

Administration von Linux-PCs

Übersicht

- Einführung
Historisches, Shells, Systembefehle
- Installation
Yast, Kernelkonfiguration, Bootvorgang
- Administration
Benutzerverwaltung, Cron-Jobs, Backup
- Netzwerk
Grundlagen, NFS, NIS
- Server
Druck-, FTP-, WWW-, Samba-Server
- Sicherheit
ssh, Firewall / IP-Masquerading

1 Einführung

1.1 Historisches

Im August 1991 begann ein finnischer Student eine Mitteilung an die comp.os.minix-Newsgroup mit den Worten:

```
Hello everybody out there using minix -  
I'm doing a (free) operating system (just  
a hobby, won't be big and professional  
like gnu) for 386(486) AT clones.
```

Der Student war Linus Torvalds und aus seinem „Hobby“ entwickelte sich das, was wir heute als Linux kennen, ein vollwertiges POSIX-artiges Betriebssystem, entwickelt nicht von Linus Torvalds allein, sondern von hunderten von Programmierern überall auf der Welt.

2 UNIX-Shells

Aufgaben einer Shell sind u.a.:

- Kommandozeilen-Interpretation
- Programmausführung
- Dateinamen-Erweiterung
- Ein-/Ausgabeumlenkung
- Umgebungskontrolle

Verbreitet eingesetzt werden:

- Bourne-Shell (sh)
- C-Shell (csh) mit C-ähnlicher Script-Sprache
- Korn-Shell (ksh)
- Bourne-Again-Shell (bash) des GNU-Projekts mit Elementen aus csh und ksh
- Die tcsh als Erweiterung der C-Shell

Welche Shell beim Einloggen aufgerufen wird, ist in der Passwort-Datei durch den Systemadministrator festgelegt.

Je nach Shell werden verschiedene Konfigurationsdateien durchlaufen:

Shell	sh	bash	ksh	csch	tcsh
login	/etc/profile			/etc/csh.cshrc	
	~/.profile			~/.cshrc	
	—			/etc/csh.login	
	—	~/.bashrc	\$ENV	~/.login	
script	/etc/csh.cshrc			/etc/csh.cshrc	
	—	~/.bashrc	\$ENV	~/.cshrc	

Die Endung „rc“ steht für Run Command. Run-Command-Dateien werden bei jedem Start einer Shell abgearbeitet.

2.1 Programmausführung

- Ausführung eines Befehls durch die Shell in selbständigem Prozeß.
- Um Warten auf Beedingung des Prozesses zu vermeiden, starten im Hintergrund:

Befehl &

- Laufenden Prozeß in Hintergrund versetzen:

`Ctrl-Z`

`bg`

- Laufende Prozesse anzeigen:

`ps aux`

darin:

a Prozesse aller Benutzer

u Anzeige von Benutzer und Startzeit

x auch Prozesse ohne zugeordnetes Terminal

- Hintergrundprozeß in den Vordergrund bringen:

`fg PID`

- Starten eines Prozesses, so daß er beim Ausloggen nicht beendet wird:

nohup *Befehl*

Werden keine Ausgabedateien angegeben, so werden sowohl `stdout` als auch `stderr` in die Datei `nohup.out` umgeleitet.

`nohup` startet Prozesse stets mit etwas erniedrigter Priorität.

Bei Verwendung der `csch` wird durch das Starten eines Prozesses im Hintergrund bereits sichergestellt, daß er beim Logout nicht beendet wird. Dies gilt jedoch nicht für die `tcsh`!

- `bash` und `csch` führen Protokoll über die zuletzt eingegebenen Befehle (die Anzahl dieser Befehle kann für die `bash` in `/etc/profile` mit `HISTSIZE=n`, für die `csch` in `/etc/csch.cshrc` mit `set history=n` gesetzt werden).

Eine Liste dieser Befehle liefert das Kommando

history

Auf einen dieser Befehle kann wie folgt zurückgegriffen werden:

!!	führt den letzten Befehl erneut aus
!-n	führt den <i>n</i> -ten vorhergehenden Befehl aus
!n	führt den Befehl mit Nummer <i>n</i> aus
!Text	führt den letzten Befehl aus, der mit <i>Text</i> beginnt
!?Text?	führt den letzten Befehl aus, der <i>Text</i> enthält
^a^b	wiederholt den letzten Befehl und ersetzt dabei <i>a</i> durch <i>b</i>

2.2 Dateinamen-Erweiterungen und Ersetzungen

Die Shell führt folgende Dateinamen-Erweiterungen durch:

- * steht für eine beliebig lange Zeichenkette (auch kein Zeichen)
- ? steht für genau ein beliebiges Zeichen
- [...] steht für ein Zeichen aus der angegebenen Menge, z.B. bezeichnet `a[bi-n]z` eine Zeichenkette, die mit `a` beginnt, gefolgt von `b` oder einem der Zeichen zwischen `i` und `n`, und die mit `z` endet.
- {...} beschreibt mehrere Varianten, z.B.: liefert `test.{c,o}` \Rightarrow `test.c test.o`

Außerdem werden folgende Ersetzungsregeln verwendet:

`\` hebt die Wirkung eines nachfolgenden Sonderzeichens (`*` `\` `|` `&` `(` `)` `{` `}`) auf.

`$variable` interpretiert *variable* als Shellvariable und setzt ihren Wert textuell ein.

``Befehl`` setzt die Ausgabe von *Befehl* an der betreffenden Stelle textuell ein.

`"Text"` klammert eine als Einheit zu betrachtende Zeichenkette. Lediglich „\$“, „\“ und „`“ behalten ihre besondere Bedeutung.

`´Text´` Klammert eine als Einheit zu betrachtende Zeichenkette, in der keinerlei Ersetzungen vorgenommen werden sollen.

2.3 Ein-/Ausgabeumlenkung

Befehl < Datei

der Inhalt der *Datei* wird zur Standardeingabe von *Befehl*

Befehl > Datei

die Standardausgabe von *Befehl* wird in die angegebene *Datei* geschrieben. Ist die Variable `$noclobber` gesetzt, wird eine Fehlermeldung ausgegeben.

Befehl >> Datei

die Standardausgabe von *Befehl* wird an die genannte *Datei* angefügt.

Befehl > & Datei

Standardausgabe und Standardfehlerausgabe von *Befehl* werden in dieselbe *Datei* umgelenkt. (Unter der `bash` äquivalent zu: *Befehl > Datei 2>&1*).

Befehl >> & Datei

Standardausgabe und Standardfehlerausgabe von *Befehl* werden gleichermaßen an die genannte *Datei* angefügt. (Unter der `bash`: *Befehl >> Datei 2>&1*).

Befehl₁ | ***Befehl₂***

Die Standardausgabe von *Befehl₁* wird Standardeingabe von *Befehl₂*.

Unter der `csch` wird im Gegensatz zur `sh` ein im Hintergrund gestarteter Prozeß blockiert, sobald er von der Standardeingabe lesen will, und der Benutzer darüber informiert. Er kann dann den Prozeß mit `fg` in den Vordergrund holen.

2.4 bash-Shellprogrammierung

- Parameter und Variablen
- Ablaufsteuerung
- Fehlerbehandlung
- Shellfunktionen

Parameter

Sonderparameter

- \$0 Name der Prozedur
- \$# Anzahl angegebener Argumente
- \$* angegebene Argumente als *eine* Zeichenkette
- @ angegebene Argumente als \$# Zeichenketten
- \$? Returncode des vorangegangenen Befehls
- \$\$ die PID des aktuellen Prozesses

Bedingte Ersetzung von Parametern

$\${Parameter}$

falls *Parameter* definiert, *Parameter*, sonst leere Zeichenkette.

$\${Parameter-Argument}$

falls *Parameter* definiert, *Parameter*, sonst *Argument*.

$\${Parameter+Argument}$

falls *Parameter* definiert, *Argument*, sonst leer.

$\${Parameter=Argument}$

wie $\${Parameter-Argument}$, zusätzlich erhält *Parameter* den Wert *Argument*, falls *Parameter* nicht definiert war.

$\${Parameter?}$

falls *Parameter* nicht definiert, Abbruch.

$\${Parameter?Meldung}$

falls *Parameter* nicht definiert, Abbruch mit *Meldung*.

Ein `:` hinter *Parameter* behandelt eine leere Zeichenkette wie einen undefinierten *Parameter*.

Beispiel: Default-Parameter

```
#!/bin/sh
echo -n Anzahl der Einträge in ${1-$HOME}:
ls ${1-$HOME} | wc -w
```

Setzen positionaler Parameter (\$1, \$2,...)

```
set string1 string2 ...
```

Nur die positionalen Parameter 1-9 können unmittelbar angesprochen werden, die weiteren lassen sich über `shift` erreichen

```
shift [n]
```

Variablen

Variable=Text

definiert die *lokale Variable* mit *Text* als Wert (keine Leerzeichen!).

\$Variable

interpretiert *Variable* als Shellvariable und setzt ihren Wert textuell ein.

set zeigt die Werte aller lokalen Variablen an.

unset *Variable*

löscht die *lokale Variable*.

export *Variablen*

Sollen Variablen nicht nur in der aktuellen sondern auch in einer *untergeordneten Shell* zur Verfügung stehen, müssen sie exportiert werden. In eine Subshell werden standardmäßig lediglich kopiert:

- Parameterwerte
- Werte einiger Sondervariablen, z.B.: PATH, IFS, PS1, PS2
- Name des aktuellen Verzeichnisses
- umask-Wert

Eine Subshell kann einen Wert lediglich per `echo` an eine aufrufende Shell zurückliefern.

alias [*Kürzel*=[*Befehl*]]

Ordnet *Kürzel* den angegebenen *Befehl* zu. Fehlt die Angabe von *Befehl*, wird die Definition von *Kürzel* geliefert, wird `alias` ohne Argumente aufgerufen, so werden alle definierten Kürzel aufgelistet.

Ablaufsteuerung

```
if Bedingung ; then
    Befehle
else
    Befehle
fi
```

```
case Text in
    Muster1 ) Befehle ;;
    :
    Mustern ) Befehle ;;
esac
```

```
for Index in Liste ; do
    Befehle
done
```

```
for Index; do
    Befehle
done
```

äquivalent zu:

```
for Index in $*; do
    Befehle
done
```

```
while Bedingung; do
    Befehle
done
```

```
until Bedingung; do
    Befehle
done
```

```
:Argumente
# Leerkommando: nur Argumentauswertung
```

Kommandos

<code>exit</code>	beendet die aktuelle Shell mit dem Returncode des letzten Kommandos
<code>break</code>	beendet die aktuelle Schleife und fährt nach dem nächsten <code>done</code> fort.
<code>continue</code>	bricht den aktuellen Schleifendurchlauf ab und fährt mit dem nächsten fort.
<code>true</code>	Returncode 0
<code>false</code>	Returncode != 0

Bedingungen

	wahr, falls
<code>[-d <i>Verzeichnis</i>]</code>	(directory) <i>Verzeichnis</i> vorhanden
<code>[-f <i>Datei</i>]</code>	(file) <i>Datei</i> vorhanden
<code>[-s <i>Datei</i>]</code>	(size) <i>Datei</i> nicht leer ist
<code>[-r <i>Datei</i>]</code>	(read) <i>Datei</i> lesbar ist
<code>[-w <i>Datei</i>]</code>	(write) <i>Datei</i> beschreibbar ist
<code>[-x <i>Datei</i>]</code>	(execute) <i>Datei</i> ausführbar ist

[*String*] *String* nicht leer ist

[*String*₁=*String*₂] *String*₁ und *String*₂
identisch (analog mit !=)

[*Zahl*₁ -eq *Zahl*₂] *Zahl*₁ und *Zahl*₂ gleich
(analog mit -ne, -gt,
-ge, -lt, -le)

Verknüpfungen sind mit -a (and), -o (or) und ! (not) möglich.

