatik
Institut für Robotik

9.11.2007

Wiederholung vom letzten Mal

- Umleiten von Ein- und Ausgaben (`<`, `>`)
- Bytes, Zeichen, Wörter und Zeilen zählen (`wc`)
- Abarbeitungsgeschwindigkeiten messen (`time`)
- Der Alias-Mechanismus
- Dateien suchen und finden (`find`, `locate`, `whereis`, `which`)

Heute

- Einführung für Linux/UNIX-Anwender (Teil 5)
 - Zeitgesteuerte Kommandoausführung (cron)
 - Kommandos zu einer späteren Zeit ausführen (at)
 - System- und Prozessüberwachung (top)
 - Regelmäßige Programmausführung (watch)
 - Systeminformationen ausgeben (uname)
 - Sortieren (sort)
 - Umgebungsvariablen anzeigen (env, printenv)
 - Umgebungsvariablen setzen und löschen (export, set, unset)

Kommandos zeitgesteuert ausführen mit cron

- Der Dämon cron ist ein Dienst zur zeitgesteuerten Jobsteuerung.
- Perfekt geeignet für regelmäßige, wiederkehrende Aufgaben.
- Beispiele für typische cron-Aufgaben:
 - Inhalts von /tmp in festen Abständen löschen.
 - Regelmäßige Backups von bestimmten Verzeichnissen.
 - Erinnerungs-Email vor dem Geburtstag wichtiger Menschen.
- Die auszuführenden Befehle stehen in einer Tabelle, der Crontabelle.
- Jeder Benutzer hat eine eigene Crontabelle.
- Die eigene Crontabelle bearbeitet man mit dem Kommando crontab.

crontab -l	Die crontab eines Benutzers ausgeben lassen (<i>list</i>).
crontab -e	Die crontab eines Benutzers bearbeiten (<i>edit</i>).
crontab -r	Die komplette crontab eines Benutzers löschen (<i>remove</i>).

Kommandos zeitgesteuert ausführen mit cron (2)

- Das Kommando `crontab` ruft einen Editor auf (Standardmäßig `vi`) und öffnet die eigene Crontabelle.
- Soll die Crontabelle mit einem alternativen Editor geöffnet werden, muss die Umgebungsvariable `EDITOR` gesetzt sein und den Namen und eventuell Pfad des alternativen Editors enthalten.

```
user@server:~$ export EDITOR=/usr/bin/joe
```

- Der Systemadministrator kann die Crontabellen aller Benutzer einsehen, ändern und löschen. \implies `crontab -u Benutzername`
- Beispiel: Crontabelle des Benutzers Student ausgeben:

```
server:~# crontab -u Student -l
```

Syntax der Crontabelle

- Die Crontabelle besteht aus 6 Spalten.
- Die ersten fünf Spalten legen den Ausführungszeitpunkt des Kommandos fest. In der sechsten Spalte ist das auszuführende Kommando.
- Die Spalten werden durch Leerzeichen oder Tabulatoren getrennt:
 1. Spalte: Minute (**0-59** oder *****)
 2. Spalte: Stunde (**0-23** oder *****)
 3. Spalte: Tag (**1-31** oder *****)
 4. Spalte: Monat (**1-12**, **Jan-Dec**, **jan-dec** oder *****)
 5. Spalte: Wochentag (**0-6**, **Sun-Sat**, **sun-sat** oder *****)
 6. Spalte: Auszuführendes Kommando. Eventuell mit Pfad
- Einträge in der Crontabelle dürfen keine Zeilenumbrüche enthalten!
- Kommentare beginnen in der Crontabelle immer mit einer Raute #.

Beispiele zu cron

- An jedem Werktag um 7:10 Uhr mit dem Lieblingslied wecken lassen:

```
10 7 * * 1,2,3,4,5 /usr/bin/mpg123 -b 1024 /pfad/zu/song.mp3
```

- Inhalt von /tmp jeden Sonntag und Mittwoch um 13 Uhr löschen:

```
0 13 * * Wed,Sun rm -rf /tmp > /dev/null
```

Der Zusatz > /dev/null legt fest, dass die Ausgabe des Jobs nicht per Email geschickt wird, sondern nach /dev/null weitergeleitet wird.

