# Dokumentation des Schulservers

# Dokumentation des Schulservers

Veröffentlicht 30.04.2005 15:56:26

Copyright © 2001, 2002, 2003, 2004, 2005 Andreas Dangel

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

# Inhaltsverzeichnis


Vorwort ..... vii
1. Allgemeines zum Netzwerk ..... 1
1.1. Topologie ..... 1
1.2. Technik ..... 1
1.3. Eingesetzte Hardware ..... 1
1.4. IP-Adressen ..... 1
2. Installation des Servers ..... 2
2.1. Debian installieren ..... 2
2.1.1. Debian besorgen ..... 2
2.1.2. Installation starten ..... 2
2.1.3. Partitionierung ..... 2
2.1.4. Weiterer Verlauf ..... 2
2.2. Kernel machen ..... 3
2.3. Software-RAID ..... 4
2.3.1. Voraussetzungen ..... 4
2.3.2. Schritt für Schritt-Anleitung ..... 5
2.3.3. Fehlerfall ..... 8
2.4. APC SmartUPS (USV) ..... 8
2.4.1. Zweck ..... 9
2.4.2. Möglichkeiten ohne Treiber ..... 9
2.4.3. Installation des Treibers ..... 9
2.4.4. Anzeigetools ..... 10
2.5. Netzwerkkarten konfigurieren ..... 11
2.6. T-DSL einrichten ..... 11
2.6.1. Logdatei ..... 12
2.7. PPP-Filter ..... 13
2.8. apt-get konfigurieren ..... 13
2.8.1. apt-get verwenden ..... 13
3. Dienste ..... 15
3.1. Grundlegende Dienste ..... 15
3.1.1. DNS-Server ..... 15
3.1.2. dhcp-Server ..... 16
3.1.3. Mail Transfer Agent (SMTP/UUCP) ..... 17
3.2. Fernwartung ..... 21
3.2.1. SSH-Server ..... 21
3.2.2. dyndns ..... 21
3.3. Dienste für Clients ..... 22
3.3.1. Webserver ..... 22
3.3.2. pop3-Server ..... 24
3.3.3. Proxy und Filter ..... 24
3.3.4. Drucker ..... 27
3.3.5. Samba ..... 28
3.4. Sicherheit ..... 32
3.4.1. Firewall ..... 32
3.4.2. inetd ..... 32
3.4.3. netstat und nmap ..... 32
3.4.4. Backup ..... 33
3.4.5. sudoers ..... 34
3.5. Zusätzliches ..... 35
3.5.1. Homepage hochladen ..... 35
3.5.2. Quota ..... 36
3.5.3. ntpdate ..... 36
3.5.4. Transparenter Proxy ..... 36
3.5.5. SMART ..... 37
4. Verwaltung und Wartung ..... 38
4.1. Benutzerverwaltung ..... 38
4.1.1. Benutzer ..... 38

4.1.2. Gruppen ..... 38
4.2. Benutzermanager ..... 39
4.2.1. Benutzer löschen ..... 39
4.2.2. Passwort ändern ..... 39
4.2.3. Rechte eines Benutzers ändern ..... 39
4.2.4. Benutzer sperren bzw. freigeben ..... 40
4.2.5. Neuer Benutzer erstellen ..... 40
4.2.6. Informationen über die Benutzer sammeln ..... 41
4.3. proxymanager ..... 41
4.3.1. Logdatei ..... 42
4.4. Info-Tools ..... 42
4.4.1. Festplattenplatz ..... 42
4.4.2. uptime ..... 43
4.5. Weitere CGI-Scripte ..... 43
4.5.1. Passwort ändern ..... 43
4.5.2. Mail-Weiterleitung konfigurieren ..... 43
4.5.3. Benutzerverzeichnisse ..... 43
5. Clients ..... 44
5.1. Windows 98SE ..... 44
5.1.1. Zusammenarbeit mit dem Netzwerk und Samba ..... 44
5.1.2. Netzwerkinstallation von best. Programmen ..... 45
5.1.3. Partitionsmanager ..... 45
5.1.4. Windows beschränken ..... 46
5.2. Windows 2000 Professional / Windows XP Professional ..... 48
5.2.1. Benutzerprofile, Roaming Profiles ..... 51
5.2.2. Mozilla-Profileinstellungen beim Login kopieren ..... 51
6. Linux-Grundlagen ..... 53
6.1. Einloggen/Ausloggen ..... 53
6.2. Linux herunterfahren (shutdown) ..... 53
6.3. Dateiverwaltung ..... 53
6.3.1. Dateien auflisten ..... 53
6.3.2. Datei/Verzeichnis erstellen ..... 54
6.3.3. Verzeichnis wechseln ..... 54
6.3.4. Datei/Verzeichnis löschen ..... 54
6.3.5. Dateien kopieren ..... 54
6.3.6. Dateien verschieben ..... 54
6.3.7. Textdateien anzeigen ..... 54
6.4. Benutzerverwaltung ..... 55
6.4.1. Benutzer erstellen ..... 55
6.4.2. Benutzer bearbeiten ..... 55
6.4.3. Benutzer löschen ..... 55
6.4.4. Gruppe erstellen ..... 55
6.4.5. Gruppe löschen ..... 55
6.5. Paketverwaltung ..... 55
6.6. Editor ..... 55
6.6.1. Datei öffnen bzw. neue Datei erstellen ..... 55
6.6.2. Dateien bearbeiten und speichern ..... 55
6.7. Prozessmanagement ..... 56
6.8. Linux Support ..... 56
A. Anhang ..... 57
A.1. Firewall-Script ..... 57
A.2. proxy-manager ..... 60
A.2.1. squid-redirector ..... 66
A.2.2. internet/index.cgi ..... 67
A.2.3. inactive.cgi ..... 69
A.3. mail-config/index.cgi ..... 70
A.3.1. Hilfsprogramm change-aliases ..... 71
A.4. passwd/index.cgi ..... 72
A.5. admin/adduser.cgi ..... 74
A.5.1. make_mozilla_config.pl ..... 77
A.6. generate-apache-auth ..... 78
A.7. homepage/mirror.cgi ..... 79
A.7.1. homepage/index.html ..... 80

A.7.2. Hilfsprogramm mirror-hp.sh ..... 81
A.8. usermanager3.cgi ..... 81
A.8.1. Hilfsprogramm user-status3 ..... 96
A.9. benutzer ..... 98
A.10. netzstatus.cgi ..... 99
A.10.1. Hilfsprogramm ping.pl ..... 100
A.11. smbstatus.cgi ..... 100
A.12. samba-logger.cgi ..... 100
B. GNU Free Documentation License ..... 104
B.1. English (original) version ..... 104
B.2. Inoffizielle deutsche Übersetzung ..... 108

# Vorwort

Diese Dokumentation beschreibt den Aufbau des Netzwerkes am Gymnasium Münsingen [http://www.gm.rt.schule-bw.de] Der Schwerpunkt liegt beim Linux-Server. Um die Dokumentation umsetzen zu können, sind Erfahrungen mit Linux notwendig. Möglicherweise hilft auch das sehr, sehr knapp gehaltene Kapitel über Linux-Grundlagen weiter.

Auf der Homepage unter http://www.adabolo.de/projects/server-docu/index.shtml gibt es immer eine aktuelle Version der Dokumentation.

# Kapitel 1. Allgemeines zum Netzwerk

## 1.1. Topologie

Sternförmig

## 1.2. Technik

Technik: Ethernet

## 1.3. Eingesetzte Hardware

*Switch im Informatikraum (R104).* An diesem Switch sind alle Rechner, die im Informatikraum stehen, angeschlossen. Außerdem ist der *Server* für das Schulnetz an diesem Switch über eine 1-GBit/s-Ethernet-Verbindung angeschlossen. An dem Switch hängt ein weitere Switch, der die Rechner in R023, R024, Biologie-, Physik- und Chemie-Sammlung anbindet. Durch den Schulhaus-Erweiterungsbau ist dieser Switch über eine Glasfaser-Verbindung mit einem zusätzlichen Switch verbunden, über den die Rechner im Neubau angebunden sind.

*Verwaltungsnetz.* Das Verwaltungsnetz im Sekretariat besitzt einen eigenen Server und einen eigenen Switch und ist somit komplett vom Schulnetz abgekoppelt.

## 1.4. IP-Adressen

| | |
|---|---|
| Netzwerk | `192.168.0.0` |
| Netzmaske | `255.255.255.0` (kurz: `/24`) |
| Broadcast | `192.168.0.255` |
| Server-IP | `192.168.0.1` |
| Netzdrucker R104 | `192.168.0.254` |
| Oki-Page Netzwerkdrucker R023 | `192.168.0.139` |
| Farblaserdrucker Minolta R023 | `192.168.0.138` |
| Servername | server |
| Domain | gm.rt.schule-bw.de |

# Kapitel 2. Installation des Servers

## 2.1. Debian installieren

### 2.1.1. Debian besorgen

Auf dem Server soll Debian 3.0 ("Woody") installiert werden. Die Debian-Linux-Distribution wird komplett unter der GNU GPL 2 entwickelt. Daher lassen sich auch die CDs ganz legal aus dem Internet herunterladen. Wo und wie das geht, steht auf http://www.debian.de. Insgesamt müssen 7 CDs heruntergeladen werden.

### 2.1.2. Installation starten

Um die Installation zu starten, muss von der ersten CD gebootet werden. Im Bootmenü der CD **`bf24`** eingeben und die Eingabetaste drücken. So wird der 2.4-Kernel gebootet. Dieser wird benötigt, damit wir das neue ReiserFS-Dateisystem verwenden können.

### 2.1.3. Partitionierung

Debian wird auf die erste Festplatte installiert. Diese muss allerdings noch partitioniert werden. Es wird von einer 80 GB großen Festplatte ausgegangen. Auf dieser werden drei primäre Partitionen benötigt:

1. `/dev/hda1`: die root-Partition (/); ca. 70 GB; Dateisystem: ReiserFS
2. `/dev/hda2`: die var-Partition (/var); ca. 10 GB; Dateisystem: ReiserFS; hier liegen verschiedene Daten, z.B. Emails.
3. `/dev/hda3`: die swap-Partition; ca. 500 MB

### 2.1.4. Weiterer Verlauf

Einfach den Anweisungen am Bildschirm folgen. Spezielle Module (= Treiber) brauchen nicht installiert werden. lilo kann in den MBR der ersten Festplatte installiert werden. Nach einem Neustart geht die Installation weiter. Wir verwenden keine MD5 verschlüsselte Passwörter, da wir die Benutzerdatenbank des alten Servers weiterverwenden wollen. Danach kommt die Frage nach dem root-Passwort; dieses Passwort unbedingt sehr gut merken. Einen normalen Benutzer legen wir nicht an. Die Programme tasksel und dselect verwenden wir nicht. Wir installieren die Programme, die benötigt werden, bei Bedarf. So haben wir eine minimalste Software-Auswahl und keine unnötige Programme. Wenn die Konfiguration des Mail-Servers exim kommt, die Option 5 wählen: Wir konfigurieren den Mail-Server später.

Nachdem die Installation vorüber ist, installieren wir zunächst noch einige wichtige Programme: less, vim, bzip2 und aptitude. less dient zum Anzeigen von Dateien, vim ist ein einfacher Editor, mit bzip2 lassen sich Dateien komprimieren und aptitude ist ein einfach zu bedienendes Installationsprogramm für die Debian-Pakete. Die Installation dieser Pakete funktioniert als root mit **apt-get install less vim bzip2 aptitude**.

Um als root bei der Ausgabe von **ls** Farben zu erhalten, muss die Datei `/root/.bashrc` geändert werden. Es müssen lediglich die Kommentar-Zeichen vor einigen Zeilen entfernt werden, sodass die Datei ungefähr so aussieht:

```
#
# /root/.bashrc
#

[...]

# You may uncomment the following lines if you want 'ls' to be
# colorized:
export LS_OPTIONS='--color=auto'
```

```
eval `dircolors`
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'

[...]
```

# 2.2. Kernel machen

Hier wird beschrieben, wie ein Kernel aus den Quellcodes gemacht wird. Dazu werden mehrere Schritte benötigt.

1. Zunächst einmal muss dafür gesorgt werden, dass die benötigten Programme und Bibliotheken installiert sind. Wenn der Kernel zum ersten Mal nach der Installation von Debian gemacht wird, sind noch folgende Programme nachzuinstallieren: binutils, cpp, g++, gcc, make, patch, libncurses5-dev. Das geht mit folgendem Befehl in einem Rutsch (als root): **apt-get install binutils cpp g++ gcc make patch libncurses5-dev**

2. Jetzt muss der Kernel noch besorgt werden. Den aktuellen Quellcode gibt's unter http://www.kernel.org. Die Datei heißt z.B. `linux-2.4.21.tar.bz2` und ist ca. 30 MB groß.

3. Um den Hardware-Sensor (Temperaturanzeige) anzusteuern, werden noch extra Treiber benötigt. Der Kernel muss gepatcht werden. Die dafür notwenigen Dateien sind unter http://secure.netroedge.com/~lm78/ zu finden. Es werden zwei Dateien benötigt. Sie heißen etwa `i2c-2.8.0.tar.gz` und `lm_sensors-2.8.0.tar.gz`.

   Außerdem wird ein Patch benötigt, damit Quota unter dem ReiserFS-Dateisystem funktioniert. Mit Quota lässt sich die Festplattenbenutzung pro Benutzer begrenzen. Patches für aktuelle Kernel sind unter ftp://www.suse.com/pub/people/mason/patches/data-logging/ zu finden. Im Verzeichnis `2.4.21/` sind verschiedene Patches für den Kernel der Version 2.4.21 vorhanden. Für Quota werden nur `07-quota-v2-2.4.21.diff.gz`, `08-reiserfs-quota-28.diff.gz` und `09-kinoded-8.diff.gz` benötigt. Näheres steht in der `README`-Datei.

4. Nachdem jetzt alle benötigten Quellcodes zusammen sind, muss der Kernel zuerst mal entpackt werden. Das geschieht in einem Unterverzeichnis im Homeverzeichnis von root (`/root/kernel/`) mit diesen Befehlen:

   ```
   bash# cd /root
   bash# mkdir kernel
   bash# cd kernel
   bash# tar xfjv /root/linux-2.4.21.tar.bz2
   ```

   Dabei wird angenommen, dass alle Dateien, die heruntergeladen wurden, sich im Homeverzeichnis von root (`/root/`) befinden.

   Nun kommen die Patches für den Hardware-Sensor dran. Folgende Schritte sind auszuführen, um die Treiber zu entpacken und den Kernel anschließend zu patchen:

   ```
   bash# cd /root/kernel
   bash# tar xfzv /root/i2c-2.8.0.tar.gz
   bash# cd i2c-2.8.0
   bash# mkpatch/mkpatch.pl . ../linux > ../i2c-patch
   bash# cd ../linux
   bash# patch -p1 -E < ../i2c-patch

   bash# cd /root/kernel
   bash# tar xfzv /root/lm_sensors-2.8.0.tar.gz
   bash# cd lm_sensors-2.8.0
   bash# mkpatch/mkpatch.pl . ../linux > ../lm_sensors-patch
   bash# cd ../linux
   bash# patch -p1 -E < ../lm_sensors-patch
   ```

Jetzt muss noch der ReiserFS-Quota-Patch verwendet werden. Das geht mit folgenden Befehlen:

```
bash# cd /root/kernel/linux
bash# zcat /root/07-quota-v2-2.4.21.diff.gz | patch -p1
bash# zcat /root/08-reiserfs-quota-28.diff.gz | patch -p1
bash# zcat /root/09-kinoded-8.diff.gz | patch -p1
```

5. Jetzt kann der Kernel endlich konfiguriert werden. Das Konfigurationsprogramm wird mit **make menuconfig** im Verzeichnis `/root/kernel/linux/` gestartet.

   Bei der Konfiguration muss sichergestellt werden, dass unter "Multi-device support (RAID and LVM)" "RAID-1 (mirroring) mode" fest in den Kernel eingebunden wird. Für Software-RAID siehe auch Abschnitt 2.3, „Software-RAID“.

   Wichtig ist außerdem auch noch, dass unter "File systems" das ReiserFS-Dateisystem fest eingebunden wird. Die Treiber für die Netzwerkkarten werden als Module ausgewählt: Unter "Network device support", "Ethernet (10 or 100Mbit)" die Karten "EtherExpressPro/100 support (e100..." und unter "3COM cards" "3c590/3c900 series..." auswählen.

6. Nach der Konfiguration kann der Kernel mit den Modulen compiliert werden. Einfach **make dep bzImage modules modules_install** eingeben.

7. Der neue Kernel muss jetzt noch installiert werden. Mit **cp /root/kernel/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.4.21** das Kernel-Image an die richtige Stelle kopieren. Danach muss `/etc/lilo.conf` bearbeitet werden: Die erste Zeile, die mit "image=" anfängt, muss in "image=/boot/vmlinuz-2.4.21" abgeändert werden. Danach mit **lilo** lilo neu installieren und den Computer anschließend neu starten.

8. Nach dem Neustart können mit **modconf** die erstellten Module so konfiguriert werden, dass sie beim nächsten Neustart automatisch geladen werden. Wichtige Module sind hier die Module für den Hardware-Sensor (i2c-isa, w83..d), die Netzwerkkarten (e100, 3c59x) und das Modul für APM (Advanced Power Management; apm). Damit die Treiber für den Hardware-Sensor einen Sinn ergeben, wird noch das Paket lm-sensors benötigt. Es wird mit **apt-get install lm-sensors** installiert. Wenn die Treiber geladen sind, kann mit **sensors** die aktuelle Temperatur angezeigt werden.

# 2.3. Software-RAID

Software-RAID dient zur besseren Sicherheit der Daten auf dem Server. RAID kann in verschiedenen Modi betrieben werden. Der Modus, der weiter unten eingerichtet wird, heißt RAID-1. Dabei befinden sich im Server zwei Festplatten annähernd gleicher Kapazität. Der Software-RAID-Treiber von Linux sorgt dann dafür, dass sich auf beiden Festplatten exakt der gleiche Inhalt befindet. Die Festplatten werden sozusagen gespiegelt. Fällt eine Festplatte aus, so ist die andere immer noch voll funktionsfähig.

Die Anleitung weiter unten beschreibt, wie ein vorhandenes Linux-System nachträglich in ein RAID-1-System umgewandelt werden kann. Bei modernen Linux-Distributionen kann man möglicherweise schon bei der Installation Software-RAID einrichten; dann kann dieser Abschnitt komplett übersprungen werden.

## 2.3.1. Voraussetzungen

Um Software-RAID einzurichten, werden zunächst einmal die raidtools in der Version 0.9 oder höher benötigt; ob die raidtools bereits installiert sind, lässt sich mit **mkraid -V** herausfinden. Das passende Debian-Paket zum Nachinstallieren heißt "raidtools2". Dann muss sehr wahrscheinlich noch ein neuer Kernel der Version 2.4.x gemacht werden; dieser Vorgang ist genauer in Abschnitt 2.2, „Kernel machen“ beschrieben.

Natürlich werden auch zwei Festplatten benötigt, die ungefähr die gleiche Größe haben sollten. Diese sind - sofern IDE-Festplatten - idealerweise jeweils als Master angeschlossen und als `/dev/hda` und `/dev/hdc` ansprechbar.

Die Ausgangssituation ist folgende: Auf der ersten Festplatte (`/dev/hda`) ist bereits ein Linux-System installiert. Dabei ist es nicht wichtig, dass es konfiguriert ist, denn das lässt sich später noch genau so gut

konfigurieren. Es ist sogar eher noch besser, wenn Software-RAID vor der eigentlichen Konfiguration des Systems eingerichtet wird. Die Anleitung bezieht sich auf die Linux-Installation, die weiter oben beschrieben ist.

## 2.3.2. Schritt für Schritt-Anleitung

Alle folgenden Schritte werden als Superuser (root) durchgeführt.

1. Zuerst muss ein frischer Kernel gemacht werden. Dabei sind folgende Einstellungen zu machen:

   ```
   "Multi-device support (RAID and LVM)"
   [y] Multiple devices driver support (RAID and LVM)
   [y] RAID support
   [y] RAID-1 (mirroring) mode
   ```

   Wichtig ist hier, dass der RAID-Treiber fest im Kernel ist und keine Module erstellt werden. Eine genaue Anleitung für die Erstellung und Installation des neuen Kernels ist in Abschnitt 2.2, „Kernel machen“ beschrieben.

2. Jetzt muss der neue Kernel getestet werden, d.h. es muss mit dem neuen Kernel gebootet werden. Wenn alles soweit funktioniert hat, kann mit dem nächsten Schritt weitergemacht werden. Übrigens: Der Kernel kann auch später nochmal aktualisiert werden; spezielle Hardware kann also später eingerichtet werden.

3. Nach dem Neustart kann mit **cat /proc/mdstat** überprüft werden, ob der neue Kernel auch tatsächlich RAID unterstützt. Dabei sollte in der ersten Zeile folgendes erscheinen:

   ```
   Personalities : [raid1]
   ```

4. Als nächstes muss die zweite Festplatte so wie die erste Festplatte partitioniert werden. Dabei dürfen die Partitionen auf der zweiten Festplatte auf keinen Fall größer sein als die Partitionen auf der ersten Festplatte. Sie müssen entweder gleich groß oder kleiner sein.

   Die Partitionierung kann z.B. mit dem Programm **cfdisk** vorgenommen werden. Am besten speichert man die Partitionstabelle der ersten Festplatte mit **cfdisk -P t /dev/hda** > **hda-table** in die Datei `hda-table`. Das sieht dann so aus:

   ```
   Partition Table for /dev/hda

   ---Starting---        ----Ending----    Start Number of
   # Flags Head Sect Cyl   ID  Head Sect Cyl    Sector   Sectors
   -- ----- ---- ---- ---- ---- ---- ---- ---- --------- ---------
   1 0x80     1    1    0 0x83   15   63 1023        63 135742257
   2 0x00    15   63 1023 0x83   15   63 1023 135742320  19531008
   3 0x00    15   63 1023 0x82   15   63 1023 155273328   1028160
   4 0x00     0    0    0 0x00    0    0    0         0         0
   ```

   Wichtig sind die zwei letzten Spalten (Start Sector und Number of Sectors). Mit **cfdisk /dev/hdc** wird die zweite Festplatte partitioniert; es müssen genau die gleichen Partitionen erstellt werden. Das geht am einfachsten über die Sektor-Anzahl. Wenn das Programm nach der Größe fragt, einfach **`12345S`** eingeben, also die Anzahl der Sektoren mit angehängtem S. Zum Schluss müssen die Partitionstabellen der beiden Festplatten nocheinmal verglichen werden, damit auch kein Fehler unterlief; also: Partitionstabelle der zweiten Festplatte wie oben in eine Datei speichern.

   Hier sollte dann auch auf der zweiten Festplatte gleich das Bootflag gesetzt werden (auf `/dev/hdc1`). Außerdem müssen alle Partitionen außer dem swap-Partition den Partitionstyp `fd` (Linux raid autodetect) haben.

5. Jetzt muss die Konfigurationsdatei für den Software-RAID-Treiber geschrieben werden. Diese steht in `/etc/raidtab`. Im Moment ist dabei die erste Festplatte (`/dev/hda`) als fehlerhaft zu markieren.

   ```
   #
   # /etc/raidtab
   #
   ```

```
# /dev/md0 ist die root-Partition
raiddev                 /dev/md0
raid-level              1               # RAID-1 (mirroring)
nr-raid-disks           2
nr-spare-disks          0
persistent-superblock   1
chunk-size              32

device                  /dev/hdc1
raid-disk               0
# dies ist die erste Festplatte, die hier zunächst als fehlerhaft
# markiert ist
device                  /dev/hda1
failed-disk             1


# /dev/md1 ist die var-Partition
raiddev                 /dev/md1
raid-level              1               # RAID-1 (mirroring)
nr-raid-disks           2
nr-spare-disks          0
persistent-superblock   1
chunk-size              32

device                  /dev/hdc2
raid-disk               0
# die erste Festplatte ist hier genauso als fehlerhaft markiert
device                  /dev/hda2
failed-disk             1
```

6. Danach können dieraid-Geräte gemacht werden:

   ```
   bash# mkraid /dev/md0
   bash# mkraid /dev/md1
   ```

   Mit **cat /proc/mdstat** kann nachgeschaut werden, ob die Geräte registriert sind. Die erste Festplatte ist hier als fehlerhaft eingetragen.

7. Anschließend können die raid-Geräte formatiert und gemountet werden. Da die erste Festplatte ja als fehlerhaft markiert ist, wird sie nicht formatiert, sondern nur die zweite Festplatte. Die Daten auf der ersten Festplatte bleiben also vorhanden.

   Das Formatieren funktioniert so:

   ```
   bash# mkreiserfs /dev/md0
   bash# mkreiserfs /dev/md1
   ```

   Für das Mounten der Dateisysteme müssen folgende Befehle ausgeführt werden:

   ```
   bash# mount /dev/md0 /mnt
   bash# mkdir /mnt/var
   bash# mount /dev/md1 /mnt/var
   ```

8. Jetzt können die Dateien der ersten Festplatte auf die zweite Festplatte kopiert werden. Das geht ganz einfach mit folgenden Befehlen:

   ```
   bash# cd /
   bash# find . -xdev -name "*" -o -name ".*" | cpio -pmv /mnt
   bash# cd /var
   bash# find . -xdev -name "*" -o -name ".*" | cpio -pmv /mnt/var
   ```

9. Danach muss wieder getestet werden. Wir aktualisieren dafür zuerst folgende Datei:

```
#
# /mnt/etc/fstab
#

/dev/md0    /       reiserfs    defaults    0   0
/dev/md1    /var    reiserfs    defaults    0   0
[...]
```

Wir ersetzen also `/dev/hda1` mit `/dev/md0` und `/dev/hda2` mit `/dev/md1`.

Beim nächsten Start mit dem neu eingerichteten RAID werden so die raid-Geräte automatisch verwendet.

10. Nun muss noch eine Bootdiskette erstellt werden. Dazu wird eine leere, formatierte und fehlerfreie Diskette benötigt. Die Diskette in Laufwerk A: einlegen und folgende Befehle ausführen:

```
bash# dd if=/boot/vmlinuz-2.4.21 of=/dev/fd0 bs=18k
bash# rdev /dev/fd0 /dev/md0
bash# rdev -r /dev/fd0 0
bash# rdev -R /dev/fd0 1
```

Auf der so erstellten Bootdiskette ist der neue Kernel mit Software-RAID-Unterstützung (`/boot/vmlinuz-2.4.21` ist das Kernel-Image). Diese Bootdiskette kann später vielleicht mal sehr wichtig werden, z.B. wenn der Server aus irgendeinem Grund nicht mehr von der Festplatte booten kann. Deshalb sollte die Bootdiskette gut aufbewahrt werden und bei jedem Kernel-Update auch aktualisiert werden.

11. Jetzt kann der Server neu gestartet werden. Dazu müssen aber erst die raid-Geräte gestoppt werden:

```
bash# umount /mnt/var
bash# umount /mnt
bash# raidstop /dev/md0
bash# raidstop /dev/md1
```

Wenn die Bootdiskette eingelegt wurde, kann mit **shutdown -r now** der Server neu gestartet werden.

12. Nach dem Neustart muss geprüft werden, ob alle Daten auf die zweite Festplatte kopiert wurden. Wenn der Server ohne Probleme neugestartet ist, sollte dies der Fall sein. Wenn nicht, kann der Server immer noch ohne Diskette von der ersten Festplatte gestartet werden. Dann müssen wie in Punkt 7 beschrieben die raid-Geräte wieder gemountet werden, damit eventuell vergessene Dateien kopiert werden können.

13. Wir haben von der Diskette gebootet. Jetzt muss die erste Festplatte, die im Moment ja noch als fehlerhaft markiert ist, dem raid-Array hinzugefügt werden.*Dabei gehen alle Daten auf dieser Festplatte verloren!*

Als erstes müssen mit **cfdisk /dev/hda** die Partitionstypen der Partitionen auf der ersten Festplatte zu `fd` (Linux raid autodetect) geändert werden (die swap-Partition bleibt swap). Auch das Bootflag sollte nochmal frisch auf `/dev/hda1` gesetzt werden.

Dann muss in der Datei `/etc/raidtab` zweimal `failed-disk` durch `raid-disk` ersetzt werden.

Und schließlich wird mit folgenden Befehlen die erste Festplatte dem raid-Array hinzugefügt:

```
bash# raidhotadd /dev/md1 /dev/hda2
bash# raidhotadd /dev/md0 /dev/hda1
```

Mit **cat /proc/mdstat** kann der Fortschritt der Synchronisation angezeigt werden.

14. Während der Synchronisationsvorgang noch läuft, kann das System schon bootfähig gemacht werden. Dazu muss `/etc/lilo.conf` editiert werden. Die Datei sieht dann so aus:

```
#
# /etc/lilo.conf
#

boot=/dev/md0
```

```
raid-extra-boot=/dev/hda,/dev/hdc

read-only
prompt
timeout=50
root=/dev/md0

image=/boot/vmlinuz-2.4.21
label=Linux
```

Jetzt muss noch lilo auf beiden Festplatten installiert werden. Dies geschieht einfach durch Aufruf von **lilo**.

15. Schließlich muss noch die swap-Partition auf der zweiten Festplatte eingerichtet werden. Dazu muss `/etc/fstab` verändert werden.

```
#
# /etc/fstab
#

[...]
/dev/hda3   none    swap    sw,pri=1    0   0
/dev/hdc3   none    swap    sw,pri=1    0   0
```

Außerdem muss die swap-Partition noch mit **mkswap /dev/hdc3** formatiert werden. Beim nächsten Neustart sind dann beide swap-Partitionen aktiviert.

16. Wenn der Synchronisationsvorgang beendet ist, kann neu gestartet werden, um die Installation zu testen.

Falls das Booten von der Festplatte nicht funktioniert, kann immer noch auf die Bootdiskette zurückgegriffen werden.

## 2.3.3. Fehlerfall

Fällt einmal tatsächlich eine Festplatte aus, muss diese ersetzt werden. Im Folgenden wird angenommen, dass die erste Festplatte (`/dev/hda`) ausfällt. Die einzelnen Schritte sind dann wie folgt:

1. Erst muss der Server heruntergefahren werden, wenn er noch läuft.
2. Dann kann die defekte Festplatte ausgetauscht werden.
3. Jetzt muss gebootet werden. Wenn der Server nicht automatisch von der zweiten (intakten) Festplatte bootet, muss eine Bootdiskette verwendet werden (siehe voriger Abschnitt, Punkt 10).
4. Die neue Festplatte muss nun partitioniert werden. Die swap-Partition nicht vergessen: Mit **mkswap /dev/hda3** die swap-Partition formatieren.
5. Mit dem Befehl **raidhotadd** den Synchronisationsvorgang starten:

```
bash# raidhotadd /dev/md1 /dev/hda2
bash# raidhotadd /dev/md0 /dev/hda1
```

6. Mit **lilo** lilo neu installieren.
7. Nach dem Synchronisationsvorgang kann von der Festplatte neu gebootet werden.

*Hinweis*: Falls das BIOS nicht von der zweiten Festplatte booten kann, wird unbedingt eine Bootdiskette benötigt.

# 2.4. APC SmartUPS (USV)

# 2.4.1. Zweck

Eine USV (Unterbrechungsfreie Stromversorgung) soll, wie der Name schon sagt, die Stromversorgung sicherstellen. Dabei kann man sich die USV als einen sehr großen Akku vorstellen. Fällt der Strom aus, dann wird der Server weiterhin mit Strom versorgt - aus dem Akku. Außerdem schützt die USV den Server vor Überspannung.