Logische Verknüpfungen

***Kommando*₁ && *Kommando*₂**

*Kommando*₁

if \$? == 0; then *Kommando*₂; fi

Kommando*₁ || *Kommando

*Kommando*₁

if \$? != 0; then *Kommando*₂; fi

***Kommando*₁ && *Kommando*₂ || *Kommando*₃**

Falls entweder *Kommando*₁ oder *Kommando*₂ nicht fehlerfrei ausgeführt werden können wird *Kommando*₃ ausgeführt; dabei wird *Kommando*₂ nur ausgeführt, wenn *Kommando*₁ fehlerfrei ausgeführt werden konnte.

Arithmetik

i=`expr \$i + 1`

Ausführung einer Prozedur in der aktuellen Shell

Jedes Kommando außer den internen Shellkommandos und jede Prozedur startet normalerweise eine eigene Shell. Der Punktbefehl erlaubt die Ausführung einer Prozedur in der aktuellen Shell:

.Datei

Eingabeumlenkung in Shell-Prozeduren

Die Standardeingabe für ein Kommando kann in der betreffenden Datei selbst enthalten sein:

```
#!/bin/sh
echo "gleich kommt's."
cat <<EOF
Hier der eingebettete Text.
EOF
echo "das war's."
```

Fehlerbehandlung

Flags zur Fehlersuche

- v (**verbose**) jede Kommandozeile wird ohne Interpretation ausgegeben.
- x (**execute**) die Interpretation jedes ausgeführten Kommandos wird ausgegeben.

Signale

In folgenden Fällen werden Signale generiert, die (mit Ausnahme von Signal 9) in einer Shellprozedur abgefangen werden können:

- `Ctrl-C` (Signal 2), `exit`- oder `kill`-Kommando
- illegales Kommando oder Argument
- Hardwarefehler
- Ende eines Subprozesses

Zum Abfangen eines Signals dient:

```
trap 'Befehle' Signalnummer
```

Ein Trap kann zurückgesetzt werden mit:

```
trap Signalnummer
```

Shellfunktionen

Eine Shellfunktion kann in der Form

```
Funktion() {  
    Befehle  
}
```

erklärt werden. Ihr Aufruf erfolgt mit

```
Funktion Argumente
```

Eine solche Funktion befindet sich wie vordefinierte Shellfunktionen stets im Hauptspeicher, ist aber nur in der Shell sichtbar, in der sie definiert wurde (nicht in untergeordneten Shells).

2.5 tcsh-Shellprogrammierung

Vordefinierte Variablen

`$argv[0]` oder `$0` : Name des Scripts
`$argv[n]` oder `$n` : n -tes Argument
`$argv[*]` oder `$*` : gesamte Argumentliste
`$#argv` : Anzahl der Argumente
`$status` : Ergebniswert des zuletzt ausgeführten Befehls.

Benutzerdefinierte Variablen

`set Variable = Text`
definiert die *lokale Variable* mit *Text* als Wert.

`set Feldvariable = (Elemente)`
definiert die *lokale Feldvariable* und weist ihr die (durch Leerzeichen getrennten) *Elemente* der Reihe nach zu.

`@ Variable = Ausdruck`
erklärt die *lokale, numerische Variable* und weist ihr den Wert von *Ausdruck* zu.

unset *Variable*

löscht die *lokale Variable*.

setenv *Variable Text*

definiert die *globale Variable* mit *Text* als Wert. Eine mit `setenv` erklärte Variable ist auch in einer untergeordneten Shell bekannt.

unsetenv *Variable*

löscht die *globale Variable*.

alias [*Kürzel* [*Befehl*]]

Ordnet *Kürzel* den angegebenen *Befehl* zu. Fehlt die Angabe von *Befehl*, wird die Definition von *Kürzel* geliefert, wird `alias` ohne Argumente aufgerufen, so werden alle definierten Kürzel aufgelistet.

`$Variable`

interpretiert *Variable* als Shellvariable und setzt ihren Wert textuell ein.

`$?Variable`

liefert 1, falls *Variable* definiert ist, andernfalls 0.

Mit `set` werden die Werte der lokalen, mit `setenv` die der Umgebungsvariablen angezeigt.

Ablaufsteuerung

```
if (Ausdruck) Befehl
```

```
if (Ausdruck) then  
  Befehle  
endif
```

```
if (Ausdruck) then  
  Befehle  
else  
  Befehle  
endif
```

```
foreach Index (Liste)  
  Befehle  
end
```

```
while (Ausdruck)  
  Befehle  
end
```

```
repeat n Befehl
```

```
break  
#Sprung aus einer foreach-, while oder repeat-Schleife
```

```
continue  
#Sprung an das Ende einer foreach-, while oder repeat-Schleife
```

```
switch (Text)  
  case Muster1: Befehle; breaksw  
  :  
  case Mustern: Befehle; breaksw  
  default: Befehle #optional  
endsw
```

```
goto marke
```

```
onintr marke
```

2.6 Der vi-Editor

Der vi kennt drei Modi:

- Kommandomodus
- Eingabe in der Statuszeile
- Eingabe- oder Ersetzungsmodus

Einige wichtige vi-Befehle

Eingabe	Bedeutung
vi <i>Datei</i>	Aufruf
:wq	Abspeichern und Verlassen
:q!	Verlassen <i>ohne</i> abzuspeichern
Esc	Verlassen des Eingabemodus
i	Einfügen <i>vor</i> dem Cursor
I	Einfügen am Zeilenanfang
a	Einfügen <i>hinter</i> dem Cursor
A	Einfügen am Zeilenende
R	Überschreiben des Textes
C	Rest der Zeile ersetzen
s	Zeichen unter dem Cursor ersetzen
S	Ersetzen der Zeile
ndd	<i>n</i> Zeilen ab der aktuellen Zeile löschen
D	Rest der aktuellen Zeile löschen

2.7 Systembefehle

find *Verzeichnisse* *Ausdrücke*

Durchsucht die angegebenen Verzeichnisse und deren Unterverzeichnisse nach Dateien, die den Kriterien der *Ausdrücke* entsprechen. Dabei erlauben Seiteneffekte der *Ausdrücke* weitreichende Aktionen.

Typischer Aufruf:

```
find . -name "*.tex" -print      oder  
find . -type d -exec chmod 755 {} \;
```

Der erste Aufruf listet alle Dateien im aktuellen Verzeichnis und dessen Unterverzeichnissen auf, die auf `.tex` enden.

Der zweite Aufruf führt für das aktuelle Verzeichnis und jedes seiner Unterverzeichnisse `v` den Befehl `chmod 755 v` aus.

Mögliche *Ausdrücke*:

`-name Datei`

trifft auf jede gleichnamige *Datei* zu.

`-print`

trifft auf alle Dateien zu. Als Seiteneffekt wird der relative Pfadname der betreffenden Datei ausgegeben.

-type *Dateityp*

trifft auf jede Datei zu, die den angegebenen *Dateityp* besitzt.

b für „block special files“

c für zeichenorientierte „special files“

d für Verzeichnisse

f für reguläre Dateien

l für symbolische Links

-exec *Befehl*

trifft auf alle Dateien zu. Als Seiteneffekt wird der angegebene *Befehl* ausgeführt. In dem *Befehl* wird „{ }“ durch den Namen der betreffenden Datei ersetzt. Der *Befehl* muß durch „\ ;“ abgeschlossen sein.

`awk Optionen awk-Skript Dateiopt`
`awk Optionen -f awk-Skriptdatei Dateiopt`

Der Reportgenerator `awk` zerlegt die Eingabedatei in Zeilen und diese in Felder. Eine eigene, an C angelehnte Sprache erlaubt es, nach bestimmten Feldern zu suchen, sie zu modifizieren, ggf. mit ihnen zu rechnen und sie neuformatiert auszugeben.

Typischer Aufruf:

```
awk -F: '$3 > 499 {print $1}' /etc/passwd
```

Optionen:

`-Ffs` Feldtrennzeichen

Das `awk`-Skript enthält Angaben der Form

Muster{Anweisungen}

Das einleitende *Muster* entscheidet, auf welche Zeilen die nachfolgenden Anweisungen angewandt werden. Ein Muster kann aus regulären Ausdrücken, arithmetischen Bedingungen, und logischen Verknüpfungen bestehen. Zwei Muster, durch ein Komma getrennt, geben einen Bereich an.

Besondere Bedeutung besitzen die Muster

BEGIN

die folgenden Anweisungen werden ausgeführt, bevor die erste Zeile der Eingabedatei gelesen wird.

END die folgenden Anweisungen werden ausgeführt, nachdem die letzte Zeile der Eingabedatei gelesen wurde.

Anweisungen können beinhalten:

- Ausgabebefehle
- Zuweisungen
- Kontrollanweisungen

```
if(Bedingung)  
    Anweisung  
else  
    Anweisung
```

```
while(Bedingung)  
    Anweisung
```

```
for(Ausdruck; Bedingung; Ausdruck)  
    Anweisung
```

Auf die Felder einer Zeile wird mit \$1, \$2,... zugegriffen.

Anmerkungen:

- Bei Eingabe vom Terminal aus muß das awk-Skript in Hochkommata eingeschlossen werden, um eine Interpretation durch die Shell zu verhindern.
- Ohne Angabe eines *Musters* werden sämtliche Zeilen bearbeitet.
- Ohne Angabe von *Anweisungen*, werden die betreffenden Zeilen nach `stdout` geschrieben.
- Fehlt die Angabe der *Datei*, wird von `stdin` gelesen.

Muster-Beispiele:

/Text/

Es werden alle Zeilen ausgewählt, in denen *Text* vorkommt.

\$i ~ /Text/

Es werden die Zeilen ausgewählt, in denen *Text* im *i*-ten Feld auftritt.

\$1 >= "s" && \$1 < "t"

Es werden alle Zeilen ausgewählt, deren erstes Feld mit *s* beginnt.

Ein ausführlicheres awk-Script:

```
BEGIN
{ s1 += $2; s2 += $3; s3 += $4}
END { print "Gesamtplattenplatz:", s1
      print "davon benutzt: ", s2
      print "noch frei: ", s3}
```

Aufruf mit

```
df | awk -f awk-Scriptdatei
```

`sed` Optionen [*Datei*]

Der Streameditor `sed` liest die angegebene *Datei* und gibt sie modifiziert als Standardausgabe aus. Fehlt die Angabe der *Datei*, wird die Standardeingabe gelesen.

Optionen:

- n Ausgabe nur bei explizitem `p`-Befehl.
- e *Script*
 (zusätzliche) Verwendung der angegebenen Anweisungen.
- f *Script-Datei*
 (zusätzliche) Verwendung der Anweisungen der angegebenen Datei.

`sed`-Anweisungen besitzen die Form

[Adresse₁[, Adresse₂]]Funktion [Argumente]

Grundsätzlich werden alle Anweisungen für jede Zeile der *Datei* ausgeführt. Durch Angabe einer *Adresse* kann eine einzelne Zeile, durch Angabe zweier, durch Komma getrennter *Adressen* ein Zeilenbereich ausgewählt werden.

Als Adresse kann eine Zeilennummer, \$ für die letzte Zeile oder ein Muster in der Form */Text/* angegeben werden.

Ersetzungsfunktion

s/Muster/Text/Modus

2¹

Textteile, auf die *Muster* paßt, werden durch *Text* ersetzt. Die Klammerung von *Muster* und *Text* kann durch beliebige Zeichen außer \ erfolgen.

Modus

n Es wird nur das *n*-te passende Textstück ersetzt.

g Es werden alle passenden Textstücke ersetzt.

p Im Fall einer Ersetzung wird der neue Text ausgegeben.

w Datei Im Fall einer Ersetzung wird der Text des Bereichs an das Ende der *Datei* geschrieben.

