



# **Linuxseminar - Serveradministration**

©2005 Rainer Kulhanek





# Inhaltsverzeichnis

<a href="#">_Bootloader.....</a>	<a href="#">1</a>
<a href="#">_System V Init.....</a>	<a href="#">5</a>
<a href="#">/etc/inittab.....</a>	<a href="#">9</a>
<a href="#">_Runlevel.....</a>	<a href="#">11</a>
<a href="#">_FileSystemTABelle.....</a>	<a href="#">13</a>
<a href="#">_mounten von Dateisystemen.....</a>	<a href="#">15</a>
<a href="#">_AutoFileSystem.....</a>	<a href="#">17</a>
<a href="#">_Benutzer.....</a>	<a href="#">19</a>
<a href="#">_Benutzer II.....</a>	<a href="#">21</a>
<a href="#">_Gruppenverwaltung.....</a>	<a href="#">23</a>
<a href="#">_Zugriffsrechte.....</a>	<a href="#">25</a>
<a href="#">_AccessControllist.....</a>	<a href="#">27</a>
<a href="#">_Module erstellen und installieren.....</a>	<a href="#">29</a>
<a href="#">_Neuen Kernel erstellen und installieren.....</a>	<a href="#">31</a>
<a href="#">_Prozesse.....</a>	<a href="#">33</a>
<a href="#">_Network File System.....</a>	<a href="#">35</a>
<a href="#">_DNS.....</a>	<a href="#">37</a>
<a href="#">_Samba.....</a>	<a href="#">39</a>
<a href="#">_Samba - Client.....</a>	<a href="#">43</a>
<a href="#">_Samba-Benutzer.....</a>	<a href="#">45</a>
<a href="#">_NIS-Client NIS-Client.....</a>	<a href="#">47</a>
<a href="#">_NIS-Server.....</a>	<a href="#">49</a>
<a href="#">_NTP - Dienst.....</a>	<a href="#">51</a>
<a href="#">_net - Optionen (samba 3.0).....</a>	<a href="#">53</a>



# Inhaltsverzeichnis

Netzwerk.....55



# Bootloader

Gebräuchlich sind LILO und GRUB

## LILO

Konfigurationsdatei: /etc/lilo.conf

Beispiel:

```
# LILO configuration file
# Start LILO global Section
# If you want to prevent console users to boot with init=/bin/bash,
# restrict usage of boot params by setting a passwd and using the option
# restricted.
#password=bootpwd
#restricted
append="max_scsi_luns=1"
boot=/dev/hda
#compact # faster, but won't work on all systems.
vga=normal
message=/boot/message
menu-scheme=Wg:kw:Wg:Wg
read-only
prompt
timeout=100
# End LILO global Section
#
image = /boot/vmlinuzv root = /dev/hda2
label = Linux
initrd = /boot/initrd
#
image = /boot/vmlinuz.park
root = /dev/hda2
label = 2.4.18-4GB
initrd = /boot/initrd
#
image = /boot/vmlinuz
root = /dev/hda2
label = 70
initrd = /boot/initrd
```

Aufruf: lilo

Schreibt den Bootloader an den in /etc/lilo.conf mit **boot** festgelegten Ort.

## GRUB

## **Titel: Das Bootloader-Konzept in der SuSE Linux Version 8.1**

Original unter [http://sdb.suse.de/de/sdb/html/fhassel\\_bootld\\_cpt.html](http://sdb.suse.de/de/sdb/html/fhassel_bootld_cpt.html)

Bezieht sich auf SuSE Linux: Version 8.1

### **Anliegen**

In diesem Artikel soll kurz erläutert werden, warum ab SuSE Linux Version 8.1 als Bootmanager Grub anstelle von Lilo verwendet wird.

### **Hintergrund**

Ab SuSE Linux 8.1 hat sich das Bootloader-Konzept grundlegend geändert. Bei einer Neuinstallation wird als Bootloader nun Grub eingerichtet. Folgende Gründe sprachen für einen Wechsel zu Grub:

1. Grub bietet bei Bedarf noch vor dem Booten eine Betriebssystem-ähnliche Umgebung
2. Grub kann eine Vielzahl von Betriebssystemen booten (neben Linux, Windows, OS/2 und BeOS auch einige freie Unix-Betriebssysteme)
3. Durch Direktzugriff auf Dateisysteme ist eine Neuinstallation von Grub nicht erforderlich, wenn die Bootloader-Konfiguration bzw. Kernel und Initrd geändert werden
4. Noch vor dem Booten ist Zugriff auf Daten möglich
5. Grub kann die zum Booten erforderlichen Dateien bei Bedarf über das Netzwerk laden
6. Grub bietet für Terminals ohne Bildschirm eine Kontrolle über die serielle Leitung
7. Grub ist für unsere Entwickler einfacher zu pflegen
8. Die United Linux Partner benutzen Grub als Bootloader

Bei einem Update von einer älteren SuSE Linux Version wird Lilo als Bootmanager beibehalten. Dies ist ebenso der Fall, sofern die Root-Partition auf einem Raid System (auch bei Software-Raid oder LVM) installiert wird.

Obwohl Grub nun der Standard ist, können Sie dennoch, falls Sie wünschen, zurück zu Lilo wechseln. Die Vorgehensweise dazu ist beschrieben im Artikel "LILO anstatt GRUB in SuSE 8.1 als Bootloader benutzen" ([http://sdb.suse.de/de/sdb/html/fhassel\\_grub\\_lilo.html](http://sdb.suse.de/de/sdb/html/fhassel_grub_lilo.html)).

Hintergründe zur Funktionsweise des Bootmanagers Grub sowie Hinweise zur Konfiguration sind erläutert im Artikel "Der Bootmanager GRUB" ([http://sdb.suse.de/de/sdb/html/fhassel\\_grub\\_overview.html](http://sdb.suse.de/de/sdb/html/fhassel_grub_overview.html)). Hier finden Sie auch Verweise auf weitere Artikel im Zusammenhang mit Grub.

Konfigurationsdateien:

**/etc/grub.conf:**

```
root (hd0,4)
install --stage2=/boot/grub/stage2 /grub/stage1 d (fd0) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
quit
```

**/boot/grub/menu.lst:**

```
# Modified by YaST2. Last modification on Sat Nov 23 23:58:59 2002
```

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
vga 791
```

```
title linux
kernel (hd0,4)/vmlinuz root=/dev/hda9
initrd (hd0,4)/initrd
```

```
title windows
root (hd0,0)
makeactive
chainloader +1
```

```
title dos
root (hd1,0)
makeactive
chainloader +1
```

```
title floppy
root (fd0)
chainloader +1
```

```
title failsafe
kernel (hd0,4)/vmlinuz.shipped root=/dev/hda9 ide=nodma apm=off acpi=off vga=normal nosmp
maxcpus=0 disableapic 3
initrd (hd0,4)/initrd.shipped
```

**/boot/grub/device.map:**

```
(fd0) /dev/fd0
(hd0) /dev/hda
(hd1) /dev/sda
```





# System V Init

## BIOS

Nach dem Ablauf verschiedener Selbsttests wird das (evtl. im BIOS einstellbare) Bootmedium gelesen. Hier wird bei der Intelarchitektur aus dem sog. Masterbootrecord (MBR) ein Teil eines sogenannten Bootloaders (Ladeprogramm) gelesen und ausgeführt. Dieser liest den restlichen Teil des Bootloaders. Hier sind je nach Bootloader verschiedene Auswahlmöglichkeiten etc. üblich. Letztendlich wird der Kernel in den Speicher geladen und ausgeführt.

## Kernel und init

### Kernel

Der Kernel startet den ersten Prozess mit dem Namen init.

### Initrd

In dieser Datei befindet sich ein Dateisystem [\\*1](#) mit einigen für den Bootvorgang notwendigen Dateien. Diese sind in einer normalen Verzeichnisstruktur abgelegt. Hierdurch besteht die Möglichkeit, Funktionalitäten die **nicht** im Kernel enthalten sind wie z.B. spezielle Netzwerktreiber, aber auch Treiber für Dateisysteme zu benutzen obwohl die entsprechenden Module noch nicht geladen werden können, da sie sich auf einem Device (Netzwerk oder Partition) befinden das noch nicht gemountet wurde. So könnte z.B. die Kerneldatei vmlinuz und die initrd auf einer Partition liegen, welche nach dem Beenden des Bootvorganges dem Verzeichnis /boot entspricht. Die root-Partition ist dem Filesystem Reiserfs und die /usr-Partition mit xfs eingerichtet. Beide Dateisysteme würden nicht vom Kernel unterstützt. Sofern die Treiber sich als Module in initrd befinden, können beide Partitionen angesprochen und gemountet werden.

Die Datei initrd befindet sich im Verzeichnis /boot. Es handelt sich um einen Link auf die jeweils gültige Datei mit Namen initrd-[Kernelversion]-Verwendung.

Beispiele:

```
lrwxrwxrwx 1 root root 26 23. Mär 20:55 initrd -> initrd-2.6.18.2-34-default
-rw-r--r-- 1 root root 2907802 23. Mär 20:55 initrd-2.6.18.2-34-default
-rw-r--r-- 1 root root 2813566 23. Mär 20:55 initrd-2.6.18.2-34-xen
lrwxrwxrwx 1 root root 22 23. Mär 20:30 initrd-xen -> initrd-2.6.18.2-34-xen
```

Übung: Den Inhalt der Datei /boot/initrd ansehen [\\*3](#).

## Der Prozess mit der Nummer 1 - init

Der erste wird der letzte sein oder der Vater aller Dinge.

Dieser Prozess ist der Vater aller weiteren Prozesse und bis zum endgültigem Halt des Systems vorhanden. Die Konfiguration erfolgt durch die Datei

## /etc/inittab

(Beispiel s. [inittab](#)). Hier finden sich Angaben in welchen [Runlevel](#) das System booten soll. Wieviele virtuelle Konsolen angesteuert werden, welche Programme bei Nachrichten einer evtl. vorhandenen USV (Unterbrechungsfreie Stromversorgung) gestartet werden sollen oder auch was die Tastenkombination Steuer,Alternative,Entfernen bewirken soll.

## /etc/init.d/boot

Die erste Datei, die der init-Prozess ausführt. Hier findet man u.a. Aufrufe von Startskripten wie z.B. boot.ipconfig (Setzt IP-Adresse etc.). Der Mechanismus der die Aufrufreihenfolge bestimmt ist der gleiche wie bei den Run-Level-Skripten [\\*2](#).

## /etc/init.d/boot.local

Da die Datei `/etc/init.d/boot` bei einem Update etc. geändert oder neu erstellt werden könnte, sollte man eigene Eintragungen in die Datei `boot.local` schreiben. Diese Datei wird bei Änderungen des Systems nicht beeinträchtigt. `boot.local` wird **nach** den o.g. boot-Dateien jedoch vor den Wechsel in den eigentlichen Runlevel abgearbeitet.

## /etc/fstab

... ist [dort](#) beschrieben.

## Runlevel

... ist [dort](#) beschrieben.

## Module

noch nicht erstellt.

\*1 D. h. die Datei ist als xxx-system strukturiert. Siehe auch [unter Dateien als Dateisysteme](#).

\*2 Softlinks in `/etc/init.d/boot.d/` mit der Namenskonvention `S[Zahl][Name]` werden in Reihenfolge der Zahlen gestartet. Siehe auch [Runlevel](#)

\*3 Arbeitsverzeichnis erstellen: `mkdir /work`

**Kopie** von der Orginaldatei erstellen: `cp /boot/initrd /work/initrd`

in Arbeitsverzeichnis wechseln: `cd /work`

(gültig bis Suse-Version 9.2)

Als gzipte Datei behandeln: `mv initrd initrd.gz`

Dekomprimieren: `gzip -d initrd.gz`

Datei als Dateisystem mounten: `mount -o loop initrd /mnt`

Inhalt ansehen: `ls /mnt`

(ab 9.3 handelt es sich um cpio-Archiv)

Dekomprimieren und das cpio-Archiv entpacken:

`gunzip -c -9 /boot/initrd | cpio -i -d -H newc --no-absolute-filenames`

|| \*







## /etc/inittab

In der Datei /etc/inittab wird u.a. definiert:

Auszug:

```
# The default runlevel is defined here
```

```
id:5:initdefault: Hier wird der Runlevel, in welchen gebootet wird, definiert
```

```
# First script to be executed, if not booting in emergency (-b) mode
```

```
si::bootwait:/etc/init.d/boot Das erste Script, welches gestartet wird. Hier meist eine Verzweigung zu boot.local u.a.
```

```
# /etc/init.d/rc takes care of runlevel handling Die einzelnen Level
```

```
#
```

```
# runlevel 0 is System halt (Do not use this for initdefault!)
```

```
# runlevel 1 is Single user mode
```

```
# runlevel 2 is Local multiuser without remote network (e.g. NFS)
```

```
# runlevel 3 is Full multiuser with network
```

```
# runlevel 4 is Not used
```

```
# runlevel 5 is Full multiuser with network and xdm
```

```
# runlevel 6 is System reboot (Do not use this for initdefault!)
```

```
#
```

```
Die Anzahl und "Belegung" der virtuellen Konsolen
```

```
l0:0:wait:/etc/init.d/rc 0
```

```
l1:1:wait:/etc/init.d/rc 1
```

```
l2:2:wait:/etc/init.d/rc 2
```

```
l3:3:wait:/etc/init.d/rc 3
```

```
#l4:4:wait:/etc/init.d/rc 4
```

```
l5:5:wait:/etc/init.d/rc 5
```

```
l6:6:wait:/etc/init.d/rc 6
```

```
Der Singleuser-Modus
```

```
# what to do in single-user mode
```

```
ls:S:wait:/etc/init.d/rc S
```

```
~~:S:respawn:/sbin/sulogin
```

```
.....
```

```
# getty-programs for the normal runlevels
```

```
# <id>:<runlevels>:<action>:<process>
```

```
# The "id" field MUST be the same as the last
```

```
# characters of the device (after "tty")
```

```
1:2345:respawn:/sbin/mingetty --noclear tty1
```

```
2:2345:respawn:/bin/login -f rk </dev/tty2 2>&1 Hier wird der Benutzer rk automatisch eingeloggt.
```

```
3:2345:respawn:/bin/su root -c top </dev/tty3 2>&1 Hier wird das Programm top unter dem Benutzer root gestartet.
```

```
4:2345:respawn:/sbin/mingetty tty4
```

```
5:2345:respawn:/sbin/mingetty tty5
```

```
6:2345:respawn:/sbin/mingetty tty6
```





## Runlevel

Durch Zusammenfassen verschiedener Dienste und Funktionen zu einem Runlevel kann einfach die Rolle und der Zustand eines Rechners definiert werden. (Beispiele: Runlevel 1 = Single User Modus, nur zur Administration durch den Systemverwalter, oder Einsatz als Mailserver ohne eigene Nutzerverwaltung etc.). Die Runlevel sind in der Datei /etc/inittab definiert.

Die einzelnen Definitionen unterscheiden sich also nur dadurch, welche Dienste gestartet werden. Das Starten und Stoppen (beim Wechsel des Runlevels ) wird durch eine Verknüpfung (Softlink) zum eigentlichem Skript im Verzeichnis /etc/inid.d/rc[RUNLEVEL].d festgelegt. Die Skripten selbst befinden sich im übergeordneten Verzeichnis init.d. Beispiel:

```
lrwxrwxrwx 1 root root 10 Nov 25 21:00 S05network -> ../network
```

Beim **Starten** werden alle Einträge mit S in der Reihenfolge der darauf folgenden Zahlen abgearbeitet. Beim Stoppen bzw. Verlassen eines Runlevels werden alle Einträge mit K (**K**ill) in Reihenfolge der darauf folgenden Zahlen abgearbeitet. Der Befehl init ruft Einträge mit "S" mit der Option start und Einträge mit "K" mit stop auf.

<b>S/K</b>	<b>Reihenfolge</b>	<b>Name des</b>
<b>Start oder Stop</b>	<b>[00..99]</b>	<b>Skripts</b>
S	05	network* <a href="#">1</a>
K	10	network* <a href="#">1</a>

Ein Initskript sollte zumindest die Optionen start, stop und status verarbeiten. Ein Beispielskript das auch als Grundlage für eigene Initskripte verwendet werden kann ist bei Suse unter /etc/init.d/skeleton zu finden.

Optionen für die durch diese Skripte gesteuerten Dienste werden in den, den Dienstnamen tragenden Dateien unter /etc/sysconfig eingetragen. [\\*2](#)

---

\*1 Dieser Eintrag im Verzeichnis rc3.d bedeutet: Beim Betreten dieses Runlevels wird das Script ../network als fünftes gestartet.

\*2 ..und hier wird das Netzwerk gestoppt.

\*3 Siehe auch [kpresenter-Präsentation](#)





# FileSystemTAbelle

## Die Datei /etc/fstab

Die Datei /etc/fstab beschreibt pro Zeile ein Dateisystem in folgender Form:

Gerätedatei oder Netzbezeichnung, Mountpunkt, Dateisystem, Optionen, Überprüfung, Dump.

Die mit "auto" bzw. "defaults" in der Optionenspalte bezeichneten Dateisysteme werden während des Bootvorganges automatisch eingehängt. Soll ein noch nicht gemountetes Dateisystem eingehängt werden, kann der Mount Befehl verkürzt werden wenn der Mountpunkt in dieser Datei aufgeführt ist. Statt also ein Verzeichnis mit dem Befehl

```
mount -t smbfs -o username=rainer,password=geheim //scenic4/rainer /homelocal/rainer
```

einzuhängen, reicht ein schlichtes

```
mount /homelocal/rainer
```

aus. Alles ist eine Datei! Auch Netzwerkverbindungen oder Dateien können also ins Dateisystem eingehängt werden. Bei neueren Suseversion siehe nfs und smbfs (-> Auslagerung der mountfunktion)

## Beispiel:

/dev/hda9	/	ext3	defaults	1 1
/dev/hda5	/boot	ext3	defaults	1 2
/dev/hda11	/homelocal	ext3	defaults	1 2
/dev/hda1	/windows/C	vfat	users,gid=users,umask=0002,icharset=iso8859-15,code=437	0 0
/dev/hda7	swap	swap	pri=42	0 0
devpts	/dev/pts	devpts	mode=0620,gid=5	0 0
proc	/proc	proc	defaults	0 0
usbdevfs	/proc/bus/usb	usbdevfs	noauto	0 0
/dev/cdrom	/media/cdrom	auto	ro,noauto,user,exec	0 0
/dev/fd0	/media/floppy	auto	noauto,user,exec	0 0
/dev/sda1	/media/sda1	auto	noauto,user,exec	0 0
/dev/mem0c0b	/media/memcard	auto	noauto,user,exec	0 0
//scenic4/rainer	/homelocal/rainer	smbfs	noauto,username=rainer,password=geheim,user,exec	0 0
WAS	WOHIN	TYP	OPTIONEN	0 0

## Geräte

Als Gerätedatei wird (wie beim entsprechenden Mountbefehl) die dem Gerät entsprechende Gerätedatei im Verzeichnis /dev angegeben. Zum Beispiel /dev/hdc4 für die vierte Partition (4) auf der dritten (c) Festplatte am IDE-Controller (hd). Als Dateisystem wird der Typ angegeben mit dem diese Partition formatiert wurde. So z.B. ext3 oder msdos oder xfs.

## Dateien als Dateisysteme

Als Gerätedatei wird die im Filesystem vorhandene Datei und als Dateisystem der Typ mit dem diese Datei strukturiert ist angegeben. Als Option ist -o loop mitanzugeben.

Beispiel:

```
dd if=/dev/zero of=/home/testuser/Datei count=2880
mke2fs /home/testuser/Datei
mount -t ext2 -o loop /home/testuser/Datei /mnt
```

Hier wird eine diskettengrosse (2880 x 512 Bytes), mit Nullen gefüllte Datei in einem Verzeichnis erzeugt. Anschließend wird diese Datei mit der Struktur eines ext2-Dateisystems versehen und zum Schluss in das Dateisystem an das Verzeichnis /mnt eingehängt.

## NFS-Verbindungen

Als Gerätedatei wird Rechnername:Verzeichnis und als Dateisystem der Typ nfs angegeben. Unter Optionen können Optionen wie beim Mountbefehl für nfs angegeben werden. Bei älteren Versionen werden diese Dateisysteme bei Starten automatisch gemountet sofern auto oder default gesetzt ist. Bei neueren Versionen ist hierfür der Dämon nfsd zuständig.

## SaMBa-Verbindungen

Als Gerätedatei wird //Rechnername/sharename (wie beim entsprechenden Mountbefehl) und als Dateisystem der Typ smbfs angegeben. Unter Optionen können Optionen wie beim Mountbefehl angegeben werden. Da im Gegensatz zu NFS die Freigabe eines Shares fast immer von einer Authentifizierung abhängt, kann man die Angaben zu Name und Passwort in der Optionenspalte ablegen (siehe obige Beispieldatei). Sinnvoll ist dies meist nicht, da /etc/passwd für alle lesbar sein muss. Daher sollten diese Angaben in einer credentials-Datei\*<sup>1</sup> mit entsprechend eingeschränkten Rechten vermerkt sein. In der fstab-Datei wird dann unter Optionen nur ein entsprechender Verweis aufgenommen.

## Automatisches Ein- und Aushängen von Dateisystem nach Bedarf

Hierfür ist das "AutoFilesystem" zuständig. Siehe [dort](#).

## Hotplug-Verbindungen

Dateisysteme die über die Hotplugfunktion (z.B. Digitalkameras, Speicherstifte etc.) eingebunden werden, trägt das System selbst in die fstab ein und entfernt diese Einträge auch automatisch wieder.

---

\*1 Beispiel credentials

# mounten von Dateisystemen



## Mount einer NFS-Freigabe:

```
mount -t nfs rechner:/freigabe /mountpoint
```

## Mount in /etc/fstab:

```
//rechner/share /mountpoint nfs defaults 0 0
```

## Mount einer Datei, die ein Filesystem beinhaltet

```
mount -t filesystemtyp -o loop dateiname /mountpoint
```

## Mount in /etc/fstab:

```
dateiname /mountpoint filesystemtyp auto,loop 0 0
```

## Mount eines SMB-Shares:

```
mount -t smbfs -o username=name,password=passwort //rechner/share /mountpoint
```

## Mount in /etc/fstab:

```
//rechner/share /mountpoint smbfs auto,gid,rmask=0660,dmask=0770,ioccharset=iso8859-15,  
code=437,credentials=/etc/samba/creds 0 0
```

## Mount von verschlüsselten Dateisystemen

### Mount einer verschlüsselten Partition

```
losetup -e twofish /dev/loop0 /dev/hda3  
mount -t reiserfs /dev/loop0 /mountpoint
```

### Mount einer Datei als verschlüsseltes Dateisystem

```
losetup -e twofish /dev/loop1 /cryptdatei  
mount -t filesystemtyp /dev/loop1 /mountpoint
```

### Erstellen der Datei

```
dd if=/dev/urandom of=/cryptdatei bs=1024 count=20000  
losetup -e twofish /dev/loop1 /cryptdatei  
mke2fs /dev/loop1
```

**Mount in /etc/fstab:**

Wird beim Hochfahren automatisch eingehängt, Passwort wird abgefragt:

```
/dev/loop0 /dev/hda3 /mountpoint filesystemtyp twofish defaults 0 0  
/dev/loop1 /cryptdatei /encrypt_file filesystemtyp twofish defaults 0 0
```

Wird beim Systemstart dieses Dateisystem nicht automatisch gemountet, kann jedoch auch von Benutzern gemountet werden:

```
/cryptdatei /mountpoint filesystemtyp loop,encryption=twofish,noauto,user 0 0
```

Das Passwort kann bei verschlüsselten Dateisystemen nach dem Anlegen nicht mehr geändert werden.

# AutoFileSystem



## Einbinden von Dateisystemen nach Bedarf.

Mit dem **Autofs**-Dienst können Dateisysteme automatisch bei Bedarf eingehängt und nach einer gewissen Zeit der Nichtbenutzung (Leerlaufzeit)<sup>\*1</sup> ausgehängt werden. Je Verzeichnis (als sog. Mountpoint) das vorhanden sein muss, wird in der Datei /etc/auto.master eine Datei angegeben. In dieser Datei wird spezifiziert welche Mountvorgänge im Bedarfsfall (= Zugriff auf ein Verzeichnis unterhalb des "Mountpointverzeichnisses") durchgeführt werden. Diese Verzeichnisse werden "virtuell" beim Mountvorgang nach Bedarf angelegt und beim Aushängen gelöscht. Beim Mountvorgang werden angegebene Optionen berücksichtigt. Es sind die Optionen wie bei einem manuellem Mountvorgang möglich<sup>\*2</sup>.

Beispiel:

auto.master:

```
# $Id: auto.master,v 1.1 2001/04/17 11:43:02 arvin Exp arvin $
# Sample auto.master file
# Format of this file:
# mountpoint map options
# Also see variable AUTOFS_OPTIONS in /etc/sysconfig/autofs
# For details of the format look at autofs(8).
```

```
#/net /etc/auto.net
/media /etc/auto.media
/home /etc/auto.home
```

Hier werden für die Verzeichnisse /media und /home spezielle Dateien angegeben.

auto.media:

```
# $Id: auto.misc,v 1.1 2001/04/17 11:43:02 arvin Exp arvin $
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage
```

```
#cdrom -fstype=auto,ro /dev/cdrom
#floppy -fstype=auto,sync /dev/fd0
server -fstype=nfs server.local:/export
fileshare -fstype=smbfs //newcomer/fileshare
daten -fstype=nfs,rsize=8192,wsiz=8192 siedler:/daten
#was wie von_wo
```

Hier werden für die "virtuellen" Mountpoints cdrom, floppy und daten in dem Verzeichnis /media angegeben was mit welchen Optionen gemountet werden soll. Wird nun z.B. ein ls /net/daten ausgeführt so wird der virtuelle Mountpoint daten erzeugt und an diesen mit dem Dateisystem nfs

die Freigabe /daten des Rechners siedler mit der Option Lese- und Schreibblockgrösse 8kb eingehängt.

## Einbinden von Homeverzeichnissen

Eine der elegantesten Arten autofs anzuwenden:

Hierbei wird auf dem Client im Verzeichnis /home (muss existieren) ein "virtueller" Mountpoint mit dem Namen des Benutzers angelegt und das Verzeichnis des/r Benutzer(s) eingehängt (mehrere zugleich möglich):

auto.home:

```
# $Id: auto.misc,v 1.1 2001/04/17 11:43:02 arvin Exp arvin $
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage
```

```
* -fstype=nfs siedler:/home/&
```

Auf dem Server muss in /etc/exports das Verzeichnis /home freigegeben sein.

## Verteilen von auto.xxx-Dateien

Dateien wie auto.home, auto.net etc. können über den NIS-Dienst verteilt werden. Hierzu sind, neben den sonstigen Voraussetzungen, in der Datei /var/yp/Makefile die entsprechenden Dateien mit aufzunehmen bzw. das Kommentarzeichen zu entfernen und in der Datei /etc/nsswitch.conf muss die Zeile

```
automount : files, nis
```

vorhanden sein.

---

\*1 Die Leerlaufzeit wird durch die Angabe von AUTOFS\_OPTIONS="-t 1" (z.B. fr 1 sec) in der Datei /etc/sysconfig/autofs eingestellt.

\*2 Mögliche Optionen siehe man-page zum Befehl mount.



# Benutzer

## Der Benutzer

Ein Benutzer, das ist:

1. Einem "Datensatz" in einer "Datenbank" wie zum Beispiel der Datei /etc/passwd. Natürlich kann es sich auch um einen NIS-Eintrag, einen LDAP-Eintrag etc. handeln.
2. Einem "Homeverzeichnis". Welches Verzeichnis dies ist bestimmt sich nach dem o.g. Datensatz.

## Beispiel

```
root:x:0:0:/:root:/bin/bash
.....
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
.....
rainerk:x:500:100:Rainer Kulhanek:/homelocal/rk:/bin/bash
gast:x:9999:65533:fuer Zugriffe von aussen:/home/gast:/bin/bash
.....
Name:Passwort:UID:GID:frei*1:Heimatverzeichnis:Shell
```

Feld	Bedeutung	s.a.
Name	Benutzername	keine führenden Ziffern
Früher das verschlüsselte Passwort, heute ein ....	x - Passwort ist in der Datei /etc/shadow abgelegt,	
UID	Benutzernummer	üblicherweise beginnen Benutzernummern ab 500 bzw. 1000. Der Bereich darunter ist für Systemdienste etc. reserviert.
GID	Nummer der Hauptgruppe des Benutzers	Zuordnung der Gruppennummer zu den GID in der Datei /etc/group
frei	Dieses Feld steht zur freien Verfügung <a href="#">*1</a>	
Homeverzeichnis	Das Heimatverzeichnis des Benutzers	Hier startet die eingestellte Shell
Shell	Die unter der UID des Benutzers gestartete Shell.	Meist die /bin/bash. Der Eintrag /bin/false verhindert das der Benutzer eine Shell starten kann.

Beispiel /etc/shadow:

Feld	Bedeutung	s.a.
loginname	verschlüsseltes Passwort <a href="#">*4</a>	
Passwortalter	Alter des Passwortes <a href="#">*5</a>	
Mindestalter	Tage bis geändert werden darf	

Maximalalter	Tage bis geändert werden muss
Warnung	Tage vor ungültig werden des Passwortes Tage nach denen das Passwort endgültig abläuft*6
invalid	Tage ab 1.1.1970 bis zum ablaufen:Reservefeld

Bis auf die beiden ersten sind die Einträge optional.

## Anlegen eines Benutzers

Mehrere Möglichkeiten:

- Manuell
- [useradd](#) - Kommando

Kommandosyntax:

```
useradd [-u uid [-o]] [-g group] [-G group,...] [-d home] [-s shell] [-c comment*1 ] [-m [-k
template*2]] [-f inactive] [-e expire ] [-p passwd] [-r] name
```

- graphische Werkzeuge
- Yast
- Programme wie z.B. webmin

---

\*1 Hier können organisatorische Merkmale eingetragen werden. Wird vom Programm finger (heute aus Sicherheitsbedenken kaum mehr im Einsatz) genutzt.

\*2 Eine Vorlage zur Verwendung beim Anlegen des Benutzers ist unter /etc/skel zu finden.

\*4 Ein Stern bedeutet das kein Passwort existiert. Wird vor allem bei Sytemdiensten verwendet.

\*5 ab 1.1.1970 bis zur letzten Änderung

\*6 kann als Puffer verwendet werden



## Benutzer II

### Anmelden

Programm mingetty wartet auf Anmeldeinformationen (Benutzername und Passwort)

Überprüfung der Anmeldeinformationen durch pam\_module,

Vorgaben über Anzahl der Loginversuche, Zeitdauer etc. in /etc/login.defs

### Homeverzeichnis

Musterverzeichnis für neu zu erstellende Benutzer -> /etc/skel

Umask für Homeverzeichnis bei Neuanlage von Benutzern -> /etc/login.defs

Vorgaben für YAST bei Neuanlage von Benutzern -> /etc/default/useradd

### Profil

/etc/profile, \$HOME/.profile, evtl. Vorgaben durch pam-Modul pam\_env.



# Gruppenverwaltung



## Gruppen

### Hauptgruppe

Jeder Benutzer hat eine Hauptgruppe und kann weiteren Gruppen angehören. Ein Gruppenwechsel ist mit ... möglich.

### Gruppenzugehörigkeit

Mitglieder einer Gruppe (Benutzer deren Hauptgruppe nicht diese Gruppe ist) werden in der Datei /etc/group aufgeführt.