# 2.4.2. Möglichkeiten ohne Treiber

Wenn die USV einfach so, ohne Treiber an den Server angeschlossen wird, dann springt sie automatisch ein, wenn der Strom ausfällt. Wenn der Akku leer ist, fällt der Strom komplett aus; der Server wird nicht heruntergefahren! Die USV ohne Treiber ist also nur für kurze Stromausfälle geeignet.

# 2.4.3. Installation des Treibers

Wenn der Treiber installiert ist, meldet die USV den Stromausfall. Dann kann der Server entsprechend reagieren. Reagiert er nicht, dann fährt die USV den Server herunter, sobald die Akkuladung einen kritischen Wert erreicht hat. Der Server kann aber auch reagieren und z.B. nach 60 Sekunden herunterfahren. Dadurch kann verhindert werden, dass der Akku beim möglicherweise etwas langen Herunterfahren versagt.

Die USV wird über die erste serielle Schnittstelle (COM1) an den Server angeschlossen. Zur Installation des Treibers wird das Programmpaket *nut* benötigt. Die Version, die bei Debian Woody (stable) dabei ist, ist offenbar zu alt und funktioniert nicht richtig. Deshalb verwenden wir eine neuere Version. Unter http://www.logic.at/debian/i386/ gibt es neuere Versionen für Woody. Hier die Dateien `nut_1.4.2-2woody1_i386.deb` und `nut-cgi_1.4.2-2woody1_i386.deb` (o.ä. je nach Versionsnummer) herunterladen. Diese Dateien lassen sich mit **dpkg -i *.deb** installieren. Dann ist die neue Version installiert.

Konfiguriert wird der Treiber über mehrere Dateien. In `/etc/nut/ups.conf` müssen zunächst folgende Zeilen hinzugefügt werden:

```
[myups]
    driver = apcsmart
    port = /dev/ttyS0
    cable = 940-0095B
```

Danach muss mit **chown root.nut /dev/ttyS0** dem Treiber Zugriff auf die COM1-Schnittstelle erlaubt werden. Dann muss `/etc/nut/upsd.conf` bearbeitet werden: Die Zeile mit "#ACCESS grant master localhost [<upsmon-password>]" suchen und ändern in:

```
ACCESS grant master localhost passwort
```

Damit wird dem Monitorprogramm erlaubt, den Treiber zu befragen. Dabei wird das Passwort "passwort" verwendet. Der Treiber und das Monitorprogramm müssen aber auf demselben Rechner laufen, sodass das Passwort nicht so sicherheitsrelevant ist.

In der Datei `/etc/nut/upsmon.conf` muss im ersten Abschnitt noch folgende Zeile ergänzt werden:

```
MONITOR myups@localhost 1 username passwort master
```

In `/etc/nut/upsd.users` wird noch ein Benutzer definiert:

```
[username]
    password = passwort
    allowfrom = localhost
    upsmon master
```

Damit ist die einfache Konfiguration abgeschlossen. Mit **/etc/init.d/nut start** wird der Treiber geladen.

Damit der Server nach einer bestimmten Zeit nach dem Stromausfall automatisch herunterfährt, ist weitere Arbeit nötig. In der Datei `/etc/nut/upsmon.conf` müssen in den entsprechenden Abschnitten folgende

Zeilen hinzugefügt werden:

```
NOTIFYCMD /sbin/upssched

NOTIFYFLAG ONLINE SYSLOG+EXEC

NOTIFYFLAG ONBATT SYSLOG+WALL+EXEC
```

Wenn der Strom ausfällt (ONBATT) oder wieder da ist (ONLINE) wird das Programm **/sbin/upssched** ausgeführt. Jetzt muss die Datei `/etc/nut/upssched.conf` mit folgendem Inhalt erstellt werden:

```
#
# /etc/nut/upssched.conf
#

CMDSCRIPT /usr/local/sbin/go-down
PIPEFN /var/run/upssched.pipe
AT ONBATT * START-TIMER onbattwarn 40
AT ONLINE * CANCEL-TIMER onbattwarn
```

Mit dieser Konfiguration wird bei Stromausfall (ONBATT) ein 40-Sekunden-Timer gestartet, der nach Ablauf das Script `/usr/local/sbin/go-down` ausführt. Wird die Stromversorgung vor Ablauf des Timers wiederhergestellt (ONLINE), wird der Timer abgebrochen.

Das Script `/usr/local/sbin/go-down` sieht so aus:

```
#!/bin/sh
#
# /usr/local/sbin/go-down
#
upsmon -c fsd
```

Die Zugriffsrechte für dieses Script werden mit **chmod 700 /usr/local/sbin/go-down** richtig gesetzt. Mit **/etc/init.d/nut restart** wird die neue Konfiguration verwendet. Damit ist die Konfiguration komplett abgeschlossen.

Jetzt sollte die USV noch getestet werden. Wird das Stromkabel an der USV herausgezogen, sollte in `/var/log/syslog` eine Meldung erscheinen. Nach 40 Sekunden sollte der Server herunterfahren.

# 2.4.4. Anzeigetools

Die Anzeigetools sind CGI-Programme für einen Webserver. Sie können also erst richtig genutzt werden, wenn ein Webserver läuft. Trotzdem können sie auch schon jetzt eingerichtet werden.

Die Programme wurden schon im letzten Abschnitt installiert. Zur Konfiguration muss zunächst die Datei `/etc/nut/hosts.conf` erstellt werden:

```
#
# /etc/nut/hosts.conf
#
MONITOR myups@localhost "Local UPS"
```

Weiter muss die Datei `/etc/nut/upsset.conf` erstellt werden. Diese Datei ist nötig, um dem upsset.cgi-Programm mitzuteilen, dass die CGI-Programme auf dem Webserver gesichert sind und nur vom internen Netzwerk zu erreichen sind. Das muss bei der späteren Konfiguration des Webserver beachtet werden! Die Datei hat folgenden Inhalt:

```
#
# /etc/nut/upsset.conf
#
I_HAVE_SECURED_MY_CGI_DIRECTORY
```

Die CGI-Scripte liegen in `/usr/lib/cgi-bin/nut/` und sollten, wenn der Webserver konfiguriert ist, in ein entsprechendes cgi-bin-Verzeichnis kopiert werden.

# 2.5. Netzwerkkarten konfigurieren

Im Server sind zwei Netzwerkkarten. Eine für das Netzwerk und die andere für den DSL-Anschluss. Damit die Netzwerkkarten konfiguriert werden können, müssen zunächst einmal die Treiber geladen sein. Der Befehl **ifconfig -a** gibt Aufschluss über die installierten Netzwerk-Geräte:

```
bash# ifconfig -a
eth0    Link encap:Ethernet  HWaddr 00:10:DC:CD:61:59
        BROADCAST MULTICAST  MTU:1500 Metric:1
        [...]

eth1    Link encap:Ethernet  HWaddr 00:50:DA:90:8E:A9
        BROADCAST MULTICAST  MTU:1500 Metric:1
        [...]

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        [...]
```

Hier sind also zwei Ethernet-Netzwerkkarten (`eth0` und `eth1`) und ein loopback-Gerät installiert (`lo`). `eth0` ist die Netzwerkkarte onboard, also die mit dem Treiber *e100*. Dieser Treiber wurde vor dem der zweiten Netzwerkkarte `eth1` geladen (*3c59x*).

Zur Konfiguration: Die Einstellungen wie IP-Adresse stehen in der Datei `/etc/network/interfaces`. Diese muss so aussehen:

```
#
# /etc/network/interfaces
#

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.0.1
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255

auto eth1
iface eth1 inet static
    address 192.168.22.1
    netmask 255.255.255.0
    network 192.168.22.0
    broadcast 192.168.22.255
```

Der Server erhält also die IP-Adresse `192.168.0.1`. Die zweite Netzwerkkarte (`eth1`) wird mit einer fiktiven IP-Adresse konfiguriert, die nie wirklich benutzt wird. Sie ist vielmehr ein Platzhalter. Über diese zweite Netzwerkkarte läuft später der DSL-Verkehr.

Um die Konfiguration zu übernehmen muss noch der Befehl **/etc/init.d/networking restart** ausgeführt werden. Beim nächsten Neustart wird die Konfiguration automatisch übernommen.

# 2.6. T-DSL einrichten

Zunächst muss das DSL-Modem mit der zweiten Netzwerkkarte verbunden werden. Die zweite Netzwerkkarte ist bereits wie oben beschrieben konfiguriert (IP-Adresse: `192.168.22.1`). Um den DSL-Anschluss zu konfigurieren, werden drei Pakete benötigt, die mit **apt-get install pppoe pppoeconf pppstatus** installiert werden.

Eine weitere Voraussetzung für den DSL-Zugang sind bestimmte Treiber, die als Kernel-Module geladen werden können. Es wird auf jeden Fall das Modul *ppp_async* benötigt. Das Modul *pppoe* wird nicht benötigt; die Aufgabe dieses Treibers wird von der DSL-Software übernommen.

Nach der Installation der Software-Pakete startet automatisch das Programm **pppoeconf**; falls nicht, muss es manuell gestartet werden: einfach **pppoeconf** eingeben. Im ersten Dialog sollten beide Netzwerkkarten (`eth0` und `eth1`) zu sehen sein. Das Programm sucht dann an beiden Netzwerkkarten automatisch nach einem DSL-Modem. Einfach den Schritten auf dem Bildschirm folgen.

Der Benutzername für die DSL-Einwahl wird aus den Daten von T-Online aufgebaut:

Anschlusskennung: **0001111111111**

T-Online-Nummer: **22222222222**

Mitbenutzer-Suffix: **0001**

Daraus ergibt sich der Benutzername: **000111111111122222222220001@t-online.de**. Dann wird natürlich noch das Passwort benötigt. Die Frage, ob `/etc/resolv.conf` beim Verbindungsaufbau mit DNS-Servern vom Provider gefüllt werden soll, ist mit nein zu beantworten. Wir verwenden später unseren eigenen DNS-Server. Nach einigen weiteren Fragen, die wie empfohlen beantwortet werden sollten, ist die Konfiguration abgeschlossen.

Mit dem Programm **pppstatus** kann der Status der DSL-Verbindung abgefragt werden. Mit **pon dsl-provider** wird eine Verbindung aufgebaut, mit **poff** wieder abgebaut.

Um "dial on demand" einzurichten, also die Einwahl bei Bedarf, muss die Datei `/etc/ppp/peers/dsl-provider` bearbeitet werden. In "Section 2" müssen folgende zwei Zeilen stehen:

```
demand
idle 120
```

Dann wird die Verbindung bei Bedarf aufgebaut und wenn 120 Sekunden lang keine Daten gesendet oder empfangen werden, wird die Verbindung getrennt.

Damit der ppp-Dämon bei Systemstart geladen wird und somit die Internetverbindung verfügbar ist, muss im Verzeichnis `/etc/ppp/` mit dem Befehl **ln -s ppp_on_boot.dsl ppp_on_boot** ein symbolischer Link erstellt werden. Außerdem sollte in der Datei `/etc/ppp/ppp_on_boot.dsl` die Variable *INTERFACE* mit `eth1` belegt sein; dies ist gegebenenfalls anzupassen.

# 2.6.1. Logdatei

Soll mitgeloggt werden, wann der Server online geht und wann wieder offline, muss in `/etc/ppp/ip-up.d/` und `/etc/ppp/ip-down.d/` jeweils ein Script installiert werden. Das erste Script heißt `/etc/ppp/ip-up.d/log` und muss mit folgendem Inhalt erstellt werden:

```
#!/bin/sh
#
# /etc/ppp/ip-up.d/log
#

logfile="/var/log/dsl-einwahl"
date=`date`
echo $date up >> $logfile
```

Das zweite Script heißt `/etc/ppp/ip-down.d/log` und muss mit folgendem Inhalt erstellt werden:

```
#!/bin/sh
#
# /etc/ppp/ip-down.d/log
#

logfile="/var/log/dsl-einwahl"
date=`date`
echo $date down >> $logfile
```

Beide Scripte werden mit **chmod 755 /etc/ppp/ip-*.d/log** ausführbar gemacht. In der Datei `/var/log/dsl-einwahl` steht dann, wann sich der Server entweder eingewählt oder ausgewählt hat.

Damit die Logdatei nicht ins Unendliche anwächst, kann noch logrotate konfiguriert werden. Dafür wird die Datei `/etc/logrotate.d/dsl-einwahl` mit folgendem Inhalt erstellt:

```
/var/log/dsl-einwahl {
    rotate 7
    weekly
    compress
    missingok
    notifempty
}
```

Jetzt wird jede Woche die Logdatei "rotiert", d.h. die Datei wird umbenannt und komprimiert. Es werden maximal 7 Dateien behalten, also man kann die DSL-Einwahl 7 Wochen zurückverfolgen.

# 2.7. PPP-Filter

Der Internet-Zugang ist oben so konfiguriert worden, dass sich der Server automatisch (bei Bedarf) einwählt und wenn 120 Sekunden lang keine Daten mehr gesendet oder empfangen wurden, beendet er die Verbindung wieder. Das Problem dabei sind heute die sogenannten Peer-to-Peer-Tauschbörsen wie eDonkey. So kommen relativ oft unerwünschte Pakete über den T-Online-Zugang am Server an. Sie stellen zwar kein Sicherheitsrisiko dar, zählen aber trotzdem zum Traffic. Linux registriert, dass Daten empfangen wurden und beendet die Verbindung nicht. Um dieses Szenario zu verhindern, muss Linux so eingerichtet werden, dass es nur noch ausgehende Datenpakete zählt. Da aber die Pakete trotzdem empfangen werden, reagiert Linux normalerweise mit einer Fehlerpaket als Antwort; dies muss später noch mit einer Firewall verhindert werden.

Um den Filter einzurichten, muss am Ende der Datei `/etc/ppp/options` folgende Zeile hinzugefügt werden:

```
active-filter 'outbound'
```

Mit dem Befehl **/etc/init.d/ppp restart** wird diese Konfiguration übernommen.

*Hinweis:* Dabei wird eine bestehende Internetverbindung getrennt.

# 2.8. apt-get konfigurieren

Das Programm **apt-get** wird immer benutzt, wenn neue Software installiert werden soll. Da wir über eine DSL-Verbindung ins Internet verfügen, kann apt-get so eingerichtet werden, dass es die neue Software automatisch aus dem Internet lädt. Somit entfällt das lästige CD-Einlegen. Außerdem lässt sich so das Debian-System einfach aktuell halten.

Zunächst löschen wir die alte Konfiguration: **rm /etc/apt/sources.list**. Dann rufen wird das Setup-Programm auf: **apt-setup**. Hier wählen wir bei der ersten Frage *http* aus. Die nächste Frage beantworten wir mit *No* (wir wollen keine Software verwenden, die nicht frei ist). Im nächsten Dialog wählen wir den Mirror-Server aus: *Germany, ftp.de.debian.org*. Die nächste Frage kann vorerst unbeantwortet bleiben. Wir verwenden im Moment noch keinen Proxy-Server. Später, wenn der Proxy-Server eingerichtet ist, sollte apt-setup nochmal aufgerufen werden und als Proxy-Server sollte *http://localhost:3128* eingegeben werden. Danach wollen wir keine weiteren Quellen mehr hinzufügen (*No*). Die letzte Frage bezieht sich auf Sicherheitsupdates. Sie sollte unbedingt mit *Yes* beantwortet werden.

Ein weiteres, sehr nützliches Programm ist **aptitude**. Mit diesem Programm kann man herausfinden, welche Software-Pakete zur Verfügung stehen und welche Software upgedatet werden kann. Falls es noch nicht installiert ist, kann dies mit **apt-get install aptitude** nachgeholt werden.

## 2.8.1. apt-get verwenden

Mit **apt-get update** werden die Paketlisten neu heruntergeladen. Mit **apt-get upgrade** werden alle auf dem System installierten Pakete auf die aktuelle Version gebracht. (Davor sollte immer **apt-get update** aufgerufen werden.) Mit **apt-get install paketname** wird das Programmpaket mit dem Namen *"paketname"* heruntergeladen und installiert. Mit **apt-get remove paketname** wird ein Programmpaket wieder entfernt (deinstalliert). Mit **apt-get check** wird überprüft, ob sich irgendwelche Unstimmigkeiten in den Paketen finden,

z.B. Abhängigkeitsprobleme. Diese können dann wahrscheinlich mit **apt-get -f install** behoben werden.

# Kapitel 3. Dienste

## 3.1. Grundlegende Dienste

### 3.1.1. DNS-Server

Der Domain-Name-Service (dns) ist für die Umwandlung von Internetadressen (z.B. www.debian.org) in numerische IP-Adressen (198.186.203.20) zuständig. Auch die umgekehrte Richtung ist möglich. Ohne diesen Dienst müsste man immer die IP-Adresse des Servers kennen, auf den man z.B. beim Surfen zugreifen will. Zusätzlich zur IP-Adresse kann der dns-Server auch den Mailserver, der für eine Domain zuständig ist, speichern (sog. MX-Record). Weitere Infos wie Hardware-Ausstattung sind auch möglich, aber eher unüblich. Jede Domain ist in mindestens zwei dns-Servern eingetragen. Welche das sind, kann mit dem Programm **whois** herausgefunden werden.

Der gebräuchlichste dns-Server für Linux heißt *bind*. Mit **apt-get install bind9** wird Version 9 installiert. Der Server selber wird mit der Datei `/etc/bind/named.conf` konfiguriert. Im *options*-Abschnitt müssen folgende Zeilen hinzugefügt werden:

```
options
{
    forwarders {
        217.237.151.161;
        194.25.2.129;
    };

    allow-query {
        127.0.0.0/8;
        192.168.0.0/24;
    };

    allow-transfer {
        none;
    };

    listen-on
    {
        127.0.0.1;
        192.168.0.1;
    };
};
```

Damit ist der dns-Server nur vom internen Netzwerk erreichbar. Schließlich ist er auch nur für das interne Netzwerk zuständig. Am Ende von `/etc/bind/named.conf` müssen noch folgende Zeilen hinzugefügt werden:

```
zone "gm.rt.schule-bw.de" IN {
    type master;
    file "gm.rt.schule-bw.de-zone";
};

zone "0.168.192.in-addr.arpa" IN {
    type master;
    file "192.168.0-zone";
};
```

Jetzt müssen noch die sogenannten Zonen-Dateien erstellt werden. In diesen Dateien liegen die eigentlichen Informationen, die der dns-Server zur Umandlung von Domains in IP-Adressen benötigt. Zuerst erstellen wir die Datei `/var/cache/bin/gm.rt.schule-bw.de-zone`:

```
;
; /var/cache/bind/gm.rt.schule-bw.de-zone
```

```
;
$TTL 2D
@       IN      SOA     dns     root.gm.rt.schule-bw.de. (
                2003081201  ; serial JJJJMMTTNN
                8H          ; refresh rate
                2H          ; retry rate
                1W          ; expiration date
                2D )        ; minimum ttl

        IN      NS      dns
        IN      MX      10 mail

dns     IN      A       192.168.0.1
mail    IN      A       192.168.0.1
pop     IN      A       192.168.0.1
news    IN      A       192.168.0.1
server  IN      A       192.168.0.1
proxy   IN      A       192.168.0.1
www     IN      NS      dns1.belwue.de.
                ; www.gm.rt.schule-bw.de. wird von unserem
                ; dns-server nicht verwaltet, aber von belwue.
```

Diese Datei sorgt für die Auflösung eines Namens in seine IP-Adresse. Die andere Zonen-Datei ist für die umgekehrte Richtung da. `/var/cache/bin/192.168.0-zone` muss folgenden Inhalt haben:

```
;
; /var/cache/bind/192.168.0-zone
;
$TTL 2D

@       IN      SOA         dns.gm.rt.schule-bw.de. root.gm.rt.schule-bw.de. (
                2003081201  ; serial JJJJMMTTNN
                8H          ; refresh rate
                2H          ; retry rate
                1W          ; expiration date
                2D )        ; minimum ttl

        IN      NS          dns.gm.rt.schule-bw.de.

1       IN      PTR         server.gm.rt.schule-bw.de.
```

Mit **/etc/init.d/bind reload** wird die neue Konfiguration übernommen.

Damit unser Server seinen eigenen dns-Server auch benutzt, muss noch die Datei `/etc/resolv.conf` bearbeitet werden. Sie muss folgenden Inhalt haben:

```
#
# /etc/resolv.conf
#
search gm.rt.schule-bw.de
nameserver 192.168.0.1
```

Damit ist die Konfiguration des dns-Servers abgeschlossen.

## 3.1.2. dhcp-Server

Mit dem dhcp-Protokoll lassen sich die Clients automatisch konfigurieren. Da unser Netz die Adresse 192.168.0.0 hat und 192.168.0.1 und 192.168.0.254 schon belegt sind (Server und Netzdrucker), bleibt für die Clients im Prinzip der Adressbereich 2-253 übrig. Um aber für besondere Fälle (z.B. weitere Netzdrucker) noch einige Adressen frei zu haben, sollten die Clients nur folgenden Bereich nutzen: 140-253.

Zuerst muss der dhcp-Server wahrscheinlich installiert werden. Das geht mit dem Befehl **apt-get install dhcp**. Die Konfigurationsdatei heißt `/etc/dhcpd.conf`. Sie muss folgenden Inhalt haben:

```
#
# /etc/dhcpd.conf
```

```
#
default-lease-time 604800; # 7 days
max-lease-time 2592000;    # 30 days

option domain-name "gm.rt.schule-bw.de";
option domain-name-servers          dns.gm.rt.schule-bw.de;
option lpr-servers                server.gm.rt.schule-bw.de;
option netbios-name-servers       server.gm.rt.schule-bw.de;
option time-servers               server.gm.rt.schule-bw.de;
option smtp-server                  mail.gm.rt.schule-bw.de;
option pop-server                    pop.gm.rt.schule-bw.de;
option www-server                 server.gm.rt.schule-bw.de;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;

subnet 192.168.0.0 netmask 255.255.255.0
{
    range 192.168.0.140 192.168.0.253;
}
```

In der Datei `/etc/default/dhcp` steht, auf welchen Netzwerkkarten der dhcp-Server aktiv ist. Hier muss folgende Zeile stehen:

```
INTERFACES="eth0"
```

Mit **/etc/init.d/dhcp restart** wird die Konfiguration übernommen.

# 3.1.3. Mail Transfer Agent (SMTP/UUCP)

In diesem Abschnitt wird die Konfiguration des Mailsystems beschrieben. Zunächst muss *uucp* eingerichtet werden. *uucp* wird zum Abholen der Mails vom Provider verwendet (in unserem Fall: *belwue*). Dann muss der eigentliche MTA (Mail Transfer Agent) eingerichtet werden, der SMTP-Server. Dieser Server erhält über uucp die Mails der Domain *gm.rt.schule-bw.de* und verteilt diese an die Benutzer. Andererseits nimmt der SMTP-Server neue Mails von den Clients an und versendet sie über uucp.

## 3.1.3.1. uucp einrichten

Zunächst einmal muss die uucp-Software installiert werden: **apt-get install uucp**. Alle Konfigurationsdateien liegen in `/etc/uucp/`. Am besten werden in diesem Verzeichnis zuerst alle Konfigurationsdateien gelöscht.

Wir bearbeiten zuerst die Datei `call`. Hier werden drei Informationen benötigt: Der Systemname des Providers (hier: belwue), Login-Name und Passwort.

```
#
# /etc/uucp/call
#
# Format:
# system login passwd

belwue loginname passwort
```

*"loginname"* und *"passwort"* müssen natürlich noch ersetzt werden.

Die nächste Datei heißt `config`. Hier steht der UUCP-Name des lokalen Rechners.

```
#
# /etc/uucp/config
#
nodename loginname
```

Auch hier muss *"loginname"* ersetzt werden.

In der Datei `sys` werden die bekannten Systeme beschrieben, die verwendet werden sollen. Diese Datei sollte folgenden Inhalt haben:

```
#
# /etc/uucp/sys
#

###
# Globale Einstellung für alle Systeme

# Loginnamen und Passwort aus der Datei 'call' lesen
call-login *
call-password *

# Keine Einschränkung der Zugriffszeit
time any

###
# Systemspezifische Einstellungen
system belwue
    called-login    loginname
    commands        rnews rmail rsmtp rcsmtp crsmtp
    myname          loginname
    port type       tcp
    address         news.belwue.de
```

Hier muss wieder *"loginname"* ersetzt werden.

In der Datei `/etc/hosts` muss noch eine Zeile ergänzt werden:

```
129.143.4.4 news.belwue.de belwue
```

Neue Mails werden dann letztendlich mit folgendem Befehl abgeholt: **uucico -r1 -sbelwue -f**.

Die Konfiguration kann natürlich auch getestet werden. Dazu die folgenden Befehle eingeben; die Ausgabe sollte ähnlich sein.

```
server:~# su - uucp
uucp@server:~$ /usr/lib/uucp/uuchk
Local node name loginname
Spool directory /var/spool/uucp
Public directory /var/spool/uucppublic
Lock directory /var/lock
Log file /var/log/uucp/Log
Statistics file /var/log/uucp/Stats
Debug file /var/log/uucp/Debug
Global debugging level
uucico -l will strip login names and passwords
uucico will strip UUCP protocol commands
Start uuxqt once per uucico invocation

System: belwue
 Caller must log in as loginname
 Will use loginname as name of local system
 Call out using a specially defined port
 The port is defined as:
  Port name system belwue port
   Port type tcp
   TCP service uucp
   Characteristics: eight-bit-clean reliable end-to-end fullduplex
 Remote address news.belwue.de
 Chat script "" \r\c ogin:-BREAK-ogin:-BREAK-ogin: \L word: \P
 Chat script timeout 10
 Chat script incoming bytes stripped to seven bits
 Login name loginname
 Password passwort
 At any time may call if any work
```

```
 May retry the call up to 26 times
 May make local requests when calling
 May make local requests when called
 May send by local request: /
 May send by remote request: ~
 May accept by local request: ~
 May receive by remote request: ~
 May execute rnews rmail rsmtp rcsmtp crsmtp
 Execution path /bin /usr/bin /usr/local/bin /usr/sbin
 Will leave 50000 bytes available
 Public directory is /var/spool/uucppublic
 Will use any known protocol
```

Eine weitere Möglichkeit, die Konfiguration zu testen, ist, die Mails abzuholen und dabei Debug-Ausgaben erzeugen zu lassen. Das geht mit **uucico -r1 -x9 -sbelwue -f** recht einfach. In `/var/log/uucp/Debug` stehen die ausführlichen Debug-Ausgaben. Die Datei `/var/log/uucp/Log` wird immer erweitert, wenn uucp benutzt wird.

## 3.1.3.2. exim (MTA) einrichten

Exim ist wahrscheinlich schon installiert. Mit **eximconfig** wird das Konfigurationsprogramm aufgerufen. Bei der ersten Frage muss Option 1 ausgewählt werden (*Internet site*). Die Antwort auf die nächste Frage (*"visible mail name of your system"*) lautet *gm.rt.schule-bw.de*. Dann noch vier mal *"none"* also Antwort verwenden und die grobe Konfiguration ist abgeschlossen.

Jetzt kommt die feinere Konfiguration in der Datei `/etc/exim/exim.conf`. Um uucp mit exim verwenden zu können, müssen im Abschnitt *Transports* folgende Zeilen hinzugefügt werden:

```
#########################################################################
#                      TRANSPORTS CONFIGURATION                         #
#########################################################################
#                       ORDER DOES NOT MATTER                           #
#     Only one appropriate transport is called for each delivery.       #
#########################################################################
uucp_pipe:
  driver = pipe
  command = "uux - -a$sender_address -r $host\!rmail ($local_part@$domain)"
  pipe_as_creator
  restrict_to_path
  path = "/usr/bin:/bin"
  return_output
```

Im Abschnitt *Router* müssen folgende Zeilen am Anfang des Abschnitts hinzugefügt werden (die Reihenfolge spielt eine Rolle):

```
#########################################################################
#                      ROUTERS CONFIGURATION                            #
#               Specifies how remote addresses are handled              #
#########################################################################
#                          ORDER DOES MATTER                            #
#  A remote address is passed to each in turn until it is accepted.     #
#########################################################################
uucp:
  driver = domainlist
  transport = uucp_pipe
  route_list = "* belwue byname"
```

Damit wird die ausgehende Mail zuerst über uucp verschickt.

Im ersten Abschnitt (*main configuration settings*) muss die Variable *"trusted_users"* gesucht werden. Folgendes muss geändert werden:

```
trusted_users = mail:uucp
trusted_groups = uucp
```

Im selben Abschnitt müssen noch zwei Optionen gesucht und geändert bzw. überprüft werden:

```
host_accept_relay = 127.0.0.1
never_users = root
```

Damit können lokale Prozesse, die den Mailserver über die Adresse 127.0.0.1 ansprechen, Nachrichten mit beliebigem Absender und Empfänger verschicken. Die zweite Option verhindert, dass dem Benutzer *root* (also dem Systemadministrator) direkt Mails zugestellt werden. Die Mails an root müssen an einen anderen Benutzer weitergeleitet werden. Das passiert weiter unten.

Damit man jetzt aber auch von allen Computern im Netzwerk aus Mails verschicken kann, wird noch eine Authentifikation eingerichtet. Diese Konfiguration verhindert, dass mögliche Viren sich nicht selber über den Server weiterverschicken können. Dazu muss im Abschnitt *MAIN CONFIGURATION SETTINGS* die Option

```
exim_group = shadow
```

eingestellt werden. Damit läuft der Mail-Server unter der Gruppe "shadow", sodass er die Datei `/etc/shadow`, in der die Passwörter der Benutzer stehen, lesen darf. Zusätzlich muss im selben Abschnitt folgende Option gesetzt werden:

```
host_auth_accept_relay = 192.168.0.0/24
```

Damit wird es den Rechner aus dem Netzwerk erlaubt, den Mail-Server als Mail-Relay zu benutzen, nachdem sie sich authentifiziert haben. Die Authentifikation wird in der Konfigurationsdatei ganz am Ende eingerichtet, unter dem Abschnitt *AUTHENTICATION CONFIGURATION*. Hier müssen folgende Zeilen stehen:

```
plain:
   driver = plaintext
   public_name = PLAIN
   server_condition = "${if crypteq{$3}{${extract{1}{:}{${lookup{$2} \
                       lsearch{/etc/shadow}{$value}{*:*}}}}}{1}{0}}"
   server_set_id = $2
```

Die Einstellungen sind sofort aktiv; exim muss nicht neugestartet werden. Damit man jetzt Mails verschicken kann, müssen die Clients entsprechend konfiguriert sein ("Benutzername und Passwort verwenden").

Zur Mailkonfiguration gibt es noch eine zweite, sehr wichtige Datei: `/etc/aliases`. Hier können z.B. Weiterleitungen definiert werden. **eximconfig** hat schon eine neue Datei erstellt. Am Ende dieser Datei sollten folgende Zeilen hinzugefügt werden:

```
info:               verwaltung, homepage
homepage:           thomas, alexander, jmueller, bohn, andreas
verwaltung:         root
sekretariat:        poststelle@xxxxxxxx.schule.bwl.de, \sekretariat
proxymaster:        root
root:               andreas

# Anmerkung: Hier könnten noch Weiterleitungen von Mail an externe Adressen
# eingerichtet werden, zum Beispiel für Lehrer

andreas: a.dangel@gmx.de, \andreas
```

Die letzte Zeile ist ein Beispiel für eine externe Weiterleitung.