¹max. Adreßzahl

Funktionen, die Zeilen betreffen

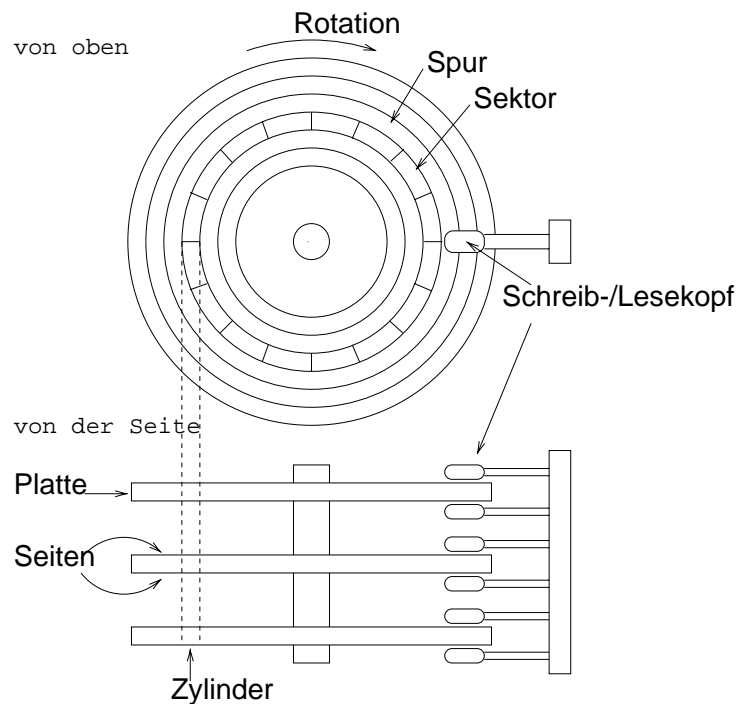
d	Zeilen löschen	2
a\ text	Zeilen anhängen	1
c\ text	Zeilen ersetzen	2

Ein-/Ausgabefunktionen

p	Ausgabe bis zum ersten Zeilenende als Standardausgabe.	2
w <i>Datei</i>	Ausgabe in die angegebene Datei.	2
r <i>Datei</i>	Einlesen der angegebenen Datei und Ausgabe ohne weitere Modifizierung als Standardausgabe.	1

3 Installation

3.1 Festplatten



Partitionen

Partitionen bedeuten eine physikalische Einteilung der Festplatte. Man unterscheidet *primäre*, *erweiterte* und *logische* Partitionen.

Nach einer physikalischen (low-level) Formatierung (durch den Hersteller) lassen sich für eine Festplatte bis zu vier *primäre* Partitionen definieren. Eine dieser Partitionen kann alternativ zu einer *erweiterten* Partition erklärt werden. Innerhalb einer erweiterten Partition lassen sich hinreichend viele *logische* Partitionen definieren.

Alle Arten von Partitionen müssen vor ihrer Benutzung zur Datenspeicherung logisch formatiert werden.

Werden mehrere primäre Partitionen definiert, kann zur selben Zeit nur eine aktiv sein (von ihr wird gebootet).

In der Vergangenheit war es nötig, für mehrere Betriebssysteme entsprechende primäre Partitionen zu definieren (DOS, Windows 3.x und Windows 95 müssen von einer aktiven Primärpartition auf dem ersten physikalischen Laufwerk booten).

Ein Betriebssystem einer primären Partition konnte nicht auf die Daten einer anderen primären Partition zugreifen.

Einfacher Bootvorgang

Mit dem Einschalten des Rechners führt die CPU die Anweisungen des ROM BIOS aus. Deren letzter Teil, die BIOS-Bootroutine, liest den physikalisch ersten Sektor der Festplatte, den Master-Boot-Record (MBR) ein.

Dieser MBR enthält ein Master-Bootprogramm sowie eine Partitionstabelle, in der die aktive Partition vermerkt ist.

Jede bootfähige Partition enthält ein Bootprogramm im ersten Sektor der Partition.

Das Master-Bootprogramm startet das Boot-Programm der aktiven Partition und damit das betreffende Betriebssystem.

Tips

- Um einen zerstörten Master-Boot-Record wieder herzustellen, kann eine DOS-Bootdiskette mit `fdisk.com` benutzt werden:

```
fdisk /mbr
```

- Um auf demselben Rechner wahlweise Linux oder Windows NT booten zu können, kann dem Windows NT-Bootmenü ein entsprechender Eintrag hinzugefügt werden:

1. den Linux Loader (LILO) in der Linux-Root-Partition installieren,
2. den Linux-Bootsektor (hier auf `/dev/hda3`)

```
/bin/dd if=/dev/hda3 bs=512 count=1 \
of=bootsec.lin
```

in eine Datei (hier `bootsec.lin`) speichern,

3. diese Datei unter Windows NT in das Hauptverzeichnis kopieren,
4. den eigentlichen Eintrag unter Windows NT in die Datei `boot.ini` vornehmen:

```
[boot loader]
timeout=10
default=multi(0)disk(0)rdisk(0)partitio
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WIN
c:\bootsec.lin="Linux"
```

3.2 Bootvorgang

Booten des Kernels

```
init (liest /etc/inittab)
```

Darin werden Shellskripte angegeben, die den weiteren Bootvorgang steuern.

SuSE 7.0:

```
si:I:bootwait:/sbin/init.d/boot  
:  
l3:3:wait:/sbin/init.d/rc 3
```

Redhat 6.1:

```
si::sysinit:/etc/rc.d/rc.sysinit
```

Konfigurationsdateien

SuSE 7.0:

```
/etc/rc.d/ -> /sbin/init.d/
```

Redhat 6.1:

```
/etc/rc.d/  
/etc/sysconfig/
```

4 Administration

4.1 Benutzerverwaltung

Einrichten eines Benutzers von Hand

1. `/etc/passwd` mit `vipw` editieren (lockt die Datei). Durch `*` als Paßwort Einloggen verhindern.
2. `/etc/group` mit `vigr` editieren. (Bei SuSE 7.0 nicht vorhanden.)
3. Home-Verzeichnis anlegen.
4. Dateien aus `/etc/skel` ins Home-Verzeichnis kopieren
5. Besitzer und Rechte setzen:

```
cd /home/Benutzer
chown -R Benutzer.Gruppe .
chmod -R go=u,go-w .
chmod go= .
```

6. Paßwort mit `passwd` setzen.

Ändern eines Accounts

`chfn` Namensfeld ändern
`chsh` Login-Shell ändern
`passwd` Paßwort setzen

Benutzer löschen

```
find / -user Benutzer
```

findet alle Dateien des *Benutzers*, auch außerhalb seines Home-Verzeichnisses.

Deaktivieren eines Accounts

Eintragen eines geeigneten Programmes als Shell (die Ausgabe beginnt in der 2. Zeile):

```
#!/usr/bin/tail +2  
Ihr Account wurde gesperrt.  
Bitte rufen Sie unter 123456 an.
```

4.2 Festplattenquoten

Durch Festplattenquoten können die Anzahl der Inodes (Dateien) und der belegten Blöcke je Benutzer/Gruppe und je Filesystem beschränkt werden.

Konfiguration

Kernelkonfiguration:

```
Filesystems
  Quota support [y]
```

Quota-Paket installieren.

Start durch `/sbin/init.d/quota` (SuSE 7.0) bzw. `/etc/rc.d/rc.sysinit` (RedHat 6.1):

```
if [ -x /sbin/quotacheck ]; then
  echo "Checking root filesystem quotas"
  /sbin/quotacheck -v /
fi

if [ -x /sbin/quotaon ]; then
  echo "Turning on user and group quotas \
      for local filesystems"
  /sbin/quotaon -a
fi
```

(Nach Mounten der betreffenden Dateisysteme)

In `/etc/fstab`:

```
/dev/hda2 /partition ext2 \  
defaults,usrquota,grpquota 1 1
```

```
touch /partition/quota.user  
touch /partition/quota.group  
chmod 600 /partition/quota.user  
chmod 600 /partition/quota.group
```

Setzen von Quoten

Für einen Benutzer:

```
edquota -u Benutzer
```

```
/dev/hda2: blocks in use: 2594, \  
limits (soft = 5000, hard = 6500)  
inodes in use: 356, \  
limits (soft = 1000, hard = 1500)
```

blocks in use: Anzahl belegte Blöcke in KByte

inodes in use: Anzahl Dateien

```
edquota -g Gruppe
```

Setzen der Quoten für alle Benutzer mit einer UID
>= 500 entsprechend *Benutzer*

```
edquota -p Benutzer \  
'awk -F: '$3 > 499 {print $1}' /etc/passwd'
```

Setzen einer zeitlichen Beschränkung für die Über-
schreitung des Softlimits

```
edquota -t
```

```
Time units may be: days, hours, minutes, \  
or seconds  
Grace period before enforcing soft limits \  
for users:  
/dev/hda2: block grace period: 0 days, \  
file grace period: 0 days
```

4.3 Cron-Jobs

Der Cron-Daemon (`crond`) wird gestartet aus `/etc/rc.d/rc3.d/S21cron` (SuSE 7.0).

Es können Dateien aus `/var/spool/cron/tabs` benutzt werden, benannt nach Benutzern, oder die Datei `/etc/crontab`:

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:\
    /usr/lib/news/bin
MAILTO=root

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Die führenden fünf Felder der `run-parts` bezeichnen Minute, Stunde, Tag des Monats, Monat und Tag der Woche, der Rest der Zeile wird ausgeführt.

4.4 Backup

Kriterien zur Auswahl von Backupmedien:

- Kosten
- Zuverlässigkeit
- Geschwindigkeit
- Handhabbarkeit

Standardprogramme:

tar, cpio, dump

tar

```
# tar -create -file /dev/ftape /etc
```

```
# tar -cMf /dev/fd0H1440 /etc
```

```
tar: Removing leading / from absolute path \
names in the archive
Prepare volume #2 for /dev/fd0H1440 and hit \
return:
```

```
# tar -compare -verbose -f /dev/ftape
```

```
# tar -create -newer '8 Sep 1995' -file \
/dev/ftape /etc -verbose
```

```
tar: Removing leading / from absolute path \
names in the archive
```

5 Netzwerk

Literatur:

The Linux Network Administrators' Guide

`/usr/share/doc/howto/en/html/NET3-4-HOWTO.html`

5.1 Grundlagen

Adressen

Internet Protocol (IP)-Adressen bestehen aus vier Bytes, geschrieben als vier durch Punkte getrennte Dezimalzahlen. Im allgemeinen besitzt jedes Interface eines Hosts oder Routers eine eigene IP-Adresse.

Eine IP-Adresse gliedert sich in mehrere Teile:

Host-Adresse	192.168.110.23
Netzwerkmaske	255.255.255.0
Netzwerkteil	192.168.110.
Host-Teil	.23
Netzwerkadresse	192.168.110.0
Broadcast-Adresse	192.168.110.255

Aus organisatorischen Gründen werden verschiedene Klassen von Netzwerken unterscheiden:

Klasse	Netzmaske	Netzwerk-Adressen
A	255.0.0.0	0.0.0.0 - 127.255.255.255
B	255.255.0.0	128.0.0.0 - 191.255.255.255
C	255.255.255.0	192.0.0.0 - 223.255.255.255
Multicast	240.0.0.0	224.0.0.0 - 239.255.255.255

Für private Netzwerke ohne Anschluß an das Internet, sind folgende IP-Adressen reserviert:

Klasse	Netzmaske	Netzwerk-Adressen
A	255.0.0.0	10.0.0.0 - 10.255.255.255
B	255.255.0.0	172.16.0.0 - 172.31.255.255
C	255.255.255.0	192.168.0.0 - 192.168.255.255

Konfigurationsdateien

RedHat 6.1:

Beschreibung:

```
/etc/sysconfig/network
```

Skripte:

```
/etc/sysconfig/network-scripts
```

Steuerung:

```
/etc/rc.d/init.d/network
```

Konfigurationstool:

```
/usr/bin/netcfg
```

Netzwerk-Interfaces

Unter Linux werden Netzwerk-Interfaces dynamisch erzeugt, ohne Device-Dateien. Im allgemeinen werden Netzwerk-Devices durch den Device-Treiber automatisch angelegt (`eth0`), wenn während der Installation die betreffende Hardware erkannt wird. Ausnahmen bilden SLIP und PPP.

