3.Übung

Systemsoftware (SYS)

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

Hochschule Mannheim – Fakultät für Informatik
Institut für Robotik

19.10.2007

Wiederholung vom letzten Mal

- Die Eingabeaufforderung (Prompt)
- Kommandos (Aufbau)
- Linux/UNIX-Verzeichnisstruktur
- Aktuelles Verzeichnis ausgeben (`pwd`) und wechseln (`cd`)
- Verzeichnisse anlegen (`mkdir`) und löschen (`rmdir`)
- Inhalte von Verzeichnissen ausgeben (`ls`)
- Das Hilfesystem von Linux: Die Manualseiten (`man`)
- Verschiedene Dateiarten unter Linux
- Leere Dateien anlegen (`touch`)
- Dateien ausgeben und verknüpfen (`cat`)
- Dateien rückwärts ausgeben und verknüpfen (`tac`)
- Inhalte von Dateien anzeigen (`more` und `less`)
- Anfang (`head`) und Ende (`tail`) von Dateien anzeigen
- Dateien kopieren (`cp`), verschieben/umbennen (`mv`) und löschen (`rm`)

Heute

- Einführung für Linux/UNIX-Anwender (Teil 2)
  - Dateirechte ändern (`chmod`)
  - Ändern des Passworts (`passwd`)
  - Shell beenden bzw. Benutzer abmelden (`exit`)
  - System neu starten oder herunterfahren (`halt`, `reboot`, `shutdown`)
  - Benutzeraccounts anlegen (`useradd`)
  - Benutzeraccounts löschen (`userdel`)
  - Benutzeraccounts ändern (`usermod`)
  - Gruppen anzeigen (`groups`)
  - Gruppen löschen (`groupdel`)
  - Gruppen anlegen (`groupadd`)
  - Gruppen ändern (`groupmod`)
  - Eigentümer und Gruppenzugehörigkeit ändern (`chown`, `chgrp`)

Dateirechte (1)

Dateityp	Besitzer			Gruppe			Andere		
-/d/l	r	w	x	r	w	x	r	w	x
	4	2	1	4	2	1	4	2	1

- Dateitypen:
  - `-` $\Longrightarrow$ Datei
  - `d` $\Longrightarrow$ Verzeichnis
  - `l` $\Longrightarrow$ symbolischer Link (Verweis)
  - `b` $\Longrightarrow$ Blockorientiertes Gerät (Device)
  - `c` $\Longrightarrow$ Zeichenorientiertes Gerät
  - `p` $\Longrightarrow$ FIFO-Datei (named pipe)
  - `s` $\Longrightarrow$ UNIX domain socket
- Rechtebits:
  - `r` $\Longrightarrow$ lesender Zugriff erlaubt
  - `w` $\Longrightarrow$ schreibender Zugriff erlaubt
  - `x` $\Longrightarrow$ Datei darf ausgeführt werden

- Kommando zum Ändern der Dateirechte $\Longrightarrow$ `chmod`

Dateirechte (2)

- Beispiele:

bel. Ausgangslage	$\Longrightarrow$	`chmod 000 <dateiname>`	$\Longrightarrow$	`----------`
bel. Ausgangslage	$\Longrightarrow$	`chmod 644 <dateiname>`	$\Longrightarrow$	`-rw-r--r--`
bel. Ausgangslage	$\Longrightarrow$	`chmod 666 <dateiname>`	$\Longrightarrow$	`-rw-rw-rw-`
bel. Ausgangslage	$\Longrightarrow$	`chmod 744 <dateiname>`	$\Longrightarrow$	`-rwxr--r--`
bel. Ausgangslage	$\Longrightarrow$	`chmod 755 <dateiname>`	$\Longrightarrow$	`-rwxr-xr-x`
bel. Ausgangslage	$\Longrightarrow$	`chmod 777 <dateiname>`	$\Longrightarrow$	`-rwxrwxrwx`
`----------`	$\Longrightarrow$	`chmod a+r <dateiname>`	$\Longrightarrow$	`-r--r--r--`
`----------`	$\Longrightarrow$	`chmod a+rwx <dateiname>`	$\Longrightarrow$	`-rwxrwxrwx`
`----------`	$\Longrightarrow$	`chmod u+rwx <dateiname>`	$\Longrightarrow$	`-rwx------`
`----------`	$\Longrightarrow$	`chmod g+rwx <dateiname>`	$\Longrightarrow$	`----rwx---`
`----------`	$\Longrightarrow$	`chmod o+rwx <dateiname>`	$\Longrightarrow$	`-------rwx`
`-rwxrwxrwx`	$\Longrightarrow$	`chmod a-rwx <dateiname>`	$\Longrightarrow$	`----------`

Spezielle Zugriffsrechte – Das t-Bit (1)

- Zu den speziellen Zugriffsrechten gehört das **t-Bit**, das auch als **Sticky Bit** bezeichnet wird.

```
user@rechner:~$ ls -ld /tmp/
drwxrwxrwt 18 root root 1080 2007-04-24 09:16 /tmp/
```

- Das t-Bit steht für **save program text on swap device** und kann auf Dateien und Verzeichnisse angewendet werden.
- Das t-Bit bewirkt, dass alle Daten, die in dieses Verzeichnis geschrieben werden sollen, so lange wie möglich im Hauptspeicher oder in der Swap-Partition existieren sollen.
- Die Daten in diesem Verzeichnis sollen nach Möglichkeit erst beim Herunterfahren des Systems tatsächlich auf die Festplatte zurückgeschrieben werden.