- Am 10. jeden Monats um 11:45 Uhr das Skript skript.sh aufrufen und die Ausgabe an die Datei mylog.log anhängen:

```
45 11 10 * * /usr/bin/skript.sh >> /var/log/mylog.log
```

Kommandos zu einer späteren Zeit ausführen mit `at`

- Mit dem Kommando `at` können Kommandos zu einem `at`-Job zusammengefasst und zu einer bestimmten Zeit und optional an einem bestimmten Datum ausgeführt werden.
- Im Gegensatz zu `Cron` führt `at` einen Job immer nur einmal aus.
- Die Zeit kann als Zahlenwert (Stunde und optional Minuten) oder als Schlüsselwort `midnight`, `noon` `now` oder `teatime` angegeben werden.
- Ein optionales Datum kann Monat, Tag, oder die Schlüsselworte `today` oder `tomorrow` enthalten.
- Das Kommando `at` kann immer vom Benutzer `root` ausgeführt werden. Alle anderen Benutzer müssen in der Datei `/etc/at.allow` stehen, wenn diese existiert. Wenn die Datei nicht existiert, dürfen die Benutzer nicht in der Datei `/etc/at.deny` stehen. Existieren beide Dateien nicht, darf nur `root` mit `at` arbeiten.

Mit at arbeiten (1)

- Die Eingabe der Kommandos wird durch ein EOF (End of File) mit dem Tastenkürzel Strg-D beendet.
- Mit dem Kommando at werden at-Jobs definiert.
- Mit atq werden alle wartenden Jobs des Benutzers aufgelistet. Führt root das Kommando atq aus, werden alle at-Jobs aller Benutzer aufgelistet.

```
$ atq
4      Thu Oct 25 17:37:00 2007 a testuser
3      Thu Oct 25 18:10:00 2007 a testuser
5      Mon Oct 29 15:14:00 2007 a testuser
```

- Mit atrm <jobID> können at-Jobs gelöscht werden.
- Die Zeitangaben können in der Form **h**, **hh** oder **hhmm** erfolgen. Beispiele sind: 7, 0740, 7:30 oder 19:35.

Mit at arbeiten (2)

- Durch am oder pm basiert die Zeitangabe auf der 12-Stunden-Uhr.
- Datumsangaben sind optional. Die Beschreibung ist wie folgt:
 - Der **Tag** wird mit dem englischen Namen beschrieben. Entweder abgekürzt mit den ersten drei Buchstaben oder ausgeschrieben.
 - Der **Monat** wird mit dem englischen Namen beschrieben. Entweder abgekürzt mit den ersten drei Buchstaben oder ausgeschrieben. Alternativ kann der Monat auch als Zahl angegeben werden.
 - Das **Jahr** wird vierstellig angegeben.
- Einige gültige at-Zeitangaben:
 - at 2015 Nov 10
 - at 8:15 pm November 10
 - at 8 am Saturday
 - at teatime tomorrow

Systemressourcen und Prozesse überwachen – top

top [Option] ...

- Das Kommando top hat eine ähnliche Funktionalität wie ps.
- Mit top überwacht man Rechen- und Speicherressourcen sowie Prozesse.
- Die Ausgabe von top wird automatisch alle 2 Sekunden aktualisiert. Der Zeitraum kann mit der Option -s <Sekunden> festgelegt werden.
- Die Ausgabe von top ist in zwei Abschnitte geteilt:
 1. Fünf Kopfzeilen mit Informationen zur Uptime, mittleren Last, Anzahl der Prozesse, Speicherverbrauch, CPU- und SWAP-Auslastung, usw.
 2. Tabelle der aktuell laufenden Prozesse mit Informationen zur PID, Benutzer, Priorität, Speicherverbrauch, CPU-Auslastung, Zeit seit Erstellung, Kommandoname, usw.

