R|R|Z|N

Leibniz Universität Hannover

# Unix-Dateirechte

Mark Heisterkamp

heisterkamp@rrzn.uni-hannover.de

18. Juni 2008

# Nutzer und Gruppen

- alle Nutzer sind in Gruppen organisiert
- Gruppen und Nutzer werden abgekürzt:
  - u Eigentümer
  - g Gruppe
  - o andere
  - a alle
- Dateien und Verzeichnisse gehören genau einem Nutzer
- Dateien und Verzeichnisse sind genau einer Gruppe zugeordnet

# Hinzufügen von Nutzern

```
adduser [Optionen] <Name>
```

einige Optionen:

| | |
|---|---|
| `-h` | Kurze Hilfe |
| `--system` | Systembenutzer anlegen (UID=100–999) |
| `--no-create-home` | Kein Heimatverzeichnis anlegen |
| `--home <DIR>` | `DIR` als Heimatverzeichnis anlegen |
| `--uid <UID>` | `UID` als UID setzen |

Konfigurationsdatei: `/etc/adduser.conf`

Das Heimatverzeichnis wird mit den Daten aus `/etc/skel` gefüllt.

# Entfernen von Nutzern

```
deluser [Optionen] <Name>
```

Optionen:

| | |
|---|---|
| `-h` | Hilfe |
| `--system` | Nur Systembenutzer löschen |
| `--remove-home` | Heimat- und Mailspoolverzeichnis löschen |
| `--remove-all-files` | Alle Daten löschen, die dem Nutzer gehören |
| `--backup` | Nutzerdaten nach `/Name.tar.gz\|bz2` sichern |

Konfigurationsdatei: `/etc/deluser.conf`

# Welche Nutzer sind auf dem System?

```
/etc/passwd
```

enthält sieben durch ':' getrennte Felder:

1. Nutzername
2. Passwort oder `x`
3. UID
4. GID
5. Kommentar
6. Heimatverzeichnis
7. Login-Shell

`x` an zweiter Stelle bedeutet, dass das Passwort in der Datei `/etc/shadow` verschlüsselt gespeichert wird.

```
root:x:0:0:root:/root:/bin/bash
```

# Passwortverschlüsselung – /etc/shadow

`/etc/shadow`

enthält neun durch ':' getrennte Felder:

1. Nutzername
2. verschlüsseltes Passwort
3. Tage nach dem 1.1.1970 bis zur letzten Passwortänderung
4. Mindestalter des Passwortes in Tagen für eine Passwortänderung
5. Höchstalter des Passwortes
6. Tage vor Ablauf des Passwortes mit Warnung des Nutzers
7. Tage nach Ablauf des Paswortes, nach denen der Account gesperrt wird
8. Tage nach dem 1.1.1970 bis zur Sperrung des Accounts
9. reserviert

...

# Passwortverschlüsselung – /etc/shadow

...

Name und Passwort müssen eingetragen sein, alle anderen Felder können frei bleiben.

Ist das Mindestalter des Passwortes größer als das Höchstalter, so kann der Nutzer sein Passwort nicht ändern.

```
gdm:*:14047:0:99999:7:::
sshd:*:14047:0:99999:7:::
messagebus:*:14047:0:99999:7:::
avahi:*:14047:0:99999:7:::
polkituser:*:14047:0:99999:7:::
haldaemon:*:14047:0:99999:7:::
rrzn:$1$MCc0tzJ6$AvLvxkGoktqwjqvX.ftY20:14047:0:99999:7:::
```

# Passwortverschlüsselung

**Standard** DES-Verschlüsselung, d.h. maximale Passwortlänge ist acht Zeichen.

**erweitert** Beginnt die Passwortsequenz mit '`$1$`', so ist das Passwort MD5-verschlüsselt und kann 255 Zeichen lang sein.

**crypt** Die `crypt`-Bibliothek beschreibt die genaue Syntax der Passwort-Zeichenkette und deren Interpretation bzgl. des Verschlüsselungsverfahrens.