```
.....  
dialout:x:16:rk,test  
gruppe:passwort:nummer:Mitglied,....  
.....
```

Hier sind die Benutzer rk und test Mitglieder der Gruppe dialout. Alles ist eine Datei, also auch Geräte wie z. B. ein Modem. Da Zugriffsrechte an Dateien gebunden sind und Benutzer als solche keine Rechte haben werden die Zugriffe durch die Rechte der Gerätedateien in /dev gesteuert. Da die Gruppe dialout Schreibzugriff an der (Geräte-)datei /dev/modem hat, können somit diese beiden Benutzer das Modem benutzen.

## Lokale und Globale Gruppen

### Network Information Service

Gruppendefinitionen (wie z.B. in /etc/group) können über den [NIS-Dienst](#) netzweit verbreitet werden.

### Netzgruppen (netgroup)

### Gruppenzugehörigkeit beim Einloggen durch pam-Modul

--

### LDAP

--

### SaMBa/Windows

Übernahme von Windowsgruppen

"Sambagruppen" ab Samba 3.0

--



# Zugriffsrechte



Beispielsausgabe eines /home-Verzeichnisses:

```
drwx----- 2 root root 1024 Jun 26 1999 adabas
drwx----- 2 root smbuser 1024 Mai 30 2000 administrator
drwx----- 14 ccs users 1024 Mai 8 2000 ccs
drwx----- 7 informix informix 1024 Apr 20 2000 informix
drwx----- 2 root root 12288 Nov 13 1999 lost+found
drwx----- 9 mails users 1024 Aug 13 1999 mails
drwx----- 2 root root 1024 Jan 4 1999 news
drwxrwxr-- 2 root smbgrp 1024 Jul 25 19:11 public
drwx---r-x 76 rainer users 13312 Jan 7 16:59 rainer
drwx----- 12 scug users 1024 Okt 29 12:04 scug
drwx----- 2 root smbuser 1024 Mai 30 2000 smbuser
drwxr-xr-x 2 root root 1024 Jan 7 17:06 test
drwx----- 2 testo users 1024 Jun 18 2000 testo
```

Art/Rechte Ln# UID GID Grösse Datum -- Time Name  
Bedeutung der Angaben in einer Verzeichnisübersicht

## Art des Eintrages

Mit dem ersten Buchstaben wird die Art der Datei gekennzeichnet:

Typ	Bedeutung	Bemerkungen
D	Verzeichnis	Ebenfalls eine Datei. Durch das d wird diese Datei als Verzeichnisliste behandelt.
-	normale Datei	
L	Softlink	Sogenannte harte Links gelten als selbstständiger Eintrag --> keine Kennzeichnung
B	Blockorientiertes Gerät	üblicherweise nur im Verzeichnis /dev. Z.B. Festplatten
S	Socket	
C	Zeichenorientiertes Gerät	üblicherweise nur im Verzeichnis /dev. Z.B. Tastatur

## Anzahl der Links (ln#)

Anhand dieses Wertes kann man erkennen wieviele "Hardlinks" auf einen Eintrag weisen. Erst wenn der Wert gleich "1" ist, wird bei "rm" oder "mv" der Dateieintrag tatsächlich gelöscht.

## Drei Rechtearten:

Rechtetyp	Datei	Verzeichnis
r	lesen	lesen
w	schreiben	schreiben
x	ausführen	betreten

Lesen bedeutet bei Verzeichnissen das Recht den Inhalt der Verzeichnisdatei (Liste der Dateinamen) lesen zu können. Ausführen bedeutet bei Verzeichnissen das Recht "in" diese Verzeichnisse wechseln zu können (= Zugriff auf die I-Nodes haben).

Weitere Zugriffsrechte siehe .... (Seite noch nicht erstellt)

## Eigentümer und Gruppe des Eintrags

Die Zugriffsrechte werden vergeben für den Dateieigentümer (UID), die Gruppe (GID) und den Rest der Welt\*[1](#). Die Zuordnung Name zu UID/GID findet sich in den Dateien /etc/passwd und /etc/group

Bei der Überprüfung wird festgestellt ob der aktuelle Benutzer mit dem Eigentümereintrag übereinstimmt; wenn ja gelten diese Rechte, ansonsten erfolgt ein Vergleich auf Übereinstimmung der Gruppen (Hauptgruppe, sonstige Gruppen, evtl. netgroups). Sofern auch dieser negativ ausfällt gelten die Rechte für den Rest der Welt (others). Wurde eine Übereinstimmung bei Eigentümer oder Gruppe gefunden wird nicht weiter geprüft. In ein Verzeichnis

```
-rwx---r-x tom users
```

kann die Benutzerin anna users nicht wechseln, obwohl es für "others" erlaubt ist. anna ist nicht gleich dem Eigentümer der Datei, jedoch der Gruppe users zugehörig und somit nicht eine "andere"! Es gelten die Rechte der Gruppe. "others" wird nicht mehr geprüft.!

Eine Erweiterung dieses Zugriffsschemas ist mit [acls](#) möglich.

## Befehle zum Verändern der Rechte bzw. Eigentümer

- chown = Eigentümer ändern\*[2](#)
- chgrp = Gruppe ändern
- chmod = Zugriffsrechte ändern

---

\*1 Letzteres heisst nicht alle, sondern alle ausser dem Eigentümer und der genannten Gruppe!!

\*2 Entgegen früheren Unices kann der Benutzer unter Linux den Eigentümer auch seiner eigenen Dateien **nicht** ändern, da sonst jegliche Quotierung unterlaufen würde.

# AccessControllist



## Erweiterte Zugriffsrechte

ACL ermöglicht eine Erweiterung hinsichtlich der Benutzer und Gruppen an einer Datei. Mit dem gebräuchlichem Schema unter Linux kann nur für einen Benutzer und eine Gruppe Rechte vergeben werden. Durch ACL wird dies auf weitere Benutzer und Gruppen erweitert. Es stellt keine neuen Rechtetypen zur Verfügung! Die beim Anlegen zu vergebenden Rechte können festgelegt und im Verzeichnisbaum nach unten vererbt werden.

### Erstellen

Notwendig ist die Unterstützung durch das Dateisystem (bei den meisten gebräuchlichen gegeben) **und** die Option "acl" beim mounten des Dateisystems.

### Beispiel

Eintrag in der fstab:

```
/dev/hda3 /home ext3 defaults,acl 0 0
```

ACLs werden mit dem Kommando setfacl vergeben und mit getfacl angezeigt.

Anzeige der Datei xntp.html:

```
-rw-r--r-- 1 rk users 17677 Jun 25 2003 xntp.html
```

Der Eigentümer rk hat Lese- und Schreibzugriff, die Gruppe users und alle anderen nur Lesezugriff.

Vergabe zusätzlicher Rechte (rwx) an User angela mit

```
setfacl -m u:angela:rwx xntp.html
```

Bei erneuter Anzeige weist ein "+" auf die zusätzlichen Rechte hin:

```
-rw-rwxr--+ 1 rk users 17677 Jun 25 2003 xntp.html
```

Beachte: Der Wert der Gruppenrechte entspricht nicht mehr den Rechten der Gruppe users, sondern den sogenannten mask-Rechten<sup>[\\*1](#)</sup>! Anzeige mittels getfacl:

```
# file: xntp.html
# owner: rk
# group: users
user::rw-
user:angela:rwx
group::r--
mask::rwx
```

```
other::r--
```

Hier sieht man das der Benutzer angela sowohl Lese- und Schreibrecht als auch das Recht zum ausführen der Datei erhalten hat. WÄ¼rde die Maske nun auf rw- gesetzt, würde der Benutzer die Datei trotzdem nicht ausführen können.

## Vererbung

Bei Verzeichnissen kann mit default Rechten angegeben werden, welche Rechte neu angelegte Dateien und Verzeichnisse erhalten.

### Beispiel

Mit `setfacl -R -m d:linuxseminar:rw linuxsem` wird der Verzeichnisdatei "linuxseminar" und allen Unterverzeichnissen (-R) ein Defaultrechte (rw) für den Benutzer linuxseminar mitgegeben.

Die Anzeige mit `getfacl linuxsem` zeigt nun neben Einträgen für den Eigentümer (default:user::), sonstige Benutzer (default:user:linuxsem:), Hauptgruppe (default:group::) und sonstige Gruppen (default:group::) sowie den Sonstigen (other::) auch eine Default-Maske an. Diese Default-Maske gibt an welche Rechte ein Benutzer oder eine Gruppe **maximal** haben kann.