Einige Tastenkürzel von top

- Das Kommando top kennt viele Tastenkürzel. Einige hilfreiche sind:

Leertaste	Sofortige Aktualisierung der Ausgabe.
c	Umschalten zwischen Befehl und Kommandozeile.
i	Inaktive Prozesse anzeigen oder ausblenden.
l	Mittlere Last (<i>load level</i>) anzeigen oder ausblenden.
m	Speicheranzeige anzeigen oder ausblenden.
t	Anzeige der Prozess- und CPU-Zustände anzeigen oder ausblenden.
A	Sortiert die Prozesse nach Alter (jüngster Job zuerst).
M	Sortiert die Prozesse nach Speicherverbrauch.
N	Sortiert die Prozesse nach Prozess-ID.
P	Sortiert die Prozesse nach CPU-Belastung.
T	Sortiert die Prozesse nach gelaufener Zeit.
W	Schreibt die aktuellen Einstellungen nach ~/.toprc.

Regelmäßige Programmausführung mit watch

- Häufig beobachtet man Echtwicklungen, wie z.B. Anwachsen von Dateien, Systemauslastung, Belegung der Partitionen, usw.
- Mit dem Kommando watch können Kommandos in gewünschten Abständen automatisch ausgeführt werden.
- Die Voreinstellung ist 2 Sekunden.
- Bei jedem Durchgang löscht watch zuerst den Bildschirm.
- `$ watch df`

```
Every 2,0s: df                                     Thu Oct 25 16:48:08 2007

Dateisystem      1K-Blöcke  Benutzt Verfügbar Ben% Eingehängt auf
/dev/hda3         67406640  58012324  9394316   87% /
tmpfs              647744    0         647744    0% /dev/shm
/dev/hda1         9267572   8100280  1167292   88% /mnt/windows
tmpfs              10240     76        10164    1% /dev
```

Das Kommando `watch` bedienen

- Mit der Option `-n <Sekunden>` kann der Zeitabstand zwischen den Kommandoausführungen von `watch` eingestellt werden.
- Um die Unterschiede zwischen den Ausgaben von `watch` hervorzuheben und damit die Lesbarkeit zu erhöhen, existiert die Option `-d`.
- Mit dem Tastenkürzel `Strg-C` wird die Ausführung von `watch` beendet.

Systeminformationen ausgeben – uname

- Das Kommando `uname` kann einige wichtige Systeminformationen ausgeben:
- Mit der Option `-a` gibt `uname` alle Informationen aus.

```
$ uname -a
Linux olymp 2.6.18 #3 PREEMPT Tue Jan 2 16:58:43 CET 2007 i686
GNU/Linux
```

- Die Informationen umfassen:
 - Kernelname (Linux)
 - Hostname (olymp)
 - Release-Nummer des Kernels (2.6.18)
 - Kernelversion (#3 PREEMPT Tue Jan 2 16:58:43 CET 2007)
 - Architektur (i686)
 - Name des Betriebssystems (GNU/Linux)

Einzelne Informationen von `uname` ausgeben

- Das Kommando `uname` kennt mehrere Optionen, um die Systeminformationen auch einzeln ausgeben zu können.
 - `--help` Eine Hilfe ausgeben.
 - `-a` Alle bekannten Informationen ausgeben.
 - `-i` Die Hardwareplattform ausgeben.
 - `-m` Die Rechnerarchitektur ausgeben.
 - `-n` Den Hostnamen ausgeben.
 - `-o` Den Namen des Betriebssystems ausgeben.
 - `-p` Den Typ des Prozessors ausgeben.
 - `-r` Die Release-Nummer des Kernels ausgeben.
 - `-s` Den Kernelnamen ausgeben.

Sortieren mit sort

- Das Kommando sort teilt jede Zeile seiner Eingabe in Felder ein.
- Als Trenner sind Leerzeichen (Blank) und Tab als Trenner voreingestellt.
- Jede Zeile der Eingabe wird über die Felder von links nach rechts sortiert.

```
$ cat sortieren
```

```
BCDE STUV
```

```
ABCD EFGH
```

```
WXYZ HIJK
```

```
DEFG MNOP
```

```
ABCD LMNO
```

```
$ sort sortieren
```

```
ABCD EFGH
```

```
ABCD LMNO
```

```
BCDE STUV
```

```
DEFG MNOP
```

```
WXYZ HIJK
```

```
$ sort -r sortieren
```

```
WXYZ HIJK
```

```
DEFG MNOP
```

```
BCDE STUV
```

```
ABCD LMNO
```

```
ABCD EFGH
```