## 3.1.3.3. Regelmäßig Mails abholen

Soll der Server regelmäßig neue Mails abholen, muss der *cron*-Dienst entsprechend konfiguriert werden. In der Datei `/etc/cron.d/getEmail` steht eine neue cron-Konfiguration. Sie muss folgenden Inhalt haben:

```
#
# /etc/cron.d/getEmail
#

# min hour(s)        day month dow user  command
0 7,10,13,15,17,19,21 *    *    *  root  /usr/local/sbin/getEmail
```

Die Datei `/usr/local/sbin/getEmail` sieht so aus:

```
#!/bin/sh
#
# /usr/local/sbin/getEmail
#

# Internetverbindung aufbauen: ping an news.belwue.de
ping -c 5 129.143.4.4 > /dev/null

# Mailqueue abarbeiten
/usr/sbin/exim -q

# neue Mails holen
/usr/sbin/uucico -r1 -sbelwue -f
```

Mit **chmod 700 /usr/local/sbin/getEmail** wird das Script ausführbar gemacht.

Standardmäßig wird die Mail-Warteschlange von exim alle 15 Minuten ausgeführt. Dies ist unnötig und verursacht nur häufige Internet-Einwahlen. Die Mail-Queue wird im Script `/usr/local/sbin/getEmail` auch ausgeführt (**exim -q**). Wir deaktivieren deshalb die Standardeinstellung. Dazu wird der Eintrag in der vierten Zeile in `/etc/cron.d/exim` mit einem "#" auskommentiert.

# 3.2. Fernwartung

## 3.2.1. SSH-Server

Wenn ein *ssh*-Server installiert ist, kann man sich über das Netzwerk auf den Server einloggen. Dabeiersetzt ssh den alten telnet-Dienst; ssh verwendet im Gegensatz zu telnet eine Verschlüsselung. ssh-Clients gibt es für Linux und für Windows. Ein guter Client für Windows heißt "*putty*".

Mit **apt-get install ssh** wird der Server installiert. In `/etc/ssh/` liegen die Konfigurationsdateien. Die Datei `sshd_config` ist für den ssh-Server zuständig; die wichtigste Option lautet *PermitRootLogin*; sie sollte auf "*no*" gestellt sein. Damit kann sich zwar root nicht mehr direkt einloggen (was erstens ein Sicherheitsrisiko ist und zweitens nicht nötig ist), aber als normaler Benutzer wird man mit **su** zum root (natürlich nach Eingabe des root-Passwortes).

Um weiter an Sicherheit zu gewinnen (der ssh-Zugang ist, wenn der Server online ist, im ganzen Internet verfügbar...), sollte noch der Standard-Port geändert werden. Das geschieht in der Datei `/etc/ssh/sshd_config` mit der Option *Port*. Der Standard-Port ist 22, wir verwenden aber ab jetzt den Port *1222*.

Die Konfiguration wird mit **/etc/init.d/ssh reload** übernommen. Mit **ssh -p 1222 user@server** kann man sich nun über das Netzwerk auf den Server einloggen.

## 3.2.2. dyndns

Bei jeder Einwahl über den T-DSL-Anschluss von T-Online bekommt der Server eine andere IP-Adresse, über die er im Internet erreichbar ist. Damit aber trotzdem Fernwartung über das Internet möglich ist, muss die IP-Adresse bekannt sein. Der kostenlose *dyndns*-Dienst stellt einen DNS-Eintrag zur Verfügung, der bei jeder Einwahl aktualisiert wird. Somit ist der Server dann immer über den gleichen Hostnamen erreichbar.

Mit **apt-get install ddclient** wird die dazu notwendige Software installiert. Die Konfigurationsdatei heißt `/etc/ddclient.conf` und sollte folgenden Inhalt haben:

```
#
# /etc/ddclient.conf
#
pid=/var/run/ddclient.pid
protocol=dyndns2
wildcard=yes
use=if, if=ppp0
server=members.dyndns.org
```

```
login=loginname
password=passwort
host.dyndns.org
```

*"loginname"*, *"passwort"* und *"host.dyndns.org"* müssen natürlich noch angepasst werden.

Damit die IP-Adresse bei jeder Einwahl auch aktualisiert wird, muss noch `/etc/default/ddclient` angepasst werden. Die zwei entscheidenden Einträge lauten:

```
run_ipup="true"
run_daemon="false"
```

Jetzt muss noch dafür gesorgt werden, dass der Server regelmäßig online geht. Zum Beispiel kann der Server regelmäßig neue Mails herunterladen. Wie das Einzurichten ist, steht in Abschnitt 3.1.3.3, „Regelmäßig Mails abholen“.

# 3.3. Dienste für Clients

## 3.3.1. Webserver

Der Webserver ist quasi die komfortable Bedienoberfläche für das Schulnetz. Über diesen Server werden einzelne CGI-Scripte ausgeführt, mit denen man zum Beispiel einen neuen Benutzer anlegen kann. Ansonsten lässt sich die Homepage der Schule anschauen und jeder Benutzer kann im Ordner `public_html/` in seinem Homeverzeichnis seine eigenen Seiten im Schulnetz veröffentlichen.

Als Webserver wird **apache** verwendet. Falls apache noch nicht installiert ist, lässt sich dies mit **apt-get install apache** nachholen. Die Konfigurationsdatei heißt `/etc/apache/httpd.conf`. Diese muss an einigen stellen angepasst werden. Der folgende Ausschnitt aus der Konfigurationsdatei soll einen Anhaltspunkt geben, was gegenüber der Standardkonfigurationsdatei von Debian verändert werden muss:

```
#
# /etc/apache/httpd.conf
#

### Section 1: Global Environment
# Keine Änderungen nötig

### Section 2: 'Main' server configuration
ServerAdmin webmaster@gm.rt.schule-bw.de
ServerName server.gm.rt.schule-bw.de
DocumentRoot "/usr/local/httpd/htdocs"

<Directory />
    AuthUserFile   /etc/apache/passwd
    AuthGroupFile  /etc/apache/group
    Order allow,deny
    Options None
    AllowOverride None
</Directory>

#

# This should be changed to whatever you set DocumentRoot to.
#
<Directory /usr/local/httpd/htdocs/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# Einzelne Verzeichnisse
<Directory /usr/local/httpd/htdocs/admin/>
    AllowOverride None
```

```
    Options ExecCGI FollowSymLinks Indexes
    DirectoryIndex index.html index.cgi

    AuthType Basic
    AuthName "Systemverwaltung"
    AuthUserFile /etc/apache/passwd
    AuthGroupFile /etc/apache/group
    require group admin
</Directory>

<Directory /usr/local/httpd/htdocs/internet/>
    AllowOverride None
    Options ExecCGI FollowSymLinks
    DirectoryIndex index.html index.cgi

    AuthType Basic
    AuthName "Internet-Einwahl"
    AuthUserFile /etc/apache/passwd
    AuthGroupFile /etc/apache/group
    require group internet
</Directory>

<Directory /usr/local/httpd/htdocs/organisation/>
    AllowOverride None
    Options Indexes
    DirectoryIndex index.html index.htm

    AuthType Basic
    AuthName "Organisation"
    AuthUserFile /etc/apache/passwd
    AuthGroupFile /etc/apache/group
    require group lehrer
</Directory>

<Directory /usr/local/httpd/htdocs/passwd/>
    AllowOverride None
    Options ExecCGI FollowSymLinks
    DirectoryIndex index.html index.cgi

    AuthType Basic
    AuthName "Passwortaenderung"
    AuthUserFile /etc/apache/passwd
    AuthGroupFile /etc/apache/group
    require group users
</Directory>

<Directory /usr/local/httpd/htdocs/mail-config/>
    AllowOverride None
    Options ExecCGI FollowSymLinks
    DirectoryIndex index.html index.cgi

    AuthType Basic
    AuthName "Mail-Weiterleitung"
    AuthUserFile /etc/apache/passwd
    AuthGroupFile /etc/apache/group
    require group users
</Directory>

<Directory /usr/local/samba/homepage/>
    AllowOverride None
    Options Indexes FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>

ScriptAlias /cgi-bin/ /usr/local/httpd/cgi-bin/

<Directory /usr/lib/cgi-bin/>
    AllowOverride None
```

```
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_mime.c>
    AddHandler cgi-script .cgi .sh .pl
</IfModule>

<IfModule mod_userdir.c>
    UserDir /home/public_html
</IfModule>

<Directory /home/public_html/*/>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
```

Es wurden für einige Verzeichnisse die Zugriffsrechte gesetzt. Oft muss man sich erst anmelden (*AuthType Basic*), wenn man so ein Verzeichnis betreten will.

Mit **/etc/init.d/apache restart** wird die neue Konfiguration übernommen. Jetzt wird noch ein Script benötigt, das die Dateien `/etc/apache/passwd` und `/etc/apache/group` erzeugt. Diese sind für eine Anmeldung notwendig. Das Script mit dem Dateinamen `/usr/local/sbin/generate_apache_auth` ist im Anhang zu finden.

Dieses Script wird mit **chmod 700 /usr/local/sbin/generate_apache_auth** ausführbar gemacht. Jedesmal, wenn ein neuer Benutzer angelegt wird, wenn ein Benutzer gelöscht wird oder wenn ein Passwort geändert wurde, muss dieses Script ausgeführt werden.

# 3.3.2. pop3-Server

Über einen POP3-Server können sich die Benutzer ihre Mail komfortabel über einen Mailclient (z.B. Mozilla-Mail, Outlook Express) abholen. Der POP3-Server verwendet die Mailboxen der lokalen Benutzer auf dem Server. Mit **apt-get install qpopper** wird ein einfacher POP3-Server installiert. Die Standardkonfiguration genügt; dieser Server muss nicht konfiguriert werden. Falls doch: Die Konfigurationsdatei heißt `/etc/qpopper.conf`.

# 3.3.3. Proxy und Filter

## 3.3.3.1. squid

Über den Proxy-Server squid erhalten die Clients Zugang zum Internet. Alle Rechner im Schulnetz greifen also nicht direkt auf das Internet zu, sondern nur auf den Proxy-Server, der dann die gewünschten Daten liefert. Zuerst muss die Software installiert werden: **apt-get install squid**. Die Konfigurationsdatei heißt `/etc/squid.conf`. In der Datei müssen einige Einstellungen angepasst werden. Hier sind die wichtigen Einstellungen:

```
#
# /etc/squid.conf
#

ftp_user webmaster@gm.rt.schule-bw.de

# ACCESS CONTROLS
# ---------------
```

```
[...]

# INSERT YOUR OWN RULE(S) HERE...
acl schulnetz src 192.168.0.0/255.255.255.0
acl server dst 192.168.0.1
http_access allow schulnetz

append_domain .gm.rt.schule-bw.de

always_direct allow server
```

Mit **/etc/init.d/squid restart** wird die Konfiguration übernommen.

## 3.3.3.2. squidGuard

Der Filter, der als Jugendschutzfilter eingerichtet werden kann, sperrt bestimmte Internetadressen. Aktuelle Listen erhält man unter http://www.squidguard.org/blacklist/.

Die Filtersoftware heißt *squidguard* und wird mit **apt-get install squidguard** installiert. Damit der Filter im Proxy integriert wird, muss `/etc/squid.conf` bearbeitet werden. Es müssen folgende zwei Einträge existieren:

```
#
# /etc/squid.conf
#
[...]

redirect_program /usr/bin/squidGuard
redirect_children 5
```

Dann müssen neue Blacklisten unter der oben genannten Adresse heruntergeladen werden. Mit **cd / var/lib/squidguard/db; tar xfzv ~/blacklists.tar.gz** wird das Archiv ins richtige Verzeichnis entpackt. Danach muss noch **chmod -R 777 /var/lib/squidguard/db/blacklists** ausgeführt werden. Die Konfigurationsdatei heißt `/etc/squid/squidGuard.conf`. Sie muss folgenden Inhalt haben:

```
#
# /etc/squid/squidGuard.conf
#
logdir /var/log/squid
dbhome /var/lib/squidguard/db

src localnet {
    ip 127.0.0.0/8
    ip 192.168.0.0/24
}

dest ads {
    domainlist blacklists/ads/domains
    urllist blacklists/ads/urls
}

dest aggressive {
    domainlist blacklists/aggressive/domains
    urllist blacklists/aggressive/urls
}

dest audio-video
{
    domainlist blacklists/audio-video/domains
    urllist blacklists/audio-video/urls
}

dest drugs {
    domainlist blacklists/drugs/domains
    urllist blacklists/drugs/urls
```

```
}

dest gambling {
    domainlist blacklists/gambling/domains
    urllist blacklists/gambling/urls
}

dest hacking {
    domainlist blacklists/hacking/domains
    urllist blacklists/hacking/urls
}

dest mail {
    domainlist blacklists/mail/domains
}

dest porn
{
    domainlist blacklists/porn/domains
    urllist blacklists/porn/urls
    #expressionlist blacklists/porn/expressions
}

dest proxy {
    domainlist blacklists/proxy/domains
    urllist blacklists/proxy/urls
}

dest violence {
    domainlist blacklists/violence/domains
    urllist blacklists/violence/urls
    #expressionlist blacklists/violence/expressions
}

dest warez
{
    domainlist blacklists/warez/domains
    urllist blacklists/warez/urls
}

acl {
    localnet {
        pass !aggressive !porn !violence all
        redirect http://server/cgi-bin/blocked.cgi?clientaddr=%a&url=%u
    }

    default {
        pass none
        redirect http://server/cgi-bin/blocked.cgi?clientaddr=%a&url=%u
    }
}
```

Damit werden die Adresslisten *"aggressive"*, *"porn"* und *"violence"* abgeblockt. Wer eine solche Seite aufruft, bekommt das Script `blocked.cgi` zu sehen. Dieses Script gibt eine Fehlermeldung aus oder, wenn der Filter eine Bilddatei blockiert (z.B. Werbung), dann wird `/usr/local/httpd/htdocs/blocked.gif` angezeigt. Das Script `blocked.cgi` muss noch im Verzeichnis `/usr/local/httpd/cgi-bin/` erstellt werden:

```
#!/usr/bin/perl
#
# /usr/local/httpd/cgi-bin/blocked.cgi
#
use CGI;
$q = new CGI;
$clientaddr = "";
$url="";
```

```
$clientaddr = $q->param('clientaddr') if defined($q->param('clientaddr'));
$url = $q->param('url') if defined($q->param('url'));

if ($url =~ /\.(gif|jpg|jpeg|mpg|mpeg|avi|mov)$/i) {
    print "Content-type: image/gif\n";
    ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = gmtime($time);
    printf "Expires: %s, %02d-%s-%02d %02d:%02d:%02d GMT\n\n", $day[$wday],
                                $mday,$month[$mon],$year,$hour,$min,$sec;
    open(GIF, "$ENV{'DOCUMENT_ROOT'}/blocked.gif");
    while (<GIF>) {
        print;
    }
    close(GIF);
} else {
    print "Content-type: text/html\n";
    printf "Expires: %s, %02d-%s-%02d %02d:%02d:%02d GMT\n\n", $day[$wday],
                                $mday,$month[$mon],$year,$hour,$min,$sec;
    print <<EOF;
<html>
<head>
    <title>403 Forbidden - Zugriff verweigert</title>
</head>
<body>
<center>
<h1>403 Forbidden - Zugriff verweigert</h1>
Der Zugriff wurde Ihnen aus einem der folgenden Gründe verweigert:
<table border="0">
<tr><td>
<ul>
    <li>Ihre IP-Adresse ist nicht aus dem Bereich des Schulnetzes</li>
    <li>Der Jugendschutzfilter ist aktiviert. Die Filterkategorien sind:
    <ul>
        <li>aggressive</li>
        <li>porn</li>
        <li>violence</li>
    </ul>
</ul>
</td></tr>
</table>
Wenn diese Seite zu Unrecht gesperrt wurde, schreiben Sie eine Mail an<br>
<a href="mailto:proxymaster\@gm.rt.schule-bw.de">
proxymaster\@gm.rt.schule-bw.de</a>
</center>
</body>
</html>
EOF
}
exit 0;
```

Mit **chmod 755 /usr/local/httpd/cgi-bin/blocked.cgi** wird das Script ausführbar gemacht. Damit *squidGuard* später schneller startet, wird **squidGuard -C all** einmal ausgeführt. Dabei werden die Adresslisten in ein Datenbankformat konvertiert. Mit **/etc/init.d/squid restart** wird der Filter letztendlich aktiviert.

## 3.3.4. Drucker

Damit der Netzwerkdrucker später über Samba den Windows-Clients zur Verfügung gestellt werden kann, muss der Netzwerkdrucker (der übrigens auch lokal am Druckerport des Servers angeschlossen sein könnte) zunächst auf dem Server eingerichtet werden. Dazu wird das "Common UNIX Printing System" benötigt, das mit **apt-get install cupsys cupsys-client** noch nachinstalliert werden muss. Konfiguriert wird das Druck-System über einen Webbrowser. Als Text-Browser eignet sich lynx. Möglicherweise muss dieser mit **apt-get install lynx** nachinstalliert werden. Mit **/etc/init.d/cupsys start** wird das Druck-System gestartet.

Mit **lynx http://localhost:631/** wird der Webbrowser zur Konfiguration gestartet. Hier muss zunächst *Manage Printers* ausgewählt werden, danach *Add Printer*. Der Browser fragt nun nach einem Benutzernamen. Einfach *"root"* und das *root-Passwort* verwenden. Auf der nächsten Seite muss der Freigabe-Name eingegeben werden, also z.B. *"netzdrucker"*. *Location* und *Description* sind weniger wichtig. Mit *Continue* geht's weiter. Auf der

folgenden Seite muss jetzt als *Device "AppSocket/HP JetDirect"* ausgewählt werden. Im Feld *Device URI* auf der nächsten Seite muss *"socket://192.168.0.254:9100"* eingetragen werden. *Model/Driver* ist *"Raw"* und *"Raw Queue (en)"*. Das bedeutet, dass die Druckdaten genau so an den Drucker weitergeleitet werden, wie der Windows-Client sie verschickt. Unter Windows muss also noch ein passender Druckertreiber installiert werden. Das war's soweit. Der Rest muss unter Samba konfiguriert werden.

## 3.3.5. Samba

Samba stellt den Windows-Clients Speicherplatz zur Verfügung: Ein Homeverzeichnis, ein Verzeichnis für temporäre Daten, ein öffentliches Verzeichnis und ein Verzeichnis für Windows-Programme. Für bestimmte Benutzergruppen gibt es weitere Verzeichnisse: Homepage und Admin beispielsweise. Außerdem gibt es noch ein sogenannten "netlogon-script", das beim Anmelden bei Samba auf dem Client gestartet wird und die Verzeichnisse automatisch einbindet. Zusätzlich gibt Samba noch den Drucker frei, der im vorigen Abschnitt eingerichtet wurde.

Zunächst muss samba installiert werden; das geht mit **apt-get install samba**. Die Konfigurationsdatei heißt `/etc/samba/smb.conf`. Sie hat folgenden Inhalt:

```
#
# /etc/samba/smb.conf
#
[global]
    workgroup = WORKGROUP
    netbios name = server
    server string = Samba Server
    comment = Samba Server

    hosts allow = 192.168.0.

    load printers = yes
    printcap name = cups
    printing = CUPS

    guest account = nobody

    #log file = /var/log/samba/log.%m # für jede Maschine eine eigene Logfile
    log file = /var/log/samba/log.smb
    log level = 1
    max log size = 1000 # in KB
    syslog = 0

    security = user

    encrypt passwords = yes
    smb passwd file = /usr/local/samba/private/smbpasswd
    passwd program = /usr/bin/passwd %u
    unix password sync = yes

    socket options = TCP_NODELAY
    keepalive = 30

    interfaces = eth0

    local master = yes
    os level = 65
    domain master = yes
    preferred master = yes
    domain logons = yes

    logon script = logon.bat
    # für Roaming Profiles (WinNT)
    #logon path = \\SERVER\%U\profile
    # für Roaming Profiles (Win9X)
    #logon home = \\SERVER\%U\profile

    wins support = yes

    # alles in eine Zeile!
```

```
    invalid users = root daemon bin sys sync games man lp mail news uucp proxy
       postgres www-data backup operator list irc gnats identd sshd gdm telnetd
       ftp nut faxmaster partimag mysql homepage internet
    valid users = +users nobody
    admin users = +admin

[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
    public = yes
    guest ok = yes
    writeable = no
    printable = yes

[netlogon]
    path = /usr/local/samba/netlogon
    write list = +admin
    guest ok = yes

[homes]
    comment = Heimatverzeichnis von %U
    # wegen public_html: www-data
    force group = www-data
    read only = no
    create mask = 0755
    browseable = no

[tmp]
    comment = Temporaere Dateien
    path = /usr/local/samba/temp
    force group = users
    read only = no
    create mask = 0664
    directory mask = 0775
    guest ok = yes

[pub]
    comment = Oeffentliches Verzeichnis
    path = /usr/local/samba/pub
    write list = +lehrer
    force group = lehrer
    create mask = 0664
    force create mode = 0664
    directory mask = 0775
    force directory mode = 0775
    guest ok = yes

[homepage]
    comment = Gymnasium-Muensingen Homepage
    path = /usr/local/samba/homepage
    write list = +homepage
    force group = homepage
    create mask = 0664
    force create mode = 0664
    directory mask = 0775
    force directory mode = 0775

[programme]
    comment = Programme fuer Windows
    path = /usr/local/samba/programme
    write list = +admin
    force group = admin
    create mask = 0664
    force create mode = 0664
    directory mask = 0775
    force directory mode = 0775
    guest ok = yes
```

```
[programdata]
    comment = Programm-Daten
    path = /usr/local/samba/programdata
    # jeder darf schreiben (Windows-Programme brauchen das ;-)
    write list = +users nobody
    force group = users
    create mask = 0664
    force create mode = 0664
    directory mask = 0775
    force directory mode = 0775
    guest ok = yes

[admin]
    comment = Admin-Verzeichnis
    path = /usr/local/samba/admin
    valid users = +admin
    force group = admin
    read only = no
    create mask = 0660
    force create mode = 0660
    directory mask = 0770
    force directory mode = 0770
```

Die angegebenen Verzeichnisse müssen natürlich existieren (alles unter `/usr/local/samba/`). Außerdem müssen für diese Verzeichnisse die passenden Zugriffsrechte gesetzt sein. Mit **/etc/init.d/samba restart** werden die Einstellungen übernommen.

Für die Druckfunktion muss das Verzeichnis `/var/spool/samba` existieren. Außerdem muss es den Mitglieder der Gruppe *"users"* erlaubt sein, in dieses Verzeichnis zu schreiben (sonst dürfen diese Benutzer nicht drucken). Dies ist zu erreichen mit den folgenden zwei Befehlen:

```
chgrp users /var/spool/samba
chmod g+wx /var/spool/samba
```

Hier ist das Netlogon-Script. Es verbindet einige Netzlaufwerke, sodass diese im Explorer zur Verfügung stehen. Außerdem wird die Uhrzeit des Clients nach der Server-Uhr gestellt. Damit es keine Probleme mit Mozilla-Profilen gibt, wird eine Datei bei jedem Einloggen auf den Client kopiert; in dieser Datei steht, dass das Mozilla-Profil unter "H:\Mozilla" zu finden ist.

Das Script heißt `logon.bat` und muss in der *netlogon*-Freigabe liegen.

```
@echo off
net use h: \\server\homes /yes
net use i: \\server\homepage /yes
net use v: \\server\pub /yes
net use t: \\server\tmp /yes
net use p: \\server\progs /yes
net time \\server /set /yes

rem Mozilla-Profil-Konfiguration:
copy \\server\netlogon\registry.dat C:\WINDOWS\Anwendungsdaten\Mozilla /Y
```

## 3.3.5.1. Logdateien

Samba benützt drei Logdateien: In `/var/log/samba/log.nmbd` protokolliert der nmbd-Dämon seine Zugriffe. In `/var/log/samba/log.smbd` steht, wann der smbd-Dämon gestartet wurde. Und in `/var/log/samba/log.smb` steht, wer auf irgendwelche Freigaben zugegriffen hat. Diese Logdateien sollen auch mit *logrotate* rotiert werden. Damit dies zuverlässig klappt, wird vor dem Rotieren der Samba-Dämon gestoppt und danach wieder gestartet.

Die Datei `/etc/logrotate.d/samba` muss folgenden Inhalt haben:

```
/var/log/samba/log.smb {
```

```
    weekly
    missingok
    rotate 7

    prerotate
        /etc/init.d/samba stop || true
        killall -q smbd || true
    endscript

    compress
    notifempty
}

/var/log/samba/log.smbd {
    weekly
    missingok
    rotate 7

    prerotate
        /etc/init.d/samba stop || true
        killall -q smbd || true
    endscript

    compress
    notifempty
}

/var/log/samba/log.nmbd {
    weekly
    missingok
    rotate 7

    prerotate
        /etc/init.d/samba stop || true
        killall -q nmbd || true
    endscript

    postrotate
       /etc/init.d/samba start
    endscript

    compress
    notifempty
}
```

## 3.3.5.2. Tägliches Neustarten

Da es offenbar öfters vorkommt, dass der Samba-Server, nachdem er ein paar Tage ununterbrochen lief, nicht mehr reagierte, wird der Samba-Server nun mittels eines Cron-Jobs täglich neugestartet. Dazu wird im Verzeichnis `/etc/cron.daily/` die Datei `samba-restart` erstellt:

```
#!/bin/sh
/etc/init.d/samba stop
sleep 4
/etc/init.d/samba start
sleep 4

# überprüfen, ob nmbd läuft
ps `cat /var/run/samba/nmbd.pid` > /dev/null
if [ $? -eq 1 ]; then
  # nmbd läuft nicht...
  /etc/init.d/samba stop
  sleep 4
  /etc/init.d/samba start
fi
```

Die Datei muss noch mit **chmod 755 /etc/cron.daily/samba-restart** ausführbar gemacht werden.

# 3.4. Sicherheit

Ein Computer ist umso sicherer, je weniger Dienste auf ihm laufen. Dann hat er eine geringere Angriffsfläche. Am besten ist es, wenn die nicht benötigten Dienste ganz deaktiviert werden und man sich nicht komplett auf die Firewall verlässt. Außerdem nützt die Firewall gar nichts, wenn der verwendete Server eine Sicherheitslücke hat. Daher sollte in regelmäßigen Abständen die Software upgedatet werden.

## 3.4.1. Firewall

Die Firewall, die auf dem Server eingerichtet werden soll, schützt den Server zum einen vor dem Internet und zum anderen vor dem lokalen Netzwerk. Immer wenn der Server online ist, ist er auch im Internet sichtbar - und daher auch angreifbar. Die Firewall soll so konfiguriert sein, dass sie nur die unbedingt benötigten Dienste erlaubt. Alles andere wird blockiert.

Das Firewall-Script wird hier nicht abgedruckt. Es ist aber im Anhang zu finden. Zunächst einmal wird das Script nach `/usr/local/sbin/firewall` kopiert. `/etc/init.d/firewall` ist ein symbolischer Link darauf. Und `/etc/rc[2345].d/S10firewall` sind ebenfalls symbolische Links, aber relative mit `../init.d/firewall` als Ziel.

Jetzt ist das Firewall-Script fertig installiert. Beim Systemstart wird es automatisch geladen. Manuell kann dies mit **/etc/init.d/firewall start** nachgeholt werden. Die Firewall kann mit **/etc/init.d/firewall stop** komplett deaktiviert werden.

Die Firewall besteht unter Linux aus mindestens zwei Teilen: INPUT und OUTPUT. Ein ankommendes Paket kommt immer nach INPUT, ein vom Server generiertes und ihn verlassendes Paket durchläuft immer die Firewallregeln in OUTPUT. Für beide Teile kann eine Standardreaktion eingestellt werden, wenn keine Firewallregel zutrifft. Bei INPUT werden alle Pakete verworfen und bei OUTPUT wird alles erlaubt.

Die Firewall startet mit sich gleich den Proxy-Manager (`/usr/local/sbin/proxy-manager`). Der Proxy-Manager verwaltet die Internet-Freigabe und schreibt in die Datei `/var/state/internet-freigabe`, welche Computer bis zu welchem Zeitpunkt freigeschaltet sind. Das selbstgeschriebene squid-redirector-Script (`/usr/local/sbin/squid-redirector`) benutzt diese Information und leitet entweder zur richtigen Internet-Adresse weiter (genauer: zu squidGuard) oder zeigt eine Fehlermeldung an. Siehe auch Abschnitt 4.3, „proxymanager“.

## 3.4.2. inetd

Über den speziellen Server *"inetd"* werden bei Bedarf weitere Server gestartet. Da viele Dienste gar nicht benötigt werden, sollten sie ganz deaktiviert werden. Die Konfigurationsdatei zu "inetd" heißt `/etc/inetd.conf`. Mit einem "#" am Anfang der Zeile werden bestehende Einträge auskommentiert. Mindestens zwei Einträge sollten aber belassen werden: der *smtp*-Dienst und der *pop3*-Dienst. Ohne diese Dienste können die Clients keine Mails mehr verschicken und empfangen. Eine minimale Konfigurationsdatei sieht daher so aus:

```
#
# /etc/inetd.conf
#

smtp   stream  tcp  nowait  mail  /usr/sbin/exim   exim -bs
pop-3  stream  tcp  nowait  root  /usr/sbin/tcpd
                                  /usr/sbin/in.qpopper -f /etc/qpopper.conf
```

Mit **/etc/init.d/inetd restart** wird die Konfiguration übernommen.

## 3.4.3. netstat und nmap

**netstat** und **nmap** sind zwei Tools, mit denen die Sicherheit ein wenig ausgelotet werden kann. nmap muss gegebenenfalls mit **apt-get install nmap** nachinstalliert werden.

### 3.4.3.1. netstat

**netstat** zeigt aktuelle Netzwerkverbindungen und geöffnete Ports an. Dazu muss **netstat -tuanp** aufgerufen werden. "-t" steht dabei für das Protokoll TCP, "-u" für UDP, "-a" zeigt alle Einträge an (nicht nur bestehende Verbindungen) und "-n" zeigt alle Ports und IP-Adressen numerisch an. "-p" zeigt zusätzlich - wenn vorhanden - das Server-Programm an.

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address    Foreign Address  State   PID/Program name
tcp        0      0 0.0.0.0:37       0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:9        0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:139      0.0.0.0:*        LISTEN  213/smbd
tcp        0      0 0.0.0.0:13       0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:110      0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:79       0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:111      0.0.0.0:*        LISTEN  113/portmap
tcp        0      0 0.0.0.0:80       0.0.0.0:*        LISTEN  240/apache
tcp        0      0 0.0.0.0:113      0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:21       0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:631      0.0.0.0:*        LISTEN  206/cupsd
tcp        0      0 0.0.0.0:23       0.0.0.0:*        LISTEN  201/inetd
tcp        0      0 0.0.0.0:25       0.0.0.0:*        LISTEN  201/inetd
udp        0      0 0.0.0.0:517      0.0.0.0:*                201/inetd
udp        0      0 0.0.0.0:518      0.0.0.0:*                201/inetd
udp        0      0 192.168.0.3:137  0.0.0.0:*                208/nmbd
udp        0      0 0.0.0.0:137      0.0.0.0:*                208/nmbd
udp        0      0 0.0.0.0:9        0.0.0.0:*                201/inetd
udp        0      0 192.168.0.3:138  0.0.0.0:*                208/nmbd
udp        0      0 0.0.0.0:138      0.0.0.0:*                208/nmbd
udp        0      0 0.0.0.0:111      0.0.0.0:*                113/portmap
```

Bei *TCP* steht unter State *"LISTEN"*, das heißt, dass hier ein Dienst auf neue Verbindungen wartet.

### 3.4.3.2. nmap

**nmap** ist ein sogenannter Portscanner. Er kann zum Beispiel dazu benutzt werden, eine Firewall zu testen. nmap zeigt dabei dann alle offene Ports an. Am besten wird nmap von verschiedenen Rechnern im Intranet und Internet ausgeführt. Ein Beispielaufruf:

```
server:~# nmap 192.168.0.1

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on server.adabolo.homelinux.net (192.168.0.1):
(The 1545 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
37/tcp     open        time
53/tcp     open        domain
80/tcp     open        http
139/tcp    open        netbios-ssn
444/tcp    open        snpp
4557/tcp   open        fax
4559/tcp   open        hylafax

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Auf den angezeigten Ports ist eine Verbindung möglich. Das heißt, dass dieser Dienst erreichbar ist.