```
# file: linuxsem
# owner: rk
# group: users
user::rwx
user:linuxseminar:rwx
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:linuxseminar:rw-
default:group::r-x
default:mask::rwx
default:other::r-x
```

Diese Defaultrechte werden an neu angelegte Verzeichnisse vererbt!

---

\*1 Der Wert mask gibt an welche Rechte ein Benutzer oder eine Gruppe maximal haben kann.



# Module erstellen und installieren

## Module

Module sind Teile des Systems, die nicht im Kernel vmlinux enthalten sind, sondern "nachgeladen" werden können. Insbesondere Routinen die hardwareorientiert sind werden als Module kompiliert um z.B. nur den Treiber für die in die Hardware eingebauten Netzwerkkarten benutzen zu können, ohne auch Treiber für andere Karten im Speicher halten zu müssen.

Sofern Module erst noch kompiliert werden sollen, muss der Kernelsource installiert sein, da die Module darauf Bezug nehmen.

Fertige Module sind unter `/lib/modules/[Kernelname]/...` zu finden. Bis Kernel 2.4.xx lautet die Endung `.so`, ab 2.6.xx `.ko`. Welche Module wann für was geladen werden und welche Optionen hierbei u.U. verwendet werden ist in der Datei `/etc/modules.conf` (bis Kernel 2.4.xx) bzw. `/etc/modprobe.conf` (Kernel ab 2.6.xx) vermerkt.

## Module auf der Kommandozeile...

- *lsmod* Anzeige der geladenen Module
- *rmmod Modulname* entladen eines Moduls
- *insmod Modulname* Modul laden
- *modprobe Modulname* Modul laden, Fehlermeldungen werden angezeigt





# Neuen Kernel erstellen und installieren

## Ablauf:

1. make xmenuconfig
2. make dep
3. make clean
4. make bzImage
5. make modules
6. make modules install
7. kopieren des neuen Kernel nach boot
8. einbinden in den Bootmanager
9. depmod -a

## Im Einzelnen:

- make xmenuconfig

Kerneleinstellungen festlegen

- make dep

Abhängigkeiten erstellen

- make clean

Zwischendateien erstellen (kann u.U. entfallen)

- make bzImage

Kernel erstellen

- make modules

Module erstellen (müß  $\frac{1}{2}$  bei jedem Kernel neu erfolgen)

- make modules install

Module in Modulverzeichnis kopieren, neue Modulliste erstellen

- kopieren des neuen Kernel nach boot

möglichst unter einem anderen Namen speichern, s.u.

- einbinden in den Bootmanager

Der neue Kernel sollte unter einem anderem Namen eingetragen werden. So kann man bei evtl.

auftretenden Fehlern immer noch den bisherigen Kernel zum Booten benutzen.

- `depmod -a`

neue Modulliste erstellen



# Prozesse

## Zustände

Prozesse können einen von drei Zuständen haben:

1. running - Der Prozess ist aktiv bzw. wird von der CPU bearbeitet
2. ready - Der Prozess ist bereit von der CPU bearbeitet zu werden
3. blocked - Der Prozess wartet auf ein Ereignis z.B. einen Tastendruck

Ein Prozess kann weitere Prozesse starten [\\*1](#). Gestartete Prozesse geben beim Beenden einen Ergebnis-Code (errorlevel) an den aufrufenden Prozesse zurück. Als Zombie Prozesse bezeichnet man beendete Prozesse, die jedoch noch einen Prozess-Tabellen-Eintrag belegen.

Z.B.: Der einen weiteren Prozess (=Kind) aufrufende (Eltern-)Prozess ist nicht mehr vorhanden oder nimmt den Abschlusswert nicht an. Der Kindprozess hat sich beendet, den von ihm belegten Speicherplatz freigegeben und wartet darauf den Abschlusswert zurückzugegeben. Solange sich der Prozess in diesem Wartezustand befindet, belegt er in der Prozesstabelle einen Eintrag.

## Kenndaten

Die Kenndaten können sehr schön mittels

```
cat /proc/[prozessnr]/status
```

ausgegeben werden [\\*2](#):

```
Name: ypbind
State: S (sleeping) Tgid: 726
Pid: 726
PPid: 724
TracerPid: 0
Uid: 0 0 0 0
Gid: 0 0 0 0
FDSize: 32
Groups:
VmSize: 17884 kB
VmLck: 0 kB
VmRSS: 736 kB
VmData: 16476 kB
VmStk: 12 kB
VmExe: 28 kB
VmLib: 1304 kB
SigPnd: 0000000000000000
SigBlk: 0000000080014407
SigIgn: 8000000000000000
SigCgt: 0000000380014407
CapInh: 0000000000000000
CapPrm: 00000000fffffeff
```

CapEff: 00000000fffffeff

Ausgabe des Befehls top:

3:20pm up 6:48, 8 users, load average: 0.07, 0.19, 0.22

141 processes: 138 sleeping, 2 running, 1 zombie, 0 stopped

CPU states: 5.6% user, 5.6% system, 0.0% nice, 88.7% idle

Mem: 247668K av, 243600K used, 4068K free, 0K shrd, 26052K buff

Swap: 385520K av, 12812K used, 372708K free 83500K cached

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
6537	rk	16	0	1016	1016	764	R	2.1	0.4	0:00	top
3446	root	15	0	29456	20M	4392	S	1.9	8.5	6:23	X
3536	rk	15	0	10928	10M	9848	S	1.9	4.4	4:01	kdeinit
4865	puretec	15	0	9992	9992	9116	S	1.9	4.0	2:47	kdeinit
4642	rk	15	0	32572	31M	18044	S	0.7	13.1	2:47	quanta
6004	rk	15	0	12428	12M	11044	S	0.7	5.0	0:01	kdeinit
13	root	15	0	0	0	0	SW	0.3	0.0	0:10	kjournald
3561	rk	15	0	11964	11M	10736	S	0.1	4.8	0:12	kdeinit
1	root	15	0	240	240	204	S	0.0	0.0	0:05	init
2	root	15	0	0	0	0	SW	0.0	0.0	0:06	keventd
3	root	15	0	0	0	0	SW	0.0	0.0	0:00	kapmd
4	root	34	19	0	0	0	SWN	0.0	0.0	0:00	ksoftirqd_CPU0v
5	root	15	0	0	0	0	SW	0.0	0.0	0:03	kswapd
6	root	15	0	0	0	0	SW	0.0	0.0	0:00	bdflush
7	root	15	0	0	0	0	SW	0.0	0.0	0:00	kupdated
8	root	15	0	0	0	0	SW	0.0	0.0	0:00	kinoded
9	root	23	0	0	0	0	SW	0.0	0.0	0:00	mdrecoveryd

\*1 Überblick auf der Kommandozeile mittels pstree

\*2 Für jeden Prozess existiert ein Verzeichnis im Prozessdateisystem (/proc). Im einzelnen werden folgende Daten für jeden Prozess abgebildet:

Dateiname	Inhalt
cmdline	aufrufende Kommandozeile
cwd	Link zum Arbeitsverzeichnis des Prozesses
environ	Umgebungsvariablen
exe	Programm
fd	die geöffneten Dateien
maps	Memorymapping
mem	Speicherverbrauch
root	
stat	Prozessstatus
statm	belegter Speicher
status	Tabelle des belegten Speichers

# Network File System



Network File System

## Client:

Einbinden mit mount-Befehl:

```
mount -t nfs -o ro*1 rechner:/name/des/verzeichnisses/wie/freigegeben /lokaler/mountpunkt
```

oder

Aufnahme in die /etc/fstab \*3:

```
rechner:/name/des/verzeichnisses/wie/freigegeben /lokaler/mountpunkt nfs defaults*2 0 0
```

oder

mittels [autofs](#)

Es müssen die Dienste portmap und nfslock gestartet sein.

## Server:

Konfigurationsdatei: /etc/exports

```
# See the exports(5) manpage for a description of the syntax of this file.  
# This file contains a list of all directories that are to be exported to  
# other computers via NFS (Network File System).  
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details  
# on how make changes in this file effective.
```

```
/media/cdrom          192.168.101.0/255.255.255.0                               (ro,sy  
/data1                192.168.101.0/255.255.255.0(rw,sync,no_root_squash,no_all_squash)  
/spezial              192.168.101.10/255.255.255.255(rw,sync,no_all_squash)  
/PFAD/ZUM/VERZEICHNIS RECHNER/NETZ/NETZMASKE(OPTIONEN)
```

Es ist nicht möglich einen "Sharenamen" zu verwenden. Bei Freigabe und beim Mounten ist der gesamte Pfad anzugeben.

Achtung, Zugriffe erfolgen **nicht** unter Benutzernamen und Gruppennamen, sondern mit UID und GID! Ein User kann auf verschiedenen Systemen verschiedene UIDs haben! Daher sollte auf jeden Fall eine netzweite Benutzer- und Gruppenverwaltung eingesetzt werden. Alternativ kann man den Zugriff für root bzw. für alle User mit root\_squash bzw. all\_squash auf den Benutzer nobody der Gruppe nogroup "mappen". Da dieser Benutzer im System keine Rechte an irgendeinem Dateieintrag hat (somit als "Rest der Welt" auftritt) kann der Zugriff ein wenig beschränkt werden.

---

1) Ein als rw (lesen und schreiben) freigegebenes Verzeichnis kann von seiten des Clients auch ro

(nur lesen) gemountet werden. Die Angabe ro an dieser Stelle bezieht sich auf die NFS-Freigabe. D.h. wenn das Verzeichnis auf dem Server für den jeweiligen Benutzer schreibbar wäre, so ist bei ro über NFS kein Schreibzugriff möglich.

2) Oder sonstige, in fstab zulässige Optionen

3) siehe auch [fstab.html](#)



## DNS

# Sample configuration for BIND9