# Wer ist eingeloggt?

```
who -H
```

zum Beispiel:

```
NAME  LINE   TIME              COMMENT
rrzn  tty7   2008-06-17 14:32 (:0)
rrzn  pts/0  2008-06-17 14:32 (:0.0)
rrzn  pts/1  2008-06-17 18:08 (kursserv.rrzn.uni-hannover.de)
rrzn  pts/2  2008-06-17 18:44 (kursserv.rrzn.uni-hannover.de)
rrzn  pts/3  2008-06-17 19:13 (kursserv.rrzn.uni-hannover.de)
```

Der Nutzer `rrzn` ist auf dem Terminaltype 7 (`tty7`, X-Windows) seit der entsprechenden Zeit (`TIME`) eingeloggt. `(:0)` ist die Display-Variable.

`pts` sind sogenannte Pseudoterminals.

Der Nutzer `rrzn` ist dreimal vom Rechner `kursserv` eingeloggt (per Shell).

# Wer war zuletzt eingeloggt? (I)

```
last [Optionen]
```

Optionen:

`-t YYYYMMDDHHMMSS` alle Logins seit

- `YYYY` Jahr vierstellig
- `MM` Monat zweistellig
- `DD` Tag zweistellig
- `HH` Stunde zweistellig
- `MM` Minute zweistellig
- `SS` Sekunde zweistellig

`-x` Zeige auch alle Runlevel-Wechsel an.

# Wer war zuletzt eingeloggt? (II)

Ausgabe von `last -x`:

```
rrzn     pts/1        kursserv.rrzn.un Tue Jun 17 18:08   still logged in
rrzn     pts/2        :0.0             Tue Jun 17 14:39 - 14:41  (00:01)
rrzn     pts/1        localhost        Tue Jun 17 14:33 - 14:33  (00:00)
rrzn     pts/0        :0.0             Tue Jun 17 14:32   still logged in
rrzn     tty7         :0               Tue Jun 17 14:32   still logged in
runlevel (to lvl 2)   2.6.24-19-generi Tue Jun 17 16:31 - 20:05  (03:33)
reboot   system boot  2.6.24-19-generi Tue Jun 17 16:31 - 20:05  (03:33)
```

# Hinzufügen von Gruppen

```
addgroup [Optionen] <Group>
```

Optionen:

| | |
|---|---|
| `-h` | Hilfe |
| `--system` | Systemgruppe anlegen (GID=100–999) |
| `--gid <GID>` | `GID` als GID setzen |

Konfigurationsdatei: `/etc/adduser.conf`

# Entfernen von Gruppen

```
delgroup [Optionen] <Group>
```

Optionen:

| | |
|---|---|
| `-h` | Hilfe |
| `--system` | Nur Systemgruppen löschen |
| `--only-if-empty` | Nur leere Gruppen löschen |

Konfigurationsdatei: `/etc/deluser.conf`

# Welche Gruppen sind auf dem System?

```
/etc/group
```

enthält alle Gruppen, ihre GID und die ihnen zugeordneten Nutzernamen:

```
floppy:x:25:rrzn
tape:x:26:
sudo:x:27:
audio:x:29:rrzn
dip:x:30:rrzn
www-data:x:33:
```

Das '`x`' steht für kein Passwortschutz.

# Nutzer einer Gruppe hinzufügen / entfernen

```
adduser <Name> <Gruppe>
```

```
deluser <Name> <Gruppe>
```

# Welchen Gruppen gehört ein Nutzer an?

```
groups [Optionen] [Name]
```

Wird der `Name` nicht angegeben, so wird der Name des aktuellen Nutzers genommen.

Optionen:

`--help` Hilfe

`--version` Versionsanzeige

# Zugriffsrechte für Dateien und Verzeichnisse

`r` für `read` also Leserecht.

`w` für `write` also Schreibrecht.

`x` für `execute` also Ausführungsrecht.

Außerdem gelten noch die Abkürzungen:

`d` für `directory` also Verzeichnis.

`l` für `link` also Verweis bzw. Verknüpfung.

`t/T` für `Sticky Bit`

`s/S` für `Set Group ID` (SGID) oder `Set User ID` (SUID)

# Anzeigen der Zugriffsrechte

Eingabe von

```
ls -l
```

erzeugt die Ausgabe:

```
drwxr-x--x 2 rrzn rrzn 4096 2008-06-18 06:13 dir
-rwxr----- 1 rrzn rrzn    0 2008-06-18 06:13 file
```

Die ersten 10 Zeichen geben Auskunft über die Zugriffsrechte. Die erste Stelle steht für

- `-` Datei,
- `d` Verzeichnis oder
- `l` Link.