Netzwerk-Konfiguration

Die Konfiguration eines Netzwerk-Interfaces besteht darin, einem Netzwerk-Device geeignete Adressen zuzuweisen, typischerweise mittels `ifconfig`:

```
ifconfig eth0 192.168.0.1 \  
netmask 255.255.255.0 up  
ifconfig eth0 down
```

Namen

Form:

```
Host.Sub-Domain.Top-Level-Domain
```

Dateien:

`/etc/resolv.conf`

```
domain rz.tu-harburg.de  
search rz.tu-harburg.de tu-harburg.de  
nameserver 192.168.10.1  
nameserver 192.168.12.1
```

/etc/host.conf

(Beschreibung mit `man resolv`)

```
order hosts,bind
multi on
```

/etc/hosts

```
127.0.0.1    localhost loopback
192.168.0.1  this.host.name
```

Loopback-Device

```
ifconfig lo 127.0.0.1
route add -host 127.0.0.1 lo
```

5.2 Routing

IP-Routing stellt den Prozeß dar, in dem ein Host mit mehreren Netzwerkverbindungen entscheidet, wohin ein IP-Datagramm zu schicken ist.

Jeder Rechner unterhält eine spezielle Routing-Tabelle, die sich wie folgt anzeigen läßt:

```
cat /proc/net/route  
/sbin/route -n  
netstat -r
```

Einfach gesagt, arbeitet das Routing wie folgt:

Die Zieladresse eines ankommenden Datagramms wird untersucht und mit jedem Eintrag in der Tabelle verglichen. Der am besten passende Eintrag wird gewählt und das Datagramm an das entsprechende Interface weitergeleitet.

Ist ein Gateway eingetragen, wird das Datagramm an den betreffenden Rechner geschickt, sonst wird angenommen, die Zieladresse befindet sich innerhalb des Netzwerks, das über das Interface erreichbar ist.

Beispiel:

```
ifconfig eth0 192.168.1.10 \  
netmask 255.255.255.0 up
```

```
route add -net 192.168.1.0 \  
netmask 255.255.255.0 eth0
```

```
route add default gw 192.168.1.1 eth0
```

5.3 NIS

Literatur:

`/usr/share/doc/howto/en/html/NIS-HOWTO.html`

NIS (Network Information Service) stellt einen Datenbankdienst dar, der netzwerkweiten Zugriff auf Informationen aus `/etc/passwd`, `/etc/shadow` u.a. ermöglicht.

Die Daten werden von einem NIS-Server bereitgestellt, ggf. bei Änderungen in Kopie an NIS-Slaves gegeben und bei Bedarf von NIS-Clients abgefragt. Slaves können die Funktion des Masters bei dessen Ausfall oder Überlastung übernehmen.

Neben NIS gibt es eine Weiterentwicklung NIS+ der Firma Sun, die erhöhten Sicherheitsanforderungen (Verschlüsselung und Authentifizierung über Secure RPC) genügt, jedoch deutlich aufwendiger zu administrieren ist.

Unter Linux bietet die glibc 2.0.x gute Unterstützung für NIS, auch mit Shadow-Paßwortdatei. Die

Unterstützung von NIS+ befindet sich noch in der Entwicklung, man benötigt einen aktuellen Snapshot der glibc.

Pakete:

```
yp-tools 2.2 arbeitet mit jeder libc  
ypbind 3.3 arbeitet mit allen Bibliotheken
```

Installation eine NIS-Clients

1. `ypbind` nach `/usr/sbin`
`ypwhich`, `ypcat`, `ypoll`, `ypmatch` nach
`/usr/bin`.
2. Konfigurationsdatei `/etc/ypconf` erstellen.

```
ypserver aaa.bbb.ccc.ddd
```

3. Domainnamen setzen

```
/bin/domainname mainzelmann
```

4. `/usr/sbin/portmap` starten.
5. Verzeichnis `/var/yp` anlegen.
6. `/usr/sbin/ypbind` starten.

7. `rpcinfo -p localhost`
sollte liefern

```
program vers proto port
 100000     2   tcp    111  portmapper
 100000     2   udp    111  portmapper
 100007     2   udp    637  ypbind
 100007     2   tcp    639  ypbind
 300019     1   udp    660
```

8. `rpcinfo -u localhost ypbind`
sollte liefern

```
program 100007 version 2 ready and waiting
```

9. In `/etc/passwd` als letzte Zeile ergänzen:

```
+:::~:::
```

Benutzereinträge können, eingeleitet durch + oder -, hinzugefügt bzw. gesperrt werden; die Shell-Angabe läßt sich überschreiben.

Installation eine NIS-Servers

1. Anpassen der Dateien `/var/yp/securenets` und `/etc/ypserv.conf`
2. Starten des Servers `ypserv`

3. Erzeugen der NIS Datenbank durch Aufruf von

```
/usr/lib/yp/ypinit -m
```

auf dem Server.

Ein NIS-Slave ist zunächst als NIS-Client zu konfigurieren. Danach ist

```
/usr/lib/yp/ypinit -s masterhost
```

zu starten.

5.4 DNS

Literatur:

`/usr/share/doc/howto/en/html/DNS-HOWNTO.HTML`

Ein caching only Nameserver

Datei `/etc/named.conf`:

```
options {
    directory "/var/named";
};

zone "." {
    type hint;
    file "root.hint";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "pz/127.0.0";
};
```

Datei /var/named/root.hint:

```
; formerly NS.INTERNIC.NET
;
. 3600000 IN NS A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. \
 3600000 A 198.41.0.4
;
; formerly NS1.ISI.EDU
;
. 3600000 NS B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. \
 3600000 A 128.9.0.107
```

Datei /var/named/pz/127.0.0:

```
@ IN SOA ws99.rz.tu-harburg.de. \
        root.localhost. (
        1          ; Serial
        8H         ; Refresh
        2H         ; Retry
        1W         ; Expire
        1D)        ; Minimum TTL
1 NS     ws99.rz.tu-harburg.de.
PTR     localhost.
```

Datei /etc/resolv.conf:

```
search rz.tu-harburg.de
nameserver 127.0.0.1
```

In /etc/nsswitch.conf:

```
hosts: files dns
```

In /etc/host.conf:

```
order hosts,bind
```

Start des Nameservers:

```
/usr/sbin/ndc start
```

Ein einfacher Nameserver

In /etc/named.conf:

```
zone "linux.bogus" {
    notify no;
    type master;
    file "pz/linux.bogus";
};

zone "196.168.192.in-addr.arpa" {
    notify no;
    type master;
    file "pz/192.168.196";
};
```

Datei /var/named/pz/linux.bogus:

```
@ IN SOA ns linux.bogus. \
        hostmaster linux.bogus. (
        199802151 ; serial
        8H       ; refresh, seconds
        2H       ; retry, seconds
        1W       ; expire, seconds
        1D )     ; minimum, seconds
;
    TXT "Linux.Bogus, your DNS consultants"
    NS  ns           ; Name server
    NS  ns.friend.bogus.
    MX  10 mail     ; Primary Mail Exchanger
    MX  20 mail.friend.bogus.
                          ; Secondary Mail Exchanger

localhost      A      127.0.0.1

gw              A      192.168.196.1
               HINFO  "Cisco" "IOS"
               TXT    "The router"

ns              A      192.168.196.2
               MX     10 mail
               MX     20 mail.friend.bogus.
               HINFO  "Pentium" "Linux 2.0"

www             CNAME  ns

donald          A      192.168.196.3
               MX     10 mail
               MX     20 mail.friend.bogus.
               HINFO  "i486"  "Linux 2.0"
               TXT    "DEK"

mail            A      192.168.196.4
               MX     10 mail
```

```

                MX      20 mail.friend.bogus.
                HINFO   "386sx" "Linux 1.2"

ftp            A      192.168.196.5
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "P6" "Linux 2.1.86"

```

Testausgabe unter nslookup mit:

```
ls -d linux.bogus
```

Datei /var/named/pz/192.168.196:

```

@ IN SOA ws99.rz.tu-harburg.de.
    root.localhost. (
        199802151 ; Serial, todays date + to-
days serial
        8H      ; Refresh
        2H      ; Retry
        1W      ; Expire
        1D)     ; Minimum TTL
NS  ns.linux.bogus.

1   PTR gw.linux.bogus.
2   PTR ns.linux.bogus.
3   PTR donald.linux.bogus.
4   PTR mail.linux.bogus.
5   PTR ftp.linux.bogus.

```

Testausgabe unter nslookup mit

```
ls -d 196.168.192.in-addr.arpa
```

Klassenlose Sub-Netze

auf Seiten des Nameservers:

```
129.125.127.203.in-addr.arpa.  IN CNAME \  
    129.128.125.127.203.in-addr.arpa.  
130.125.127.203.in-addr.arpa.  IN CNAME \  
    130.128.125.127.203.in-addr.arpa.  
(etc)
```

lokal:

```
129.128.125.127.203.in-addr.arpa.  IN PTR \  
    aaa.linux.bogus.  
130.128.125.127.203.in-addr.arpa.  IN PTR \  
    bbb.linux.bogus.  
(etc)
```

6 Linux-System als Server

6.1 Druck-Server

Literatur:

iX 6/98 S.144

iX 4/96 S. 184 (Blockgerätetreiber)

c't 17/98 S.190

Hardware

Die Druckausgabe erfolgt über eine serielle oder parallele Schnittstelle.

serielle Schnittstellen:

Device	I/O-Adresse	DOS-Name
/dev/ttyS0	0x03f8	com1(irq = 4)
/dev/ttyS1	0x02f8	com2(irq = 3)
/dev/ttyS2	0x03e8	com2(irq = 4)
/dev/ttyS3	0x02e8	com2(irq = 3)

parallele Schnittstellen:

Device	I/O-Adresse	DOS-Name
/dev/lp1	0x3bc	lpt1
/dev/lp2	0x378	lpt2
/dev/lp0	0x278	lpt3

Fehlende Devices können wie folgt erzeugt werden:

```
mknod -m 660 /dev/lp2 c 6 2
```

darin:

-m 660 Setzen der Zugriffsrechte

/dev/lp2 Name des Devices

c Art des Device

6 Major Device Number

2 Minor Device Number

Rechte setzen:

```
chown root.lp /dev/lp[2]
```

Interrupt-Einstellungen:

```
tunelp
```

Test:

```
cat Textdatei > /dev/lp1
```

Kernel

Kernelkonfiguration:

```
Loadable module support
  Enable loadable module support [y]
  Kernel daemon support[y]
General setup
  Network support [y]
Networking options
  TCP/IP networking [y]
Character devices
  Standard/generic serial support [y]
  Parallel printer support [m]
```

- Das Drucker-Spooling benötigt TCP/IP-Netzwerkunterstützung.
- Der Parallelport kann nicht gleichzeitig zum Drucken und für andere Dienste (PLIP, Zip-Laufwerk) genutzt werden.

Programme

Zum Spooling wird verbreitet die `plp`-Software (Public Line Printer) eingesetzt, eine Weiterentwicklung des auf TCP/IP basierenden BSD-Drucksystems.

<code>lpd</code>	line printer daemon
<code>lpc</code>	line printer control Programm
<code>lpr</code>	Druckbefehl
<code>lpq</code>	Anzeige einer Drucker-Queue
<code>lprm</code>	Entfernen von Druckjobs
<code>klpq</code>	KDE-Version von <code>lpq</code> , <code>lprm</code>
<code>kcmprinter</code>	KDE-Tool zur Bearbeitung von <code>/etc/printcap</code>

Rechte:

```
cd /usr/bin
chown root lpr lprm
chmod u+s lpr lprm
```

Konfigurationsdateien

`lpd` liest nach seinem Start in der Bootphase die Datei `/etc/printcap`.