## 3.4.4. Backup

Um die Datensicherheit des Servers zu verbessern (wenn Software-RAID eingerichtet wurde, besteht schon eine Sicherheit), soll wöchentlich ein Backup der Home-Verzeichnisse (alles unterhalb von `/home/`) erstellt werden. Dieses Backup kann dann auf ein externes Medium gesichert werden (z.B. DVD).

Zuerst muss das Script `/usr/local/sbin/userdir-backup.sh` erstellt werden:

```
#!/bin/sh
```

```
# Liste der alten Backup-Dateien erstellen...
filelist=`ls /usr/local/samba/admin/userdir-backup/userdir-backup-*`

tmpfile= `mktemp /tmp/backup-filelist.XXXXXX`

echo 'Backup home-directories (/home) ...'
echo
tar cv /home 2> $tmpfile | bzip2 | split -b 2047m - \
  /usr/local/samba/admin/userdir-backup/userdir-backup-`date +%Y%m%d`.tar.bz2.

if [ $? -ne 0 ]; then
  # es ist ein Fehler aufgetreten
  cat $tmpfile
else
  # kein Fehler
  # jetzt die (ganz) alten Backups löschen
  rm /usr/local/samba/admin/userdir-backup/old/*
  # die alten Backups verschieben
  mv $filelist /usr/local/samba/admin/userdir-backup/old/
fi

echo `cat $tmpfile | wc -l` files backup up.
rm $tmpfile

df -h
```

Zunächst muss dieses Script noch mit **chmod 700 /usr/local/sbin/userdir-backup.sh** ausführbar gemacht werden. Wenn dieses Script ausgeführt wird, dann werden unterhalb von `/usr/local/samba/admin/userdir-backup/` (also im Verzeichnis "*userdir-backup*" in der Samba-Freigabe "*admin*") mehrere Dateien erstellt, die eine maximale Größe von 2 GB haben (2047 MB). Die Namen der Dateien sind folgendermaßen aufgebaut: *userdir-backup-%datum.tar.bz2.%endung*. Datum ist z.B. "*20031019*", also zuerst das Jahr, der Monat und dann der Tag. Die Endung wird hochgezählt und beginnt mit "*aa*". Die erste Datei hat also die Endung "*aa*", die zweite "*ab*", die dritte "*ac*" usw.

Um das Backup zurückzusichern müssen diese Dateien zunächst wieder zu einer sehr großen einzelnen Datei zusammengeführt werden. Diese Datei ist aber möglicherweise zu groß, deshalb werden - wie beim Sichern - mehrere Befehle auf einmal angewandt: Mit **cat userdir-backup-%datum.tar.bz2.* | bunzip2 | tar xv** kann das Backup zurückgesichert werden. Diese Befehlskette muss aber im Root-Verzeichnis (`/`) ausgeführt werden oder man muss das entpackte Home-Verzeichnis entsprechend verschieben.

Damit das Backup-Script nun wöchentlich ausgeführt wird, muss ein sogenannter Crontab-Eintrag erstellt werden. Mit folgendem Crontab-Eintrag wird das Script jeden Sonntag (5. Wert: 0) um 04:00 Uhr (2. Wert: 4; 1. Wert: 0) als Benutzer "*root*" ausgeführt.

```
0 4 * * 0 root /usr/local/sbin/userdir-backup.sh
```

Dieser Eintrag wird der Datei `/etc/crontab` als letzte Zeile hinzugefügt. Damit ist das Backup-Script fertig installiert.

# 3.4.5. sudoers

Da Linux ein Multiuser-Betriebssystem ist, haben gewöhnliche Benutzer nur wenige Rechte. Im Gegensatz dazu hat der Superuser, der Administrator oder "root", alle Rechte. So darf beispielsweise nur der Administrator das Passwort eines x-beliebigen Benutzers ändern. Manche CGI-Programme für die Weboberfläche benötigen genau dieses Recht (Passwort ändern oder Passwort zurückstellen). Damit dies trotzdem realisierbar ist, gibt es ein Programm, mit dessen Hilfe man andere Programme als root ausführen kann. Dieses Programm heißt "**sudo**" und muss wahrscheinlich mit **apt-get install sudo** noch nachinstalliert werden.

Alle CGI-Programme laufen unter dem Benutzer *www-data*. Diesem Benutzer soll nun erlaubt werden, einige Programme als root auszuführen. Mit **visudo** wird die Konfigurationsdatei `/etc/sudoers` bearbeitet. Sie muss folgenden Inhalt haben:

```
#
# /etc/sudoers
#
```

```
# Host alias specification
Host_Alias LOCAL = 127.0.0.1, 192.168.0.1

# User alias specification
User_Alias WWW = www-data

# Runas alias specification
Runas_Alias USERS = root, homepage

# Cmnd alias specification
Cmnd_Alias COMMANDS = /usr/sbin/usermod, /usr/local/sbin/generate_apache_auth,\
                      /usr/bin/smbpasswd, /usr/local/sbin/change-aliases,\
                      /usr/local/sbin/proxy-manager, /usr/local/sbin/firewall,\
                      /usr/local/sbin/mirror-hp.sh, /usr/sbin/groupadd,\
                      /usr/sbin/useradd, /bin/chown, /bin/ln, /bin/mkdir,\
                      /usr/sbin/setquota,\
                      /usr/local/sbin/make_mozilla_config.pl, /usr/bin/passwd,\
                      /usr/local/sbin/chpw.pl, /usr/local/sbin/rmusr.pl,\
                      /usr/local/sbin/user-status, \
                      /usr/local/sbin/user-status3, /usr/sbin/userdel, \
                      /usr/sbin/groupdel, /bin/rm, /bin/mv

# User privilege specification
root ALL = (ALL) ALL
WWW LOCAL = (USERS) NOPASSWD: COMMANDS
#www-data ALL=(ALL) NOPASSWD: ALL
```

Der Benutzer *"www-data"* kann jetzt mit **sudo /usr/sbin/useradd neuer_benutzer** zum Beispiel einen neuen Benutzer hinzufügen, was sonst nur root dürfte.

# 3.5. Zusätzliches

## 3.5.1. Homepage hochladen

Zunächst befindet sich die Homepage im Laufwerk *"Homepage"* unter Samba. Und dort genauer im Ordner *"intranet"*. Alles was sich in diesem Ordner befindet, wird hochgeladen. Das Programm zum Hochladen heißt **weex** und wird mit **apt-get install weex** nachinstalliert. Damit der Server die Homepage halbautomatisch hochladen kann (nach dem Passwort wird noch gefragt), wird ein Benutzer mit dem Namen *"homepage"* erstellt. Sein Homeverzeichnis ist `/home/homepage/`. Die Konfigurationsdatei für weex heißt `/home/homepage/.weex/weexrc`. Sie muss mit folgendem Inhalt erstellt werden:

```
#
# /home/homepage/.weex/weexrc
#
[Belwue]
FtpPassive = true
HostName = www.belwue.de
LoginName = loginname
SrcDir = /usr/local/samba/homepage/intranet
DestDir = /htdoc
IgnoreLocalFile = FINDER.DAT
IgnoreLocalDir = RESOURCE.FRK
IgnoreRemoteDir = {
        /htdoc/wusage
        /htdoc/stat
}
IgnoreRemoteFile = /htdoc/homepage.count

[default]
Monochrome = True
```

"*loginname*" muss natürlich noch ersetzt werden.

Der Superuser (root) kann mit **su -c '/usr/bin/weex Belwue' - homepage** die Homepage updaten. Dabei wird er

nach dem FTP-Passwort gefragt.

## 3.5.2. Quota

Wenn Quota aktiviert ist, dann kann man den freien Speicherplatz für jeden Benutzer einzeln begrenzen. Dabei protokolliert das Quota-System in einer Datei mit, wieviel Plattenplatz ein beliebiger Benutzer verbraucht.

Um Quota einzurichten, müssen zunächst zwei Programmpakete installiert werden: **apt-get install quota quotatool**. Außerdem muss der Kernel quota unterstützen. Im Standardkernel ist bisher nur Unterstützung für das Dateisystem ext2fs enthalten. Um mit ReiserFS Quota zu benützen, muss der Kernel gepatcht werden. Näheres steht in Abschnitt 2.2, „Kernel machen“.

Sind die Grundvoraussetzungen (Software und Kernel) vorhanden, muss die Datei `/etc/fstab` verändert werden. In der Zeile, die mit "*/dev/md0*" beginnt, muss noch die Mount-Option *usrquota* ergänzt werden. Die Zeile sieht dann so aus:

```
/dev/md0    /       reiserfs    defaults,usrquota   0   0
```

Nach einem Neustart muss mit **init S** in den Single-User-Mode gewechselt werden. Hier kann dann die Quota-Protokoll-Datei mit **quotacheck -avu -F vfsold** erstellt werden. Anschließend kann mit **init 2** wieder in den normalen Runlevel zurückgewechselt werden. Quota ist nun aktiviert.

Mit dem Programm **setquota** kann die Begrenzung des Festplattenplatzes für einen bestimmten Benutzer verändert werden. Die Aufruf-Syntax des Programms ist folgende:

```
setquota -u user <limit> <limit> 0 0 /dev/md0
```

Mit "*limit*" ist die Begrenzung in Kilobytes (1 KB = 1024 Bytes) gemeint.

## 3.5.3. ntpdate

Mit **ntpdate** lässt sich die Server-Uhr nach einer Atomuhr im Internet stellen. Zuerst muss mit **apt-get install ntpdate** das Programm installiert werden. Bei der Installation wird nach einem NTP-Server gefragt. Einfach nichts eingeben.

Am besten ist es, wenn bei jeder Internet-Einwahl die Uhr neu gestellt wird. Dazu wird die Datei `/etc/ppp/ip-up.d/ntpdate` erstellt:

```
#!/bin/sh
#
# /etc/ppp/ip-up.d/ntpdate
#

ntpdate -s ntp1.ptb.de
hwclock -w
```

Die Datei muss mit **chmod 755 /etc/ppp/ip-up.d/ntpdate** noch ausführbar gemacht werden.

## 3.5.4. Transparenter Proxy

Damit mitgebrachte Computer (z.B. Notebooks) einfach das Internet über den Server nutzen können, wird ein sogenannter Transparenter Proxy eingerichtet. Dazu braucht der Client nur über DHCP konfiguriert werden. Im Browser muss kein Proxy-Server eingestellt werden, und trotzdem funktioniert der Internet-Zugriff. Die Anfragen an das Internet werden über die Firewall automatisch dem Proxy-Server zugespielt und dieser geht dann ins Internet und holt die Daten ab und schickt sie wieder zurück zum Client. Transparent heißt das deshalb, weil der Benutzer davon überhaupt nichts mitbekommt.

Um den Transparenten Proxy einzurichten, muss zuerst die DHCP-Server-Konfiguration angepasst werden. Der DHCP-Server soll nämlich den Client so konfigurieren, dass der Server das Default-Gateway wird. In der Datei `/etc/dhcpd.conf` muss noch folgende Zeile eingefügt werden:

```
option routers 192.168.0.1;
```

Da sich normale HTTP-Anfragen und Proxy-Anfragen unterscheiden, muss noch der Proxy-Server konfiguriert werden. Die Clients wissen ja nicht, dass sie mit einem Proxy-Server kommunizieren und verwenden daher normale HTTP-Anfragen.

In der Datei `/etc/squid.conf` müssen an der entsprechenden Stelle folgende Zeilen eingefügt werden:

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Mit **/etc/init.d/dhcp restart** und **/etc/init.d/squid restart** wird die neue Konfiguration übernommen.

Jetzt muss noch die Firewall angepasst werden, damit sie die Pakete, die normalerweise an Port 80 gesendet werden, an den Proxy-Port 3128 weiterleitet. Im Firewall-Script muss folgende Zeilen ergänzt werden:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp -d ! 192.168.0.1 --dport 80 \
-j REDIRECT --to-port 3128
```

Mit **/etc/init.d/firewall restart** die Firewall neu starten.

Jetzt sollte ein Browser auch ohne Proxy-Konfiguration ins Internet kommen.

# 3.5.5. SMART

Das S.M.A.R.T.-System dient zur Überwachung der Festplatten. Dabei protokollieren die Festplatten selber verschiedene Aktionen mit, sodass sie mehr oder weniger sichere Voraussagen machen können, wann sie kaputt gehen werden. Um dieses Feature der Festplatten nutzen zu können, muss zuerst das Paket *smartsuite* mit **apt-get install smartsuite** installiert werden. Dann können mit dem Befehl **smartctl** verschiedene Eigenschaften der Festplatten abgefragt werden. Beispiel:

```
server:~# smartctl -a /dev/hda
Device: WDC WD800JB-00CRA1 Supports ATA Version 5
Drive supports S.M.A.R.T. and is enabled
Check S.M.A.R.T. Passed.

[...]

Vendor Specific SMART Attributes with Thresholds:
Revision Number: 16
Attribute                   Flag     Value Worst Threshold Raw Value
( 1)Raw Read Error Rate     0x000b   200   200   051       0
( 3)Spin Up Time            0x0007   096   096   021       4291
( 4)Start Stop Count        0x0032   100   100   040       35
( 5)Reallocated Sector Ct   0x0033   199   199   140       1
( 7)Seek Error Rate         0x000b   200   200   051       0
( 9)Power On Hours          0x0032   090   090   000       7995
( 10)Spin Retry Count       0x0013   100   253   051       0
( 11)Calibration Retry Count 0x0013  100   253   051       0
( 12)Power Cycle Count      0x0032   100   100   000       35
(196)Reallocated Event Count 0x0032  199   199   000       1
(197)Current Pending Sector 0x0012   200   200   000       0
(198)Offline Uncorrectable  0x0012   200   200   000       0
(199)UDMA CRC Error Count   0x000a   200   253   000       0
(200)Unknown Attribute      0x0009   200   200   051       0
SMART Error Log:
SMART Error Logging Version: 1
No Errors Logged
```

Hier kann man sehen, dass die Festplatte `/dev/hda` schon 7995 Stunden insgesamt lief ("Power On Hours"). Sie wurde bisher 35 Mal angeschaltet ("Power Cycle Count"). Ganz am Schluss wieder protokollierte Fehler aufgelistet werden, aber noch gibt es keine ("No Errors Logged").

Ganz am Anfang steht übrigens, ob die Festplatte denkt, dass sie noch weiterhin fehlerfrei laufen wird: "Check S.M.A.R.T. Passed". Das ist ein gutes Zeichen. Diesen Test kann man auch im BIOS aktivieren, sodass man dann gleich im BIOS gewarnt wird, wenn die Festplatte möglicherweise nicht mehr korrekt funktionieren wird.

# Kapitel 4. Verwaltung und Wartung

## 4.1. Benutzerverwaltung

Linux ist im Gegensatz zu Windows 9x/ME ein Mehrbenutzerbetriebssystem. Es gibt verschiedene Benutzer, die verschiedene Rechte haben können. Unter Windows hat jeder Benutzer alle Rechte, d.h. er darf z.B. alle Dateien löschen oder die Dateien, die ein anderer Benutzer angelegt hat, selber öffnen und bearbeiten - oder auch löschen. Auch unter Linux gibt es ein Benutzer, der alles darf: Der Administrator (Superuser). Er heißt "root" und für ihn gelten nicht die üblichen Grenzen.

Neben den Benutzern gibt es auch noch verschiedene Gruppen. Jeder Benutzer ist Mitglied in mindestens einer Gruppe. Durch verschiedene Gruppen ist es möglich, gleiche Rechte für eine ganze Reihe von Benutzern zu gewähren. Dadurch lässt sich z.B. Teamarbeit realisieren: Man erstellt eine neue Gruppe und jedes Teammitglied wird Gruppenmitglied. Dann gewährt man dieser Gruppe (und nur dieser) Zugriff auf bestimmte Dateien.

Die Einhaltung der Rechte erfolgt unter Linux fast ausschließlich über Dateien: Jede Datei hat einen Besitzer und eine Gruppe, die diese Datei besitzt. Man kann für den Besitzer der Datei, der Gruppe und dem Rest (also allen anderen Benutzern) jeweils eigene Rechte zuteilen. Es gibt für Dateien drei Arten von Zugriffsrecht: Ausführen (eXecute), lesen (Read) und Schreiben (Write). Die Rechte einer Datei kann man mit dem Befehl **ls -l datei** anzeigen lassen. Ein Beispiel:

```
andreas@server:~$ ls -l Makefile
-rw-r--r--    1 andreas  andreas       411 Oct 21 18:05 Makefile
 \_/\_/\_/         |        |
  |  |  |         |        Gruppe
  |  |  |        Besitzer
  |  |  Rechte für alle anderen Benutzer (außer Besitzer und Gruppe)
  |  Rechte für die Gruppe
  Rechte des Besitzers
```

In diesem Beispiel gehört die Datei "*Makefile*" dem Benutzer *andreas*. Die Datei gehört auch der Gruppe mit dem Namen *andreas*. Der Besitzer darf die Datei lesen (r) und bearbeiten, d.h. schreiben (w). Die Mitglieder der Gruppe dürfen die Datei nur lesen (r). Alle anderen Benutzer dürfen die Datei auch nur lesen (r).

### 4.1.1. Benutzer

Alle Benutzer haben eine Nummer, die sogenannte User-ID. Der Administrator (root) hat immer die UID 0. Benutzer mit UIDs unter 100 sind Systembenutzer, d.h. keine realen Benutzer. Diese Systembenutzer werden für bestimmte Dienste benötigt. Zum Beispiel läuft der Webserver als Benutzer www-data (UID 33). Auch alle CGI-Scripte laufen dann unter diesem Benutzer. Das ist sinnvoll, denn wenn alle Programme unter root laufen, hätten die Programme auch alle Rechte. Am Schluss der Skala gibt es noch einen Benutzer nobody (UID 65534). Dieser Benutzer wird für den Gastzugang unter Samba verwendet. Ab UID 500 kommen die realen Benutzer.

### 4.1.2. Gruppen

Durch eine Mitgliedschaft in einer Gruppe bekommen die Benutzer mehr Rechte unter Linux. Wie bei den Benutzern gibt es zunächst eine Gruppe root (Gruppen-ID 0) und dann weitere Systemgruppen. Es existiert auch die Gruppe nogroup (GID 65534), was die Entsprechung zum Benutzer nobody darstellt. Zusätzlich gibt es jetzt eine Gruppe users (GID 100). In dieser Gruppe ist jeder reale Benutzer Mitglied. Alle Mitglieder der Gruppe internet (GID 101) dürfen das Internet aktivieren. Die Mitglieder der Gruppe admin (GID 102) haben unter Samba Administratoren-Rechte, d.h. sie dürfen jede Freigabe beschreiben. Jeder Schüler ist in der Gruppe schueler (GID 103) und jeder Lehrer in der Gruppe lehrer (GID 104). Jeder Benutzer, der die Homepage bearbeiten darf, muss Mitglied der Gruppe homepage (GID 105) sein.

Zusätzlich existiert für jeden einzelnen Benutzer eine eigene Gruppe, die denselben Namen wie der Benutzername trägt (GID und UID müssen nicht identisch sein).

# 4.2. Benutzermanager

Der Benutzermanager ist ein CGI-Programm für den Webserver. Es ist in Perl geschrieben und wird vom Browser aus angewendet. Die Bedienung ist selbsterklärend. Der Quellcode ist im Anhang abgedruckt. Hier werden nur die Befehle genannt, die im Benutzermanager jeweils ausgeführt werden.

## 4.2.1. Benutzer löschen

Jeder Benutzer existiert als Benutzer unter Linux und als Samba-Benutzer. Zunächst wird der Benutzer aus der Samba-Benutzerdatenbank gelöscht:

```
$ smbpasswd -x $login
```

Als nächstes wird das `public_html`-Verzeichnis des Benutzer gesichert werden. Es wird einfach in das Verzeichnis "oldhomes" auf dem Webserver verschoben. Außerdem müssen die richtigen Rechte gesetzt werden:

```
$ mv /home/public_html/$login /usr/local/httpd/htdocs/oldhomes/
$ chown -h -f -R root.www-data /usr/local/httpd/htdocs/oldhomes/$login
$ chmod -R 755 /usr/local/httpd/htdocs/oldhomes/$login
```

Danach kann der Benutzer, sein Homeverzeichnis und seine Gruppe gelöscht werden:

```
$ userdel -r $login
$ groupdel $login
```

Zum Schluss muss noch eine neue Passwort-Datei für apache erstellt werden, damit der Benutzer auch hier nicht mehr existiert.

```
$ /usr/local/sbin/generate_apache_auth
```

## 4.2.2. Passwort ändern

Das Passwort eines Benutzers muss an drei Stellen geändert werden: In der Linux-Benutzerdatenbank, bei Samba und dann in der apache-Passwort-Datei. Drei Befehle müssen dazu ausgeführt werden. Bei den ersten beiden muss das neue Passwort zweimal eingegeben werden.

```
$ passwd $login
$ smbpasswd $login
$ /usr/local/sbin/generate_apache_auth
```

## 4.2.3. Rechte eines Benutzers ändern

Zu den Rechten zählen zum einen die verschiedenen Gruppen, in denen der Benutzer Mitglied ist und zum andern die Quota-Einstellung, die angibt, wieviel Festplattenplatz der jeweilige Benutzer verbrauchen darf. Neben den Rechten kann auch noch die Login-Shell eingestellt werden: Ist die Shell `/bin/bash`, dann darf sich der Benutzer direkt (bzw. über das Netzwerk) am Server einloggen. Soll das verhindert werden, dann muss `/bin/false` eingestellt werden. Sinnvollerweise sollten sich nur Administratoren direkt einloggen dürfen. Die Gruppen können eine Kombination der folgenden Liste sein: *schueler, lehrer, internet, admin, homepage*.

Gruppenzugehörigkeit und Shell ändern:

```
$ usermod -s $shell -G users,$gruppen,$login $login
```

Quota ändern - $limit muss in KB angegeben werden:

```
$ setquota $login /dev/md0 $limit $limit 0 0
```

Zum Schluss muss wieder die apache-Passwort-Datei neu erstellt werden:

```
$ /usr/local/sbin/generate_apache_auth
```

## 4.2.4. Benutzer sperren bzw. freigeben

Ein Benutzer kann gesperrt werden und wieder freigegeben werden. Wenn ein Benutzer gesperrt ist, kann er sich nicht mehr einloggen. Sein Zugang ist deaktiviert. Um einen Benutzer zu sperren, müssen drei Befehle ausgeführt werden:

```
$ passwd -l $login
$ smbpasswd -d $login
$ /usr/local/sbin/generate_apache_auth
```

Um einen Benutzer, der gesperrt ist, wieder freizugeben, müssen auch drei Befehle ausgeführt werden:

```
$ passwd -u $login
$ smbpasswd -e $login
$ /usr/local/sbin/generate_apache_auth
```

## 4.2.5. Neuer Benutzer erstellen

Als erster Schritt muss eine neue Gruppe erstellt werden, die den gleichen Namen trägt wie der Benutzer, den man einrichten will:

```
$ groupadd $login
```

Danach kann der Benutzer erstellt werden. Ist der Benutzer ein Schüler, so ist sein Homeverzeichnis `/home/schueler/$login`; bei Lehrern heißt das Homeverzeichnis `/home/lehrer/$login`. "$name" ist der vollständige Name, "$login" ist der Login-Name. "$groups" ist eine Kombination aus *schueler*, *lehrer*, *internet*, *admin*, *homepage*. Dieser Befehl legt auch gleich das Homeverzeichnis an und kopiert alles aus `/etc/skel/` hinein. In `/etc/skel/` liegt ein halbkonfiguriertes Mozilla-Profil im Ordner `mozilla`.

```
$ useradd -c "$name" -d $home -g $login -G users,$groups,$login \
  -s $shell -m $login
```

Außerdem wird mit dem zweiten Befehl gleich das Passwort gesetzt.

```
$ passwd $login
```

Danach wird das "public_home"-Verzeichnis eingerichtet:

```
$ mkdir -m 0755 /home/public_html/$login
$ chown $login.$login /home/public_html/$login
$ ln -s /home/public_html/$login $home/public_html
```

Anschließend wird der Benutzer für Samba erstellt. Dabei muss das Passwort gleich gesetzt werden.

```
$ smbpasswd -a $login
```

Jetzt muss noch der Quota-Wert gesetzt werden. Ein gewöhnlicher Benutzer darf maximal 50 MB benutzen:

```
$ setquota $login /dev/md0 51200 51200 0 0
```

Damit der Benutzer über den Webserver sein Passwort ändern kann, muss die apache-Passwortdatei neu generiert werden:

```
$ /usr/local/sbin/generate_apache_auth
```

Zum Schluss muss die Mozilla-Konfiguration angepasst werden. Hier werden der Name des Benutzer und seine EMail-Adresse eingerichtet. Und zwar ersetzt das Script in der Datei `prefs.js` die Zeichenfolge `%user%` mit dem vollständigen Namen des Benutzers und `%login%` mit dem Login-Namen. Bei der Erstellung einer Mozilla-Profil-Vorlage für `/etc/skel/` muss also immer `%user%` und `%login%` verwendet werden.

```
$ /usr/local/sbin/make_mozilla_config.pl $home/Mozilla "$login" "$name"
```

## 4.2.6. Informationen über die Benutzer sammeln

Mit **passwd -S $login** erfährt man, ob ein Benutzer gesperrt ist oder nicht und wann der das letzte Mal sein Passwort geändert hat:

```
$ passwd -S andreas
andreas P 03/02/2003 0 99999 7 -1
```

Hier ist der Benutzer nicht gesperrt. Bei einem gesperrten Benutzerkonto würde statt dem "P" ein "L" stehen (locked). Mit **quota -v $login** erfährt man den benutzen Festplattenplatz des Benutzers. Mit **groups $login** sieht man, in welchen Gruppen der Benutzer Mitglied ist.

Wer sich lieber die Informationen ohne Hilfe bestimmter Programme besorgen möchte, kann auch die Benutzerdatenbank von Linux direkt anschauen. Sie besteht aus drei Textdateien: In `/etc/passwd` stehen die verschiedenen Benutzer mit UID, vollständiger Name, Homeverzeichnis, Login-Shell. Das Passwort steht in der Datei `/etc/shadow`. Die Gruppenmitgliedschaft ist in der Datei `/etc/group` festgehalten.

# 4.3. proxymanager

Der Proxymanager regelt den Zugriff auf das Internet: Bevor ein Benutzer das Internet benutzen kann, muss er es zuerst freischalten. Das passiert über den Webserver mit dem Browser. Dabei wird das Internet nicht für diesen Benutzer speziell freigeschalten, sondern für diesen Computer. Die Computer werden anhand ihrer IP-Adresse unterschieden.

Die Computer greifen nie direkt auf einen Webserver im Internet zu. Alle Anfragen gehen über den Server, genauer über den Proxyserver (Der Server ist also kein Router). So wie der Proxyserver in Abschnitt 3.3.3, „Proxy und Filter“ eingerichtet wurde, werden alle Anfragen gefiltert. Wir müssen nun diesen Filter erweitern. Dazu verwenden wir einen selbstgeschriebenen Filter. Da das Schulnetz nicht besonders groß ist, spielt Geschwindigkeit keine so große Rolle und wir können Perl als Programmiersprache verwenden.

Das Filterprogramm ist im Anhang zu finden und heißt `/usr/local/sbin/squid-redirector`. Es überprüft mit Hilfe des proxymanager-Programms, das weiter unten beschrieben wird, ob das Internet für die IP-Adresse des anfragenden Rechners freigeschaltet ist. Wenn das Internet nicht freigeschaltet ist, wird im Browser eine Fehlermeldung angezeigt ("*Internet ist nicht aktiviert*"). Diese Fehlermeldung ist in Wirklichkeit ein CGI-Programm. Wenn das Internet aktiviert ist, wird die Anfrage an den Inhaltsfilter *squidGuard* weitergeleitet.

Das proxymanager-Programm ist in `/usr/local/sbin/proxy-manager` zu finden. Es ist auch ein Perl-Programm. Das Programm kann mit verschiedenen Parametern aufgerufen werden:

```
$ /usr/local/sbin/proxy-manager -h
Syntax: proxy-manager [-hdocak] [login] [ip-adresse] [zeitsekunden]

    -h   Dieser Hilfetext
    -d   der eigentliche proxy-manager wird im Hintergrund gestartet
    -o   öffnet eine Verbindung für "ip-adresse" und schließt sie
         automatisch, wenn "zeitsekunden" erreicht sind
    -c   schließt die Verbindung für "ip-adresse" manuell
    -a   prüft, ob für "ip-adresse" eine Verbindung besteht
    -k   beendet den proxy-manager und löscht alle Verbindungen
    -t   tested, ob der proxy-manager im Hintergrund läuft
         (siehe /var/run/proxy-manager.pid).

    [login]
         Der Benutzer, der eine Verbindung öffnet, etc. wird geloggt.

    [ip-adresse]
         Jede beliebige Form ist erlaubt:
         Beispiel: 192.168.0.168
         Es kann auch 'all' verwendet werden.