Die jeweils nächsten drei Stellen repräsentieren die Benutzergruppen `user`, `group` und `other`.

# Sticky Bit (t-Bit)

- spezielles Ausführungsrecht
- steht an letzter Stelle der Zugriffsrechte
- `x` wird durch `t` ersetzt
- `-` wird durch `T` ersetzt
- Programme mit Sticky Bit bleiben auch nach Beendigung im Speicher
  ➔ beschleunigter Start
- Für Verzeichnisse erlaubt das t-Bit Schreib- und Löschoperationen innerhalb des Verzeichnisses nur noch für Eigentümer (wichtig z.B. im `/tmp`-Verzeichnis). Es muss dann für `all` bzw. `other` gesetzt werden.

# SGID und SUID (s-Bit)

- spezielles Ausführungsrecht
- Ersetzt das `x` für Eigentümer (SUID) oder Gruppe (SGID)
- die entsprechende Datei wird mit den Rechten des Eigentümers oder der Gruppe ausgeführt

## Achtung: Root-Dateien mit s-Bit sind gefährlich!

Für Verzeichnisse:

1. s-Bit (SGID) für das Verzeichnis setzen
2. Alle Dateien/Verzeichnisse, die in diesem Verzeichnis angelegt werden, gehören der Gruppe des Verzeichnisses.

Ist Die Datei bzw. das Verzeichnis eigentlich nicht ausführbar, so erscheint ein `S` statt `s`.

# Ändern der Zugriffsrechte

Die Zugriffsrechte kann ausser `root` nur der Eigentümer ändern.

Der Befehl

```
chmod
```

ändert die Zugriffsrechte. Der Befehl

```
chown
```

ändert Eigentümer und Gruppe.

# chmod-Syntax

```
chmod <usergroup><+-><rwx> <Datei/Verzeichnis>
```

Beispielsweise würde die Eingabe von

```
chmod go+x file
```

die Ausgabe des vorangegangenen Beispiels ändern in:

```
-rwxr-x--x 1 rrzn rrzn    0 2008-06-18 06:13 file
```

Alle weiteren Benutzer außer des Eigentümers dürfen nun also auch die Datei `file` ausführen.

# Oktale Darstellung bei `chmod`

Es gibt noch eine weitere Syntax für `chmod`:

```
chmod ??? <Datei/Verzeichnis>
```

Dabei werden die Fragezeichen jeweils durch eine Ziffer von 0-8 ersetzt.

| | user | | | group | | | other | | |
|---|---|---|---|---|---|---|---|---|---|
| | r | w | x | r | w | x | r | w | x |
| konventionelle Darstellung: | r | w | x | r | - | x | - | - | x |
| Binärdarstellung: | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| Oktaldarstellung: | 7 | | | 5 | | | 1 | | |

# Globale Einstellung mittels `umask`

Die Syntax für `umask` lautet:

```
umask [xxx]
```

wobei `xxx` wieder die Oktaldarstellung ist. Duch `umask` werden die Rechte eingeschränkt, also abgezogen. Die gewählten Zugriffsrechte gelten ab dem Aufruf von `umask` für alle neu erstellten Dateien bis zum Ende der Sitzung.

Der Aufruf

```
umask 077
```

entfernt alle Rechte der Benutzegruppen `group` und `other`, so dass alle Rechte (`rwx`) für den Eigent ümer (`user`) gesetzt werden.

# Vierstellige Rechtedarstellung (oktal)

Es kann den drei genannten eine vierte Zahl vorangestellt werden:

1. `t`-Bit
2. `s`-Bit für die Gruppe
4. `s`-Bit für den Eigentümer

# chown-Syntax

```
chown <user:group> <Datei/Verzeichnis>
```

dabei sind die Synonyme für `user` und `group` abhängig von den Vorgaben innerhalb des jeweiligen Systems. Die Administratoren können für diese Gruppen beliebige Namen vergeben.

Dieser Befehl ist in der Regel ausschließlich `root` vorbehalten

# Access Control Lists (ACL)

Der Zugriff auf Verzeichnisse und Dateien wird durch ACLs geregelt.