```
options {  
  directory "/var/named";  
  forwarders { 194.25.2.129;217.237.159.1;10.0.0.1; };  
  listen-on-v6 { any; };  
  notify no;  
};
```

```
zone "localhost" in {  
  type master;  
  file "localhost.zone";  
};
```

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

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





# Samba

## Muster fr PDC-Erstellung:

gltig fr samba <3.0; ab 3.0 z.Teil andere [Befehle](#)

### Am Server

smb.conf:

Minimalangaben

```
# smb.conf is the main samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SuSE
# Date: 2002-09-12
[global]
workgroup = DOMAIN
netbios name = rechnername
security = user
os level = 64
encrypt passwords = yes
printing = CUPS
printcap name = CUPS
socket options = SO_KEEPALIVE IPTOS_LOWDELAY TCP_NODELAY
wins support = yes
character set = ISO8859-15
client code page = 850
veto files = /*.eml/*.nws/richted20.dll/*.*}/
domain master = yes \*1
domain logons = yes
logon home = \\%L\%U
logon script = logon.bat
; Wenn erweiterte Zugriffsrechte von Windows aus benutzt werden sollen:
; Die verwendeten Dateisysteme mssen natrlich mit acl gemountet sein!
nt acl support = yes
[homes]
comment = Home Directories
valid users = %S
browseable = no
writeable = yes
create mask = 0640
directory mask = 0750

[printers]
comment = All Printers
path = /var/tmp
printable = yes
create mask = 0600
browseable = no
```

```
[print$]
comment = Printer Drivers
path = /var/lib/samba/drivers
write list = @ntadmin root
force group = ntadmin
create mask = 0664
directory mask = 0775
```

```
[netlogon]
path = /home/netlogon
guest ok = no
writeable = yes
browseable = no
```

```
[profile]
path = /home/samba/profile
read only = no
create mode = 0700
directory mode = 0770
browseable = no
guest ok = no
```

Anlegen einer Gruppe

optional, unterstützt jedoch die Übersicht

```
groupadd smbuser
```

Anlegen der UNIX-Benutzer

```
useradd USERNAME -g smbgroup
```

Anlegen der Samba-Benutzer

```
smbpasswd -a benutzername*3
```

Anlegen von Rechnerkonten\*4

mit yast oder z.B.

```
useradd -uid 600 -g smbgroup -d -m -s /bin/false rechnername*2
```

Anlegen eines Computerkontos in Samba

```
smbpasswd -a -m rechner$*5
```

Anlegen eines "Administrators" z.B. root

```
smbpasswd -a root*6
```

Neustart des/der Dienst(z)en

rcsamba restart

Sofern rcsamba (Ein Skript das sowohl smbd als auch nmbd stoppt/startet) nicht vorhanden ist, die Dämonen einzeln mit rcsmb und rcnmb starten.

## Am Client

Bei Windows die bliche Vorgehensweise zum Beitritt in eine Domäne. Bei Linuxclient

```
smbpasswd -j DOMÄNE
```

Fertig!

Beim Beitreten in die Domäne muss das o.g. "Administratorkonto" angegeben werden.

---

\*1 Hierdurch wird der Samba-Server zum Primary Domain Controller

\*3 Kein lokales Anmelden, da als Shell /bin/false eingetragen.

\*3 wenn nicht gleichnamig mit den LINUX-Nutzern, evtl. Mapping in /etc/samba/smbuser

\*4 Mit \$ am Ende z.B. rechner\$, Passwort ist gleichgültig z.B. Rechnername, jedoch später nicht mehr ändern.

\*5 Passwort ist gleichgültig z.B. Rechnername, jedoch später nicht mehr ändern.

\*6 root ist in der passwd existent, muss jedoch auch ein SMB-verschlüsseltes Konto haben, da beim Beitritt eines Clients in die Domäne das Passwort als SMB-verschlüsselt über das Netzwerk übertragen wird. Dieses Passwort muss nicht zwingend gleich mit dem UNIX-Passwort sein!





## Samba - Client

### Mount eines SMB-Shares:

```
mount -t smbfs -o username=name,password=passwort //rechner/share /mountpoint
```

### Mount in /etc/fstab:

```
//rechner/share /mountpoint smbfs auto,gid,rmask=0660,dmask=0770,icharset=iso8859-15,  
code=437,credentials=/etc/samba/creds
```



# Samba-Benutzer



Die Benutzer müssen auch unter Linux vorhanden sein (=/etc/passwd oder nis)! Eintrag erfolgt in /etc/smbpasswd. Dies wird mit dem Programm smbpasswd erledigt:

```
smbpasswd [options] [username] [password]
```

options:

- s use stdin for password prompt
- D LEVEL debug level
- U USER remote username
- r MACHINE remote machine
- R ORDER name resolve order
- j DOMAIN join domain name
- a add user
- d disable user
- e enable user
- n set no password
- m machine trust account

Beispiele

Anlegen eines Benutzers

```
smbpasswd -a karl geheimes_wort
```

Anlegen eines Rechnerkontos

```
smbpasswd -a -m rechnername
```





## NIS-Client NIS-Client

Um mit dem NetworkInformationSystem netzweite Namensauflösung zu benutzen, muss der Client entsprechend eingerichtet werden.

Für den Betrieb als Client wird benötigt:

- RPC-Portmapper-Diemon
- ypbind-Diemon

Start/Stop durch rcypbind, Automatischer Start beim Booten durch Eintrag in entsprechende Runlevel.

- Angabe eines Rechners der den Serverdienst ausführt in der Datei yp.conf:

```
ypserver 192.168.101.8
ypserver 127.0.0.1
```

Sofern der Client auch Slave-Server des YP-Dienstes ist kann er als zweiter/letzter Rechner eingetragen werden. Netzausfälle können dadurch berichtigt werden.

- Setzen der YP-Domäne in /etc/defaultdomain
- Entsprechende Eintragungen in z.B. /etc/passwd, /etc/group:

/etc/passwd:

```
.....
rk:x:500:100:Rainer Kulhanek:/home/rk:/bin/bash
gast:x:502:100:fuer Zugriffe von aussen:/home/gast:/bin/bash
test80:x:503:100:test test:/home/teste80:/bin/bash
+:::
```

/etc/group:

```
.....
maildrop:x:59:
mailman:x:67:
fou4s::500:
+:::
```

Die letzten Zeilen (mit einem Pluszeichen an erster Stelle) bewirken, dass bei libc5-Systemen nach dem erfolglosen Durchsuchen der Datei der YP-Dienst befragt wird.

- sowie Aufnahme der NIS-Abfrage in /etc/nsswitch.conf:

/etc/nsswitch.conf:

```
#
# /etc/nsswitch.conf
#
```

```
passwd: files nis  
group: files nis
```

```
#passwd:compat  
#group: compat  
hosts: nis files dns  
networks: files dns
```

Hiermit wird festgelegt, welche Namensverzeichnisse bzw. -dienste in welcher Reihenfolge durchsucht werden.

---

\*1 Die netzweit zu verteilenden Angaben müssen keineswegs in den Standarddateien abgelegt werden. Es ist möglich und sicherer, diese in eigenen Dateien einzutragen und diese Daten netzweit anzubieten.

Vorteil: Netzbenutzer können sich am NIS-Server nicht anmelden, obwohl ihre Authentifizierung von dort erfolgt.

In diesem Fall muss die Datei /var/yp/Makefile manuell oder mittels Yast etc. angepasst werden.



## NIS-Server

Mit dem **NetworkInformationSystem** können Daten wie lokale Namensauflösung netzweit verfügbar gemacht werden. Meist werden die Benutzereinträge, Gruppendefinitionen, Rechnernamen im Netz verteilt.

<b>Angaben</b>	<b>üblicherweise in Datei:<sup>*1</sup></b>
Benutzer	/etc/passwd und /etc/shadow
Gruppen	/etc/groups
Rechnernamen	/etc/hosts
Dienste	/etc/services
Mountpoints	/etc/auto.xxx

Es können selbstverständlich weiteren Angaben netzweit angeboten werden. Siehe auch das Makefile unter /var/yp. Die einfachste Art der Einrichtung eines NIS-Servers ist die Benutzung von YAST.

---

\*1 Die netzweit zu verteilenden Angaben müssen keineswegs in den Standarddateien abgelegt werden. Es ist möglich und sicherer, diese in eigenen Dateien einzutragen und diese Daten netzweit anzubieten.

Vorteil: Netzbenutzer können sich am NIS-Server nicht anmelden, obwohl ihre Authentifizierung von dort erfolgt.

In diesem Fall muss die Datei /var/yp/Makefile manuell oder mittels Yast etc. angepasst werden.





## NTP - Dienst

### Systemweit die gleiche Zeit:

Um im gesamten System die gleiche Zeit zu haben wird die Systemzeit der einzelnen Rechner miteinander synchronisiert. Das ntp-Protokoll bietet die Möglichkeit eine sehr genaue Zeitangabe von einer hochgenauen Stelle (z.B. einer "Atomuhr") zu beziehen und an andere Rechner weiterzugeben. Die durch die Weitergabe verursachte Zeitdifferenz ebenso wie "Gangungenauigkeiten" der eingebauten Hardwareuhr werden hierbei im Laufe der Zeit immer genauer erfasst und ausgeglichen.

Da die einzelnen Stufen der Weitergabe jeweils eine Ungenauigkeit bewirken, wird der "Zeitgeber" (Server, Hardwareuhr, etc.) mit einem "stratum" klassifiziert. Je geringer die Zahl, desto "näher" ist diese Quelle an der ursprünglichen Zeitquelle.

### Eine einfache Konfigurationsdatei für einen Client.

```
restrict default noquery notrust nomodify
server 192.168.101.8
# Der Server im eigenen Netz von dem die Zeit abgefragt wird.
fudge 127.127.1.0 stratum 3
server 127.127.1.0
fudge 127.127.1.0 stratum 10
restrict 127.0.0.1
restrict 192.168.101.0 mask 255.255.255.0
driftfile /etc/ntp.drift
logfile /var/log/ntp.log
```

### Die Konfigurationsdatei des Servers

```
server ntp1.ptb.de
# Abfrage des Zeitnormals in Braunschweig
fudge ntp1.ptb.de stratum 2
# Die Atomuhr hat Stratum 1, der dortige Rechner Stratum 2
##
## Undisciplined Local Clock. This is a fake driver intended for backup
## and when no outside source of synchronized time is available.
##
server 127.127.1.0 # local clock (LCL)
fudge 127.127.1.0 stratum 10 # LCL is unsynchronized
#Die eingebaute Uhr im eigenen Rechner ist nicht allzu genau.
##
## Miscellaneous stuff
##

driftfile /var/lib/ntp/ntp.drift # path for drift file
#Hier wird die durch Synchronisation und Ungenauigkeit der eigenen
#Hardwareuhr auftretende Drift der Zeit laufend vermerkt.
logfile /var/log/ntp # alternate log file
```

```
# logconfig =syncstatus + sysevents
logconfig =all

# Localhost darf zugreifen
restrict 127.0.0.1
# Lokales Netz darf zugreifen
restrict 192.168.101.0 mask 255.255.255.0
# ...sonst niemand
#restrict default notrust nomodify nopeer
```

## net - Optionen (samba 3.0)



### Usage:

net time	to view or set time information
net lookup	to lookup host name or ip address
net user	to manage users
net group	to manage groups
net groupmap	to manage group mappings
net join	to join a domain
net cache	to operate on cache tdb file
net getlocalsid [NAME]	to get the SID for local name
net setlocalsid SID	to set the local domain SID
net changesecretpw	to change the machine password in the local secrets database only this requires the -f flag as a safety barrier
net ads	to run ADS commands
net rap	to run RAP (pre-RPC) commands
net rpc	to run RPC commands
Type "net help	
Valid targets: choose one (none defaults to localhost)	
-S or --server=	server name
-I or --ipaddress=	address of target server
-w or --workgroup=	target workgroup or domain
Valid miscellaneous options are:	
-p or --port=	connection port on target
-W or --myworkgroup=	client workgroup
-d or --debuglevel=	debug level (0-10)
-n or --myname=	client name
-U or --user=	user name
-s or --configfile=	pathname of smb.conf file
-l or --long	Display full information
-V or --version	Print samba version information
-P or --machine-pass	Authenticate as machine account





# Netzwerk

- Um das Netzwerkgerät (=Netzwerkkarte) anzusprechen wird ein passender Treiber benötigt. Dieser ist entweder
  - ◆ im Kernel einkompiliert
  - ◆ oder als Modul vorhanden. In diesem Fall muss das Modul [\\*1](#) bei Bedarf geladen werden. Die dazu notwendigen Angaben finden sich in der Datei /etc/modules.conf (bis incl. Kernel 2.4.x oder /etc/modprobe.conf (ab Kernel 2.6.x). s.a. [Module](#).
- Desweiteren ist noch als Minimum die Angabe einer Defaultroute notwendig (Zeile in /etc/sysconfig/network/routes: default 192.168.100.1 - -). Der Befehl route ergibt dann folgende Ausgabe:

```
linux:/ # route
Kernel IP Routentabelle
Ziel Router Genmask Flags Metric Ref Use Iface
default 192.168.100.1 0.0.0.0 UG 0 0 0 eth0
```

- Den Rechnernamen in die Datei /etc/HOSTNAME schreiben.
- Die Angaben zu IP-Adresse, Netzwerkmaske, etc.
  - ◆ Über YAST angeben oder
  - ◆ in den Dateien /etc/sysconfig/network/ipcfg-[Netzwerkdevice] eintragen

```
linux:/ # cat ifcfg-eth0
BOOTPROTO='static'
REMOTE_IPADDR=""
STARTMODE='onboot'
UNIQUE='37TO.3ekLfYwO5g7'
WIRELESS='no'
BROADCAST='10.4.2.255'
IPADDR='10.4.2.4'
NETMASK='255.255.255.0'
NETWORK='10.4.2.0'
```

- ◆ oder: Rechnername, IP-Adresse werden von DHCP- und [DNS](#)-Servern bezogen.

```
linux:/ # cat ifcfg-eth0
BOOTPROTO='dhcp'
REMOTE_IPADDR=""
STARTMODE='onboot'
UNIQUE='qnJ_v8YwuvTx0m6'
WIRELESS='no'
```

- "Das Netzwerk" mit rcnetwork start[restart] starten. Bei der Konfiguration durch YAST wird das Netzwerk automatisch gestartet/neu gestartet.

.... und dann gibt es Arbeit ;-)

- Je nach Aufgabe(n) des Rechners müssen nun die verschiedenen Client- bzw. Serverdienste eingerichtet werden (im den jeweiligen [Runlevel](#) starten, Konfigurationsdateien bearbeiten)werden. Bei einem Arbeitsplatzrechner evtl. [ypclient](#), [smbclient](#), etc. Bei einem Server evtl. [DNS-Server](#), [NFS-Server](#), etc.
- Desweiteren sollen evtl. die nichtlokalen Verzeichnisse welche Programme vorrätig halten eingehängt werden.

- Ein wichtiger Punkt sollte auf jeden Fall bedacht werden: Sicherheit. Sobald der Rechner in einem Netzwerk erreichbar ist, muss eine vorherige sorgfältige Planung erfolgen. Client- und Serverdienste, welche nicht notwendig sind, dürfen nicht gestartet werden. Am besten sollten sie nicht installiert werden. Ports, die nicht benötigt werden sollten auch nicht von einem Dienst bedient werden.
- 

\*1 Die Module für Netzwerkgeräte finden sich im Verzeichnis  
/lib/modules/[KERNELVERSION]/kernel/drivers/net