

Vorlesung Unix-Praktikum

3. Prozesse und Zugriffsrechte

Dirk Frettlöh

Folien: DF und Carsten Gnörlich
Technische Fakultät
Uni Bielefeld

Willkommen zur vierten Vorlesung

Was gab es beim letzten Mal?

- ▶ Ein-/Ausgabe umleiten (<, >, >>)
- ▶ Ein-/Ausgabe verketteten (|)
- ▶ Befehle verketteten (; ())

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Willkommen zur vierten Vorlesung

Was machen wir heute?

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Vorder-/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Vorder- und Hintergrund

Kommandos zum Bearbeiten von Textdateien

Abkoppeln von der Kommandozeile

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Normale Kommandozeilen-Programme

- ▶ benötigen Kommandozeilenfenster für Ein-/Ausgabe
- ▶ Beispiel: `bc` blockiert das Fenster solange es läuft

Graphische Programme haben eigene Fenster

- `$ xclock`
 - ▶ blockiert Kommandozeile bis `xclock` beendet wird
- `$ xclock &`
 - ▶ “Abkoppeln” von der Kommandozeile
 - ▶ Kommandozeile kann weiter benutzt werden während `xclock` läuft

Kommandos zum Bearbeiten von Textdateien

Nachträgliches Abkoppeln von der Kommandozeile

Falls ein graphisches Programm bereits aufgerufen worden ist:

```
$ xclock
```

```
^Z
```

```
$ bg
```

Erklärung:

- ▶ Strg-z koppelt das Programm ab und hält es an
- ▶ bg lässt das Programm im Hintergrund (abgekoppelt) weiter laufen
- ▶ fg würde das Programm wieder angekoppelt laufen lassen

Unix-

Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Mehr zu...

Dateiverwaltung

Dateiverwaltung

Freien Speicherplatz feststellen

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

df (freien Plattenplatz anzeigen)

```
$ cd
```

```
$ df -h .
```

```
Filesystem                Size  Used Avail Use% Mounted on
fileserver.rbg.TechFak.Uni-Bielefeld.DE:/export/rbg/homes/df
                                                                    591G  294G  297G   50% /homes/df
```

```
$ cd /tmp
```

```
$ df -h .
```

```
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/system-tmp
                                                                    1008M   34M  924M    4% /tmp
```

- ▶ Verfügbarer Speicher hängt vom Gerät unter dem Verzeichnis (genauer: mount point) ab!

Dateiverwaltung

Alle Nutzer teilen sich freien Speicherplatz

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

```
$ df -h .
```

```
Filesystem          Size  Used Avail Use% Mounted on
fileserver.rbg.TechFak.Uni-Bielefeld.DE:/export/rbg/homes/df
                    591G  294G  297G  50% /homes/df
```

Gesamtplatz (z.B. 297G) steht allen Nutzern zur Verfügung

- ▶ bitte Rücksicht nehmen!
- ▶ max. Verbrauch pro Nutzer begrenzt (z.Zt. 25G)
(G heißt hier Gib, also $2^{30} = 1024^3$ Byte,
und nicht GB = 1000^3 Byte)

Dateiverwaltung

Eigener Speicherverbrauch

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

du (disc usage)

```
$ cd          # wechselt ins Home-Verzeichnis
```

```
$ du -hs .   # mein gesamter Speicherverbrauch
```

```
198M .
```

```
$ du -hs *   # Verbrauch in Unterverzeichnissen
```

```
6.5M  Vorlesung01
```

```
1.8M  Vorlesung02
```

```
2.7M  Vorlesung03
```

... weitere Dateien und Unterverzeichnisse ...

Dateiverwaltung

Einzelne Dateien komprimieren

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Komprimieren

```
$ gzip datei.txt → datei.txt.gz
```

▶ Keine Sorge: verlustlose Kompression!