    [zeitsekunden]
         Anzahl der vergangenen Sekunden seit seit 00:00:00, Jan 1, 1970.
         Zu ermitteln z.B. durch "date +%s".
```

```
 Hinweise:
    Falls Fehler beim Start des proxy-managers auftreten
    sollten: die Zugriffsrechte überprüfen:
    /var/log/internet-freigabe
    /var/state/internet-freigabe

proxy-manager 1.2 (2003-08-20)
(c) 2001 Thomas Bleher <ThomasBleher@gmx.de>
(c) 2002,2003 Andreas Dangel <adabolo@adabolo.de>

proxy-manager comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; see the GNU GPL.
```

Im Firewall-Script wird der Proxymanager mit **proxy-manager -d** als Hintergrundprozess gestartet. Das ist nötig, denn das Internet kann immer nur für eine bestimmte Zeit freigeschaltet werden. Der Hintergrundprozess prüft jede Minute, ob für einen freigeschalteten Computer die Zeit abgelaufen ist und deaktiviert gegebenenfalls das Internet. Alle weiteren Parameter sind ja im Hilfetext erklärt. Das CGI-Programm, mit dem man das Internet letztendlich aktivieren kann, ruft ebenfalls den Proxymanager auf, z.B. mit **proxy-manager -o andreas 192.168.0.140 1067188761**. Daraufhin schreibt der Proxymanager in die Datei `/var/state/internet-freigabe` folgende Zeile:

```
192.168.0.140:1067188761
```

Der Hintergrundprozess überprüft dann, ob der angegbene Zeitpunkt vorbei ist und entfernt die Zeile gegebenenfalls wieder.

Damit der Proxyserver (squid, siehe Abschnitt 3.3.3, „Proxy und Filter“) unseren neuen Filter auch verwendet, muss die Datei `/etc/squid.conf` bearbeitet werden. Folgender Eintrag muss abgeändert werden:

```
redirect_program /usr/local/sbin/squid-redirector
```

Mit **/etc/init.d/squid restart** wird unser neuer Filter verwendet.

Alle hier erwähnten Programme sind im Anhang zu finden.

### 4.3.1. Logdatei

Die Logdatei für den Proxymanager heißt `/var/log/internet-freigabe`. Hier wird mitgeloggt, wer das Internet jeweils freischaltet. Damit diese Datei nicht ins Unendliche anwächst, wird wieder logrotate konfiguriert: Die Datei `/etc/logrotate.d/internet-freigabe` muss mit folgendem Inhalt existieren:

```
/var/log/internet-freigabe {
        rotate 7
        weekly
        compress
        missingok
        notifempty
}
```

## 4.4. Info-Tools

### 4.4.1. Festplattenplatz

Um den vebrauchten und den noch freien Festplattenplatz zu erfahren, wird **df** verwendet. df steht für "Disk free". Mit der Option "-h" werden die Angaben besser verständlich formatiert; es werden Angaben in MB und GB gemacht. Hier ein Beispielaufruf von **df -h**:

```
$ df -h
Filesystem             Size  Used Avail Use% Mounted on
/dev/md0                65G   24G   41G  37% /
```

```
/dev/md1               9.3G  237M  9.0G   3% /var
```

## 4.4.2. uptime

Wie lange der Server seit dem letzten Neustart schon läuft, erfährt man mit dem Befehl **uptime**:

```
$ uptime
 12:46am up 3 days, 4:31, 0 users, load average: 0.00, 0.00, 0.00
```

Der Server läuft hier also seit 3 Tagen, 4 Stunden und 31 Minuten. Außerdem zu sehen ist die durchschnittliche Auslastung (*load average*). Ein Wert von 0 bedeutet, dass der Server überhaupt nicht ausgelastet ist.

# 4.5. Weitere CGI-Scripte

Alle CGI-Scripte sind im Anhang zu finden. Beim Installieren ist zu beachten, dass die CGI-Scripte ausführbar sein müssen. Es müssen also gegebenenfalls die Attribute mit **chmod 755 script.cgi** angepasst werden.

## 4.5.1. Passwort ändern

Damit jeder Benutzer sein Passwort ändern kann, gibt es ein CGI-Script. Es ist mit dem Browser über den Webserver erreichbar. Das Script wird unter `/usr/local/httpd/htdocs/passwd/index.cgi` installiert.

## 4.5.2. Mail-Weiterleitung konfigurieren

Jeder Benutzer hat ja eine Mailadresse der Form: *login@gm.rt.schule-bw.de*. Alle Mails, die an diese Adresse geschickt werden, können auch an eine externe Mailadresse (z.B. gmx.de oder web.de) weitergeleitet werden. Mit dem CGI-Script, das unter `/usr/local/httpd/htdocs/mail-config/index.cgi` installiert wird, kann dies jeder Benutzer selber einrichten.

## 4.5.3. Benutzerverzeichnisse

Mit diesem CGI-Script lassen sich die verschiedenen `public-html`-Verzeichnisse der Benutzer anschauen. Diese Verzeichnisse sind über die Adresse *http://server/~login/* erreichbar. Das Script wird unter `/usr/local/httpd/cgi-bin/benutzer` installiert.

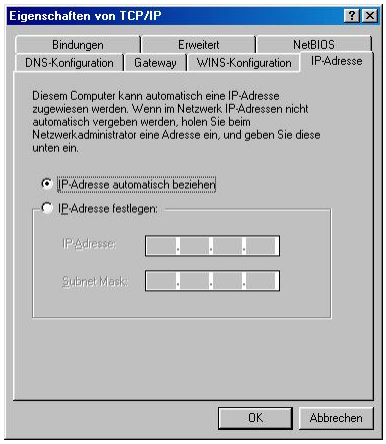
# Kapitel 5. Clients

## 5.1. Windows 98SE

In der Schule verwenden wir Windows 98 SE als Client-Betriebssystem. Windows 98 besitzt noch einen DOS-Modus, sodass ältere DOS-Programme auch weiterhin benutzt werden können. Außerdem kann so der Linux-basierte Partitionsmanager verwendet werden, der die Partition sichert und bei Bedarf wieder zurückkopiert.

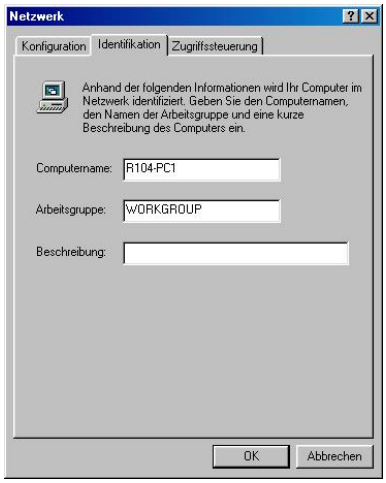
### 5.1.1. Zusammenarbeit mit dem Netzwerk und Samba

#### 5.1.1.1. Netzwerk




Zunächst muss der Windows-Rechner ins Netzwerk eingebunden werden, d.h. er muss eine passende IP-Adresse besitzen. Diese erhält er über DHCP ("*IP-Adresse automatisch beziehen*"). Zu dieser Option kommt man über: Rechtsklick auf "*Netzwerkumgebung*" und "*Eigenschaften*" wählen. Dann im Register "*Konfiguration*" Doppelklick auf "*TCP/IP*". Hier ist die Option im Register "*IP-Adresse*" zu finden.

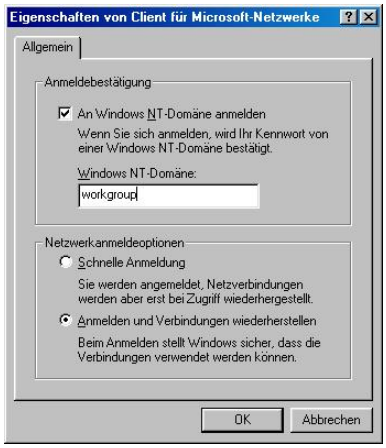
#### 5.1.1.2. Computer-Name




Um die Samba-Logdateien etwas übersichtlicher zu gestalten erhält jeder Computer einen eindeutigen Namen im Windows-Netzwerk. Die Bennenung erfolgt so: `RXXX-PC#`. "XXX" steht für die Raum-Nummer, sie ist immer dreistellig (z.B. R023). "#" ist die Nummer des PCs im Raum. Die Rechner werden im Raum einfach durchgezählt (Bsp. R023-PC2 oder R104-PC15). Zu dieser Option kommt man über: Rechtsklick auf "*Netzwerkumgebung*" und "*Eigenschaften*" wählen. Dann zum Register "*Identifikation*" wechseln. Hier den Computernamen eintragen. Die Arbeitsgruppe heißt "*workgroup*".

### 5.1.1.3. Netz-Logon


Damit sich ein Benutzer beim Start von Windows einloggen kann und damit das Passwort vom Linux-Server bestätigt wird, ist auch eine spezielle Einstellung nötig. Zur dieser Option kommt man über: Rechtsklick auf "*Netzwerkumgebung*" und "*Eigenschaften*" wählen. In diesem Dialog muss die "*Primäre Netzwerkanmeldung*" auf "*Client für Microsoft-Netzwerke*" stehen. Außerdem muss dieser "*Client für Microsoft-Netzwerke*" mit einem Doppelklick konfiguriert werden. Im sich neu geöffneten Dialog wird "*An Windows NT-Domäne anmelden*" aktiviert. Die Windows NT-Domäne heißt "*workgroup*".




## 5.1.2. Netzwerkinstallation von best. Programmen

Bestimme Programme, wie z.B. StarOffice, bieten von sich aus eine Netzwerkinstallation an. Hier muss einfach als Installationspfad `P:\Programmname` gewählt werden. Meistens geht dies auch mit jedem anderen Programm, bei dem man den Installationspfad selber wählen kann. Eine etwas flexiblere Lösung ist die, dass man als Installationspfad `//server/programme/Programmname` wählt. Dann werden die Symbole (Icons) im Startmenü bzw. auf dem Desktop entsprechend erstellt und wenn sich der Benutzer unter Windows nicht einloggt, sondern "*Abbrechen*" drückt, dann kann er die Programme trotzdem mit Doppelklick starten.

Die Programme, die im Netzwerk verfügbar sein sollen, werden also unter `//server/programme` installiert. In dieser Freigabe hat jeder Benutzer alle Rechte, d.h. ein x-beliebiger Schüler könnte hier alles löschen. Es ist also dringend ein Backup dieser Freigabe zu empfehlen.

## 5.1.3. Partitionsmanager

Der Partitionsmanager ist in Wirklichkeit ein Mini-Linux-System. Es wird mit dem Linuxloader LoadLin im DOS-Modus gestartet. Alle Programme sind auf einer Startbaren RAMDisk. Mit speziellen Argumenten, die man schon LoadLin mit auf den Weg gibt, kann die Partition z.B. vollautomatisch zurückgesichert werden. Das

Programm, das die eigentliche Arbeit leistet, heißt partimage. Hier ein paar Aufruf-Beispiele:

```
C:\PARTIMG> loadlin bzimage root=/dev/ram initrd=initrd.gz rw manual
```

Hier wird ein Interaktivmodus gestartet ("manual"). Man kann hier z.B. manuell eine Netzwerkverbindung aufbauen, etc.

```
C:\PARTIMG> loadlin bzimage root=/dev/ram initrd=initrd.gz rw restore sicher.img
```

Hier wird das Image mit dem Dateinamen `sicher.img` zurückkopiert. Das Image liegt auf dem Laufwerk D: (unter Linux: `/dev/hda5`); das ist das erste logische Laufwerk. Zurückkopiert wird das Image auf das Laufwerk C: (`/dev/hda1`). Laufwerk C: wird also wiederhergestellt. Nach Abschluss des Vorgangs wird der Rechner automatisch neugestartet.

```
C:\PARTIMG> loadlin bzimage root=/dev/ram initrd=initrd.gz rw store sicher.img
```

Mit diesem Befehl wird ein Image von Laufwerk C: erstellt. Es wird unter `D:\SICHER.IMG` gespeichert. Der Rechner wird danach ebenfalls neugestartet.

```
C:\PARTIMG> loadlin bzimage root=/dev/ram initrd=initrd.gz rw update
    http://192.168.0.1/images/sicher.img sicher.img
```

Dies ist ein Befehl und er muss in einer Zeile geschrieben werden. Das Image wird dann von der angegebenen URL heruntergeladen. Es wird dann auf den Rechner kopiert und auf dem Laufwerk D: gespeichert. Diese Funktion arbeitet möglicherweise nicht ganz ordnungsgemäß; ausprobieren. Als dritter Parameter muss der Dateiname des heruntergeladenen Images angegeben werden. Existiert das Image auf Laufwerk D: schon, wird es zuerst gelöscht und dann heruntergeladen.

Wenn ein Image schnell verteilt werden muss, dann kann im *Netlogon-Script* ein Befehl wie

```
copy \\server\pub\sicher.img D:\SICHER.IMG
```

eingefügt werden. Anschließend muss man sich nur noch unter Windows anmelden und das Image wird auf den jeweiligen Rechner kopiert.

Um das Zurückspielen des Images, das auf Laufwerk D: liegt, kann der entsprechende Befehl in eine Stapel-Datei (Batch-Datei) geschrieben werden. Danach muss `C:\CONFIG.SYS` bearbeitet werden. (Die Datei ist möglicherweise schreibgeschützt und/oder versteckt.) Die Datei muss ungefähr folgenden Inhalt haben:

```
[menu]
menuitem=W98, Windows 98 starten
menuitem=PART, Image zurückspielen
menudefault=W98,5

[W98]
DEVICE=C:\WINDOWS\setver.exe
device=C:\WINDOWS\COMMAND\display.sys con=(ega,,1)
Country=049,850,C:\WINDOWS\COMMAND\country.sys

[PART]
shell=c:\command.com /C C:\PARTIMG\RESTORE.BAT

[COMMON]
```

Damit erscheint bei jedem Windows-Start ein Menü ("*Windows 98 Startmenü*"). Wenn der Benutzer nach 5 Sekunden nichts ausgewählt hat, wird automatisch Windows gestartet. Wählt der Benutzer den Menüpunkt "*Image zurückspielen*" wird das Mini-Linux gestartet und der Computer wiederhergestellt.

Das Programm ist auf der CD zu finden.

# 5.1.4. Windows beschränken

Damit nicht jeder Benutzer in der Systemsteuerung herumspielt und etwas umstellt, wird diese deaktiviert. Das geschieht einfach mit einer Registrierungsdatei. So wird das Startmenü ein wenig kleiner und die Anzeige-Eigenschaften sind nicht mehr erreichbar. Es kann also kein Benutzer ein Bildschirmschonerpasswort setzen.

Mit einer zweiten Registrierungsdatei können die Restriktionen wieder rückgängig gemacht werden. Die Registrierungsdateien können ganz bequem über einen Doppelklick eingelesen werden und sind nach einer Neuanmeldung aktiv.

Hier die Datei `restrict.reg`.

```
REGEDIT4

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer]
"NoSetFolders"=dword:00000001
"NoSetTaskbar"=dword:00000001
"NoFolderOptions"=dword:00000001
"NoSetActiveDesktop"=dword:00000001
"NoWindowsUpdate"=dword:00000001
"NoRecentDocsMenu"=dword:00000001
"NoRecentDocsHistory"=dword:00000001
"ClearRecentDocsOnExit"=dword:00000001
"NoFavoritesMenu"=dword:00000001
"NoPrinterTabs"=dword:00000001
"NoDeletePrinter"=dword:00000001
"NoAddPrinter"=dword:00000001
"NoSaveSettings"=dword:00000001

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Winlogon]
"DontDisplayLastUserName"=dword:00000001

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Network]
"DisablePwdCaching"=dword:00000001

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network]
"HideSharePwds"=dword:00000001
"DisablePwdCaching"=dword:00000001
"NoNetSetup"=dword:00000001
"NoNetSetupIDPage"=dword:00000001
"NoNetSetupSecurityPage"=dword:00000001

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\System]
"DisableRegistryTools"=dword:00000001
"NoDispCPL"=dword:00000001
"NoDispBackgroundPage"=dword:00000001
"NoDispScrSavPage"=dword:00000001
"NoDispAppearancePage"=dword:00000001
"NoDispSettingsPage"=dword:00000001
"NoProfilePage"=dword:00000001
"NoSecCPL"=dword:00000001
"NoPwdPage"=dword:00000001
"NoAdminPage"=dword:00000001
"NoProfilePage"=dword:00000001
"NoDevMgrPage"=dword:00000001
"NoConfigPage"=dword:00000001
"NoFileSysPage"=dword:00000001
"NoVirtMemPage"=dword:00000001

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders]
"Personal"="H:\"
```

Hier die Datei `unrestrict.reg`.

```
REGEDIT4
[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer]
"NoSetFolders"=dword:00000000
"NoSetTaskbar"=dword:00000000
"NoFolderOptions"=dword:00000000
"NoSetActiveDesktop"=dword:00000000
"NoWindowsUpdate"=dword:00000000
"NoRecentDocsMenu"=dword:00000000
"NoRecentDocsHistory"=dword:00000001
```

```
"ClearRecentDocsOnExit"=dword:00000001
"NoFavoritesMenu"=dword:00000000
"NoPrinterTabs"=dword:00000000
"NoDeletePrinter"=dword:00000000
"NoAddPrinter"=dword:00000000
"NoSaveSettings"=dword:00000001

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Winlogon]
"DontDisplayLastUserName"=dword:00000001

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network]
"HideSharePwds"=dword:0000001
"DisablePwdCaching"=dword:00000001
"NoNetSetup"=dword:00000000
"NoNetSetupIDPage"=dword:00000000
"NoNetSetupSecurityPage"=dword:00000000

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\System]
"DisableRegistryTools"=dword:00000000
"NoDispCPL"=dword:00000000
"NoDispBackgroundPage"=dword:00000000
"NoDispScrSavPage"=dword:00000000
"NoDispAppearancePage"=dword:00000000
"NoDispSettingsPage"=dword:00000000
"NoProfilePage"=dword:00000000
"NoSecCPL"=dword:00000000
"NoPwdPage"=dword:00000000
"NoAdminPage"=dword:00000000
"NoProfilePage"=dword:00000000
"NoDevMgrPage"=dword:00000000
"NoConfigPage"=dword:00000000
"NoFileSysPage"=dword:0000000
"NoVirtMemPage"=dword:00000000

[HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders]
"Personal"="H:\"
```

Außerdem werden hier kleinere Feineinstellungen an Windows vorgenommen. So wird z.B. das Verzeichnis "*Eigene Dateien*" so konfiguriert, dass es auf das Homeverzeichnis (H:\) auf Samba zeigt. Da wird Windows so eingestellt, dass es beim Login-Dialog den Namen des vorigen Benutzers löscht. Außerdem wird der Passwort-Cache deaktiviert. Ist diese Funktion aktiv, dann speichert Windows in C:\WINDOWS für jeden Benutzer eine PWL-Datei, in der das Passwort gespeichert ist. Mit ein paar Hilfsprogrammen aus dem Internet kann mit dieser Datei das Passwort rekonstruiert werden.

# 5.2. Windows 2000 Professional / Windows XP Professional

Der Aufwand, Windows 2000 oder Windows XP dem Netzwerk hinzuzufügen ist etwas größer, aber machbar. Zunächst muss entschieden werden, ob Windows 2000/XP auf einer FAT32-Systempartition oder auf einer NTFS-Systempartition installiert werden soll. Da Windows 2000/XP eine Benutzer- und Rechteverwaltung hat, ist nicht mehr jeder Benutzer Administrator wie unter Windows 98, wo jeder alles darf. Wenn FAT32 gewählt wird, wird die Rechteverwaltung zum Teil aufgehoben, d.h. jeder darf überall auf der Festplatte Dateien speichern und verändern. Das spart unter Umständen einiges an Konfigurationsaufwand. Andererseits gibt es dabei auch ein Sicherheitsproblem bezüglich Viren - die dürfen dann auch überall Dateien verändern oder löschen. Bei NTFS achtet Windows 2000/XP dagegen auf die Rechte, die der angemeldete Benutzer hat. Da der angemeldete Benutzer aber nur auf dem Samba-Server existiert, darf der angemeldete Benutzer nur Dateien in seinem Homeverzeichnis speichern.

*Hinweis:* Von Windows XP gibt es zwei Versionen. Die "Home Edition" kann keiner Domäne beitreten; man kann diese Version auch nicht so konfigurieren, dass man sie wie mit Windows 98 an einer Domäne anmeldet. Windows XP Home Edition ist also für ein Netzwerk völlig unbrauchbar. Nur die Windows XP Professional-Version hat noch alle benötigten Funktionen.

Windows 2000/XP dem Netzwerk hinzufügen bedeutet, dass der Windows-PC der (Samba-) Domäne hinzugefügt werden muss. Auf dem Samba-Server muss deshalb unter anderem ein Maschinen-Account für den Rechner eingerichtet werden. Den Rechner der Domäne hinzufügen muss dann der Samba-Benutzer "*root*", der die Funktion des Domänenadministrators hat.

Aber zuerst alles der Reihe nach:

1. Auf dem Server muss mit **smbpasswd -a root** der Samba-Benutzer "root" erstellt werden. Als Passwort sollte allerdings nicht das tatsächliche root-Passwort verwendet werden - sicher ist sicher. Dieses Passwort wird später benötigt.

2. Zunächst sollte mit **cp /etc/samba/smb.conf /etc/samba/smb.conf.backup** ein Backup der Samba-Konfigurationsdatei erstellt werden. In der Konfigurationsdatei sind nämlich einige Änderungen nötig, die aber am Schluss wieder alle rückgängig gemacht werden müssen.

   Die erste Änderung betrifft den Benutzer "root". Der muss aus der Liste "invalid users" gelöscht werden und zu "valid users" hinzugefügt werden. Danach muss mit **/etc/init.d/samba restart** der Samba-Server neugestartet werden.

3. Dann muss der Maschinen-Account für den Rechner eingerichtet werden. Das sind im Prinzip normale Benutzer; der Benutzername endet allerdings auf ein Dollarzeichen und das Passwort handelt der Windows-Rechner mit dem Samba-Server selbstständig aus. Der Name des Maschinen-Accounts muss mit dem *Computernamen* des Windows-PCs identisch sein. Außerdem ist es eventuell sinnvoll, für alle Maschinen-Accounts eine extra Linux-Gruppe anzulegen, etwa **groupadd windows**. Mit **useradd -g windows -s /bin/false "RXXX-PCY$"** (das Dollarzeichen am Ende und die Anführungszeichen beachten) wird zunächst ein Benutzer unter Linux erstellt und mit **smbpasswd -a -m RXXX-PCY** wird für Samba der dazugehörige Maschinen-Account erzeugt (hier darf kein Dollarzeichen angegeben werden, das fügt smbpasswd automatisch an).

4. Jetzt kann der Windows-Rechner gestartet werden. Für die weiteren Schritte muss man als lokaler Administrator eingeloggt sein. Der Windows-Rechner muss genauso heißen, wie der Name des Maschinen-Accounts (ohne das Dollarzeichen).

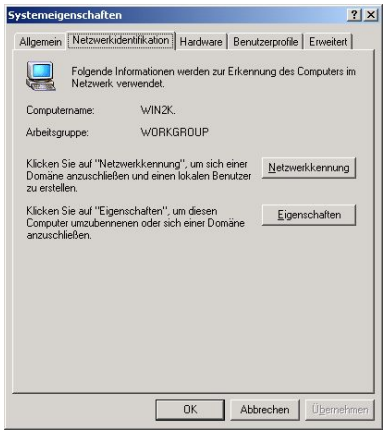
5. Das Hinzufügen des Windows-Rechners zur Domäne funktioniert nur dann, wenn erstens der Samba-Server auch als WINS-Server eingetragen ist (das sollte DHCP automatisch machen) und zweitens der Windows-Rechner noch keinerlei Verbindungen zum Samba-Server aufgebaut hat. Aber genau das macht Windows wahrscheinlich schon automatisch; und um das zu verhindern, muss der Samba-Server ein wenig umkonfiguriert werden.

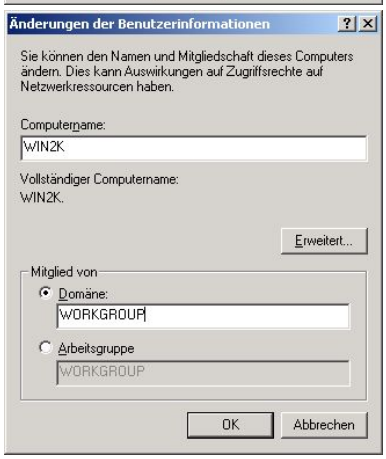
   Zuerst wird die IP-Adresse des Windows-Rechners benötigt. Diese bekommt man mit **ipconfig /all** in der MS-DOS-Eingabeaufforderung heraus. Dann kann auf dem Server mit **smbstatus -S** überprüft werden, ob diese IP-Adresse in der Liste auftaucht. Wenn ja, dann müssen in `/etc/samba/smb.conf` die einzelnen Shares (Freigaben) bearbeitet werden. Und zwar muss mit

   ```
   hosts deny = %ip-adresse%
   ```

   bei jedem Abschnitt (außer im Abschnitt "*[global]*") dem Windows-Rechner der Zugriff verboten werden. Dann muss mit **/etc/init.d/samba restart** der Samba-Server neugestartet werden. Auf dem Windows-Rechner sollte man sich kurz aus- und wieder einloggen. Jetzt sollte ein **smbstatus -S** auf dem Server keine Verbindungen mehr zu dem Windows-Rechner anzeigen.

6.






Jetzt sind alle Vorbereitungen getroffen. Jetzt unterscheiden sich Windows 2000 und Windows XP geringfügig voneinander. Auf dem Windows-2000-Rechner kommt man mit einem Rechtsklick auf *Arbeitsplatz*, dann weiter auf *Identifikation* und auf *Eigenschaften* zum gewünschten Dialog. Unter Windows XP gelangt man über Rechtsklick auf *Arbeitsplatz*, dann weiter auf *Computername* und auf *Ändern* zum gewünschten Dialog, um die Domäneneigenschaften zu ändern. Hier muss man dann *Domäne* auswählen und "workgroup" als Domäne eintragen. Danach wird nach einem Benutzernamen und Passwort verlangt. Das ist der Zugang des Domänen-Administrators (nur der darf neue Rechner der Domäne hinzufügen). Bei Samba ist das der im ersten Schritt hinzugefügte Benutzer "*root*". Nach einiger Zeit sollte dann eine Messagebox auftauchen, mit der Nachricht: "*Willkommen in der Domäne workgroup*".

7. Zum Schluss müssen die Änderungen an Samba wieder rückgängig gemacht werden. Mit **smbpasswd -x root** wird der Benutzer "root" wieder gelöscht. Mit **cp /etc/samba/smb.conf.backup /etc/samba/smb.conf** (eventuell das Überschreibung bestätigen) wird das Backup der Konfigurationsdatei wieder zurückgespielt. Jetzt noch einmal mit **/etc/init.d/samba restart** den Samba-Server neustarten. Das war's.

8. Bei Windows XP Professional ist zum Schluss noch ein Schritt nötig, damit der Login-Vorgang funktioniert. Sehr wahrscheinlich kommt beim Login folgende Fehlermeldung: "Es kann keine Verbindung mit der Domäne hergestellt werden, da der Domänencontroller nicht verfügbar ist bzw. das Computerkonto nicht gefunden werden konnte. Wieder holen Sie den Vorgang später..." Das liegt daran, dass Windows XP versucht, eine verschlüsselte Verbindung zum Samba-Server aufzubauen. Diese Funktion, die

"RequireSignOrSeal" heißt, muss deaktiviert werden. Dazu am Windows-XP-Rechner als lokaler Administrator einloggen, dann zur Systemsteuerung in klassischer Ansicht gehen. Dann auf "Verwaltung" und dort auf "Lokale Sicherheitsrichtlinie" doppelklicken. In diesem Fenster auf der linken Seite unterhalb von "Sicherheitseinstellungen/Lokale Richtlinien" die Kategorie "Sicherheitsoptionen" auswählen und in der rechten Seite dann auf die Option "Domänenmitglied: Daten des sicheren Kanals digital verschlüsseln und signieren (immer)" doppelklicken und deaktivieren. Damit sollte auch der Login-Vorgang mit Windows XP problemlos funktionieren.

## 5.2.1. Benutzerprofile, Roaming Profiles

Zunächst mal eine interessante Samba-Einstellung, die ab Windows NT (also z.B. unter Windows 2000 oder Windows XP) funktioniert:

```
logon drive = z:
```

Damit wird das Home-Verzeichnis des eingeloggten Benutzers automatisch mit dem Laufwerk "Z:" verbunden. Der Buchstabe Z ist die Standardeinstellung. Man muss also im Netlogon-Script das Home-Verzeichnis nicht mehr manuell verbinden, sondern könnte einfach "logon drive = h:" verwenden.

Jeder Benutzer unter Windows 2000/XP hat ein Benutzerprofil. In diesem Profil wird für jeden Benutzer extra der Desktop, das Startmenü, die eigenen Dateien gespeichert. Außerdem wird der benutzerspezifische Teil der Windows-Registrierung gespeichert (z.B. Desktop-Hintergrundbild-Einstellungen). "Roaming Profile" bedeutet, dass das Benutzerprofil mit seinem Benutzer den PC wechselt. D.h. es ist egal, auf welchem Rechner sich der Benutzer einloggt, er findet immer seinen Desktop mit seinen Symbolen vor. Das funktioniert folgendermaßen: Beim Einloggen wird das Benutzerprofil vom Server auf die Festplatte des jeweiligen Rechners kopiert und beim Ausloggen wieder zurück auf den Server kopiert. Damit werden alle Änderungen, die der Benutzer gemacht hat, übernommen und gesichert. Der Nachteil ist, dass der Login- bzw. Logout-Vorgang etwas länger dauert. Der entscheidende Vorteil ist aber, dass der Benutzer an jedem Rechner seine gewohnte Arbeitsumgebung vorfindet. Und es kommt beim Einloggen keine Fehlermeldung, dass das Benutzerprofil nicht gefunden werden konnte.

Um für Windows 2000/XP die Benutzerprofile zu aktivieren, ist folgende Samba-Einstellung nötig:

```
logon path = \\server\%U\profile
```

Damit wird das Profil im Homverzeichnis des jeweiligen Benutzers gespeichert (das "%U" wird durch den Login-Namen des Benutzers ersetzt). Für Windows 9x gibt es eine entsprechende Option: "logon home". Wenn im Homeverzeichnis des Benutzers noch kein Profil existiert, wird eines beim ersten Login automatisch angelegt.

Ist man allerdings als Admin-User eingeloggt (alle Mitglieder der Gruppe "admin"), dann kann es mit dem automatischen erstellen des Profils Probleme geben. Als Admin-User werden neue Dateien und Verzeichnisse grundsätzlich so angelegt, dass der (Linux-) Benutzer "root" der Eigentümer ist und nicht der Benutzer selber. Das Verzeichnis `profile/` im Homeverzeichnis des Benutzer gehört also dem Benutzer nicht selber. Dies kann man entweder unter Linux mit **chown Benutzername profile/** korrigieren oder unter Windows (Rechtsklick auf den Ordner, Eigenschaften, ...).

Welche Benutzerprofile auf dem Windows-Rechner existieren, kann man über die Systemsteuerung in klassischer Ansicht, dann System, Erweitert, Benutzerprofile und Einstellungen nachschauen. Die Profile werden ja beim Einloggen auf die Festplatte des Windows-Rechners kopiert. Sie landen im Verzeichnis `C:\Dokumente und Einstellungen\%user%\` (%user% ist entsprechend zu ersetzen). Existiert allerdings schon ein lokaler Benutzer mit gleichem Benutzernamen, so heißt der Ordner `C:\Dokumente und Einstellungen\%user%.WORKGROUP\`.

## 5.2.2. Mozilla-Profileinstellungen beim Login kopieren

Da Windows 98 noch kein Multiuser-Betriebssystem ist, war es möglich, die Mozilla-Profil-Datei `registry.dat` einfach in den entsprechenden Windows-Ordner zu kopieren und alle Benutzer hatten diese Datei dann auch verwendet. Ab Windows 2000 werden die "Anwendungsdaten" für jeden Benutzer individuell in seinem Profil gespeichert. Das Profil, das nach dem Einloggen gültig ist, liegt ja im Verzeichnis `C:\Dokumente und Einstellungen\%user%\` und heißt für jeden Benutzer anders. Damit man die

Datei trotzdem über das Logon-Script an die richtige Stelle kopieren kann, muss man die Umgebungsvariablen, die Windows setzt, bemühen. So könnte eine entsprechende Zeile in `logon.bat` so aussehen:

```
mkdir "%USERPROFILE%\Anwendungsdaten\Mozilla"
copy \\server\netlogon\registry.dat "%USERPROFILE%\Anwendungsdaten\Mozilla" /Y
```

Hier heißt die verwendete Umgebungsvariable "USERPROFILE". Wichtig sind außerdem noch die Anführungszeichen, die nötig sind, da der Pfad zum Benutzerprofil Leerzeichen enthält. Welche Umgebungsvariablen zur Verfügung stehen, erhält man in der Eingabeaufforderung mit dem Befehl **set**. Ein anderer interessanter Befehl heißt **pause**. Damit wartet das Logon-Script auf eine Tastatur-Eingabe und das Fenster schließt sich nicht automatisch. Dadurch ist es möglich, eventuelle Fehlermeldungen beim Kopieren etc. zu lesen.

# Kapitel 6. Linux-Grundlagen

## 6.1. Einloggen/Ausloggen

Nachdem linux hochgefahren ist, erscheint ein Text-basierter Login-Bildschirm:

```
Debian GNU/Linux 3.0 server tty1

server login:
```

Hier kann man dann seinen Benutzernamen eingeben. Danach wird nach dem Passwort verlangt. Ist man eingeloggt, sieht der Bildschirm ungefähr so aus:

```
Last login: Sun Nov 9 09:35:49 2003 on tty1
Linux server 2.4.22 #1 Sun Aug 17 16:51:06 CEST 2003 i686 unkown

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
andreas@server:~$
```

Die letzte Zeile ist die Eingabeaufforderung. Hier können jetzt Befehle eingegeben werden.

Will man sich wieder ausloggen, so muss man den Befehl **logout** ausführen. Es erscheint dann wieder der Login-Bildschirm.

## 6.2. Linux herunterfahren (shutdown)

Linux herunterfahren kann nur der Superuser. Man muss also als Benutzer *root* eingeloggt sein. Jetzt kann man mit **init 0** oder **halt** Linux sofort herunterfahren. Mit **shutdown -h +15** fährt Linux in 15 Minuten herunter.

Will man Linux neustarten, so muss man **init 6** bzw. **reboot** ausführen. Auch mit Shutdown geht das Neustarten: **shutdown -r +15** startet Linux in 15 Minuten neu.

## 6.3. Dateiverwaltung

### 6.3.1. Dateien auflisten

Mit dem Befehl **ls** werden Dateien aufgelistet (ls = LiSt). Dieser Befehl entspricht dem DOS-Befehl dir. Mit bestimmten Parametern kann die Ausgabe variiert werden: **ls -l** gibt eine detailreichere Liste aus (-l für long). Dabei gibt der Befehl zunächst mal das aktuelle Verzeichnis aus. Will man ein anderes Verzeichnis sehen, muss man z.B. **ls /home** aufrufen. Hier ein paar Beispiele:

```
$ ls
CD-ROOT                      source
CD-ROOT-Inhalt.txt           source-20031015-214838.tar.bz2
Makefile                     source-20031016-214533.tar.bz2
archiv1.tar.bz2              source-20031018-120615.tar.bz2
html                         source-20031018-150309.tar.bz2
passwörter.out               source-20031018-184430.tar.bz2
passwörter.pdf               source-20031020-191948.tar.bz2
server-docu-sicherung.pdf    source-20031021-183556.tar.bz2
server-docu-sicherung.ps     source-20031023-191929.tar.bz2
[...]
$ ls -l
total 7944
```

```
drwxr-xr-x    3 andreas andreas      4096 Aug 18 19:48 CD-ROOT
-rw-r--r--    1 andreas andreas      3081 Aug 19 16:03 CD-ROOT-Inhalt.txt
-rw-r--r--    1 andreas andreas       411 Oct 21 18:05 Makefile
-rw-r--r--    1 andreas andreas      7079 Aug 19 14:55 archiv1.tar.bz2
drwxr-xr-x    3 andreas andreas      8192 Nov  8 20:25 html
-rw-r--r--    1 andreas andreas         0 Sep  3 22:14 passwörter.out
-rw-r--r--    1 andreas andreas      7940 Sep  3 22:14 passwörter.pdf
-rw-r--r--    1 andreas andreas    214374 Oct 15 21:02 server-docu-sicherung.pdf
-rw-r--r--    1 andreas andreas 9991 Oct 15 21:48 source-20031015-214838.tar.bz2
[...]
$ ls /home
andreas ftp samba
```

Die erste Spalte in der Ausgabe von **ls -l** zeigt die Rechte an: Das erste Zeichen stellt den Typ das ("-" ist eine normale Datei, "d" ein Verzeichnis). Die nächsten drei Zeichen stellen die Rechte für den Besitzer der Datei/des Verzeichnisses dar. "r" heißt Leserecht, "w" Schreibrecht und "x" Ausführrecht. Die folgenden drei Zeichen stellen die Rechte für die Gruppe dar, der diese Datei gehört. Die letzten drei Zeichen sind die Rechte, die jeder andere Benutzer des Systems an der Datei hat.

In der dritten Spalte steht der Besitzer und in der vierten Spalte die Gruppe. In der fünften Spalte steht die Größe der Datei in Bytes. Danach kommt das Datum der letzten Änderung und schließlich der Dateiname.

## 6.3.2. Datei/Verzeichnis erstellen

Mit **touch dateiname** kann eine leere, 0-Bytes-große Datei erstellt werden. Ein Verzeichnis erstellt man mit **mkdir verzeichnisname**.

## 6.3.3. Verzeichnis wechseln

Das aktuelle Verzeichnis erfährt man mit **pwd** (Print Working Directory). Mit **cd neues_verzeichnis** wechselt man das aktuelle Arbeitsverzeichnis.

## 6.3.4. Datei/Verzeichnis löschen

Mit **rm dateiname** wird eine Datei gelöscht. Ein leeres Verzeichnis kann man mit **rmdir verzeichnis** löschen. Ist das Verzeichnis noch nicht leer, kann das Verzeichnis mit dem ganzen Inhalt gelöscht werden: **rm -r verzeichnis**.

*Hinweis*: Gelöschte Dateien sind in der Regel nicht mehr wiederherstellbar!

## 6.3.5. Dateien kopieren

Mit **cp quelle ziel** kann man Dateien kopieren. Der Befehl funktioniert wie der DOS-Befehl "copy". Wird als Quelle eine Datei angegeben, so kann das Ziel ebenfalls eine Datei sein oder ein Verzeichnis, in das die Datei hineinkopiert werden soll. Ist die Quelle ein Verzeichnis, so kann das Ziel auch nur ein Verzeichnis sein, in welches dann das Quellverzeichnis hineinkopiert wird.

*Hinweis*: Vorhandene Dateien werden ohne Nachfrage überschrieben!

## 6.3.6. Dateien verschieben

Mit **mv quelle ziel** kann man Dateien verschieben. Der Befehl funktioniert wie der DOS-Befehl "move". Quelle und Ziel können wie beim "cp"-Befehl entweder Dateien sein oder Verzeichnisse.

Auch hier werden vorhandene Dateien ohne Nachfrage überschrieben.

## 6.3.7. Textdateien anzeigen

Mit **less textdatei** kann man eine Textdatei (z.B. ein Konfigurationsdatei) bequem anschauen. Mit den Cursor-Tasten scrollt man durch den Text. Mit einem Tastendruck auf "q" verlässt man den Datei-Anzeiger.

# 6.4. Benutzerverwaltung

Die Benutzer unter Linux werden in einer Datenbank verwaltet. Diese Datenbank besteht aus mehreren Textdateien: In `/etc/passwd` stehen die Benutzer, deren Homeverzeichnisse, die jeweilige Shell. Die dazugehörigen Passwörter stehen in der Datei `/etc/shadow`. Sie sind verschlüsselt. Mit dem Befehl **mkpasswd** lassen sich solche verschlüsselten Passwörter von Hand herstellen. In `/etc/group` stehen die Gruppen, die es gibt. In `/etc/gshadow` stehen Passwörter für die einzelnen Gruppen; diese Datei wird in der Regel nicht benötigt. Mit dem Befehl **vipw** lässt sich `/etc/passwd` bearbeiten. Mit **vigr** lässt sich `/etc/group` bearbeiten. Allerdings muss man dazu das Dateiformat kennen.

## 6.4.1. Benutzer erstellen

Um einen neuen Benutzer anzulegen, verwendet man den Befehl **useradd. useradd neuer_user** legt beispielsweise den Benutzer *neuer_user* an. **useradd --help** gibt Auskunft über die verschiedenen Argumente, die man dem Befehl geben kann. Nachdem der Benutzer mit diesem Befehl angelegt wurde, muss noch das Homeverzeichnis unterhalb von `/home/` erstellt werden. Außerdem sollte mit **passwd neuer_user** noch ein Passwort gesetzt werden.

## 6.4.2. Benutzer bearbeiten

Ein vorhandener Benutzer wird mit **usermod** bearbeitet. Mit **usermod -s /bin/bash user** erhält der Benutzer user als Shell `/bin/bash`. **usermod --help** gibt auch hier wieder Auskunft über die verschiedenen Argumente.

## 6.4.3. Benutzer löschen

Mit **userdel user** wird der Benutzer user gelöscht. Sein Homeverzeichnis bleibt erhalten. Mit **userdel -r user** wird auch sein Homeverzeichnis gelöscht. Es gibt keine Möglichkeit, die Daten wiederherzustellen!

## 6.4.4. Gruppe erstellen

Um eine neue Gruppe anzulegen, wird der Befehl **groupadd gruppe** verwendet.

## 6.4.5. Gruppe löschen

Mit **groupdel gruppe** wird die Gruppe gruppe gelöscht.

# 6.5. Paketverwaltung

Hierzu steht schon einiges unter Abschnitt 2.8.1, „apt-get verwenden“.

# 6.6. Editor

Da unter Linux fast alle Konfigurationsdateien Textdateien sind, sollte man wenigstens einen Texteditor beherrschen. In diesem Abschnitt wird der Texteditor **vim** vorgestellt. Er ist ganz anders als ein Editor aus der Windows- bzw. DOS-Welt zu bedienen, aber vim ist sehr verbreitet.

## 6.6.1. Datei öffnen bzw. neue Datei erstellen

Um die Datei `/etc/hosts` zu bearbeiten, wird einfach **vim /etc/hosts** ausgeführt. Die Datei wird dann geöffnet. Existiert die angegebene Datei nicht, dann wird sie erstellt. So lassen sich also auch neue Dateien erstellen.

## 6.6.2. Dateien bearbeiten und speichern

Nach dem Öffnen muss man vim erstmal in den Editiermodus umschalten. Das geht einfach, indem man die Taste "i" drückt. In der untersten Zeile steht dann "-- INSERT --". Jetzt kann man wie gewohnt den Text

eingeben. Die Tasten "Pos1", "Ende", "Einfg" und "Entf" funktionieren wie erwartet. Ist man mit dem Editieren fertig, verlässt man den Editiermodus mit der "Esc"-Taste. Die letzte Zeile wird wieder leer. Jetzt muss man einfach ":w" eingeben und mit der Eingabetaste bestätigen. In der letzten Zeile wird der Befehl übrigens angezeigt. Will man den Editor beenden, gibt man ":q" und Eingabetaste ein. Speichern und gleichzeitiges Beenden geht mit ":x" und Eingabetaste.

# 6.7. Prozessmanagement

Mit **ps awx** erfährt man, welche Prozesse im Moment laufen. Außerdem wird die Prozess-ID (PID) angezeigt. Mit **kill $PID** kann man den Prozess mit der Nummer `$PID` beenden. Funktioniert das nicht, hilft vielleicht **kill -9 $PID** weiter.

# 6.8. Linux Support

Zu fast jedem Befehl gibt es unter Linux eine Handbuchseite (Manual Page). Diese Seite lässt sich mit **man befehl** ansehen. Manchmal ist auch **info befehl** aufschlussreicher.

Außerdem bietet fast jeder Befehl selber eine kleine Hilfestellung an. Diese erhält man mit dem Argument "--help" oder "-h": **befehl --help**.

Unter `/usr/share/doc/` gibt es für jedes Programmpaket ein Unterverzeichnis. In diesem Verzeichnis ist manchmal auch eine Dokumentation enthalten.

Im Internet unter http://www.tldp.org gibt es u.a. auch sogenannte HOWTOs. Das sind Kurzanleitungen, die einen ganz bestimmten Themenbereich erklären. Meistens sind es praxisnahe Anleitungen.

# Anhang A. Anhang

## A.1. Firewall-Script

Es ist unter `/usr/local/sbin/firewall` zu finden.

```
#!/bin/sh
#
# Firewall-Script für Gymnasium Münsingen
#
# /etc/init.d/firewall -> /usr/local/sbin/firewall
# /etc/rc[2345].d/S10firewall -> ../init.d/firewall
#

# Tool iptables vorhanden?
iptables=/sbin/iptables
test -x $iptables || exit 5
# ip_tables-Modul laden
modprobe ip_tables || exit 5


#####################################################################
# FUNKTIONEN
#####################################################################
function firewall_config() {
    # 1. Interfaces
    # extern -> Internet
    ext_int="ppp+"  ## nicht eth1, das wird quasi nicht verwendet...
    # intern -> Intranet
    int_int="eth0"
    # loopback
    lo_int="lo"

    # 2. IP-Adressen
    # server externe adresse
    server_ext="0/0"  # ist egal, s.o.: auf ppp+ kommen nur pakete an,
                      # die für den comp. bestimmt sind
    # server interne adresse
    server_int="192.168.0.1"
    # intranet adress-bereich
    lan="192.168.0.0/24"

    # 3. Ports
    # alle privilegierten ports
    priv_ports="0:1023"
    # alle unprivilegierten ports
    unpriv_ports="1024:65535"
}

function firewall_init() {
    # Alle regeln und chains löschen
    $iptables -F
    $iptables -t nat -F
    $iptables -X

    # default policy
    $iptables -P INPUT DROP
    $iptables -P FORWARD DROP
    $iptables -P OUTPUT ACCEPT

    # loopback interface aktivieren
    $iptables -A INPUT -i $lo_int -j ACCEPT

    # Schutz vor SYN Flooding
    for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
        echo "1" > $f
```

```
    done

    # IP Forwarding deaktivieren
    echo "0" > /proc/sys/net/ipv4/ip_forward

    # chains
    $iptables -N intranet
    $iptables -N internet


    # spezielle regeln:

    # spezielles dhcp-broadcast-paket
    $iptables -A INPUT -i $int_int -d 255.255.255.255 -p udp --dport 67 \
        -j ACCEPT


    # spezielles smb-broadcast-paket...nötig??
    $iptables -A INPUT -i $int_int -d 192.168.0.255 -p udp --sport 138 \
        --dport 138 -j ACCEPT
    $iptables -A INPUT -i $int_int -d 192.168.0.255 -p udp --sport 137 \
        --dport 137 -j ACCEPT
    $iptables -A INPUT -i $int_int -d 192.168.0.255 -p udp --dport 138 \
        -j ACCEPT
    $iptables -A INPUT -i $int_int -d 255.255.255.255 -p udp --dport 138 \
        --sport 138 -j ACCEPT

    # allgemeine regeln mit userdefined chains
    $iptables -A INPUT -i $int_int -s $lan -d $server_int -j intranet
    $iptables -A INPUT -i $ext_int -j internet

    # erlaube ankommende Pakete von bestehenden Verbindungen
    $iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
    $iptables -A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT

    # logging:
    $iptables -A INPUT -j LOG --log-prefix "firewall: " -m limit
}

function firewall_intranet () {
    #leere chain intranet
    $iptables -F intranet

    # icmp pakete in
    # see /usr/include/netinet/ip_icmp.h
    $iptables -A intranet -p icmp --icmp-type  0 -j ACCEPT
    $iptables -A intranet -p icmp --icmp-type  3 -j ACCEPT
    $iptables -A intranet -p icmp --icmp-type  4 -j ACCEPT
    $iptables -A intranet -p icmp --icmp-type  8 -j ACCEPT
    $iptables -A intranet -p icmp --icmp-type 12 -j ACCEPT
    # erlaube ssh (tcp,1222) in
    $iptables -A intranet -p tcp --sport $unpriv_ports --dport 1222 -j ACCEPT
    # erlaube smtp (tcp,25) in
    $iptables -A intranet -p tcp --sport $unpriv_ports --dport 25 -j ACCEPT
    # erlaube domain (udp,53) in
    $iptables -A intranet -p udp --sport $unpriv_ports --dport 53 -j ACCEPT
    # erlaube dhcp (udp,67/68) in
    $iptables -A intranet -p udp --sport $unpriv_ports --dport 67 -j ACCEPT
    $iptables -A intranet -p udp --sport 68 --dport 67 -j ACCEPT
    # erlaube httpd (tcp,80) in
    $iptables -A intranet -p tcp --sport $unpriv_ports --dport 80 -j ACCEPT
    # erlaube pop3 (tcp,110) in
    $iptables -A intranet -p tcp --sport $unpriv_ports --dport 110 -j ACCEPT
    # erlaube smb (tcp/udp,137:139) in
    $iptables -A intranet -p udp --sport 137 --dport 137 -j ACCEPT
    $iptables -A intranet -p udp --sport 138 --dport 138 -j ACCEPT
    $iptables -A intranet -p tcp --sport $unpriv_ports --dport 139 -j ACCEPT
    # erlaube swat (tcp/901) in
    $iptables -A intranet -p tcp --sport $unpriv_ports --dport 901 -j ACCEPT
```

```
    # reject smb (tcp/445) in
    $iptables -A intranet -p tcp --dport 445 -j REJECT
    # webmin (tcp/10000) in
    $iptables -A intranet -p tcp --dport 10000 -j ACCEPT


    ### proxy - internet-freigabe
    $iptables -A intranet -p tcp --sport $unpriv_ports --dport 3128 -j ACCEPT
    # proxy-manager aufrufen
    /usr/local/sbin/proxy-manager -d
    # transparent proxy (nur für http)
    $iptables -t nat -A PREROUTING -i eth0 -p tcp -d ! $server_int --dport 80 \
        -j REDIRECT --to-port 3128
}

function firewall_internet () {
    # leere chain internet
    $iptables -F internet

    # richtige adresse? -- prüfung entfällt (s.o.), da nur pakete ankommen,
    #                      die auch für dem comp. bestimmt sind.
    #$iptables -A internet -d ! $server_ext -j RETURN

    ### provided services! in
    # icmp pakete in
    $iptables -A internet -p icmp -d $server_ext --icmp-type  0 -j ACCEPT
    $iptables -A internet -p icmp -d $server_ext --icmp-type  3 -j ACCEPT
    $iptables -A internet -p icmp -d $server_ext --icmp-type  4 -j ACCEPT
    $iptables -A internet -p icmp -d $server_ext --icmp-type  8 -j ACCEPT
    $iptables -A internet -p icmp -d $server_ext --icmp-type 12 -j ACCEPT
    # erlaube ssh (tcp,1222) in
    $iptables -A internet -p tcp -d $server_ext --dport 1222 --sport \
        $unpriv_ports -j ACCEPT
    # erlaube http (tcp,80) in
    #$iptables -A internet -p tcp -d $server_ext --dport 80 --sport \
        $unpriv_ports -j ACCEPT
}

function firewall_stop () {
    # proxy-manager beenden
    /usr/local/sbin/proxy-manager -k

    # alle chains leeren
    $iptables -F
    $iptables -t nat -F
    # alle userdefined chains löschen
    $iptables -X
    # default policies
    $iptables -P INPUT ACCEPT
    $iptables -P OUTPUT ACCEPT
    $iptables -P FORWARD ACCEPT
}
#####################################################################
runfile=/var/run/firewall



case "$1" in
    start)
        if [ -e $runfile ]; then
            echo "Firewall already running..."
            echo "see: $runfile"
            exit 1;
        fi

        echo "Starting firewall..."
        firewall_config
        echo "config: ext_int: $ext_int"
        echo "        int_int: $int_int"
```

```
        echo "        server_ext: $server_ext"
        echo "        server_int: $server_int"
        firewall_init
        firewall_intranet
        firewall_internet
        touch $runfile
        echo "...done"
        ;;
    stop)
        if [ ! -e $runfile ]; then
            echo "Firewall is not running..."
            exit 1;
        fi

        echo -n "Stopping firewall..."
        firewall_stop
        rm $runfile
        echo "done"
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    status)
        /usr/local/sbin/proxy-manager -t
        echo -n "Checking for firewall..."
        if [ -e $runfile ]; then
          echo "running"
        else
          echo "not running!"
        fi

        ;;
    *)
        echo "Usage: $0 {start|stop|status|restart}"
        exit 1
        ;;
esac

exit 0
```

# A.2. proxy-manager

Der Proxy-Manager verwaltet die Internet-Freigaben und wird vom Firewall-Script aus gestartet.

Dateiname: `/usr/local/sbin/proxy-manager`.

```
#!/usr/bin/perl -w
#
# /usr/local/sbin/proxy-manager
#
# proxy-manager Programm zum Verwalten der Internetfreigaben
#
# (c) 2001 Thomas Bleher <ThomasBleher@gmx.de>
# (c) 2002,2003 Andreas Dangel <adabolo@adabolo.de>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
```

```
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
#


use Fcntl;   # for sysopen, file modes
use Fcntl qw(:flock);
use strict;

## Konfiguration
my $version = "1.2 (2003-08-21)";
my $pidfile = '/var/run/proxy-manager.pid';
my $logfile = '/var/log/internet-freigabe';
my $iptables = '/sbin/iptables';
my $actives = '/var/state/internet-freigabe';
my $lockfile = '/var/lock/proxy-manager.lck';
my $lockfile2 = '/var/lock/proxy-manager2.lck';
my $pid;


########################
##### SUBPROCEDURES #####
########################
sub usage() {
    print <<EOF;
Syntax: $0 [-hdocak] [login] [ip-adresse] [zeitsekunden]

    -h   Dieser Hilfetext
    -d   der eigentliche proxy-manager wird im Hintergrund gestartet
    -o   öffnet eine Verbindung für "ip-adresse" und schließt sie
         automatisch, wenn "zeitsekunden" erreicht sind
    -c   schließt die Verbindung für "ip-adresse" manuell
    -a   prüft, ob für "ip-adresse" eine Verbindung besteht
    -k   beendet den proxy-manager und löscht alle Verbindungen
    -t   tested, ob der proxy-manager im Hintergrund läuft
         (siehe /var/run/proxy-manager.pid).

    [login]
         Der Benutzer, der eine Verbindung öffnet, etc. wird geloggt.

    [ip-adresse]
         Jede beliebige Form ist erlaubt:
         Beispiel: 192.168.0.168
         Es kann auch 'all' verwendet werden.

    [zeitsekunden]
         Anzahl der vergangenen Sekunden seit seit 00:00:00, Jan 1, 1970.
         Zu ermitteln z.B. durch "date +%s".

    Hinweise:
       Falls Fehler beim Start des proxy-managers auftreten
       sollten: die Zugriffsrechte überprüfen:
       $logfile
       $actives


    proxy-manager $version
    (c) 2001 Thomas Bleher <ThomasBleher\@gmx.de>
    (c) 2002,2003 Andreas Dangel <adabolo\@adabolo.de>

    proxy-manager comes with ABSOLUTELY NO WARRANTY.
    This is free software, and you are welcome to redistribute it
    under certain conditions; see the GNU GPL.
EOF

    exit;
}
```

```
sub log() {
    my $data = shift;
    chomp($data);
    my $datum = localtime(time);
    open(S, ">>$lockfile");
    flock(S, LOCK_EX);
    open(LOG, ">>$logfile");
    print LOG "$datum: $data\n";
    close(LOG);
    close(S);
}

sub open_proxy() {
    &log("open_proxy()");
}

sub close_proxy() {
    &log("close_proxy()");
}

sub kill_proxymanager() {
    &close_proxy();
    unlink $actives;
    system("touch $actives");

    unlink($pidfile);

    exit;
}

sub open_connection() {
    my $login;
    my $ip;
    my $time;
    if (not (($login, $ip, $time) = @_)) {
        &log("open_connection(): missing arguments!");
        return;
    }
    if ($ip =~ /[^\w.]/) {
        &log("open_connection(): invalid argument! $ip");
        return;
    }

    &log("open_connection(): $login: $ip, $time");

    &add_actives($ip, $time);
}

sub active_connection() {
    my $ip;
    if (not ($ip = shift)) {
        &log("active_connection(): missing arguments!");
        return;
    }
    if ($ip =~ /[^\w.]/) {
        &log("active_connection(): invalid argument! $ip");
        return;
    }


    return &find_actives($ip);
}

sub close_connection() {
    my $ip;
    if (not ($ip = shift)) {
        &log("close_connection(): missing arguments!");
        return;
    }
```

```
    if ($ip =~ /[^\w.]/) {
        &log("close_connection(): invalid argument! $ip");
        return;
    }

    &log("close_connection() for $ip");

    &remove_actives($ip);
}

sub check_daemon() {
    if (-e $pidfile) {
        return 0;
    }
    return 1;
}

sub start_daemon() {
    if (not defined($pid = fork)) {
        die "Couldn't fork!";
    }

    if ($pid) {  #parent
        exit;
    } else {     #child
        sysopen(PIDFILE, $pidfile, O_WRONLY | O_CREAT | O_EXCL)
            || die "Instance of proxy-manager already running! See $pidfile";
        print PIDFILE $$;
        close PIDFILE;

        while(1) {
            sleep(60);
            my $time = time;
            &remove_actives_by_time($time);
        }
    }
}

sub add_actives() {
    my $ip = shift;
    my $time = shift;

    open(S, ">>$lockfile2");
    flock(S, LOCK_EX);
    open(FILE, "<$actives");
    my @data = <FILE>;
    close(FILE);

    push @data, "$ip:$time";

    open(FILE, ">$actives");
    print FILE join("\n", @data);
    close(FILE);

    close(S);

}

sub find_actives() {
    my $ip = shift;

    #&log("find_actives() for $ip");

    open(S, ">>$lockfile2");
    flock(S, LOCK_EX);
    open(FILE, "<$actives");
    my @data = <FILE>;
    close(FILE);
```

```
    my @found = grep(/$ip/, @data);
    my @found2 = grep(/all/, @data);
    push(@found, @found2);
    close(S);

    if (@found != 0) {
      my @fields = split(/:/, $found[0]);
      return $fields[0];
    }

    return;
}

sub remove_actives() {
    my $ip = shift;

    open(S, ">>$lockfile2");
    flock(S, LOCK_EX);
    open(FILE, "<$actives");
    my @data = <FILE>;
    close(FILE);

    my @data_new = grep(!/$ip/, @data);

    open(FILE, ">$actives");
    print FILE join("\n", @data_new);
    close(FILE);
    close(S);
}

sub remove_actives_by_time() {
    my $time = shift;

    open(S, ">>$lockfile2");
    flock(S, LOCK_EX);
    open(FILE, "<$actives");
    my @data = <FILE>;
    close(FILE);

    my @data_new;

    my $line;
    while ($line = shift @data) {
      chomp($line);
      my @fields = split(/:/, $line);
      my $time2 = $fields[1];
      if ($time >= $time2) {
        # nicht mehr hinzufügen
        &log("remove_actives_by_time: $fields[0]");
      } else {
        push(@data_new, $line);
      }
    }

    open(FILE, ">$actives");
    print FILE join("\n",@data_new);
    close(FILE);

    close(S);
}

sub handler {
  my $sig = shift;
  &log("Caught SIG$sig...");
  &kill_proxymanager();
  exit 0;
}

$SIG{'TERM'} = \&handler;
```

```
########################
##### MAIN PROGRAM #####
########################

# Argumente prüfen
if (@ARGV < 1) {
    &usage();
}

if ($ARGV[0] eq "-h") {
    &usage();
}



if ($ARGV[0] eq "-d") {
    die "Instance of proxy-manager already running! see $pidfile"
        if (&check_daemon() == 0);

    &open_proxy();
    &start_daemon();

} elsif ($ARGV[0] eq "-o") {
    die "missing arguments! try \"$0 -h\"" if (@ARGV != 4);

    if (&check_daemon() == 1) {
        die "no instance of proxy-manager running! see \"$0 -h\"";
    }
    &open_connection($ARGV[1], $ARGV[2], $ARGV[3]);

} elsif ($ARGV[0] eq "-c") {
    die "missing arguments! try \"$0 -h\"" if (@ARGV != 2);

    if (&check_daemon() == 1) {
        die "no instance of proxy-manager running! see \"$0 -h\"";
    }
    &close_connection($ARGV[1]);

} elsif ($ARGV[0] eq "-a") {
    die "missing arguments! try \"$0 -h\"" if (@ARGV != 2);

    if (&check_daemon() == 1) {
        die "no instance of proxy-manager running! see \"$0 -h\"";
    }
    if (my $ret = &active_connection($ARGV[1])) {
        print $ret;
    }

} elsif ($ARGV[0] eq "-k") {
    if (&check_daemon() == 1) {
        die "no instance of proxy-manager running! see \"$0 -h\"";
    }

    # sending SIGTERM(15)
    my $opid = `cat $pidfile`;
    kill 15, $opid;

    #&kill_proxymanager();

} elsif ($ARGV[0] eq "-t") {
    if (&check_daemon() == 0) {
        print "Der proxy-manager läuft...\n";
        exit 0;
    } else {
        print "Der proxy-manager läuft nicht...\n";
        exit 1;
    }
```

```
} else {
    &usage();
}

exit;
```

## A.2.1. squid-redirector

Dateiname: `/usr/local/sbin/squid-redirector`.

```
#!/usr/bin/perl
use FileHandle;
use IPC::Open2;
use Fcntl qw(:flock);
$|=1;

## Konfiguration
my $squidGuard = "/usr/bin/squidGuard";
my $proxymanager = "/usr/local/sbin/proxy-manager";
my $redirecturl = "http://server/cgi-bin/inactive.cgi";
my $logfile = "/var/log/squid/squid-redirector";

my $pid = open2(*Reader, *Writer, $squidGuard);

&log("[$$,$pid] started");

while (<>) {
    my $request = $_;
    my @fields = split / /, $request;
    my @fields2 = split /\//, $fields[1];
    my $ip = $fields2[0];

    chomp($request);
    chomp($ip);

    #&log("request: $request ($ip)");

    print &check_ip($ip, $request) . "\n";
}

&log("[$$,$pid] stopped");

exit 0;

sub log {
  my $msg = shift;
  my $date = localtime(time);
  chomp($msg);
  system("echo \"$date $msg\" >> $logfile");
}

sub check_ip {
  my $ip = shift;
  my $request = shift;

  if ($ip eq "127.0.0.1") {
      return;
  }

  my $active = `$proxymanager -a $ip`;

  #&log("proxymanager: $active");

  if ($active eq "") {
      return "$redirecturl";
  }
```

```
  print Writer "$request\n";
  my $response = <Reader>;
  chomp($response);
  return $response;
}
```

## A.2.2. internet/index.cgi

Das ist das Front-End zum proxy-manager. Damit kann man das Internet freischalten usw.

Dateiname: `/usr/local/httpd/htdocs/internet/index.cgi`.