Spezielle Zugriffsrechte – Das t-Bit (2)

- Besonders bei kurzlebigen (temporären) Daten kann das zu Performancesteigerungen führen, da zeitaufwändige Schreiboperationen vermieden werden.
- In Verzeichnissen mit dem t-Bit darf ein Benutzer immer nur seine eigenen Daten umbenennen oder löschen, aber nie die Daten eines anderen Benutzers.

Spezielle Zugriffsrechte – Das s-Bit (1)

- Das **s-Bit** steht für **set user or group ID on execution**.

```
user@rechner:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 28480 2006-08-04 23:42 /usr/bin/passwd
```

- Das s-Bit bewirkt, dass ein Programm beim Aufruf mit den Rechten des Eigentümers gestartet wird.
- Achtung! Das s-Bit unüberlegt einsetzen, kann extrem gefährlich sein.
- Wenn ein Programm dem Systemadministrator `root` gehört, hat das Programm bei seiner Ausführung hier auch immer die Rechte des Systemadministrators.

Spezielle Zugriffsrechte – Das s-Bit (2)

- Ist das **s-Bit** nicht beim Besitzer, sondern bei der Gruppe gesetzt, steht es für **set group ID on execution**.
- Das s-Bit bei den Gruppenrechten bewirkt, dass ein Programm beim Aufruf mit den Rechten der Gruppe gestartet wird.
- Auch hier muss man vorsichtig sein, denn das s-Bit stellt auch bei den Gruppenrechten ein Sicherheitsrisiko dar.

Spezielle Zugriffsrechte ändern (1)

- Beispiele:

bel. Ausgangslage	$\Longrightarrow$	`chmod 1000 <dateiname>`	$\Longrightarrow$	`---------T`
bel. Ausgangslage	$\Longrightarrow$	`chmod 1666 <dateiname>`	$\Longrightarrow$	`-rw-rw-rwT`
bel. Ausgangslage	$\Longrightarrow$	`chmod 1777 <dateiname>`	$\Longrightarrow$	`-rwxrwxrwt`
bel. Ausgangslage	$\Longrightarrow$	`chmod 2000 <dateiname>`	$\Longrightarrow$	`------S---`
bel. Ausgangslage	$\Longrightarrow$	`chmod 2666 <dateiname>`	$\Longrightarrow$	`-rw-rwSrw-`
bel. Ausgangslage	$\Longrightarrow$	`chmod 2777 <dateiname>`	$\Longrightarrow$	`-rwxrwsrwx`
bel. Ausgangslage	$\Longrightarrow$	`chmod 4000 <dateiname>`	$\Longrightarrow$	`---S------`
bel. Ausgangslage	$\Longrightarrow$	`chmod 4666 <dateiname>`	$\Longrightarrow$	`-rwSrw-rw-`
bel. Ausgangslage	$\Longrightarrow$	`chmod 4777 <dateiname>`	$\Longrightarrow$	`-rwsrwxrwx`

Spezielle Zugriffsrechte ändern (2)

- Beispiele:

`----------`	$\Longrightarrow$	`chmod u+s <dateiname>`	$\Longrightarrow$	`---S------`
`-rw-rw-rw-`	$\Longrightarrow$	`chmod u+s <dateiname>`	$\Longrightarrow$	`-rwSrw-rw-`
`-rwxrwsrwx`	$\Longrightarrow$	`chmod u+s <dateiname>`	$\Longrightarrow$	`-rwsrwxrwx`
`----------`	$\Longrightarrow$	`chmod g+s <dateiname>`	$\Longrightarrow$	`------S---`
`-rw-rw-rw-`	$\Longrightarrow$	`chmod g+s <dateiname>`	$\Longrightarrow$	`-rw-rwSrw-`
`-rwxrwxrwx`	$\Longrightarrow$	`chmod g+s <dateiname>`	$\Longrightarrow$	`-rwxrwsrwx`
`----------`	$\Longrightarrow$	`chmod o+t <dateiname>`	$\Longrightarrow$	`---------T`
`-rw-rw-rw-`	$\Longrightarrow$	`chmod o+t <dateiname>`	$\Longrightarrow$	`-rw-rw-rwT`
`-rwxrwxrwx`	$\Longrightarrow$	`chmod o+t <dateiname>`	$\Longrightarrow$	`-rwxrwxrwt`

Passwort ändern – passwd

- Mit passwd lässt sich das **eigene** Benutzerpasswort ändern.

```
$ passwd
Changing password for user
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: Kennwort erfolgreich geändert
```