Auspacken

```
$ gunzip datei.txt.gz → datei.txt
```

Dateiverwaltung

Mehrere Dateien komprimieren

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

```
$ tar czf archiv.tar.gz *.txt Fotos
```

- ▶ packt Dateien und Verzeichnisse in ein Archiv

```
$ tar tvzf archiv.tar.gz
```

- ▶ zeigt Inhalt des Archivs

```
$ tar xzf archiv.tar.gz
```

- ▶ alles wieder auspacken

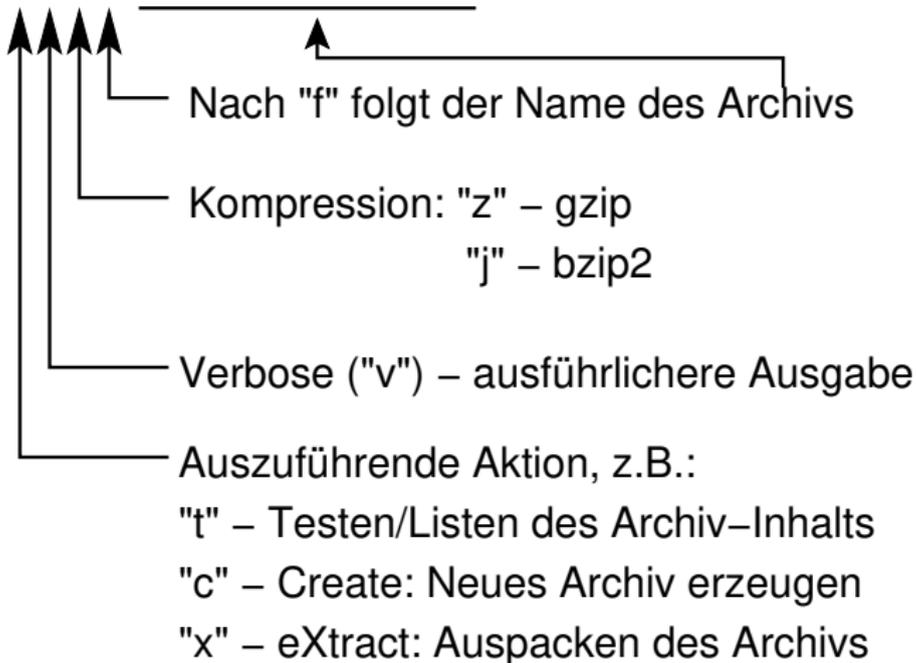
```
$ tar xzf archiv.tar.gz eins.txt
```

- ▶ nur die angegebenen Dateien wieder auspacken

Dateiverwaltung

Die Logik der tar-Kürzel

tar tvzf archiv.tar.gz ...



Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Dateiverwaltung

tar und Kompression

Warum die seltsame Dateiendung `.tar.gz`?

```
$ tar czf archiv.tar.gz *.txt Fotos
```

ist das gleiche wie:

```
$ tar cf archiv.tar *.txt Fotos
```

```
$ gzip archiv.tar
```

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Prozesse

Motivation

Ressourcenverteilung auf shell, compute, ...

shell, compute, etc. haben viele gleichzeitige Nutzer

- ▶ Wie verteilt man Ressourcen gerecht?
 - ▶ Rechenzeit
 - ▶ Speicherplatz
 - ▶ ...
- ▶ Wie schützt man Ressourcen?
 - ▶ wer darf auf meine Daten zugreifen und wer nicht?

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

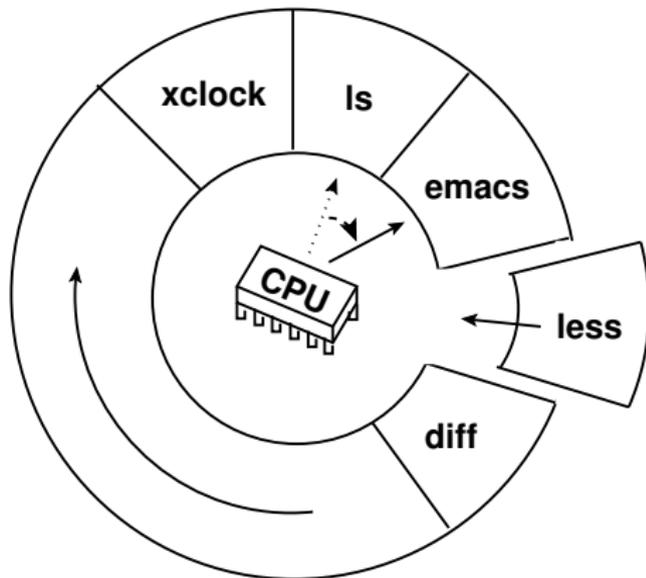
Überblick

Ausblick

Prozesse

Zeitscheibenprinzip

- ▶ ablaufende Programme = Prozesse
- ▶ Jeder Prozess wird reihum ein Stück abgearbeitet (Zeitscheibenprinzip)