/etc/printcap

```
# -----
# printcap für Linux-Systeme
#
# Einträge:
#   name:\
#       :sd=spool_directory:\
#       :lp=output_device_name:\
#       :lf=error_logging_file:\
#       :rm=remote_machine_name:\
#       :rp=remote_printer_name:\
#       :if=input_filter:\
#       :of=output_filter:\
#       :rs:\   Benutzer muß Account besitzen
#       :sh:\   kein Deckblatt
#       :mx#0: beliebige Dateilänge
#
lp|dj600:\
    :sd=/var/spool/lpd/dj600:\
    :lf=/var/spool/lpd/dj600/errs:\
    :if=/var/spool/lpd/bin/ps2dj600:\
    :lp=/dev/lp1:\
    :sh:\
    :mx#0:

lq850:\
    :sd=/var/spool/lpd/lq850:\
    :lf=/var/spool/lpd/lq850/errs:\
    :if=/var/spool/lpd/bin/ps2lq850:\
    :lp=/dev/ttyS1:\
    :br#9600:\
    :ty=ixon -imaxbel -ixany -ixoff \
        -crtsts:\
    :sh:\
    :mx#0:
```

- Zeilen, die mit einem Leerzeichen beginnen, werden ignoriert.
- Einrückungen müssen mit einem Tabulartorzeichen erfolgen.
- Soll ein Benutzer lediglich einen Account zum Drucken erhalten, genügt:

```
useradd Benutzer -s /bin/false
```

- `lpd` übergibt dem Filter als fünftes Argument `$5` den Namen des Benutzers, der den Druckauftrag erteilt hat.
- Im Gegensatz zu dem älteren BSD-Drucksystem unterstützt `plp` auch Input-Filter bei Remote-Druckern.

Einrichten von Spoolverzeichnissen:

```
#!/bin/tcsh
set printer=$1

mkdir /var/spool/lpd/$printer
chown lp:lp /var/spool/lpd/$printer
chmod 775 /var/spool/lpd/$printer

cd /var/spool/lpd/$printer
touch .seq errs status lock
chown lp:lp .seq errs status lock
chmod 664 .seq errs status lock
```

Druckfilter

Zeilenumbrüche:

```
#!/usr/bin/perl
while(<STDIN>){chop $_; print "$_\r\n";}
```

GhostScript zur PostScript-Ausgabe auf nicht Post-Script-fähige Drucker:

```
#!/bin/bash
/usr/bin/gs -q -dSAFER -dNOPAUSE \
-sPAPERSIZE=a4 -sDEVICE=djet500 -r300 \
-sOutputFile=- -
```

apsfilter

Einen universellen Druckfilter stellt `apsfilter` dar; `S.u.S.E.-apsfilter` bietet noch weitreichendere Konfigurationsoptionen, beliebig viele Druckerqueues, optionale Namen für Warteschlangen in `/etc/printcap` und ist für alle Distributionen verwendbar.

`apsfilter` benötigt:

- `a2ps` (mitgeliefert)
- `pbmplus/netpbm`
- `dvips`
- `jpeg`
- `transfig` aus `xfig`
- `gzip`

Die Konfiguration von `apsfilter` erfolgt mit

```
/var/lib/apsfilter/SETUP
```

Dabei wird nach den erforderlichen Zusatzprogrammen gesucht, alle notwendigen Parameter werden abgefragt, Spool-Verzeichnisse und Druckerschlangen erzeugt.

Konfigurationsdateien:

```
/etc/apsfilterrc
```

systemweite Konfiguration:

- Optionen für dvi-Treiber
- METAFONT-Modi
- druckerspezifische Escape-Sequenzen

`/etc/apsfilterrc.gs_device_name`

separate Einstellungen bei S.u.S.E.-apsfilter.

Wichtig: korrekte Einstellung der Auflösung
mittels `GS_RESOL`

Standardmäßig wird ASCII-Text mit zwei Textseiten
je Blatt, ohne Rahmen und Seitennumerierung ge-
druckt; Abhilfe:

```
A2PS_OPTS="-1 -F12.0 -p -8 -m -nu -nL -nP"
```

darin bedeuten:

- 1 einseitiger Druck
- p Portrait-Modus
- F12.0 Fontgröße 12 Punkte
- 8 ISO-Latin-1
- m 66 Zeilen
- nu kein Dateiname am unteren Ende der Seite
- nL kein User-Login
- nP Ausgabe auf stdout

Netzwerkkonfiguration

Server-seitig

`/etc/hosts.lpd`

Jeder Rechner, von dem Druckaufträge entgegengenommen werden sollen, muß in `/etc/hosts.lpd` eingetragen sein:

```
client.xx.tu-harburg.de
```

`/etc/hosts`

Die betreffenden Rechner des lokalen Netzwerks sollten in `/etc/hosts` definiert sein, um den Zugriff auf den Printserver nicht unnötig zu verzögern.

```
127.0.0.1    localhost
192.168.1.1  server.xx.tu-harburg.de  server
192.168.1.2  client.xx.tu-harburg.de  client
```

`/etc/host.conf`

```
order hosts,bind
multi on
```

Client-seitig

Hier ist eine ganz ähnliche Installation nötig wie Server-seitig, jedoch mit angepaßter `/etc/printcap`-Datei.

6.2 WWW-Server

Literatur:

iX 6/96 S.122

Server:

www.apache.org

Ursprünglich ein fehlerbereinigter (a patchy) NCSA-Server, hat sich der Apache-HTTP-Daemon zu dem am weitesten verbreiteten WWW-Server entwickelt.

Installation

Version 1.3.12

Dateien und Verzeichnisse

<code>/usr/sbin/httpd</code>	Server
<code>/var/run/httpd.pid</code>	Server PID
<code>/var/log/httpd.*</code>	Log-Dateien
<code>/sbin/init.d/apache</code>	Start/Stop-Script
<code>/usr/local/httpd/</code>	Server Root
<code>/etc/httpd/</code>	Konfigurationsdateien (S.u.S.E.)

Unterverzeichnisse unter `/usr/local/httpd`

<code>htdocs</code>	Dokumente
<code>icons</code>	Icons
<code>cgi-bin</code>	ausführbare Programme (CGI, Common Gateway Interface)

Im Konfigurationsverzeichnis sind anzupassen:

<code>httpd.conf</code>	Daemon
<code>srm.conf</code>	Darstellung
<code>access.conf</code>	Zugriffe

httpd.conf

- Üblich ist es, `httpd` stand alone zu starten (mit einem Script in `/sbin/init.d` und entsprechenden Links), da ein einmal gestarteter Daemon nicht bei jedem Zugriff die Konfiguration erneut einlesen muß.
- Nur wenn der Port 80 gewählt wird, muß einer URL nicht die Port Nummer angefügt werden.
- Pfadnamen für Error- und Log-Dateien können übernommen werden, ebenso die Voreinstellung zur Speicherung der Prozeßnummer.

- Eine neue Konfiguration kann eingelesen werden mit

```
kill -HUP `cat /var/run/httpd.pid`
```

Ggf. zu ändern sind:

ServerAdmin

Mail-Adresse des Webmasters.

ServerRoot

Ausgangspfad für alle wesentlichen Server-Dateien.

Servername

kann der Hostname sein, oder ein Alias, der dann in `/etc/hosts`, im Domain Name System (DNS) oder im Network Information Service (NIS) eingetragen sein muß.

srm.conf

legt die Darstellungsart von Dokumenten fest.

DokumentRoot

`/usr/local/httpd/htdocs`

UserDir

Einige Systemverwalter lehnen den Verweis von auf individuelle Verzeichnisse in \$HOME-Bereichen der Benutzer ab und legen stattdessen einzelne Verzeichnisse im DokumentRoot an.

DirectoryIndex

benennt die Dateien, die angezeigt werden sollen, falls auf ein Verzeichnis statt auf eine Datei zugegriffen wird.

Existiert diese Datei nicht, sind die im Verzeichnis liegenden Dateien und Verzeichnisse für den Aufrufer sichtbar und klickbar.

Alternativen:

```
DirectoryIndex index.html default.html \
default.shtml
```

* .shtml-Dateien kennzeichnen *Server-Side-Includes*, Skripte, die zur Aufrufzeit Daten einbinden, beispielsweise das letzte Änderungsdatum automatisch einlesen:

```
<p>Last update:  
<!--#echo var='LAST_MODIFIED' --></p>
```

Server-Site-Includes sind umstritten, da sie in jedem Fall einen Zugriff auf den Server erfordern, auch wenn die betreffende Seite bereits im lokalen Cache oder Proxy vorliegt.

Der Zugriff auf Directory-Listings kann unterdrückt werden, indem in der Konfigurationsdatei `access.conf` `Indexes` unter `Options` nicht aufgeführt wird.

AccessFileType

legt fest, in welcher Datei Zugriffsbeschränkungen für ein Verzeichnis zu finden sind.

ErrorDocument

Mögliche Arten der Fehlerbehandlung:

```
ErrorDocument 500 \  
"Serverprobleme, schade aber auch..."
```

```
ErrorDocument 404 \  
/cgi-bin/missing-handler.pl
```

```
ErrorDocument 404 \  
http://NeuerServer/url.html
```

access.conf

Beispiel:

```
#Eintrag für das cgi-bin-Verzeichnis
<Directory /usr/local/httpd/cgi-bin>
Options Index FollowSymLinks
</Directory>

# DokumentRoot-Konfiguration
<Directory /usr/local/httpd/htdocs>

# "None", "All", oder Kombinationen von
# "Includes", "FollowSymLinks", "ExecCGI",
# "MultViews"
Options Indexes FollowSymLinks

# was .htaccess überschreiben darf
AllowOverride All

# Zugriffsberechtigungen
<Limit GET>
order allow, deny
allow from all
</Limit>

</Directory>
```

<Directory>

Beschänkung des Zugriff auf ein Verzeichnis
mittels

<Limit>

Übersicht über zugelassene Benutzer:

```
<Limit GET>
order deny,allow
deny from all
allow from .tu-harburg.de aaa.bbb.ccc.
</Limit>
```

<Location>

Beschränkung des Zugriffs auf ein einzelnes Dokument

Allow Override

legt fest, welche Optionen in `.htaccess`-Dateien überschrieben werden dürfen.

Directory-Optionen

None keine der folgenden Optionen.

All alle folgenden Optionen außer `MultiViews`.

ExecCGI

CGI-Scripte dürfen ausgeführt werden.

FollowSymLinks

verfolgt symbolische Links.

Includes

Server-Side-Includes sind erlaubt.

IncludesNOEXEC

Server-Side-Includes bis auf `#exec` und `#include` (CGI-Scripte) sind erlaubt.

Indexes

Bei Anfrage nach einem Verzeichnis wird dessen Inhalt formatiert ausgegeben, wenn `index.html` o.ä. nicht vorhanden ist.

Multiviews

Inhaltsbezogene MultiViews sind erlaubt.

Dabei handelt es sich beispielsweise um mehrere Dateien in jeweils unterschiedlicher Sprache (oder Bilder in unterschiedlichen Formaten), die der Server je nach Spezifizierung des Clients sucht und verschickt.

SymLinksIfOwnerMatch

symbolische Links werden nur dann verfolgt, wenn das Ziel denselben Eigentümer besitzt wie der Link.

Zugriffsbeschränkungen

Zugriffsbeschränkungen sind auch in einzelnen Verzeichnissen mittels `.htaccess`-Dateien möglich; so kann etwa festgelegt werden,

- ob symbolische Links verfolgt werden sollen,
- ob der Inhalt des Verzeichnisses angezeigt wird, wenn kein `index.html` o.ä. existiert

Im Verzeichnis `support` der Apache-Distribution finden sich Scripte zur Benutzerverwaltung:

```
htpasswd  
dbmmanage
```

Aufruf:

```
htpasswd -c Paßwortdatei \  
adduser Name Paßwort
```

- `-c` dient zum Anlegen der Datei.
- das Paßwort ist im Klartext anzugeben.

Zusätzlich zu einer Paßwortdatei kann eine Gruppendatei angelegt werden.

Paßwort- und Gruppendatei können im Konfigurations-Verzeichnis abgelegt werden, keinesfalls sollten sie unterhalb der `DokumentRoot` oder im zu schützenden Verzeichnis liegen.

Eine `<Directory>`-Anweisung oberhalb des `<Limit>`-Abschnitts in `access.conf` oder einer `.htaccess`-Datei kann festlegen, ob Paßworteingaben erforderlich sind:

```
<Directory /usr/local/httpd/htdocs/protected>
AuthName Protected stuff
AuthType Basic
AuthUserFile /etc/httpd/passwd
AuthGroupFile /etc/httpd/group
<Limit GET>
require user anton bert
require group staff
</Limit>
```

Nur die Mitglieder der Gruppe `staff` und die Benutzer `anton` und `bert` dürfen nach Paßwortabfrage auf die Daten des Verzeichnisses `protected` zugreifen.