Optionen von sort

<code>--help</code>	Eine Hilfe ausgeben.
<code>-b</code>	Führende Leerzeichen und Tabs ignorieren.
<code>-f</code>	Klein- als Großbuchstaben behandeln.
<code>-n</code>	Numerisch sortieren (9 kommt vor 10).
<code>-r</code>	Ergebnis der Sortierung umkehren.
<code>-u</code>	Doppelte Zeilen ignorieren.
<code>-o<Dateiname></code>	Ergebnis in eine Datei schreiben.
<code>-c</code>	Nur prüfen, ob die Dateien sortiert sind.
<code>-t<Zeichen></code>	<Zeichen> als Trenner verwenden.
<code>-k<n>[,<m>]</code>	Die ersten <n> Felder werden ignoriert und die Felder n bis m-1 zum Sortieren verwendet. Die Felder werden mit 1 beginnend nummeriert.

Sortierschlüssel angeben mit `sort`

- Mit der Option `+n [-m]` werden die Zeichen zwischen `n` (ausschließlich) und `m` (einschließlich) zum Sortierschlüssel bestimmt.
- Wenn `m` fehlt, sind alle Zeichen von `n` bis zum Zeilenende Sortierschlüssel.
- Positionen werden in der Form `x.y` angegeben. Das heißt: Zeichen Nummer `y` von Feld Nummer `x`.
- Ist kein `y`-Teil angegeben, wird nach dem ersten Zeichen im Feld sortiert.
- Beispiele für Sortierschlüssel:

`+2` Linux ist **ein einfaches Betriebssystem**.

`+3.2` Linux ist **ein** einfaches Betriebssystem.

`+2 -3` Linux ist **ein** einfaches Betriebssystem.

Einsatzbeispiele zu sort

- Eine Dateiliste erzeugen und anhand der Zeilen in den Dateien sortieren:

```
wc -l * | sort -nr
```

- Die Einträge in der Datei /etc/passwd anhand der Benutzer-ID in numerischer Ordnung sortieren:

```
sort -k3 -n -t: passwd
```

Umgebungsvariablen ausgeben – env und printenv

- Umgebungsvariablen sind Variablen der Shell, in denen beliebige Daten gespeichert und wieder ausgelesen werden können.
- Welche Umgebungsvariablen aktuell gesetzt sind, erfährt man mit den Kommandos env und printenv.

```
user@server:~$ env
TERM=xterm
SHELL=/bin/bash
PS1=\u@\h:\w\$
USER=user
PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:
PWD=/home/user/
EDITOR=joe
LANG=de_DE@euro
LANGUAGE=de_DE:de:en_GB:en
DISPLAY=:0.0
...
```

Umgebungsvariablen setzen und löschen

- Umgebungsvariablen anlegen mit set oder besser export bzw. setenv.
- Umgebungsvariablen entfernen mit unset bzw. unsetenv.
- Zugriff auf Umgebungsvariablen: Immer mit \$ vor dem Namen.

```
user@server:~$ export EDITOR=/usr/bin/joe
user@server:~$ printenv | grep EDITOR
EDITOR=/usr/bin/joe
user@server:~$ echo $EDITOR
/usr/bin/joe
user@server:~$ unset EDITOR
user@server:~$ printenv | grep EDITOR
```

- **Problem:** Mit export angelegte Umgebungsvariablen sind weg, wenn die Shell beendet wird. \implies **Lösung:** Eintrag in die Datei ~/.profile

Beispiel zu Umgebungsvariablen

- Die Umgebungsvariable PATH enthält vier Pfade:

```
user@server:~$ echo $PATH  
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11
```

- Wie kann einfach und schnell der Pfad /usr/games hinzugefügt werden?
- Lösung:

```
user@server:~$ export PATH=$PATH:/usr/games  
user@server:~$ echo $PATH  
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games
```

- **Wichtig:**

Umgebungsvariablen anlegen \implies Ohne \$ vor dem Namen.

Auf Umgebungsvariablen zugreifen \implies Mit \$ vor dem Namen.

Nächste Übung:
16.11.2007