Prozesse

Typische Operationen auf Prozessen

Unix-Praktikum

Dirk Frettlöh

Vorder-/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

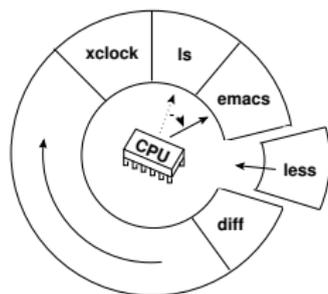
nice

Dateirechte

Berechtigungen

Überblick

Ausblick



- ▶ laufende Prozesse anzeigen
- ▶ (ungewünschte) Prozesse beenden
- ▶ Priorität bei Ausführung ändern

Prozesse

Prozesse auflisten

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse
anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

ps (process summary)

```
$ ps
```

zeigt alle Prozesse dieses Kommandozeilen-Fensters

```
$ ps -u frettlöe      (oder BSD-Stil: $ ps ux)
```

alle meine Prozesse

```
$ ps -e              (oder BSD-Stil: $ ps aux)
```

alle Prozesse (auch anderer Nutzer; Systemprozesse)

```
$ ps -f              oder $ ps aux
```

ausführlichere Anzeige (Parent, "Besitzer").

Status: R: running, S: sleeping, Z: zombie
(Dead, Terminated... gibt es, die sollte man aber nie sehen)

Prozesse

Hilfsprogramm: Zeilen/Worte zählen

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

wc (word count)

- ▶ `wc` zählt Zeilen, Worte und Bytes
- ▶ `wc -l` zählt nur die Zeilen

Beispiel:

```
# Anzahl der laufenden Prozesse  
$ ps aux | tail -n +2 | wc -l
```

```
# Anzahl meiner bash-Prozesse  
$ ps ux | grep bash | wc -l
```

Abbruch aus der Kommandozeile

- Strg-c oder Strg-d
- (bzw. Ctrl-c oder Ctrl-d auf US-Tastaturen)
- ▶ hilft aber nicht immer

kill (Prozess beenden)

- Prozess-ID (pid) mit Hilfe von ps ermitteln

\$ kill pid freundliche Version

\$ kill -9 pid unfreundliche Version

- Unix räumt vollständig auf - kein Neustart notwendig

Programme können weitere Programme starten

▶ typisches Beispiel: Die Shell (`bash`)

`$ ps -f` Prozesshierarchie der aktuellen Kommandozeile

`$ ps -efx` Prozesshierarchie systemweit

- `kill` des Prozessvaters beendet typischerweise die Kinder!
 - `kill` auf die `bash` oder das Fenstersystem löst Kettenreaktion aus
- ▶ kann gewünscht sein (oder auch nicht!)

Da große Programme viele Prozesse haben (Firefox, Office...):

`$ pkill firefox`

Beendet alles, worin „firefox“ vorkommt (Logik wie `grep`)

Prozesse

Zombie-Prozesse

Wie entstehen "Zombie"-Prozesse?

- Kind-Prozess wurde beendet
- ▶ Vater-Prozess hat das noch nicht quittiert
- ▶ erkennbar am "Z"-Flag bzw. <defunct>

Eigenschaften des Zombies:

- ▶ ist nur noch Eintrag in der Prozesstabelle (verbraucht keine Rechenzeit oder sonstigen Ressourcen)
- ▶ unkillbar, da schon tot
- ▶ verschwindet, wenn Vater-Prozess reagiert oder sich beendet

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Prozesse

Prozess-Waisen (Orphans)

Wie entstehen “Orphan”-Prozesse?

- Vater-Prozess hat “abgekoppeltes” Kind erzeugt
- Vater-Prozess ist danach (fehlerhaft) beendet worden
- ▶ Kind bekommt den Prozess mit pid 1 als neuen “Vater”
(init oder systemd oder launchd oder ...)

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Prozesse

Laufende Prozesse beobachten

top (zeigt Anfang der Prozess-Warteschlange)

\$ top

- ▶ Verlassen mit Taste q
- ▶ laufenden Prozess beenden mit Taste k,
dann pid und signal:
 - ▶ 15 für “freundlichen” Abbruch
 - ▶ 9 für “harten” Abbruch

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Prozesse

Ressourcen schonen

Unix ist ein Mehrbenutzersystem

- ▶ alle Benutzer konkurrieren um die Ressourcen
- ▶ gegenseitige Rücksichtnahme:
lange, nicht interaktive Prozesse → nice

nice (Prozess ressourcenschonend aufrufen)

```
$ nice gzip ganz_lange_datei  
was man normalerweise eingibt
```