- Nur der Superuser (root) kann die Passwörter **aller** Benutzer ändern und muss dafür nicht das alte Passwort eingeben.

```
# passwd <Benutzername>
Enter new UNIX password:
Retype new UNIX password:
passwd: Kennwort erfolgreich geändert
```

Shell beenden – `exit`

- Das Kommando `exit` beendet die aktuelle Shell bzw. meldet einen Benutzer ab.
- Akternativ: `Strg-D`

```
user@server:/tmp$ su
Password:

server:/tmp# whoami
root

server:/tmp# exit
exit

user@server:/tmp$
```

Das System neu starten und herunterfahren

- `halt` fährt das System herunter und schaltet den Rechner aus.
- `reboot` fährt das System herunter und startet den Rechner neu.
- `shutdown` fährt das System auf unterschiedliche Arten herunter.
  Beispiele:
  - Das System sofort herunterfahren und danach den Rechner neu starten:
    ```
    # shutdown -r now
    ```
  - Das System sofort herunterfahren und danach den Rechner ausschalten:
    ```
    # shutdown -h now
    ```
  - Das System in 5 Minuten herunterfahren und danach den Rechner ausschalten:
    ```
    # shutdown -h +5
    ```
  - Das System um 20:45 herunterfahren und danach den Rechner ausschalten:
    ```
    # shutdown -h 20:45
    ```
- Diese Kommandos können nur vom Superuser (root) ausgeführt werden.

Benutzeraccounts anlegen – useradd

useradd *[Option] ... Benutzer*

- Das Kommando useradd richtet ein neues Benutzerkonto ein und nimmt die notwendigen Einträge in den Systemdateien vor.
- Nur der Superuser (root) darf Benutzer anlegen.
- Mit der Option --home /home/geek wird für den neuen Benutzer automatisch ein Home-Verzeichnis /home/geek angelegt.

```
# adduser --home /home/geek geek
Lege Benutzer »geek« an ...
Lege neue Gruppe »geek« (1002) an ...
Lege neuen Benutzer »geek« (1002) mit Gruppe »geek« an ...
Erstelle Home-Verzeichnis »/home/geek« ...
Kopiere Dateien aus »/etc/skel« ...
...
```

Benutzeraccounts löschen – `userdel`

`userdel` *`[Option] Benutzer`*

- Das Kommando `userdel` löscht alle Systemeinträge eines Benutzers.
- Nur der Superuser (root) darf Benutzer löschen.
- Mit der Option `-r` wird auch das Home-Verzeichnis des Benutzers mit all seinen Dateien gelöscht.

```
# userdel -r geek
```

Benutzeraccounts ändern – `usermod`

`usermod` *`[Option] Benutzer`*

- Mit dem Kommando `usermod` können Benutzerkonten verändert werden.

`-d`	Home-Verzeichnis ändern.
`-c`	Kommentarfeld ändern.
`-l`	Login-Name ändern (Benutzer darf nicht angemeldet sein!).
`-g`	Haupt-Gruppe ändern.
`-G`	Zusätzliche Gruppenzugehörigkeiten hinzufügen oder entfernen.
`-s`	Login-Shell ändern.
`-u`	User Identification (UID) ändern (muss eindeutig sein!).

Gruppen anlegen, ändern und löschen

- Gruppen sind logische Ausdrücke einer Gliederung und fassen die Benutzer zu einem gemeinsamen Zweck zusammen.
- Jeder Benutzer kann in mehreren Gruppen Mitglied sein.
- Gruppen helfen, die Zugriffsrechte besser zu verteilen.
- Alle Benutzer einer Gruppe können Dateien, die der Gruppe gehören, lesen, schreiben und ausführen.
- Die Gruppen stehen in der Datei `/etc/group`

`groups`	Zeigt die Gruppenzugehörigkeiten des Benutzers an.
`groupadd`	Eine neue Gruppe anlegen (nur als root).
`groupdel`	Eine bestehende Gruppe löschen (nur als root).
`groupmod`	Name oder Gruppen-ID ändern (nur als root).

Eigentümer und Gruppe ändern – `chown` und `chgrp`

`chown` *`[Option] ... neuerBesitzer Datei ...`*
`chgrp` *`[Option] ... neueGruppe Datei ...`*

- Um den Eigentümer oder die Gruppenzugehörigkeit zu ändern, gibt es die Kommandos `chown` (*CHange OWNner*) und `chgrp` (*CHange GRouP*).
- Den Eigentümer einer Datei kann nur der Superuser (root) ändern.
- Die Gruppenzugehörigkeit dann der Eigentümer der Datei wechseln.

`-c`	Gibt Informationen über die veränderten Dateien aus.
`-f`	Unterdrückt die Ausgabe von Fehlermeldungen.
`-v`	Gibt Informationen über die Dateien, auf die das Kommando versucht zuzugreifen.
`-R`	Bezieht auch die Unterverzeichnisse mit ein. Arbeitet rekursiv.

Nächste Übung:

26.10.2007