`dbmmanage` leistet entsprechendes wie `htpasswd`, benutzt jedoch das rudimentäre Datenbank-Format des NIS-Umfeldes. Statt `AuthUserFile` muß in diesem Fall `AuthDBMUserFile` benutzt werden.

Server-Side-Includes

Mit Server-Site-Include können in WWW-Dokumenten Daten eingefügt werden, die auf Filesystemebene bekannt sind, oder CGI-Scripte in HTML-Seiten integriert werden.

Dazu ist in `srm.conf` die Zeile

```
AddType application/x-server-parsed-html \
shtml
```

einzuführen.

Der `Options`-Eintrag in einer `<Directory>`-Anweisung oder einzelnen `.htaccess`-Dateien muß den Parameter `Includes` oder `IncludeNOEXEC` enthalten. Letzterer verhindert, daß CGI-Scripte irgendetwas zurückschicken können.

Im Quelltext stehen Includes in der Form

```
<!--#command tag="value" -->
```

Als `command` sind zulässig: `config`, `echo`, `include`, `fsize`, `flastmod`, `exec`.

Das Datum der letzten Änderung einer Datei läßt sich so einbinden mit:

```
<!--#flastmod file="Datei.shtml"-->
```

`exec` führt ein Bourne-Shell- oder CGI-Script aus.

Variablen wie der Dokumentname oder die URL stehen allen Dokumenten zur Verfügung.

LAMP

Mit

Linux

Apache WWW-Server

MySQL Datenbank

PHP Middleware

läßt sich ein Web-Publishing-System realisieren, ähnlich einem System aus Windows NT-Server, Internet Information Server und Microsoft-SQL-Datenbank:

In HTML-Seiten lassen sich spezielle Befehle in Kommentarform einbetten. Bei Aufruf einer derartigen Seite sucht die Middleware-Komponente des

Servers Script-Anteile heraus, führt sie aus und fügt das Ergebnis an entsprechender Stelle ein.

Beispielsweise arbeiten Gruner & Jahr und der Spiegel mit PHP.

Reihenfolge bei Selbstübersetzung:

1. MySQL übersetzen, installieren und testen. Zum Einrichten der wichtigsten Accounts gibt es ein Script auf dem ftp-Server der c't.
2. GD Für Grafikunterstützung.
3. FreeType für TrueType-Unterstützung.
4. Apache unter `/usr/local/apache` entpacken.
5. PHP übersetzen.
6. Apache übersetzen.

Zum Einbinden des PHP-Moduls beim Aufruf von `configure` den Pfad des PHP-Moduls mit dem Parameter `activate-module` angeben.

6.3 FTP-Server

Literatur:

ix 2/95 S. 174 (wu-ftpd 2.4, S.u.S.E.6.0: 2.4.2-BETA-18)
Managing Internet Information Services
(O'Reilly & Associates).

wu-ftpd stellt einen frei verfügbaren, flexibel konfigurierbaren ftp-Daemon der Washington University dar mit folgenden Erweiterungen gegenüber dem Unix-Standard:

- Protokollieren von Übertragungen
- Protokollieren von Befehlen
- On-the-fly (De-)Komprimierung
- Klassifikation von Benutzern
- klassenbezogene Beschränkungen
- verzeichnisbezogene Upload-Rechte
- Beschränkungen für Gast-Benutzer
- systemweite und verzeichnisbezogene Meldungen
- Abkürzungen für Verzeichnisse
- `cdpath`
- Dateinamensfilter
- Unterstützung virtueller Hosts

Konfiguration

Konfigurationsdateien:

```
/etc/ftpaccess  
/etc/ftpconversions
```

Benutzer lassen sich in Klassen einteilen und diesen Klassen Zugriffszeiten, maximale Nutzerzahl und erlaubte Funktionen zuordnen:

```
class RZ    guest,anonymous *.rz.tu-harburg.de  
class TUHH  guest,anonymous *.tu-harburg.de  
class Welt  guest,anonymous *  
class User  real             *
```

Ein Benutzer wird entsprechend seinem Login der ersten Klasse zugeordnet, für die sich eine Übereinstimmung seines Hostnamen oder seiner IP-Adresse mit der betreffenden Filtermaske ergibt.

Für jede Klasse können mit `limit`-Anweisungen Beschränkungen getroffen werden; Benutzer einer Klasse ohne `limit`-Anweisungen können unbeschränkt zugreifen.

```
limit TUHH 10 Any0800-1800 \  
    /etc/msg.kernzeit  
limit TUHH 20 SaSu|Any1800-0800 \  
    /etc/msg.freetime  
limit Welt 0 Any /etc/msg.closed
```

Rechner lassen sich auch ganz ausschließen:

```
deny !nameserved /etc/msg.DNSwanted
```

Ab dem erfolgreichen Einloggen auf den ftp-Server, verstehen sich alle Pfadangaben relativ zu dem Home-Verzeichnis des Benutzers `ftp`.

Meldungen

wu-ftpd unterstützt drei Arten von Nachrichten an den Benutzer:

```
banner /.prompt  
message /.welcome login  
message /.message cwd=*
```

- Der Banner wird vor dem Login-Prompt angezeigt,

- nach erfolgreichem Login kann eine kurze Einweisung des Benutzers erfolgen,
- schließlich kann der Systemadministrator auf Besonderheiten von Verzeichnissen hinweisen.

Hinweise auf Informationen können durch den Meldungstyp `readme` spezifiziert werden:

```
readme /README.DOC login
readme 00-*          cwd=*
```

In Meldungen lassen sich folgende Makros einsetzen:

- `%E` EMail-Adresse aus der `email`-Zeile in `ftpass`
- `%N` Laufende Nummer in der benutzten Klasse
- `%M` Limit der benutzten Klasse
- `%T` Aktuelle Zeit des Servers
- `%C` Aktuelles Verzeichnis
- `%R` Remote-Hostname des FTP-Benutzers
- `%U` User-Name beim Einloggen auf den Server
- `%%` das Prozentzeichen selbst

Verzeichnisangaben

```
alias latex: /pub/software/latex
```

erlaubt, von überall mit `cd latex:` in das `latex-`Verzeichnis zu wechseln.

Hat ein Administrator ein Verzeichnis `/pub/links` mit Links auf themenorientierte Verzeichnisse angelegt, läßt sich dies mit

```
cdpath /pub/links
```

integrieren; falls ein entsprechender `linux`-Link existiert, genügt dann von überall `cd linux` um in das betreffende Verzeichnis zu wechseln.

On-the-fly Konvertierungen

`wu-ftpd` unterstützt es, während des Transfers Dateien oder Verzeichnisse zu komprimieren und zu dekomprimieren.

Regeln dafür finden sich in der Datei

```
ftpconversions
```

Beispiel:

```
:.Z: : :/bin/compress -d -c %s:\
T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
: : :.Z:/bin/compress -c %s:\
T_REG:O_COMPRESS:COMPRESS
.gz: : :/bin/gzip -cd %s:\
T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
: : :.gz:/bin/gzip -9 -c %s:\
T_REG:O_COMPRESS:GZIP
: : :.tar:/bin/tar -c -f - %s:\
T_REG|T_DIR:O_TAR:TAR
: : :.tar.Z:/bin/tar -c -Z -f - %s:\
T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
: : :.tar.gz:/bin/tar -c -z -f - %s:\
T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
```

Darin besitzen die einzelnen durch : getrennten Felder folgende Bedeutung:

Feld	Bedeutung
1	abzustreifendes Prefix
2	abzustreifendes Postfix
3	anzuhängendes Prefix
4	anzuhängendes Postfix
5	auszuführendes Kommando
6	erlaubte Dateitypen
7	interne Optionen
8	Beschreibung

Eine Datei `.notar` in einem Verzeichnis bewirkt, daß dieses Verzeichnis nicht gepackt werden kann.

Die nötigen Programme sind nach `~ftp/bin` zu kopieren.

Uploads

Verzeichnisbezogene Rechte lassen sich z.B. wie folgt vergeben:

```
upload /home/ftp * no
upload /home/ftp /incoming yes \
ftp daemon 0666 nodirs
```

- Die erste Zeile verbietet Uploads pauschal,
- die zweite läßt sie für das `incoming`-Verzeichnis zu.

Im zweiten Teil der Zeile werden Benutzer, Gruppe und Rechte nach dem Upload angegeben. `nodirs` verbietet das Anlegen von Unterverzeichnissen.

Gezielt kann festgelegt werden, welche Operationen ein Benutzer ausführen darf; etwa:

```
chmod      no anonymous
delete     no anonymous
overwrite  no anonymous
rename     no anonymous
umask      no anonymous
```

Gruppen

Auch geschlossene Benutzergruppen kann `wu-ftpd` unterstützen:

Zunächst ist ein entsprechenden *Benutzer* mit der Shell `/bin/true` einzurichten.

Dieser Benutzer kann dann mit

```
guestgroup Benutzer
```

bekanntgemacht werden.

Nach dem Einloggen finden ein `chroot` zum Home-Verzeichnis des Benutzers statt.

Protokolldaten

lassen sich mit `xferstats` aus der Datei `xfer-logs` gewinnen.

6.4 NFS-Server

Zu startende Dienste:

```
rpc.portmap  RPC-Portmapper
rpc.mountd   RPC-Mount
rpc.nfsd     RPC-NFS-Daemon
```

Startscripte:

```
/sbin/init.d/rpc
/sbin/init.d/nfsserver
```

Konfigurationsdatei:

```
/etc/exports
```

Beispiel:

```
/progsys/pd      anton(rw)
/fs2/suse/cdrom *.rz.tu-harburg.de\
(ro,root_squash)
```

Für jedes zu exportierende Verzeichnis wird in einer eigenen Zeile angegeben, an welche Rechner es mit welchen Rechten exportiert werden soll; mit einem Verzeichnis werden auch alle seine Unterverzeichnisse exportiert.

Die wichtigsten Optionen:

`ro` exportiert nur mit Leserechten.

`rw` exportiert mit Schreib- und Leserechten.

`root_squash`

`root` besitzt auf diesem Verzeichnis keine für `root` typischen Sonderrechte. Dazu werden Zugriffe mit der User-ID 0 auf die User-ID 65534 (-2) umgesetzt, die dem Benutzer `nobody` zugewiesen sein sollte.

`no_root_squash`

erhält `root`-Rechte.

`link_relative`

(Standard)

setzt absolute symbolische Links (beginnend mit „/“) in relative um (beginnend mit „. . /“).

`link_absolute`

erhält symbolische Links unverändert.

`map_identity`

(Standard)

verwendet auf dem Client dieselben User-IDs wie auf dem Server.

`map_daemon`

Client und Server besitzen keine übereinstimmenden User-IDs. `nfsd` erstellt eine Umsetztabelle, dazu muß der `ugidd`-Daemon aktiv sein.

6.5 Samba-Server

Literatur:

c't 20/98 S.104
c't 24/98 S.186 (Drucken)
iX 5/98 S. 56

Offizieller Mirror für Deutschland

`ftp://ftp.uni-trier.de/pub/unix/network/samba`
aktuelle Version: `samba-latest.tar.gz`

Newsgroups:

`news:comp.protocols.smb`
`news:de.com.os.unix.networking`

Einführungen:

`http://www.zbs-ilmenau.de/~lutz/samba/`
`http://www.mnd.fh-wiesbaden.de/~dreymann/linux`

- Samba stellt File-, Printserver und WINS-Dienste (Windows Internet Naming Service) bereit.
- Samba verwendet das Windows eigene SMB-Protokoll (Server Message Block).
- Die einzige Anforderung an den Client besteht darin, daß das Netzwerk auf dem mit Windows gelieferten TCP/IP aufsetzt.