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Dateirechte

...aka *permissions*

Datei- und Verzeichnisrechte

Übersicht

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Dateien verwalten

- ▶ vieles kennen wir schon: `pwd`, `ls`, `cd`, `cp`, `mv`, `rm`

Weitere typische Aufgaben

- ▶ Lese- / Schreibrechte verstehen
- ▶ ... und verwalten

Datei- und Verzeichnisrechte

Zugriffsrechte

3-stufiges System von Berechtigungen:

Besitzer (Ihr!) **Gruppe** **Alle (Vorsicht!)**

\$ls -ld

```
rw-rw-r-- df staff 1973 2019-11-03 17:12 brief.odt
rw-r----- df staff 8457 2019-10-25 11:03 pv.csv
rwxr-xr-x df staff 48 2019-08-10 09:57 ablage
```

Grundlegende Berechtigungen:

r	read	Öffnen / Lesen erlaubt
w	write	Schreibzugriff / Löschen erlaubt
x	execute	Dateien: Programmausführung erlaubt Verzeichnisse: Durchgreifen erlaubt

Datei- und Verzeichnisrechte

Zu welchen Gruppen gehöre ich?

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

groups

```
$ groups
```

```
bmstaff teachlinux vlvkinf tak ...
```

id (identity)

```
$ id
```

```
uid=22227(df) gid=12000(bmstaff) groups=...
```

- ▶ gid: primäre Gruppe

Wann bekommt man zusätzliche Gruppen?

- ▶ Maschinenbezogen (z.B. *audio* bei lokalem login an PCs)
- ▶ Statuswechsel (HiWi werden, Bachelorarbeit schreiben)

Datei- und Verzeichnisrechte

Berechtigungen ändern

chmod (change file mode)

\$ chmod g-w *datei*

kein Schreibzugriff für Gruppe

\$ chmod u+w *datei*

erlaube Schreibzugriff für sich selbst

\$ chmod o=r *datei*

erlaube nur Lesezugriff für alle

(w,x werden gelöscht)

\$ chmod go-rwx *.txt

für *.txt-Dateien alle Zugriffe

für Gruppe und alle wegnehmen

\$ chmod g=rw,o= *datei*

Gruppe darf lesen und schreiben,

andere haben keinen Zugriff

u : Berechtigung für **Besitzer** (user; erster **rw**-Block)

g : Berechtigung für **Gruppe** (group; zweiter **rw**-Block)

o : Berechtigung für **Alle** (other; dritter **rw**-Block)

Datei- und Verzeichnisrechte

Berechtigungen ändern

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

chown (change owner)

Im Prinzip so:

```
$ chown juser scan0003.pdf
chown: changing ownership of 'scan0003.pdf':
Operation not permitted
```

Das darf nur der Superuser (su)

```
$ sudo chown juser datei.txt
```

Datei- und Verzeichnisrechte

Prioritäten auf Dateiberechtigungen

Die speziellste anwendbare Berechtigung gilt:
(am Beispiel jeweils aus Sicht des Nutzers `df`)

Berechtigung	Nutzer	Gruppe	df darf lesen
<code>-r-----</code>	<code>df</code>	<code>staff</code>	<code>ja</code>
<code>---r--r--</code>	<code>df</code>	<code>staff</code>	<code>nein</code>
<code>----r--r--</code>	<code>juser</code>	<code>staff</code>	<code>ja</code>
<code>-----r--</code>	<code>juser</code>	<code>staff</code>	<code>nein</code>
<code>-----r--</code>	<code>nn</code>	<code>nn</code>	<code>ja</code>

(`df` sei Mitglied der Gruppe `staff`, aber nicht in `nn`)

Datei- und Verzeichnisrechte

Berechtigungen auf Verzeichnissen

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

'w'-Berechtigung auf Verzeichnis

- ▶ Anlegen von Dateien / Unterverzeichnissen
- ▶ Löschen von Dateien / Unterverzeichnissen

Zusammenspiel von Datei- und Verzeichnisberechtigungen

```
$ ls -ld
```

```
dr-xr-xr-x  4  df      staff  ...  .
```

```
drwxrwxr--  3  root    root   ...  ..
```

```
-rwx-r--r--  1  df      staff  ...  brief.txt
```

- ▶ `brief.txt` kann verändert werden (Dateiberechtigung)
- ▶ `brief.txt` kann *nicht* gelöscht werden (Verzeichnisber.)
- ▶ Neue Dateien können *nicht* angelegt werden (Verzeichnisberechtigung)