```
#!/usr/bin/perl -w
#
# /usr/local/httpd/internet/index.cgi
#
# browser-frontend für proxy-manager
#
# (c) 2001 Thomas Bleher <ThomasBleher@gmx.de>
# (c) 2002 Andreas Dangel
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
#

use strict;
use CGI;

my $q = new CGI;

my $login;
my $user;
my $lehrer;
my $ip;
my $welche;
my $dauer;
my $status1;
my $status2;
my $status3;

$login = $q->remote_user;
if (not $user = (getpwnam($login))[6]) {
    die "Unauthorized access to internet/index.cgi! Username: $login";
}
$lehrer = ((getgrnam('lehrer'))[3] =~ /\b\Q$login\E\b/);
$ip = $ENV{REMOTE_ADDR} or die "No REMOTE_ADDR!";

if ($q->param()) {

    if ($q->param('aktivieren')) {
        if ($lehrer) {
            if ($q->param('welche') eq 'alle') {
                $welche = 'all';
            } else {
                $welche = $ip;
            }
```

```
        } else {
            $welche = $ip;
        }

        $dauer = $q->param('dauer');
        $dauer += 0;  # In Zahl umwandeln
        if (($dauer < 1) or ($dauer > 180)) {
            $dauer = 45;
        }
        $dauer *= 60;
        $dauer += time;

# IPs freischalten
system("/usr/bin/sudo /usr/local/sbin/proxy-manager -o $login $welche $dauer");

    } elsif ($q->param('deaktivieren')) {
        if ($lehrer) {
            if ($q->param('welche') eq 'alle') {
                $welche = 'all';
            } else {
                $welche = $ip;
            }
        } else {
            $welche = $ip;
        }

        system("/usr/bin/sudo /usr/local/sbin/proxy-manager -c $welche");
    }
} else {
    # keine Parameter gegeben:
}


$status1 = `/usr/bin/sudo /usr/local/sbin/proxy-manager -a $ip`;
$status2 = `/usr/bin/sudo /usr/local/sbin/proxy-manager -a all`;
`/usr/bin/sudo /usr/local/sbin/firewall status > /dev/null 2>&1`;
$status3 = $?;

########################
##### AUSGABEN ########
########################

print $q->header;

if ( $status3 != "0" ) {
    print
        $q->start_html( -title=>'Internet-Einwahl - Internet aktiviert',
                        -author=>'webmaster@gm.rt.schule-bw.de'),
        $q->h1('Internet-Einwahl - Internet aktiviert'),
        "Da die Firewall deaktiviert ist, ist das Internet
         <strong>immer</strong> und für <strong>jeden</strong> Computer
         aktiviert!!";
} elsif ($status1 eq $ip) {
    print
        $q->start_html( -title=>'Internet-Einwahl - Internet aktiviert',
                        -author=>'webmaster@gm.rt.schule-bw.de'),
        $q->h1('Internet-Einwahl - Internet aktiviert'),
        $q->p, $q->startform,
        $q->em("Hallo $user, der Internetzugang ist <b>für diesen Computer
                aktiviert</b>.");
    if ($status2 eq "all") {
        print $q->br, $q->em("Außerdem ist der Internetzugang <b>für alle
                              aktiviert</b>.");
    }
    print $q->p;

    if ($status2 && $lehrer) {
        print $q->popup_menu('welche',['eigener', 'alle'], 'eigener'),
                'Computer';
```

```
    } elsif (not $lehrer) {
        print $q->popup_menu('welche',['eigener'], 'eigener'), 'Computer';
    }

    print $q->br, $q->submit(-name=>'deaktivieren',
                             -value=>'Zugang deaktivieren');
} elsif ($status2 eq "all") {
    print
        $q->start_html(-title=>'Internet-Einwahl - Internet für alle aktiviert',
                       -author=>'webmaster@gm.rt.schule-bw.de'),
        $q->h1('Internet-Einwahl - Internet für alle aktiviert'),
        $q->p, $q->startform,
        $q->em("Hallo $user, der Internetzugang ist <b>für alle
                aktiviert</b>.");

    if ($lehrer) {
        print $q->p;
        print $q->popup_menu('welche', ['alle'], 'alle'), 'Computer';
        print $q->br, $q->submit(-name=>'deaktivieren',
                                 -value=>'Zugang deaktivieren');
    }
} else {
    print
        $q->start_html( -title=>'Internet-Einwahl - Internet deaktiviert',
                        -author=>'webmaster@gm.rt.schule-bw.de'),
        $q->h1('Internet-Einwahl - Internet deaktiviert'),
        $q->p, $q->startform,
        $q->em("Hallo $user, der Internetzugang ist nicht aktiviert."),
        $q->p;

    print "<br>Dauer: ", $q->popup_menu('dauer',
                             [5,10,15,20,30,45,60,75,90,120,150,180],45),
                             "min.";
    print $q->br;
    if ($lehrer) {
        print $q->popup_menu('welche',['eigener', 'alle'], 'eigener'),
              'Computer';
    } else {
        print $q->popup_menu('welche',['eigener'], 'eigener'), 'Computer';
    }
    print $q->br,$q->submit(-name=>'aktivieren', -value=>'Zugang aktivieren');
}

print '<p><strong>Zur&uuml;ck zur <a href="../">Startseite</a></strong>';
print <<HERE;
<hr><strong>Anmerkung zum Internetzugang</strong>: Dass der Internetzugang
eingeschaltet ist bedeutet nicht notwendigerweise, dass eine
Verbindung ins Internet besteht. Vielmehr wird die Verbindung nur
aufgebaut, wenn Daten aus dem Internet angefordert werden und die
Verbindung wird nach einiger Zeit der Inaktivit&auml;t (zur Zeit 120
Sekunden) von selbst wieder abgebaut. Die Zeitangabe gibt an, wie lange
der Internetzugang maximal aktiv sein darf, bevor er zwangsweise beendet wird.
Er kann natürlich auch früher beendet werden.
HERE

print $q->endform,
      $q->end_html;
```

## A.2.3. inactive.cgi

Dieses Programm wird angezeigt, wenn das Internet nicht freigeschaltet sein sollte.

Dateiname: `/usr/local/httpd/cgi-bin/inactive.cgi`.

```
#!/usr/bin/perl

print "Expires: Fri, 31 Dec 1990 23:59:59 GMT\n";
```

```
print "Cache-Control: no-cache\n";
print "Content-type: text/html\n\n";

print "<html>\n";
print "<head>\n";
print "    <title>Internet nicht aktiviert!</title>\n";
print "</head>\n";
print "<body>\n";
print "<h1>Das Internet ist nicht aktiviert!</h1>\n";
print "<h3><a href=\"http://server/\">Zurück zur Startseite</a></h3>\n";
print "</body>\n";
print "</html>\n";
```

# A.3. mail-config/index.cgi

Dieses Programm dient dazu, die Mail-Weiterleitung zu ändern.

Dateiname: `/usr/local/httpd/htdocs/mail-config/index.cgi`.

```
#!/usr/bin/perl -wT
# (C) 2001 Thomas Bleher (ThomasBleher@gmx.de) under the GNU GPL
#
# Erlaubt Benutzern, eine Mail-Weiterleitung anzugeben
# Letzte Änderung: 19.01.2001
use strict;
use CGI;
use CGI::Carp;

# Initialization
$ENV{'PATH'} = '/bin:/usr/bin';
delete @ENV{'IFS', 'CDPATH', 'ENV', 'BASHENV'};
$ENV{'SHELL'} = '/bin/sh' if exists $ENV{'SHELL'};

my $login = ($ENV{'REMOTE_USER'} =~ /^([-\w.]+)$/)[0];
my $user = (getpwnam($login))[6]; # untainting ???

my ($alias, $qlogin, $qalias, $cmdline);

my $q = new CGI;
print $q->header, $q->start_html('eMail-Weiterleitung'),
          $q->h1('eMail-Weiterleitung'), $q->start_form;

if (defined ($alias = $q->param('alias'))) {
        if ($alias !~ /^([-\w.]+\@[-.a-zA-Z0-9]+)?$/) {
        # entweder gültige Mail-Adresse oder gar keine
                print <<HERE;
<h1 style="color:#FF0000;">Ung&uuml;ltige eMail-Adresse!</h1>
<p>Ihre eMail-Adresse ($alias) scheint ung&uuml;ltig zu sein! Sollte es sich
tats&auml;chlich um eine g&uuml;ltige Adresse handeln, setzen sie sich bitte
mit dem <a href="mailto:sysadmin\@gm.rt.schule-bw.de">Administrator</a> in
Verbindung, damit ihre Adresse manuell eingetragen werden kann!
HERE

        } else {
# construct commandline
#         $qlogin = quotemeta($login);
#         $qalias = quotemeta($alias);
#         $cmdline = 'undef $/; $_ = <>; s/^' .
#                    $qlogin .
#                    ':.*?\\\\'.$qlogin.'$/'.$qlogin.': ' .
#                    $qalias.', \\\\'.$qlogin.'/m or $_ .= "\n" .
#                    $qlogin.': '.$qalias.', \\\\'.$qlogin.'";
#       print;
#       system "/usr/bin/newaliases > /dev/null 2>&1";'; # some magic here ...

                system '/usr/bin/sudo','/usr/local/sbin/change-aliases',$login,$alias;
```

```
#                 or die 'Error while executing /usr/local/sbin/change-aliases';

                if ($alias) {
                        print <<HERE;
<p><b>Mail-Weiterleitung erfolgreich ge&auml;ndert; eMail an
<i>$login\@gm.rt.schule-bw.de</i> wird ab sofort an <i>$alias</i>
weitergeleitet.</b>
HERE
                } else {
                        print <<HERE;
<p><b>Mail-Weiterleitung erfolgreich gel&ouml;scht.</b>
HERE
                }
        }
} else {
        # bestimme aktuellen Alias
        open INPUT, '</etc/aliases';
        my $input = join '', <INPUT>;
        close INPUT;
        $alias = ($input =~
      /^\s*?$login:\s*?([-\w.]+\@[-.a-zA-Z0-9]+),\s*?\\$login\s*?$/m)[0] || '';

        print <<HERE;
<p>Hallo <b>$user</b>. Ihre aktuelle eMail-Adresse ist
<i>$login\@gm.rt.schule-bw.de</i>. Hier k&ouml;nnen sie eine zus&auml;tzliche
externe eMail-Adresse angeben (zum Beispiel bei GMX, T-Online, ...), um ihre
eMails auch au&szlig;erhalb der Schule zu empfangen. Alle eMails an ihre
Schuladresse werden automatisch an ihre externe Adresse weitergeleitet.
<p>eMail-Weiterleitung an: <input type="text" name="alias" value="$alias"
size="40">
<p><input type="submit" value="Mail-Weiterleitung ändern">
<hr><p>Bei Problemen melden sie sich bitte beim <a
href="mailto:sysadmin\@gm.rt.schule-bw.de">Systemadministrator</a>
HERE

}
print '<p>Zur&uuml;ck zur <a href="/">Startseite</a>';
print $q->end_form, $q->end_html;
__END__
```

## A.3.1. Hilfsprogramm change-aliases

Dateiname: `/usr/local/sbin/change-aliases`.

```
#!/usr/bin/perl -w -i.orig
my ($login, $alias) = @ARGV;
@ARGV = ('/etc/aliases');
my $qlogin = quotemeta $login;
my $qalias = quotemeta $alias;
if (not $alias) {
  while (<>) {
    s/^$qlogin:.*?\\$qlogin$//o;
    print;
  }
} else {
  while (<>) {
    s/^\s*?$qlogin:.*?\\$qlogin\s*?$/$login: $alias, \\$login/o and $done = 1;
    print;
    print "\n$login: $alias, \\$login" if eof and not $done;
  }
}
#system '/usr/bin/newaliases > /dev/null 2>&1';
exit(0);
```

# A.4. passwd/index.cgi

Dieses Programm dient dazu, sein eigenes Passwort zu ändern.

Dateiname: `/usr/local/httpd/htdocs/passwd/index.cgi`.

```
#!/usr/bin/perl -wT
# (C) 2001 Thomas Bleher <ThomasBleher@gmx.de> under the GNU GPL
# Zuletzt geändert am 22.01.2001

use CGI;# qw(:standard);
use CGI::Carp qw(fatalsToBrowser);
#use Crypt::Cracklib;

# Konfiguration:
$PASSWORD_STRENGTH_CHECK = 0; # soll Passwort auf Sicherheit überprüft werden?
                              # 1: Ja; 0: Nein
# Konfiguration Ende

my $q = new CGI;
my $login = ($ENV{REMOTE_USER} =~ /^([-\w.]+)$/)[0]; # untaint me
my $user = ((getpwnam($login))[6] =~ /^([- \wäöüß]+)$/i)[0]; # me too
                                    # (accents berücksichtigen?)

delete @ENV{'IFS', 'CDPATH', 'ENV', 'BASH_ENV'};
$ENV{'PATH'} = '/bin:/usr/bin:/usr/local/bin';

my %errors = (
        'difference' => 'Die beiden eingegebenen Passw&ouml;rter waren leider
                        nicht identisch! :-( ',
        'too short'         => 'Das Passwort ist leider zu kurz! Es muss mindestens 5
                        Zeichen lang sein! ',
        'too long'          => 'Das Passwort ist leider zu lang! Es darf h&ouml;chstens
                        8 Zeichen lang sein! ',
        'ungueltig'         => 'Das Paswort enthält ungültige Zeichen! Erlaubt sind die
                        Zeichen <b>a-z</b>, <b>A-Z</b>, <b>0-9</b> sowie
                        <b>#*,;:._-+$%&amp;/|?{[()]}</b>! ',
        'too easy'          => 'Das Passwort ist zu leicht erratbar! W&auml;hlen sie
                        ein schwierigeres Passwort!!! Genauer Fehler: ',
);

print $q->header,
        $q->start_html( -title=>'Passwortänderung',
                                        -author=>'webmaster@gm.rt.schule-bw.de'),
        $q->h1('Passwortänderung'),
        "Hallo Benutzer <b>$user</b> ($login)! (Wenn Sie nicht Benutzer
        <b>$user</b> sind, melden sie sich bitte beim <A
        HREF=\"mailto:webmaster\@gm.rt.schule-bw.de\">Systemadministrator</A>!)",
        $q->p, $q->startform;

if ($q->param()) { # Passwort aendern
        $pw1=$q->param('pw1');
        $pw2=$q->param('pw2');
# Error-checking
        $pw1 ne $pw2         && fehler($errors{'difference'});
        length($pw1) < 5 && fehler($errors{'too short'});
        length($pw1) > 8 && fehler($errors{'too long'});
        $pw1 =~ m@[^-\w#*,;:.+!$%&/|?{[()]}]@ && fehler($errors{'ungueltig'});
        if ($PASSWORD_STRENGTH_CHECK) {
                my $fehler = fascist_check($pw1, '/usr/lib/cracklib_dict');
                $fehler !~ /^ok$/ && fehler($errors{'too easy'}.'<b>"'.$fehler.'"</b>')
        }
```

```
# change passwd
# my $pw = ($pw1 =~ m@^([-\w#*,;:.+!\$\%&/|?{\[()\]}]+)$@)[0]; # untaint me
        my $pw = ($pw1 =~ /(.+)/)[0]; # was already checked
        $enc_passwd = crypt($pw,
                    join('',('.','/',0..9,'A'..'Z','a'..'z')[rand 64, rand 64]));
        system('sudo','/usr/sbin/usermod','-p',$enc_passwd,$login);
        fehler_bei_aenderung() if ($! or $?);

        system('sudo','/usr/local/sbin/generate_apache_auth');

        open PIPE, "|sudo smbpasswd -a -s $login > /dev/null"
        or fehler_bei_aenderung();
        print PIPE $pw,"\n",$pw,"\n" or fehler_bei_aenderung();
        close PIPE or fehler_bei_aenderung();

        print "<p><strong>Passwort erfolgreich geändert!</strong>";

        sub fehler_bei_aenderung {
                print "<p><strong>Bei der Passwortänderung ist leider etwas
                        schiefgegangen. (Fehlermeldung: $!). Bitte melden sie den Fehle
                        <a href=\"mailto:webmaster\@gm.rt.schule-bw.de\">Systemadminist
                        </a>!</strong>";
                ende();
        }
} else {
    print <<HERE;
Hier haben sie die M&ouml;glichkeit, ihr Passwort zu ändern.
Geben Sie es bitte zweimal ein, damit sichergestellt ist, dass sie sich
nicht vertippt haben.
<p><strong>Achtung:</strong> Ihr Passwort muss zwischen 5 und 8 Zeichen lang
sein und darf die Buchstaben <b>a-z</b>, <b>A-Z</b>, die Zahlen <b>0-9</b>
sowie die Zeichen <b>#*,;:._-+!\$\%\&/|?{[()]}</b> enthalten.

<p style="color:red"><strong>Wählen sie
ein Passwort, das nicht leicht zu erraten ist! Die
Sicherheit ihrer Daten hängt entscheidend von der
Sicherheit ihres Passworts ab!</strong>
<p style="color:red">Ihr Passwort sollte nicht nur Kleinbuchstaben, sondern
auch Großbuchstaben,
Zahlen und Sonderzeichen beinhalten!
<p>Passwort eingeben:<br>
HERE

    print $q->password_field(-name=>"pw1",-size=>"8",-maxlength=>"8"),$q->br;
    print $q->password_field(-name=>"pw2",-size=>"8",-maxlength=>"8"),$q->br;
    print $q->submit(-name=>"eingeben",-value=>'Passwort verändern');
    ende();
}

sub fehler {
        my $fehler = shift;
           print <<HERE;
<h1>Fehler</h1>
<p><strong style="color:#FF0000;">$fehler</strong>
<p>Geben Sie entweder die Passw&ouml;rter noch einmal ein oder gehen sie
zur&uuml;ck zur <a href="/">Startseite</a>.

<p><strong>Achtung:</strong> Ihr Passwort muss zwischen 5 und 8 Zeichen lang
sein und darf die Buchstaben <b>a-z</b>, <b>A-Z</b>, die Zahlen <b>0-9</b>
sowie die Zeichen <b>#*,;:._-+!\$\%\&/|?{[()]}</b> enthalten.

<p>
HERE
           print $q->password_field(-name=>"pw1",-size=>"8",-maxlength=>"8"),$q->br;
           print $q->password_field(-name=>"pw2",-size=>"8",-maxlength=>"8"),$q->br;
           print $q->submit(-name=>"eingeben",-value=>'Passwort verändern');
        ende();
}
```

```
sub ende {
        print '<p>Zur&uuml;ck zur <a href="/">Startseite</a>';
        print $q->endform, $q->end_html;
        exit;
}
```

# A.5. admin/adduser.cgi

Dieses Programm dient dazu, neue Benutzer anzulegen.

Dateiname: `/usr/local/httpd/htdocs/admin/adduser.cgi`.

```
#!/usr/bin/perl -w
# adduser.cgi: a utility to add users to the system
#
# Copyright (C) 1994 Ian A. Murdock <imurdock@shell.portal.com>
# Heavily modified by Thomas Bleher <thomas@gm.rt.schule-bw.de>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

use CGI;
use CGI::Carp qw(fatalsToBrowser);

my $q = new CGI;
print $q->header,$q->start_html(-title=>'Benutzereinrichtung');

if (not $q->param) {
# noch keine Daten eingegeben
    print $q->h1('Benutzereinrichtung');
    print $q->p,'Hier k&ouml;nnen Sie neue Benutzer einrichten. ';
    &erzeuge_eingabefeld();
    print <<'HERE';
<hr /><h2>Anmerkungen</h2>

<p><em>Vollst&auml;ndiger Name</em>: Hier sollte der wirkliche Name angegeben
werden, erlaubt sind alle Zeichen.</p>

<p><em>login-Name</em>: Unter diesem Namen muss sich der Benutzer am System
anmelden. Erlaubt sind die Buchstaben a-z, A-Z, die Zahlen 0-9 sowie die
Zeichen ".", "-" und "_". Erstes Zeichen muss ein Buchstabe sein.</p>

<p><em>Passwort</em>: Das Passwort ist auf 8 Zeichen beschr&auml;nkt. Aus
Gr&uuml;nden der Systemsicherheit muss es mindestens 5 Zeichen lang sein.
Voreingestellt ist "<i>passwort</i>". Erlaubt sind alle Zeichen.</p>

HERE

    print $q->end_html;

} else { # es wurden schon Daten eingegeben
```

```
    $login=$q->param('login');
    $name=$q->param('fullname');
    $passwd=$q->param('passwort');
    @rechte=$q->param('rechte');
    create_user();
    if (%fehler) {
        print
            "<h1>Fehlerhafte Eingabe!</h1>",
            "<p>Der Benutzer konnte leider nicht eingerichtet werden, da ihre ",
                       "Eingaben fehlerhaft waren. Korrigieren Sie bitte ihre Eingabe
                       "und versuchen sie es noch einmal.</p>",
                       "<hr />",
                       "<h2>Fehler:</h2>",
                       "<strong>";
                foreach (keys %fehler) {
                    print $_,"<br>\n";
                }
                print '</strong>';
                # Fehlermeldung: Format ungueltig
                erzeuge_eingabefeld();

        } else {

                print "<h1>Erfolgreiche Benutzereinrichtung!</h1>
                <p>Der Benutzer <strong>$name</strong> wurde erfolgreich
        eingerichtet!\n";

        print $q->p;

                $rechte = '';
                foreach (keys %rechte) {
                    $rechte .= "$_ ";
                }
                $rechte = 'Sch&uuml;ler' if (not $rechte);
                print <<HERE;
<h2>Daten</h2>
<table>
<tr><td>Vollst&auml;ndiger Name</td><td>$name</td></tr>
<tr><td>Login-Name</td><td>$login</td></tr>
<tr><td>Passwort</td><td>$passwd</td></tr>
<tr><td>Home-Verzeichnis</td><td>$dhome</td></tr>
<tr><td>Rechte</td><td>$rechte</td></tr>
<tr><td>e-Mail-Adresse</td><td>$login\@gm.rt.schule-bw.de</td></tr>
</table>
HERE
$url=$q->url(-relative=>1);
print "<p><a href=\"$url\">Neuen Benutzer einrichten</a><p>";
print $q->a({-href=>'/'},'Zur&uuml;ck zur Hauptseite');
print $q->end_html;
    }
}

sub erzeuge_eingabefeld {
    print $q->start_form(-action=>$ENV{SCRIPT_NAME});
    print "\n",$q->p,'<table><tr><td>Vollst&auml;ndiger Name: </td><td>',
            $q->textfield(-name=>'fullname',
            -default=>'',
                    -size=>'30',
                    -maxlength=>'60');
    print "\n</td></tr><tr><td>login-Name: </td><td>",
            $q->textfield(-name=>'login',
            -default=>'',
                    -size=>'16',
                    -maxlength=>'32');

    print "\n</td></tr><tr><td>Passwort: </td><td>",
                $q->password_field(-name=>'passwort',
                '-value'=>'passwort',
                         -size=>'8',
```

```
                        -maxlength=>'8');

    print "\n</td></tr></table><p><strong>Rechte: </strong>",$q->br,
            $q->checkbox_group(-name=>'rechte',
                        -values=>['Lehrer','Systemadministrator']);

    print "\n",$q->p,$q->submit('send','Benutzer erzeugen');
    print $q->end_form;
}




sub create_user {

use English;
use File::Copy;
$dhome="/home";

$login  = $q->param('login');
$name   = $q->param('fullname');
$passwd = $q->param('passwort');
@rechte_a = $q->param('rechte');
foreach (@rechte_a) {
    $rechte{$_}=1;
}
# Plausibilitaetspruefung:

if (length($passwd) > 8) {$fehler{'Passwort zu lang'}=1};
if (length($passwd) < 5) {$fehler{'Passwort zu kurz'}=1};
if ($login !~ /^[a-zA-Z][-a-zA-Z0-9._]+$/) {
    $fehler{'login-Name nicht g&uuml;ltig'}=1
};
if ($login eq '') {$fehler{'login-Name zu kurz'}=1};
if ($name eq '') {$fehler{'Name zu kurz'}=1};
if (length($login) > 16) {$fehler{'login-Name zu lang'}=1};
if (length($name) > 60) {$fehler{'Name zu lang'}=1};
if (getpwnam $login) {$fehler{'Benutzer existiert schon'}=1};
if (%fehler) {return()};

# And now the program begins:
# Checkboxen auswerten (in Hash umwandeln?)

#Home-Verzeichnis
if (defined $rechte{'Lehrer'})         {$dhome .= "/lehrer/";}
else         {$dhome .= "/schueler/";}

$dhome .= $login;

# Gruppenzugehörigkeit
$add_groups = $login;
add2group('users');
# $add_groups = 'users';
if (defined $rechte{'Lehrer'}) {
    add2group('lehrer');
    add2group('internet');
} else {
    add2group('schueler');
}
if (defined $rechte{'Systemadministrator'}) {
    add2group('admin')
}

$output = `sudo /usr/sbin/groupadd $login 2>&1`;
#print "Gruppe erzeugt (Output:\n$output\n)";
$enc_passwd = crypt ($passwd,
    join('',('.','/', 0..9, 'A'..'Z', 'a'..'z')[rand 64, rand 64]));
setgrent;
```

```
eval {
        local $SIG{ALRM} = sub {
        die "Gruppe konnte nicht eingerichtet werden! (Timeout)\n"
    };
        alarm 120;
        while (not defined($gid = getgrnam $login)) {endgrent;setgrent;}
    #warten, bis die group vom System erkannt wird...
    # Gefahr eines Deadlocks?
        alarm 0;
};
endgrent;
if ($@) {$fehler{"$@"}=1; return();};
#print "User existiert\n";
my $cmd = "sudo /usr/sbin/useradd -c \'$name\' -d $dhome -g $gid ".
          "-G $add_groups -p $enc_passwd -s \'/bin/false\' ".
          "-m $login 2>&1";
$output = `$cmd`;
#print "useradd-output:\n$output\n";
if ($?) {$fehler{'<i>useradd</i> funktioniert nicht richtig!'}=1; return();}
#print "Benutzer erzeugt(Output:\n$output)\n";
system("sudo mkdir -m 0755 /home/public_html/$login");
eval {
        local $SIG{ALRM} = sub {
        die "Benutzer konnte nicht eingerichtet werden! (Timeout)\n"
    };
        alarm 120;
        while (not defined($uid = getpwnam $login)) {}
        alarm 0;
};
if ($@) {$fehler{"$@"}=1; return();};
#print "Benutzer existiert\n";
system("sudo chown $uid:$gid /home/public_html/$login");
system("sudo ln -s /home/public_html/$login /$dhome/public_html");
#system("sudo mkdir /usr/local/samba/profiles/$login");
#quotas setzen
system("sudo /usr/sbin/setquota $login 51200 51200 0 0 /dev/md0");
#print "Verzeichnisse erzeugt\n";
open PIPE, "|sudo smbpasswd -s -a $login > /dev/null"
    or $fehler{"<i>smbpasswd</i> funktioniert nicht richtig ".
                "(on PIPE-Open) ($!)"}=1, return();
print PIPE $passwd,"\n",$passwd,"\n"
    or $fehler{"<i>smbpasswd</i> funktioniert nicht richtig ".
                "(on passwd-print) ($!)"}=1, return();
close PIPE;
#    or $fehler{"<i>smbpasswd</i> funktioniert nicht richtig ".
#                "(on PIPE-close) ($!)"}=1, return();
#print "Samba-Passwort\n";
system "sudo /usr/local/sbin/generate_apache_auth";
system "sudo /usr/local/sbin/make_mozilla_config.pl $dhome/Mozilla ".
        "\'$login\' \'$name\'";
#print "Rest erzeugt!";
}
sub add2group {
    $add_groups .= ','.$_[0];
}
__END__ of adduser.cgi
```

## A.5.1. make_mozilla_config.pl

Dateiname: `/usr/local/sbin/make_mozilla_config.pl`.

```
#!/usr/bin/perl -w
use File::Find;
my ($path,$login,$user) = @ARGV;
```

```
if (not $user) {$user =(getpwnam($login))[6]};
if (not $path) {die "Pfad nicht g&uuml;ltig";}
File::Find::find(\&change_files,$path);
sub change_files {
    my $file = $_;
    return unless -f $_;
    open(FILE, "<$file") or return;
    my $data = join('', <FILE>);
    close(FILE);
    my $orig = $data;
    $data =~ s/\%user\%/$user/g;
    $data =~ s/\%login\%/$login/g;
    if ($data ne $orig) {
        open FILE, ">$file" or return;
        print FILE $data;
        close FILE;
    }
}
```

# A.6. generate-apache-auth

Das Programm dient dazu, die Dateien `/etc/apache/passwd` und `/etc/apache/group` zu generieren. Diese Dateien dienen zur Authentifizierung auf dem Webserver und werden u.a. dazu gebraucht, sein eigenes Passwort zu ändern.

Dateiname: `/usr/local/sbin/generate_apache_auth`.

```
#!/usr/bin/perl -w
#
# /usr/local/sbin/generate_apache_auth
#
# WARNING: will run as root!!!
# Copyright (C) 1999 Thomas Bleher <thomas@gm.rt.schule-bw.de>
#  under the GNU GPL
#

umask(0027);
%valid_users = map {$_ => 1} split / /, (getgrnam('users'))[3];

open(DATEI, '/etc/shadow');
@passwords = <DATEI>;
close DATEI;

open DATEI, '>/etc/apache/passwd';
foreach (@passwords) {
    @user = split(/:/, $_);
    if (defined($valid_users{$user[0]})) {  ## nur normale user (Sicherheit)
        print DATEI "$user[0]:$user[1]\n";
    }
}
close DATEI;

open DATEI, '/etc/group';
@groups = <DATEI>;
close DATEI;

open DATEI, '>/etc/apache/group';
foreach (@groups) {
    @user = split(/:/, $_);
    $user[3] =~ s/,/ /g;
    $user[3] =~ s/root//g;
    print DATEI "$user[0]: $user[3]";
}
```

```
close DATEI;


$uid = 0;  # root
$gid = 33; # www-data
chown $uid, $gid, '/etc/apache/passwd', '/etc/apache/group';
```

# A.7. homepage/mirror.cgi

Das Programm dient dazu, die Homepage zu spiegeln, d.h. auf den Belwue-Server zu kopieren.

Dateiname: `/usr/local/httpd/htdocs/admin/homepage/mirror.cgi`.

Die Zugriffsrechte für das Verzeichnis `/usr/local/httpd/htdocs/admin/homepage/` sehen so aus, damit die Log-Dateien darin erstellt werden können:

```
drwxrwx--- 2 root www-data 9312 Jul 15 12:05 homepage
```

mirror.cgi:

```
#!/usr/bin/perl -w
# This is an experimental version ...
use strict;
use CGI;
use CGI::Carp;
use POSIX 'setsid';
# use Time::localtime;
my ($logfile,$index);

my $q = new CGI;
if (not $q->param('ftp-pw')) {
        print $q->header, $q->start_html('Fehler: Zu wenig Daten'),
                $q->h1('Fehler: Zu wenig Daten'),
                '<p>Tja... wenigstens das Passwort sollte ich haben ... :-(',
                '<p>Zurück zur <a href="/">Startseite</a>',
                $q->end_html;
        exit;
}
my $pw = $q->param('ftp-pw');


my ($min, $hour, $day, $month, $year) = (localtime())[1..5];
$year += 1900; # we want to be Y2K-compliant, don't we ...
$month++;
my $i = 1;
$i++ while -e ($logfile = sprintf 'log-%04u.%02u.%02u-%u.txt',
                $year,$month,$day,$i);
$index = sprintf '<a href="%s">%02u.%02u.%04u %02u.%02u Uhr</a>',
        $logfile,$day,$month,$year,$hour,$min;


# um das ganze etwas interessanter zu machen ...
my $pid;

if (not defined ($pid = fork)) {
        print $q->header, $q->start_html('Interner Fehler (Cannot fork)'),
                $q->h1('Interner Fehler (Cannot fork)'),
                '<p>Tja... da weiss ich auch nicht weiter ... :-(',
                $q->end_html;
                exit;
}
```

```
if ($pid) { # parent
        print $q->header,
        $q->start_html('Homepage-Spiegelung - &Uuml;bertragung gestartet'),
                $q->h1('Homepage-Spiegelung - &Uuml;bertragung gestartet'),
                '<p>Tja... noch etwas warten, dann ist es soweit ... :-)',
                $q->end_html;
                exit;
# print CHILD $pw, "\n";
# close CHILD; # or ...
        # hope we don't have to wait on the child ...