Samba-Programme und ihre Aufgaben

Der eigentliche Samba-Server besteht aus den im Hintergrund laufenden Programmen `smbd` und `nmbd`; im einzelnen:

`smbd`

Samba-Daemon, stellt Dateien und Drucker im Netz zur Verfügung und beantwortet Anfragen von Clients nach diesen Diensten; dabei wird eine Kopie des `smbd` für jeden Benutzer gestartet.

`nmbd`

NetBIOS Name Server, übersetzt NetBIOS-Namen in IP-Adressen und verwaltet als Browser eine Liste von Netzwerk-Ressourcen, die den Clients angezeigt werden.

`smbclient`

ftp ähnliches Programm, das den Zugriff auf SMB-Ressourcen erlaubt: z.B. Anzeigen der Browse-Liste eines Client,
(`smbclient -L Servername`)

Aufnehmen einer Verbindung zu einem Server.

Läuft auf den Clients winpopup, können mittels smbclient Meldungen zu einzelnen oder allen angeschlossenen Clients gesandt werden.

`nmblookup`

Testwerkzeug für `nmbd`.

`addtosmbpass`

trägt Benutzer auf dem Server ein.

`smbpasswd`

ändert das Paßwort eines eingetragenen Benutzers.

`testparm`

prüft `/etc/smb.conf` auf Syntaxfehler und gibt die Parameter aus, mit denen der Samba-Server tatsächlich aufgerufen wird.

`testprns`

Werkzeug zum Test der Druckerkonfiguration.

`smbstatus`

zeigt alle aktuell angemeldeten Benutzer und die von ihnen bearbeiteten Dateien an.

`smbtar`

für ein zentrales Backup müssen die Clients ihre Platten an den Server exportieren. `smbtar` lädt die Daten zur Sicherung herunter. Unter Linux können die Platten direkt gemountet werden, wenn bei der Kernelkonfiguration das `smbfs`-Dateisystem mit ausgewählt wurde.

`make_smbcodepage`

zur besseren Unterstützung von Dateinamen mit nationalen Sonderzeichen kann `make_smbcodepage` ladbare Codepage-Definitionen lesen und erzeugen.

`make_printerdef`

unterstützt die automatische Installation von Druckertreibern bei Windows '95 Clients.

Planung

Hardware-Anforderungen

- für 10-20 Benutzer genügt ein 486er mit 32MByte RAM
- RAM und schnelle Festplatten sind wichtiger als CPU-Leistung
- SCSI-Hardware ist empfehlenswert (*zuverlässige Backup-Systeme*)

Verzeichnisse

- Home-Verzeichnisse (`/home/...`)
- Team-Verzeichnisse (`samba/team/...`)
- globales temp-Verzeichnis (`/var/tmp/...`)
- Programm-Verzeichnis (`samba/prog`)

Installation

RPM-Paket einspielen

In `/etc/services` müssen erklärt sein:

```
netbios-ns  137/tcp  # NETBIOS Name
netbios-ns  137/udp  # Service

netbios-dgm 138/tcp  # NETBIOS Datagram
netbios-dgm 138/udp  # Service

netbios-ssn 139/tcp  # NETBIOS session
netbios-ssn 139/udp  # service
```

Standardmäßig werden die Server-Daemons gestartet, sobald der `inetd`-Prozeß Aktivitäten auf den passenden Netzwerk-Ports feststellt. Sollte dies zu lange dauern, kann mit `smbaconfig` auf „run as daemons“ umgestellt werden.

Init-Script zum automatischen Start von Samba:

```
#!/bin/sh
# Dieses Programm als /etc/rc.d/smb
# bzw. /etc/init.d/smb anlegen.
# Links nach /etc/rc.d/rc2.d/K20smb und
# S20smb bzw. /etc/rc2.d/K20smb und S20smb
SBINDIR=/usr/sbin
case "$1" in
    start)
        echo -n "Starting SMB services."
        $SBINDIR/nmbd -D
        $SBINDIR/smbd -D
        echo
        ;;
    stop)
        echo -n "Shutting down SMB services."
        killall -TERM nmbd # Linux-killall,
        killall -TERM smbd # nicht SVR4-killall
        echo
        ;;
    *)
        echo "Usage: $0 {start|stop}"
        exit 1
esac
exit 0
```

Konfiguration

Die Konfiguration von Samba erfolgt über die Datei `/etc/smb.conf`, zu *Testzwecken* etwa:

```
/etc/smb.conf
```

```
[global]
workgroup = ARBEITSGRUPPE
guest account = nobody
```

```
[test]
comment = Testshare
path = /tmp
writeable = yes
public = yes
```

- Diese Datei wird nur beim Start der Serverprogramme eingelesen.
- Nach jeder Änderung an der Konfiguration müssen die Prozesse `nmbd` und `smbd` beendet und neu gestartet werden.
- Grundsätzlich beschreibt jeder *[Abschnitt]* eine freigegebene Resource, ein sog. *Share*, z.B. ein Verzeichnis oder einen Drucker.

Abschnitte mit besonderer Bedeutung sind `[global]`, `[home]` (Benutzerverzeichnisse) und `[printers]` (Drucker).

- `[global]` definiert globale Voreinstellungen für das gesamte Samba.
- Ein Gast-Account muß vorhanden sein, damit Samba ordnungsgemäß funktioniert.
- `[test]` definiert ein sog. *File-Share*, das das `/tmp`-Verzeichnis exportiert, beschreibbar und ohne Paßwort unter dem Gast-Account verfügbar. Der Gast-Account muß auf dem Server existieren. Ein File-Share erscheint auf einem Windows-Client als Laufwerk.

Test

Server-seitig

```
smbclient -L Servername
```

muß `IPC$` und `test` auflisten.

Mit

```
smbclient \\Servername\test
```

sollte eine Verbindung ohne Paßwortabfrage möglich sein.

Client-seitig

- `ping` mit IP-Adresse und Namen
- Netzlaufwerk verbinden `\\Servername\test`
- Anzeige des Servers in der Netzwerkumgebung

Benutzerverzeichnisse

```
[homes]
comment = Home-Verzeichnis von %U
; path = wird automatisch ermittelt
browseable = no
public = no
read only = no
force user = %U
```

- Statt `homes` erscheint ein Eintrag für den jeweiligen Benutzer (`%U`) in der Browse-Liste.
- `force user` bewirkt, daß alle Schreib- und Leszugriffe unter der Kennung des Benutzers erfolgen, mit der sich der Client bei dem Server angemeldet hat.
- Samba kann Home-Shares aus NIS-Tabellen erzeugen.

Verzeichnisse, CD-ROM

```
[Name]
comment = Beschreibung
path = freizugebendes Verzeichnis
read only = [yes/no]
browseable = [yes/no]
public = [yes/no]
directory mode = 750
create mode = 640
```

```
[cdrom]
comment = CD-ROM Laufwerk
path = /cdrom
read only = yes
public = yes
allow hosts = aaa.bbb.ccc.ddd/255.255.255.0
```

Dateinamen

`case sensitive = no`

keine Unterscheidung zwischen Groß- und Kleinschreibung.

`mangle case = no`

`serve case = yes`

`preserve case = yes`

Originalschreibweise beim Abspeichern.

Datumsangaben

`dos filetime resolution on = yes`

`dos filetimes = yes`

Zwei-Sekunden-Auflösung.

`time server = yes`

Samba-Server als Zeitsynchronisationsquelle.

Drucker

Minimale `/etc/smb.conf`:

```
[global]
workgroup = ARBEITSGRUPPE
guest account = smbuser
printing = bsd
load printers = yes

[printers]
path = /tmp
printable = yes
public = yes
```

- Mit `load printers = yes` werden alle Drucker aus `/etc/printcap` verfügbar.
- Eingehende Druckaufträge werden zunächst im `path`-Verzeichnis gespeichert und von Samba weitergeleitet.
- `smbuser` muß auf dem System eingerichtet sein, Schreibrechte im Spool-Directory und dem als `path` angegebenen Verzeichnis besitzen.
- Im `[printers]`-Abschnitt ist es in Verbindung mit `printable = yes` möglich, zu setzen:

```
read only = yes
```

Um Benutzern gezielt einzelne Drucker zur Verfügung zu stellen, können Drucker-Shares definiert werden:

```
[global]
:
printing = bsd
; Drucker einzeln freigeben:
load printers = no

[Drucker]
; Name aus /etc/printcap
printer name = Drucker
comment = Druckerbeschreibung
browseable = yes
path = /tmp
printable = yes
print command = /usr/bin/lpr -P%p %s
; Zugriff
public = yes
allow hosts = aaa.bbb.ccc.ddd/255.255.255.0
```

- Mit `deny hosts =` kann ein Drucker-Share auch gezielt verwehrt werden.

Es ist möglich, Server-seitig Druckertreiber für Windows '95/'98-Clients zur Verfügung zu stellen:

```
[global]
:
printing = bsd
; Drucker einzeln freigeben:
load printers = no
; autom. Treiberinstallation:
printer driver file = \
    /usr/share/samba/printers.def

[Drucker]
; Name aus /etc/printcap
printer name = Drucker
comment = Druckerbeschreibung
browseable = yes
path = /tmp
printable = yes
print command = /usr/bin/lpr -P; Zugriff
public = yes
allow hosts = aaa.bbb.ccc.ddd/255.255.255.0
; autom. Treiberinstallation
printer driver location = \\Server\driver$
printer driver = Druckermodell

[driver$]
path=/usr/share/samba/Treiberverzeichnis
browsable=no
public=yes
writable=no
```

Vorgehensweise:

1. Verzeichnis bereitstellen, in dem die Treiberdateien abgelegt werden.
 - Das entsprechende System-Share (`driver$`), gekennzeichnet durch `$`, erscheint nicht in einer Liste freigegebener Ressourcen.
2. Benötigte Treiberdateien ermitteln:

```
recode ibmpc:latin1 DRUCKER.INF  
make_printerdef DRUCKER.INF Druckermo-  
dell >> printers.def
```

trägt den Druckertreiber in `printers.def` ein und zeigt eine Liste der benötigten Dateien an.

Diese Dateien sind in das bereitgestellte Verzeichnis zu kopieren.

3. Zusätzliche Einträge in `/etc/smb.conf` vornehmen, s.o.

Zugriffskontrolle

Benutzer können auf einem Samba-Server eingerichtet werden

- als normale Benutzer
- nur zur Benutzung von Samba mit `/bin/false` als Shell.

Da die Formate der Paßwortdateien von Unix und Samba nicht identisch sein können, sind die Windows-Paßwörter der Benutzer in einer gesonderten Datei

```
/etc/smbpasswd
```

zu hinterlegen.

`mksmbpasswd.sh` kann den Rumpf dieser Datei aus der normalen Unix-Paßwortdatei erzeugen.

Konfigurationsmöglichkeiten

```
encrypt passwords = yes
```

verschlüsselte Paßwort-Übertragung (für Windows NT 4.0 ab Service Pack 3 und Windows 98 standardmäßig erforderlich, von Windows 3.11 nicht unterstützt).

```
allow hosts =  
aaa.bbb.ccc.ddd/255.255.255.0
```

Einschränkung der zulässigen Clients auf das angegebene Netz.

Tuning

```
debug level = 1  
minimales Logging.
```

```
max log size = 1000  
Begrenzung der Größe einer Log-Datei (in  
KByte).
```

```
fake oplocks = no  
oplocks = yes
```

Opportunistische Locks, Dateisperren die nachgeben können: Der Inhaber eines Locks wird informiert, wenn von anderer Seite Operationen auf dem gesperrten Bereich durchgeführt werden sollen, und kann entscheiden, ob er sein Lock aufgeben möchte. Dies erlaubt aggressives Cachen von Schreibzugriffen.

```
read prediction = yes  
vorausselnde Lesezugriffe.
```

```
read raw = yes
write raw = yes
```

Beschleunigung von Lese- und Schreibzugriffen auf lange Dateien. Bei NE2000 Karten u.U. Beschränkung erforderlich mit

```
max xmit = 8192
```

```
soecket options = TCP_NODELAY
```

keine Pufferung kleiner Netzwerkblöcke.

```
getwd chace = yes
```

Cache für Verzeichnis-Lookups.

```
dead time = 15
```

Verbindungen nach angegebener inaktiver Zeit (in Minuten) vorübergehend trennen, um Anzahl der Serverprozesse klein zu halten. Erneuter Verbindungsaufbau erfolgt automatisch.