Datei- und Verzeichnisrechte

Berechtigungen auf Verzeichnissen

'w' auf Gruppenverzeichnis hebt Datei-Schreibschutz aus

```
drwxrwxr-x  4  df    projekt  ...  .
drwxrwxr--  3  root   root    ...  ..
-rw-r--r--  1  df    projekt  ...  brief.txt
```

Nutzer `nn` sei ebenfalls in der Gruppe `projekt`:

- ▶ `nn` kann `brief.txt` nicht editieren, aber
- ▶ `nn` kann `brief.txt` löschen und neu anlegen

Folgerung:

- ▶ *Niemals* das Home-Verzeichnis gruppen-/weltschreibbar machen!

Datei- und Verzeichnisrechte

Nerd-Variante

Bitweise Kodierung: Zahlen für gesetzte Berechtigungen addieren

$$\begin{array}{ccc} r & w & x \\ | & | & | \\ 4 & 2 & 1 \end{array} \longrightarrow \begin{array}{ccc} r & - & x \\ 4 & + & 0 & + & 1 & = & 5 \end{array}$$

Beispiel:

```
-rwxr-xr-- df staff 8457 2011-10-25 11:03 skript.sh
421401400
  ↓   ↓   ↓
  7   5   4
```

```
$ chmod 754 skript.sh
```

Erste Ziffer für User, zweite Ziffer für Group, dritte Ziffer für Others

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Datei- und Verzeichnisrechte

Berechtigungen auf Verzeichnissen

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

'r'-Berechtigung auf Verzeichnis

- ▶ erlaubt *Dateinamen* zu lesen (und sonst nichts!)

'x'-Berechtigung auf Verzeichnis

- ▶ erlaubt *Inhalt* von Dateien und Unterverzeichnissen zu lesen

Typischerweise: rx zusammen setzen oder wegnehmen

Datei- und Verzeichnisrechte

umask

Jeweils von Hand Rechte ändern ist lästig.

Rechte für neu angelegte Dateien voreinstellen mit `umask`.

$$\begin{array}{r} 777 \\ - 023 \\ \hline 754 \end{array}$$

Logik genau andersum:

`umask 023` bewirkt also für alle neuen Dateien 754:

`rwXr-Xr--`

(Lies "Maske" als Filter.)

Die meisten Programme erzeugen neue Dateien *ohne* das x-bit.
(Ausnahme z.B. Compiler, die sollen ja eine ausführbare Datei liefern)

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Datei- und Verzeichnisrechte

setuid und setgid

Auf aktuellen Linux-Rechnern sieht man oft so was:

drwsrwsr--

- ▶ Das User s: setuid führe solche Dateien mit den Rechten seines Besitzers aus (nicht mit denen des Users, der sie startet.)
- ▶ Das Groups s: setgid
 - ▶ Führe solche *Dateien* mit den Rechten seiner Gruppe aus (nicht mit der des Users, der sie startet.)
 - ▶ Neu angelegte Dateien in solchen *Ordnern* erben die Rechte dieses Ordners (nicht die der `umask` des Users, der sie erzeugt)

Großes S: das x-Bit ist nicht gesetzt, aber das setuid/setgid Bit schon. (mehr: 'sticky bit' suchen)

Etwas sinnlose Einstellung, da die Gruppe nicht in den Ordner kann.

Überblick

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

bg, fg, &, Strg-z
df, du
gzip, gunzip, tar

ps, jobs, top
wc
Strg-c, Strg-d, kill
ls -l
chmod, chown
umask

Prozesse in Vorder- bzw Hintergrund
Größe von Dateien bzw Festplatten
(De-)Komprimieren von Dateien
bzw Archiven
laufende Prozesse anzeigen
Zeichen/Worte/Zeilen zählen
Prozesse abbrechen
Dateirechte anzeigen
Dateirechte ändern
Dateirechte voreinstellen

Ausblick

Nächste Woche machen wir...

- ▶ Shell konfigurieren
- ▶ Aliase
- ▶ einfache Shellskripte

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Ende der heutigen Vorlesung

Unix-
Praktikum

Dirk Frettlöh

Vorder-
/Hintergrund

Dateiverwaltung

Prozesse

anzeigen

beenden

Hierarchie

nice

Dateirechte

Berechtigungen

Überblick

Ausblick

Vielen Dank fürs Zusehen!

Bis nächste Woche!