Es gibt zwei Typen von ACLs:

| | |
|---|---|
| minimal | enthält die normalen Rechte für `user`, `group` und `other` (entpricht dem Unix-Standard ohne ACL) |
| erweitert | hat einen `mask`-Eintrag und darf `named user` und `named groups` enthalten |

# Inhalt einer ACL

Eine ACL enthält folgende Einträge:

| | |
|---|---|
| `user` | Standard-Eigentümer |
| `named user` | Zusätzliche Nutzer, denen Zugriffsrechte eingeräumt werden |
| `group` | Standard-Gruppe des Eigentümers |
| `named group` | Zusätzliche Gruppen, denen Zugriffsrechte eingeräumt werden |
| `mask` | Rechtefilter |
| `other` | Standard-Rest |

# Rechtefilter `mask`

- wird automatisch erzeugt
  (abschaltbar mit dem Schalter `-n`)
- setzt die maximal Zugriffsrechte für die group class:
  - group
  - named user
  - named group
- werden Rechte durch die Maske verändert, werden die effektiven Rechte angezeigt.

# Installation von ACLs unter Debian

```
aptitude install acl
```

Für das Dateisystem muss die Mount-Option `acl` gesetzt sein!

Es stehen danach zwei neue Befehle zur Verfügung:

- `setfacl` Modifizieren von ACLs
- `getfacl` Abrufen von ACLs

# setfacl

```
setfacl <-m|-x> <acl> <Datei|Verzeichnis>
```

Optionen:

`-m` Modifizieren von Zugriffsrechten

`-x` Entfernen von Zugriffsrechten

ACL-Syntax:

```
u(ser):uid:perms
```

(bei leerer `uid` werden die Rechte des Eigentümers gesetzt.)

```
g(roup):gid:perms
```

(bei leerer `gid` werden die Rechte der Standard-Gruppe gesetzt.)

```
m(ask)::perms
```

```
o(ther)::perms
```

# Maske mit `chmod` setzen

Vertändert man die Gruppenrechte mittels `chmod`, so wird die `mask` entsprechend gesetzt.

# Anzeigen der ACL – `ls`

```
ls -l
```

zeigt durch ein '+' beim Ausführungsrecht für `other` an, dass für die Datei bzw. das Verzeichnis eine erweiterte ACL gilt:

```
-rw-r--r--  1 rrzn rrzn    0 2008-06-18 05:27 datei
-rw-rwxr--+ 1 rrzn rrzn    0 2008-06-18 05:27 datei-acl
drwxr-xr-x  2 rrzn rrzn 4096 2008-06-18 05:27 dir
drwxr-xr-x+ 2 rrzn rrzn 4096 2008-06-18 05:27 dir-acl
```

# Anzeigen der ACL – getfacl

```
getfacl <Datei|Verzeichnis>
```

ergibt:

```
# file: datei-acl
# owner: rrzn
# group: rrzn
user::rw-
user:rrzn2:rwx
group::r--
mask::rwx
other::r--
```

# Löschen von ACL-Einträgen

```
setfacl -x <u|g:Name> <Datei|Verzechnis>
```

Eine Regel wird entsprechend dem named user- oder named group-Attribut gelöscht.

Alle Regeln können mit dem Schalter `-b` gelöscht werden:

```
setfacl -b <Datei|Verzeichnis>
```

# Default-ACL bei Verzeichnissen

Bei Verzeichnissen kann man eine Vererbung aktivieren, die alle Zugriffsrechte auf Unterverzeichnisse und Dateien anwendet. Eine solche Vererbung wird durch die sogenannte Default-ACL erreicht.

- Dateien erben die Zugriffsliste
- Unterverzeichnisse erben die Default-ACL und die Zugriffsliste

```
setfacl -d -m <u|g:u|gid:perm> <Verzeichnis>
```

Setzt die Default-ACL.

Alle nicht gesetzten Rechte werden automatisch ergänzt.

# Anzeige einer Default-ACL

```
# file: dir-acl
# owner: rrzn
# group: rrzn
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:rrzn2:rw-
default:group::r-x
default:mask::rwx
default:other::r-x
```

# Abarbeitung einer ACL

Eine ACL wird in folgender Reihenfolge abgearbeitet:

1. owner
2. named user
3. owning group
4. named group
5. other

Trifft kein Recht zu, wird der Zugriff verweigert.

# Anwendungen und ACL

Unterstützung für ACLs:

- `mv`, `cp`, `ls` ...
- `star`
- `samba`

Andere Anwendungen können problematisch sein.