Weitere Hinweise finden sich in der Datei `Speed.txt` des `docs`-Verzeichnisses.

Wichtige Optionen der Datei `smb.conf`

`allow hosts=Rechnerliste`

Clients, die dieses Share benutzen dürfen.

`browseable=[yes/no]`

Soll das Share in der Liste der freigegebenen Ressourcen auftauchen?

`comment=String`

Kommentar in der Ressourcen-Liste.

`debug level=[0/1/2/3]` (G)²

Informationen und Fehlermeldungen werden in `/var/log/smb` und `/var/log/nmb.log` mitprotokolliert.

`deny hosts=Rechnerliste`

Clients, die dieses Share nicht benutzen dürfen.

`guest account=String` (G)

Benutzername, der bei anonymen Aufträgen benutzt wird.

`invalid users=Benutzerliste`

Benutzer, die dieses Share nicht benutzen dürfen.

²[global]-Abschnitt

`print command=String`

Unix-Kommandozeile zum Abschicken eines Druckjobs.

`%s` wird durch den Namen der Druckdatei ersetzt,
`%p` durch den der Queue

`load printers=[yes/no]` (G)

Sollen alle in `/etc/printcap` definierten Queues freigegeben werden?

`log file=Dateiname` (G)

Log-Datei.

`lpq command=String`

Unix-Kommandozeile zur Anzeige der Aufträge in der Queue.

`lprm command=String`

Unix-Kommando zum Löschen eines Druckauftrages.

`max log size=KByte` (G)

Maximale Größe der Log-Datei.

`path=Pfadname`

Verzeichnis, das in dem Share freigegeben wird.

`printable=[yes/no]`

Das Share beschreibt einen Drucker.

`printcap name=Dateiname` (G)

Pfadname der Datei `printcap`.

`printer driver=String`

Name eines Windows-Druckertreibers.

`printer driver file=Dateiname` (G)

Verzeichnis der verfügbaren Windows-Druckertreiber.

`printer driver location=SMB-Pfadname`

Verzeichnis mit Windows-Druckertreibern.

`printer name=String`

Name der Queue, in die die Druckaufträge gestellt werden.

`printing=[bsd/sysv/hpux/aix/qnx/plp]` (G)

Typ des installierten Drucksystems.

`public=[yes/no]`

Soll jeder Benutzer auf das Share zugreifen dürfen?

`server string=String` (G)

Beschreibung des Servers für die Ressourcenliste.

`valid users=Benutzeerliste`

Benutzer, die dieses Share benutzen dürfen.

`writable=[yes/no]`

Sollen Benutzer auf dieses Share schreiben dürfen?

`workgroup=String` (G)

Arbeitsgruppe, in der der Server erscheint.

Einsatz von Samba als WINS-Server

Besteht das Netz aus mehreren Broadcast-Domänen?

wenn nein,

kein WINS notwendig (aber möglich),
kein Domain Masterbrowser notwendig.

wenn ja,

WINS notwendig,
ein lokaler Masterbrowser je Subnetz notwendig,
ein Domain Masterbrowser notwendig.
In diesem Fall:

Windows NT Primary Domain Controller (PDC) im Netz?

wenn ja,

im Subnetz des PDC:

```
win server = aaa.bbb.ccc.ddd  
os level = 0  
preferred master = no  
local master = no  
domain master = no
```

in anderen Subnetzen:

```
win server = aaa.bbb.ccc.ddd  
os level = 65  
preferred master = yes  
local master = yes  
domain master = no
```

wenn nein:

Samba als Masterbrowser?

wenn ja,

in einem Subnetz:

```
wins support = yes  
os level = 65  
preferred master = yes  
local master = yes  
domain master = yes
```

in allen anderen Subnetzen:

```
win server = aaa.bbb.ccc.ddd  
os level = 65  
preferred master = yes  
local master = yes  
domain master = no
```

wenn nein,

```
wins server = aaa.bbb.ccc.ddd  
os level = 0  
preferred master = no  
local master = no  
domain master = no
```

- Der WINS-Server ist unter Windows bei Netzwerkeigenschaften einzutragen.
- In einer NT-Domain muß der Domain Masterbrowser mit dem Windows NT Primary Domain Controller identisch sein; Samba kann dort nicht zum Domain MasterBrowser werden.
- Ein `os level = 2` gibt Samba Vorrang vor Windows 95, ein Wert `> 32` Vorrang vor Windows NT.

6.6 Hinweise zu Samba

SWAT – Samba Web Administration Tool

Unter Linux:

1. in `/etc/inetd.conf` das Kommentarzeichen vor der entsprechenden Zeile entfernen.

2. `inetd` mittels

```
kill -1 pid
```

neu starten (`pid aus ps aux | grep inetd`).

3. Aufruf von:

```
http://localhost:901
```

Konfigurationsdatei

```
[global]
```

```
workgroup = SAMBA
```

Name der Domäne

```
netbios name = LINSE
```

Name des Servers

```
encrypt passwords = Yes
```

```
map to guest = Bad User
```

Für eingetragene Benutzer werden Versuche, sich mit falschem Passwort anzumelden, zurückgewiesen. Nicht eingetragene Benutzer werden als Gäste behandelt.

```
socket options = TCP_NODELAY IPTOS_LOWDELAY
```

Tuning-Optionen

```
logon script = logon.bat
```

wird nach erfolgreicher Anmeldung eines Benutzers ausgeführt.

```
logon path = \\%L\profiles\%U\%a
```

Verzeichnis, in dem die Roaming Profile (USER.DAT/USER.MAN unter Windows 95 /98) gespeichert werden. Windows-Rechner können gelegentlich Verbindungen zu dem [homes]-Share halten, selbst wenn kein Benutzer angemeldet ist. Aus diesem Grund sollte keinesfalls \\%N\HOMES\... gewählt werden.

logon drive = H:

Laufwerk, mit dem das Home-Verzeichnis des Benutzers verbunden wird (nur für Windows NT).

domain logons = Yes

Server für Windows 95/98 Domänen (erfordert nicht die Funktionalität eines PDC)

os level = 65

dieser Rechner wird (gegenüber Windows-Rechnern) vorrangig als lokaler Master gewählt.

preferred master = Yes

empfohlen in Verbindung mit domain master = Yes

domain master = Yes

In einer Domäne ohne Windows NT PDC (Primary Domain Controller) kann ein Samba-Rechner die Funktion Domänen-Controllers übernehmen. In einer Domäne mit Windows NT PDC kann dies zu Problemen führen.

wins support = Yes

Diese Variable sollte nur dann auf true gesetzt werden, wenn sich ein Netzwerk über mehrere Subnetze erstreckt. Die Variable sollte niemals auf mehreren Rechnern eines Netzwerkes auf true gesetzt werden.

dos filetime resolution = Yes

2-Sekunden-Auflösung

dos filetimes = Yes

Ändern des Dateidatums

time server = Yes

Samba-Server als Zeitserver

[homes]

```
comment = Heimatverzeichnis von %U
read only = No
create mask = 0750
browseable = No
force user = %U
```

[printers]

```
comment = alle Drucker
path = /tmp
create mask = 0700
print ok = Yes
browseable = No
```

[netlogon]

```
comment = Anmeldeskripte
path = /usr/lib/samba/logon
browsable = No
```

[profiles]

```
comment = Benutzerumgebungen
path = /usr/lib/samba/profiles
read only = No
create mask = 0600
directory mask = 0700
browsable = Yes
```

wichtig für Windows 95/98!

Einrichten eines Maschinenaccounts

Für jeden Rechner, der Teil einer Domäne werden soll, muß einmalig ein Maschinen-Account auf dem PDC eingerichtet werden.

in `/etc/passwd`:

```
rechner$:x:801:800:Beschreibung:\  
/dev/null:/bin/false
```

in `/etc/shadow`:

```
rechner$:*:10896:0:10000::::
```

für `/etc/smbpasswd` (bei laufendem Samba):

```
# smbpasswd -a -m rechner
```

(ohne „\$“)

Einrichten eines Benutzeraccounts

Übernehmen aller Benutzer aus `/etc/passwd`:

1. Gerüst:

```
# cat /etc/passwd | \  
/usr/sbin/mksmbpasswd.sh >> /etc/smbpasswd
```

2. nicht benötigte Einträge in `/etc/smbpasswd`
entfernen

3. Passwort setzen

```
# smbpasswd account
```

Einrichten eines neuen Benutzers:

1. Einrichten eines neuen Benutzers mit `yast`

2. Übernehmen des Gerüsts

```
# cat /etc/passwd | \  
/usr/sbin/mksmbpasswd.sh | grep account \  
>> /etc/smbpasswd
```

3. Passwort setzen

```
# smbpasswd account
```

7 Sicherheit

MIT-Magic-Cookie

Arbeiten auf einem entfernten Rechner

Will man die Ausgabe einer X-Anwendung eines entfernten Rechner auf dem lokalen X-Server ausgeben lassen, ist in jedem Fall auf dem entfernten Rechner die `DISPLAY`-Variable auf den lokalen X-Server zu setzen.

Lokaler Rechner: `local.rz`, XServer: `local.rz:0`
Entfernter Rechner: `remote.rz` (Systemausgaben sind durch `>` gekennzeichnet)

Ablauf mit `xhost`:

<code>local.rz</code>	<code>remote.rz</code>
<code>telnet remote.rz</code>	
<code>xhost +remote.rz</code>	
<code>> remote.rz being added to access control list</code>	
	<code>export DISPLAY=local.rz:0</code>
	<code>xterm</code>

Ablauf mit `xauth` bei gemeinsamem HOME-Verzeichnis:

local.rz	remote.rz
<code>telnet remote.rz</code>	<code>export DISPLAY=local.rz:0</code> <code>xterm</code>

Ablauf mit `xauth` bei unterschiedlichen HOME-Verzeichnissen:

(kein Zugriff auf die `.Xauthority`-Datei des Rechners mit dem XServer):

local.rz	remote.rz
<code>xauth list</code>	
<code>> local.rz:0 MIT-MAGIC-COOKIE-1 64872634827634826345</code>	
<code>> local/unix:0 MIT-MAGIC-COOKIE-1 64872634827634826345</code>	
<code>telnet remote.rz</code>	<code>export DISPLAY=local.rz:0</code> <code>xauth add \$DISPLAY . 64872634827634826345</code> <code>xterm</code>

Firewall / IP-Masquerading

Literatur:

iX 7/96 S.128
c't 21/98 S.288
c't 3/99 S.156

Siehe `/etc/rc.config.d/firewall.rc.config`

Überwachungs-/Testsoftware

Software zur Überwachung Verdächtiger:

`ttywatcher`

`ftp://coast.cs.purdue.edu/pu/tools/unix/\`
`ttywatcher`

`xkey`

`http://www.nrmc.org/files/unix/xkey.c`

`karpski`

komfortable Variante von `tcpdump` zur Anzeige von tcp-Paketen.

TAMU

Sammlung alter, zum Teil immer noch nützlicher Tools

COPS

früher weit verbreitet, lehrreich.

Saint

Weiterentwicklung von SATAN

Nessus

ähnlich Saint

Crack

effizient zum Aufspüren schwacher Paßwörter.

Empfehlungen

- in `/etc/inetd.conf` alle Dienste auskommentieren, die nicht unbedingt erforderlich sind!
- Verbindung zu entfernten Rechnern:
 - Nie `xhost +` einsetzen.
 - Auch `xhost +Rechner` vermeiden.

- Besser `xauth`-Mechanismus verwenden.
- Empfehlenswert ist der Einsatz der `ssh`.
- `tcpwrapper` für `inetd`-kompatible Dienste benutzen; für nicht RPC-Dienste Port Mapper.
- `tripwire` kann zur Prüfung der Systemintegrität eingesetzt werden; damit lassen sich hinzugefügte, modifizierte oder gelöschte Dateien feststellen. Graphische Oberfläche: Merlin. Anleitung bei DFN-CERT.