} else { # child
        open STDIN, '/dev/null';
        open STDOUT, ">>$logfile";
        setsid;
        open STDERR, '>&STDOUT';
        open CMD, '| /usr/bin/sudo /usr/local/sbin/mirror-hp.sh'
        or die "Cannot open pipe";
        print CMD $pw,"\n" or die "Cannot write to pipe";
        close CMD;

        system("perl -i -pe 's:<ul>:<ul>\\n<li>$index</li>:i;' index.html");
}
```

# A.7.1. homepage/index.html

Dateiname: `/usr/local/httpd/htdocs/admin/homepage/index.html`.

Diese Datei hat folgende Zugriffsrechte, damit sie vom cgi-script verändert werden kann:

```
-rw-rw---- 1 www-data www-data 15714 Jul 15 12:05 index.html
```

index.html:

```
<html>
<head>
<title>Homepage-Spiegelung</title>
</head>
<body>
<h1>Homepage-Spiegelung</h1>
<p>Hier k&ouml;nnen sie die Homepage-Spiegelung von <tt>I:\intranet</tt> auf
<a href="http://www.gm.rt.schule-bw.de">http://www.gm.rt.schule-bw.de</a>
starten. Dazu geben sie bitte unten das Passwort für den FTP-Server ein.
Die Spiegelung l&auml;uft im Hintergrund.
<form action="mirror.cgi" method="post">
<p>Passwort: <input type="password" name="ftp-pw" size="8">
<input type="submit" value="Spiegelung starten">
</form>
<p>Zur&uuml;ck zur <a href="/">Startseite</a>
<hr>
<h1>Logdateien</h1>
<p>Hier finden sie die Logdateien der &Uuml;bertragungen. Die Datei erscheint
in der Liste, sobald die &Uuml;bertragung abgeschlossen wurde.

<p>

<ul>
<li><a href="log-2004.07.15-1.txt">15.07.2004 11.51 Uhr</a></li>
<li><a href="log-2004.07.15-3.txt">15.07.2004 11.56 Uhr</a></li>
</ul>
</body>
</html>
```

## A.7.2. Hilfsprogramm mirror-hp.sh

Dateiname: /usr/local/sbin/mirror-hp.sh.

```
#!/bin/bash

# erst verbindungsaufbau erzwingen...
ping -c 1 www.belwue.de > /dev/null 2>&1

# spiegelung starten
su -c '/usr/bin/weex Belwue' - homepage
```

# A.8. usermanager3.cgi

Das ist der (neue) Benutzermanager.

Dateiname: /usr/local/httpd/htdocs/admin/usermanager3.cgi.

```
#!/usr/bin/perl -wT

#   usermanager
#
#   Copyright (c) 2001 Thomas Bleher <ThomasBleher@gmx.de>
#   Copyright (c) 2002,2003 Andreas Dangel <adabolo@adabolo.de>
#
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2 of the License, or
#   (at your option) any later version.
#
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#   GNU General Public License for more details.
#
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
#

use strict;
use CGI;
use User::pwent;
use CGI::Carp qw(fatalsToBrowser);

# ============================================================================
#                          KONFIGURATION
# ============================================================================
my $user_status_script = "sudo /usr/local/sbin/user-status3";
$ENV{"PATH"} = "/bin:/usr/bin";

# ============================================================================
#                           GLOBALE VARIABLEN
# ============================================================================
my $q;          # für CGI
my @users;      # speichert die ausgewählten Benutzer (login-names)
my $this_script;
```

```
my $now_time_string = localtime;
my $error = "";

# ============================================================================
#                              SUBS
# ============================================================================
sub main;
sub print_header;
sub print_seite0;
sub print_page_end;
sub print_page_data;
sub get_userlist;

######
main;
exit 0;
######

# ============================================================================
#                              MAIN
# ============================================================================
sub main {
  $q = new CGI;

  $this_script = $q->url();

  # wurden Daten gesendet?
  if ($q->param) {
    #&print_header("Parameter vorhanden!");
    #print $q->Dump;
    #&print_page_end;

    if (defined($q->param("user"))) {
      &print_user_data($q->param("user"));
    } elsif ($q->param("seite") == 0) {
      &print_page_data;
    } elsif ($q->param("seite") == 1) {
      if (defined($q->param("action"))) {
        if ($q->param("action") eq "lock") {
          &do_lock_users;
        } elsif ($q->param("action") eq "pw") {
          &ask_pw_users;
        } elsif ($q->param("action") eq "rights") {
          &ask_rights_users;
        } elsif ($q->param("action") eq "delete") {
          &ask_delete_users;
        }
      } else {
        $error = "Falscher Befehl!";
        &print_page_data;
      }
    } elsif ($q->param("seite") == 2) {
      &do_pw_users;
    } elsif ($q->param("seite") == 3) {
      &do_rights_users;
    } elsif ($q->param("seite") == 4) {
      &do_delete_users;
    } elsif ($q->param("seite") == 5) {
      &do_save_user_data;
    }

  } else {
    &print_seite0;
  }

}

# ============================================================================
```

```
#                              PRINT_HEADER
# ============================================================================
sub print_header {
  my $title = shift;
  print $q->header;
  print qq(<html>\n);
  print qq(<head>\n);
  print qq(  <title>$title</title>\n);
  print qq(</head>\n);
  print qq(<body>\n);
  print qq(<h1>$title</h1>\n);
}


# ============================================================================
#                               PRINT_SEITE0
# ============================================================================
sub print_seite0 {
  &print_header("Benutzermanager");

  print qq(<form method="post">\n);
  print qq(<h3>Welche Benutzer sollen angezeigt werden?</h3>\n);
  print qq(<input type="radio" name="auswahl" value="alle" checked>Alle<br>\n);
  print qq(<input type="radio" name="auswahl" value="passwort">Nur Benutzer,
               die als Passwort "passwort" haben<br>\n);
  print qq(<input type="radio" name="auswahl" value="gesperrt">Nur gesperrte
               Benutzer<br>\n);
  print qq(<input type="radio" name="auswahl" value="aktiv">Nur aktivierte
               Benutzer<br>\n);
  print qq(<input type="radio" name="auswahl" value="jahrgang">Nur bestimmter
               Abi-Jahrgang (nur Schüler):\n);
  print qq(  <input type="text" name="abijahr" value="" size="2"
               maxlength="2"><br>\n);
  print qq(<input type="radio" name="auswahl" value="gruppen">Nur bestimmte
               Gruppen:\n);
  print qq(  <input type="checkbox" name="gruppe_schueler">Schüler\n);
  print qq(  <input type="checkbox" name="gruppe_lehrer">Lehrer\n);
  print qq(  <input type="checkbox" name="gruppe_internet">Internet\n);
  print qq(  <input type="checkbox" name="gruppe_admin">Administratoren\n);
  print qq(  <input type="checkbox" name="gruppe_homepage">Homepage<br>\n);
  print qq(<input type="radio" name="auswahl" value="name">Bestimmter
               Benutzer:\n);
  print qq(  <input type="text" name="loginname" value="" size="15">\n);
  print qq(<p>\n);
  print qq(<input type="hidden" name="seite" value="0">\n);
  print qq(<input type="submit" value="Anfrage starten..."\n);
  print qq(</form>\n);
  &print_page_end;
}

# ============================================================================
#                              PRINT_PAGE_END
# ============================================================================
sub print_page_end {
  print qq(<hr>\n);
  print qq(<a href="http://server/">zur Startseite</a>\n);
  print qq( - <a href="http://server/admin/">Administration</a>\n);
  print qq( - <a href="$this_script">Benutzermanager</a>\n);
  print qq( - $now_time_string\n);
  #print qq( - <i>&copy; adabolo.de network solutions</i>\n);
  print qq(</body>\n);
  print qq(</html>\n);
}

# ============================================================================
#                               PRINT_PAGE_DATA
# ============================================================================
sub print_page_data {
  &print_header("Benutzermanager");
```

```
if ($error ne "") {
  print qq(<font color="green">$error</font>\n);
}

my @userlist = &get_userlist;
my $anzahl = @userlist;

# Benutzerliste sortieren...
my @sorted = sort {
  my $c = $a;
  my $d = $b;

  # Lehrer:
  if ($c =~ m/\A(\w)\.([\w\-]+)\Z/) { $c = "000.$2.$1." };
  if ($d =~ m/\A(\w)\.([\w\-]+)\Z/) { $d = "000.$2.$1." };
  # Sonstige:
  if ($c =~ m/\Abo\.vhs(\d\d)\Z/)   { $c = "EEE.bo.vhs.$1." };
  if ($d =~ m/\Abo\.vhs(\d\d)\Z/)   { $d = "EEE.bo.vhs.$1." };
  if ($c =~ m/\Abl\.vhs(\d+)\Z/)    { $c = "FFF.bl.vhs.$1." };
  if ($d =~ m/\Abl\.vhs(\d+)\Z/)    { $d = "FFF.bl.vhs.$1." };
  if ($c =~ m/\Akghs(\d+)\Z/)       { $c = "GGG.kghs.$1." };
  if ($d =~ m/\Akghs(\d+)\Z/)       { $d = "GGG.kghs.$1." };
  if ($c =~ m/\Abo\.kgrt(\d+)\Z/)   { $c = "HHH.bo.kgrt.$1." };
  if ($d =~ m/\Abo\.kgrt(\d+)\Z/)   { $d = "HHH.bo.kgrt.$1." };
  # Schüler
  if ($c =~ m/\A(\w)(\w+)(\d\d)\Z/) { $c = "AAA.$3.$2.$1." };
  if ($d =~ m/\A(\w)(\w+)(\d\d)\Z/) { $d = "AAA.$3.$2.$1." };

  $c cmp $d;
} @userlist;

# JavaScript Programm:
print qq(\n<script type="text/javascript" language="javascript">\n);
print qq(function mark_all() {\n);
print qq(  var b = window.document.forms[0].checker.checked;\n);
print qq(  var count = 0;\n);
print qq(  while (count < $anzahl) {\n);
print qq(    window.document.forms[0].elements[count++].checked = b;\n);
print qq(  }\n);
print qq(}\n);
print qq(</script>\n\n);

# Formular starten
print $q->start_form;
# Tabelle starten
print qq(<table border="1">\n);
print qq(<tr><th>Login</th><th>Name</th><th>Gruppen (Rechte)</th>);
print qq(<th>Quota</th><th>"passwort" (letzte Änderung)</th><th>Status</th>);
print qq(<th>Auswahl</th></tr>\n);

foreach (@sorted) {
  my $login = $_;
  my $name = &get_fullname($_);
  my $gruppen = &get_groups($_);
  my $quota = &get_quota($_);
  my $passwort = &get_password($_);
  my $status = &get_status($_);

  my $color = "#FFFFDD";

  if ($passwort =~ m/JA/) {
    $color = "#FF0000";
  }
  if ($status eq "deaktiviert") {
    $color = "#EEEEEE";
  }

  print qq(<tr bgcolor="$color">);
```

```
    print qq(<td><a href="$this_script?user=$_">$_</a></td>);
    print qq(<td>$name</td>);
    print qq(<td>$gruppen</td>);
    print qq(<td>$quota</td>);
    print qq(<td>$passwort</td>);
    print qq(<td>$status</td>);
    print qq(<td><input type="checkbox" name="users" value="$_"></td></tr>\n);
  }

  print qq(<tr><td align="right" colspan="6">Alle angezeigten
           Benutzer markieren:</td>);
  print qq(<td><input type="checkbox" name="checker" onclick="mark_all();">
           </td></tr>\n);
  print qq(</table>\n);
  print qq(Angezeigte Benutzer: $anzahl\n);
  print qq(<p>\n);

  print qq(<h2>Aktion</h2>\n);
  print qq(<input type="radio" name="action" value="lock">Benutzer
           sperren/freigeben<br>\n);
  print qq(<input type="radio" name="action" value="pw">Passwort ändern<br>\n);
  print qq(<input type="radio" name="action" value="rights">Rechte
           ändern<br>\n);
  print qq(<input type="radio" name="action" value="delete">Benutzer
           löschen<br>\n);
  print qq(<input type="hidden" name="seite" value="1">\n);

  &print_seite1_values;

  print qq(<input type="submit" value="Ausführen"> <input type="reset"
           value="Abbrechen">\n);

  print qq(</form>\n);

  &print_page_end;
}

# ============================================================================
#                              GET_USERLIST
# ============================================================================
sub get_userlist {
  my @users = split / /, (getgrnam('users'))[3];

  if ($q->param("auswahl") eq "alle") {
    return @users;
  } elsif ($q->param("auswahl") eq "passwort") {
    my @userlist = qw();
    foreach (@users) {
      # für jeden Benutzer prüfen, ob "passwort" das Passwort ist...
      # das Script gibt den Benutzernamen zurück, wenn das Passwort
      # unsicher ist. Es gibt "---secure" zurück, wenn das Passwort
      # sicher ist.
      my $ret = `$user_status_script -passwd $_`;
      if ($ret ne "---secure") {
        push(@userlist, $_);
      }
    }
    return @userlist;
  } elsif ($q->param("auswahl") eq "gesperrt") {
    my @userlist = qw();
    foreach (@users) {
      # für jeden Benutzer prüfen, ob er gesperrt ist...
      my $ret = `$user_status_script -locked $_`;
      if ($ret ne "---unlocked") {
        push(@userlist, $_);
      }
    }
    return @userlist;
  } elsif ($q->param("auswahl") eq "aktiv") {
```

```
    my @userlist = qw();
    foreach (@users) {
      # für jeden Benutzer prüfen, ob er "unlocked" ist...
      my $ret = `$user_status_script -locked $_`;
      if ($ret eq "---unlocked") {
        push(@userlist, $_);
      }
    }
    return @userlist;
  } elsif ($q->param("auswahl") eq "jahrgang") {
    # erst Abi-Jahrgang herausfinden
    my $abi = $q->param("abijahr");
    # keine zwei Ziffern?
    if ($abi !~ m/\d\d/) {
      # dann alle Benutzer anzeigen...
      return @users;
    }
    # jeden Benutzer durchgehen und in @userlist speichern
    my @userlist = qw();
    foreach (@users) {
      if ($_ =~ m/\w+$abi/) {
        push(@userlist, $_);
      }
    }
    return @userlist;
  } elsif ($q->param("auswahl") eq "gruppen") {
    # wenn gar nichts gewählt...
    if ($q->param("gruppe_schueler") ne "on" &&
        $q->param("gruppe_lehrer") ne "on" &&
        $q->param("gruppe_internet") ne "on" &&
        $q->param("gruppe_admin") ne "on" &&
        $q->param("gruppe_homepage") ne "on") {
      return @users;
    }
    my %userlist;

    my @group_users;

    if ($q->param("gruppe_schueler") eq "on") {
      @group_users = split / /, (getgrnam('schueler'))[3];
      foreach (@group_users) { $userlist{$_} = 1; }
    }
    if ($q->param("gruppe_lehrer") eq "on") {
      @group_users = split / /, (getgrnam('lehrer'))[3];
      foreach (@group_users) { $userlist{$_} = 1; }
    }
    if ($q->param("gruppe_internet") eq "on") {
      @group_users = split / /, (getgrnam('internet'))[3];
      foreach (@group_users) { $userlist{$_} = 1; }
    }
    if ($q->param("gruppe_admin") eq "on") {
      @group_users = split / /, (getgrnam('admin'))[3];
      foreach (@group_users) { $userlist{$_} = 1; }
    }
    if ($q->param("gruppe_homepage") eq "on") {
      @group_users = split / /, (getgrnam('homepage'))[3];
      foreach (@group_users) { $userlist{$_} = 1; }
    }

    return keys(%userlist);
  } elsif ($q->param("auswahl") eq "name") {
    my $login = $q->param("loginname");
    $login = quotemeta($login);
    return grep(/$login/, @users);
  } else {
    # in sonstigen Fällen, die eigentlich nicht auftreten sollten:
    # Alle Benutzer anzeigen...
    return @users;
  }
```

```
}

# ============================================================================
#                        GET_FULLNAME
# ============================================================================
sub get_fullname {
  my $login = shift;
  my $user = getpwnam($login);
  return $user->gecos;
}

# ============================================================================
#                        GET_HOMEDIR
# ============================================================================
sub get_homedir {
  my $login = shift;
  my $user = getpwnam($login);
  return $user->dir;
}

# ============================================================================
#                         GET_SHELL
# ============================================================================
sub get_shell {
  my $login = shift;
  my $user = getpwnam($login);
  return $user->shell;
}

# ============================================================================
#                         GET_GROUPS
# ============================================================================
sub get_groups {
  my $login = shift;

  # untaint
  $login =~ m/([\w\.\-]+)/;
  $login = $1;

  my $groups = `id -nG $login`;
  return $groups;
}

# ============================================================================
#                          GET_PASSWORD
# ============================================================================
sub get_password {
  my $login = shift;
  my $ret = `$user_status_script -passwd $login`;
  my $answer = "";
  if ($ret eq "---secure") {
    $answer .= "nein ";
  } else {
    $answer .= "JA ";
  }

  $answer .= "(" . `$user_status_script -change $login` . ")";

  return $answer;
}

# ============================================================================
#                             GET_STATUS
# ============================================================================
sub get_status {
  my $login = shift;

  # untaint
  $login =~ m/([\w\.\-]+)/;
```

```
  $login = $1;

  my $ret = `$user_status_script -locked $login`;
  if ($ret eq "---unlocked") {
    return "aktiviert";
  } else {
    return "deaktiviert";
  }
}

# ============================================================================
#                            GET_QUOTA
# ============================================================================
sub get_quota {
  my $login = shift;

  # untaint
  $login =~ m/([\w\.\-]+)/;
  $login = $1;

  my $ret = `$user_status_script -quota $login`;
  return $ret;
}


# ============================================================================
#                             PRINT_USER_DATA
# ============================================================================
sub print_user_data {
  my $login = shift;

  # untaint
  $login =~ m/([\w\.\-]+)/;
  $login = $1;

  &print_header("Benutzermanager - $login");

  my $fullname = &get_fullname($login);
  my $homedir = &get_homedir($login);
  my $shell = &get_shell($login);
  my $rights = &get_groups($login);
  my $quota = &get_quota($login);
  $quota =~ m/(\d+)\/(\d+)/;
  my $quota_used = $1;
  my $quota_limit = $2;
  my $status = &get_status($login);
  my $change = `$user_status_script -change $login`;



  print $q->start_form;
  print qq(<table border="0">\n);
  print qq(<tr><td><b>Login</b></td><td>$login</td></tr>\n);
  print qq(<tr><td><b>Homeverzeichnis</b></td><td>$homedir</td></tr>\n);
  print qq(<tr><td><b>Shell</b></td><td>$shell</td></tr>\n);
  print qq(<tr><td><b>Name</b></td><td><input type="text" name="fullname"
           value="$fullname"></td></tr>\n);


  print qq(<tr><td><b>Gruppen</b></td><td>);
  print qq(<input type="checkbox" name="rights" value="schueler");
  print qq( checked) if ($rights =~ m/schueler/);
  print qq(>Schüler );
  print qq(<input type="checkbox" name="rights" value="lehrer");
  print qq( checked) if ($rights =~ m/lehrer/);
  print qq(>Lehrer );
  print qq(<input type="checkbox" name="rights" value="internet");
  print qq( checked) if ($rights =~ m/internet/);
  print qq(>Internet );
```

```
  print qq(<input type="checkbox" name="rights" value="admin");
  print qq( checked) if ($rights =~ m/admin/);
  print qq(>Administratoren );
  print qq(<input type="checkbox" name="rights" value="homepage");
  print qq( checked) if ($rights =~ m/homepage/);
  print qq(>Homepage );
  print qq(</td></tr>\n);

  print qq(<tr><td><b>Quota</b></td><td>);
  print qq($quota_used / <input type="text" size="4" value="$quota_limit"
           name="quota"> MB);
  print qq(</td></tr>\n);

  print qq(<tr><td><b>Status</b></td><td>);
  print qq(<input type="radio" name="lock" value="gesperrt");
  print qq( checked) if ($status eq "deaktiviert");
  print qq(>gesperrt );
  print qq(<input type="radio" name="lock" value="freigegeben");
  print qq( checked) if ($status eq "aktiviert");
  print qq(>freigegeben );
  print qq(</td></tr>\n);

  print qq(<tr><td><b>Passwort</b></td><td>);
  print qq(<input type="password" name="passwort" value="">);
  print qq( \(Letzte Änderung: $change\));
  print qq(</td></tr>\n);

  print qq(</table>\n);

  print qq(<input type="submit" name="save" value="speichern">\n);
  print qq(<input type="submit" name="abort" value="abbrechen">\n);

  print qq(<input type="hidden" name="edit_user" value="$login">\n);
  print qq(<input type="hidden" name="seite" value="5">\n);
  #&print_seite1_values;
  print qq(<input type="hidden" name="auswahl" value="name">\n);
  print qq(<input type="hidden" name="loginname" value="$login">\n);

  print qq(</form>\n);

  &print_page_end;
}

# ============================================================================
#                          DO_LOCK_USERS
# ============================================================================
sub do_lock_users {
  my @users = qw();
  if (defined($q->param("users"))) {
    @users = $q->param("users");
  } else {
    $error = "<strong>Fehler:</strong> Keine Benutzer ausgewählt!";
    return;
  }

  $error = "Folgende Benutzer wurden gesperrt bzw. freigegeben:<br>\n<b>";

  foreach (@users) {
    my $login = $_;

    # untaint
    $login =~ m/([\w\.\-]+)/;
    $login = $1;

    my $status = &get_status($login);
    if ($status eq "aktiviert") {
      # sperre Benutzer $login
      `sudo /usr/bin/passwd -l $login`;
      fehler("Benutzer sperren (passwd) [$login]") if ($? != 0);
```

```
      `sudo /usr/bin/smbpasswd -d $login`;
      fehler("Benutzer sperren (smbpasswd) [$login]") if ($? != 0);
    } else {
      # Gebe Benutzer $login frei
      `sudo /usr/bin/passwd -u $login`;
      fehler("Benutzer freigeben (passwd) [$login]") if ($? != 0);
      `sudo /usr/bin/smbpasswd -e $login`;
      fehler("Benutzer freigeben (smbpasswd) [$login]") if ($? != 0);
    }

    $error .= "$_ ";
  }

  $error .= "</b>";

  `sudo /usr/local/sbin/generate_apache_auth`;


  &print_page_data;
}

# ============================================================================
#                       ASK_PW_USERS
# ============================================================================
sub ask_pw_users {
  my @users = qw();
  if (defined($q->param("users"))) {
    @users = $q->param("users");
  } else {
    $error = "<strong>Fehler:</strong> Keine Benutzer ausgewählt!";
    &print_page_data;
    return;
  }

  &print_header("Benutzermanager - Passwort ändern");

  print qq(Von folgenden Benutzern soll das Passwort geändert werden: <b>);
  print join(" ", @users);
  print qq(</b>\n<p>\n);

  print $q->start_form;
  print qq(Bitte geben Sie das neue Passwort ein:\n);
  print qq(<input type="password" name="password" size="8" maxlength="8"
           value="passwort">\n);

  print qq(<input type="hidden" name="seite" value="2">\n);
  foreach (@users) {
    print qq(<input type="hidden" name="users" value="$_">\n);
  }
  &print_seite1_values;

  print qq(<p>);
  print qq(<input type="submit" name="do_it" value="Passwort ändern">\n);
  print qq(<input type="submit" name="abort" value="Abbrechen">\n);
  print qq(</form>\n);

  &print_page_end;

}

# ============================================================================
#           PRINT_SEITE1_VALUES
# ============================================================================
sub print_seite1_values {
  print qq(<input type="hidden" name="auswahl" value=");
  print $q->param("auswahl");
  print qq(">\n);
  print qq(<input type="hidden" name="loginname" value=");
  print $q->param("loginname");
```

```
  print qq(">\n);
  print qq(<input type="hidden" name="abijahr" value=");
  print $q->param("abijahr");
  print qq(">\n);
  print qq(<input type="hidden" name="gruppe_schueler" value=");
  print $q->param("gruppe_schueler");
  print qq(">\n);
  print qq(<input type="hidden" name="gruppe_lehrer" value=");
  print $q->param("gruppe_lehrer");
  print qq(">\n);
  print qq(<input type="hidden" name="gruppe_internet" value=");
  print $q->param("gruppe_internet");
  print qq(">\n);
  print qq(<input type="hidden" name="gruppe_admin" value=");
  print $q->param("gruppe_admin");
  print qq(">\n);
  print qq(<input type="hidden" name="gruppe_homepage" value=");
  print $q->param("gruppe_homepage");
  print qq(">\n);
}

# ============================================================================
#                       DO_PW_USERS
# ============================================================================
sub do_pw_users {
  my @users = $q->param("users");
  my $password = $q->param("password");

  if (defined($q->param("abort"))) {
    &print_page_data;
    return;
  }

  #untaint
  $password =~ m/([-\w#*,;:.+!$%&\/|?{\[()\]}]+)/;
  $password = $1;

  my $enc_pw = crypt($password,
            join("", (".", "/", 0..9, "a".."z", "A".."Z")[rand 64, rand 64]));

  $error = "Von folgenden Benutzern wurde das Passwort geändert:<br>\n<b>";

  foreach (@users) {
    my $login = $_;

    #untaint
    $login =~ m/([\w\.\-]+)/;
    $login = $1;

    `sudo /usr/sbin/usermod -p $enc_pw $login`;
    fehler("Passwortänderung (usermod) [$login]") if ($? != 0);

    open PIPE, "| sudo /usr/bin/smbpasswd -s $login > /dev/null"
        or fehler("Passwortänderung (pipe open) [$login]");
    print PIPE "$password\n$password\n"
        or fehler("Passwortänderung (pipe write) [$login]");
    close PIPE
        or fehler("Passwortänderung (pipe close) [$login]");

    $error .= "$login ";
  }

  $error .= "</b>";

  `sudo /usr/local/sbin/generate_apache_auth`;

  &print_page_data;
}
```

```
# ============================================================================
#                       ASK_RIGHTS_USERS
# ============================================================================
sub ask_rights_users {
  my @users = qw();
  if (defined($q->param("users"))) {
    @users = $q->param("users");
  } else {
    $error = "<strong>Fehler:</strong> Keine Benutzer ausgewählt!";
    &print_page_data;
    return;
  }

  &print_header("Benutzermanager - Rechteänderung");

  print qq(Von folgenden Benutzern sollen die Rechte geändert werden: <b>);
  print join(" ", @users);
  print qq(</b>\n<p>\n);

  print $q->start_form;
  print qq(<h2>Rechte</h2>\n);
  print qq(<input type="checkbox" name="rights" value="schueler">Schüler<br>\n);
  print qq(<input type="checkbox" name="rights" value="lehrer">Lehrer<br>\n);
  print qq(<input type="checkbox" name="rights" value="internet">Internet
           <br>\n);
  print qq(<input type="checkbox" name="rights" value="admin">Administratoren
           <br>\n);
  print qq(<input type="checkbox" name="rights" value="homepage">Homepage
           <br>\n);
  print qq(Quota (0 = unbegrenzt): <input type="text" name="quota" value="50"
           size="4">MB<br>\n);

  print qq(<input type="hidden" name="seite" value="3">\n);
  foreach (@users) {
    print qq(<input type="hidden" name="users" value="$_">\n);
  }
  &print_seite1_values;

  print qq(<p>);
  print qq(<input type="submit" name="do_it" value="Rechte ändern">\n);
  print qq(<input type="submit" name="abort" value="Abbrechen">\n);
  print qq(</form>\n);

  &print_page_end;

}

# ============================================================================
#                        DO_RIGHTS_USERS
# ============================================================================
sub do_rights_users {
  my @users = $q->param("users");

  if (defined($q->param("abort"))) {
    $error = "Abbruch!";
    &print_page_data;
    return;
  }

  # Rechte sammeln und überprüfen
  my @new_rights;
  push(@new_rights, "users");

  my @rights = $q->param("rights");
  foreach (@rights) {
    push(@new_rights, "schueler") if ($_ eq "schueler");
    push(@new_rights, "lehrer") if ($_ eq "lehrer");
    push(@new_rights, "internet") if ($_ eq "internet");
    push(@new_rights, "admin") if ($_ eq "admin");
```

```
    push(@new_rights, "homepage") if ($_ eq "homepage");
  }

  my $rights_string = join(",", @new_rights);

  # Quota
  $q->param("quota") =~ m/(\d+)/;
  my $quota = $1 * 1024; # Anzahl der KB-Blöcke

  # Shell
  my $shell = "/bin/false";
  if ($rights_string =~ m/admin/) {
    $shell = "/bin/bash";
  }

  $error = "Von folgenden Benutzern wurden die Rechte auf <i>$rights_string</i>
            geändert:<br>\n<b>";

  foreach (@users) {
    my $login = $_;

    #untaint
    $login =~ m/([\w\.\-]+)/;
    $login = $1;

    `sudo /usr/sbin/usermod -s $shell -G $rights_string,$login $login`;
    fehler("Rechteänderung (usermod) [$login]") if ($? != 0);
    `sudo /usr/sbin/setquota $login $quota $quota 0 0 /dev/md0`;
    fehler("Rechteänderung (setquota) [$login]") if ($? != 0);

    $error .= "$login ";
  }

  $error .= "</b>\n";

  `sudo /usr/local/sbin/generate_apache_auth`;

  &print_page_data;
}

# ============================================================================
#                      ASK_DELETE_USERS
# ============================================================================
sub ask_delete_users {
  my @users = qw();
  if (defined($q->param("users"))) {
    @users = $q->param("users");
  } else {
    $error = "<strong>Fehler:</strong> Keine Benutzer ausgewählt!";
    &print_page_data;
    return;
  }

  &print_header("Benutzermanager - Benutzer löschen");

  print qq(<font color="red"><b>ACHTUNG!</b></font> Folgende Benutzer werden
                gelöscht: <b>);
  print join(" ", @users);
  print qq(</b>\n<p>\n);

  print qq(Auch die Homeverzeichnisse und alle persönlichen Daten werden
            gelöscht );
  print qq(und <strong>können nicht wiederhergestellt werden</strong>. Nur
            die );
  print qq(<tt>public_html</tt>-Verzeichnisse bleiben erhalten.<p>\n);

  print qq(Wollen Sie wirklick fortfahren?<p>\n);

  print $q->start_form;
```

```
  print qq(<input type="hidden" name="seite" value="4">\n);
  foreach (@users) {
    print qq(<input type="hidden" name="users" value="$_">\n);
  }
  &print_seite1_values;

  print qq(<p>);
  print qq(<input type="submit" name="do_it" value="Benutzer löschen!">\n);
  print qq(<input type="submit" name="abort" value="Abbrechen">\n);
  print qq(</form>\n);

  &print_page_end;

}

# ============================================================================
#                       DO_DELETE_USERS
# ============================================================================
sub do_delete_users {
  my @users = $q->param("users");

  if (defined($q->param("abort"))) {
    $error = "Abbruch!";
    &print_page_data;
    return;
  }


  $error = "Folgende Benutzer wurden gelöscht:<br>\n<b>";

  foreach (@users) {
    my $login = $_;

    #untaint
    $login =~ m/([\w\.\-]+)/;
    $login = $1;

    # Samba-User löschen
    `sudo /usr/bin/smbpasswd -x $login`;
    fehler("Benutzer löschen (smbpasswd) [$login]") if ($? != 0);
    # public_html-Verzeichnis verschieben
    `sudo /bin/mv /home/public_html/$login /usr/local/httpd/htdocs/oldhomes/`;
    fehler("Benutzer löschen (move public_html) [$login]") if ($? != 0);
   `sudo /bin/chown -hfR root:www-data /usr/local/httpd/htdocs/oldhomes/$login`;
    fehler("Benutzer löschen (chown public_html) [$login]") if ($? != 0);
    `sudo /bin/chmod -R 755 /usr/local/httpd/htdocs/oldhomes/$login`;
    fehler("Benutzer löschen (chmod public_html) [$login]") if ($? != 0);
    # Benutzer und Homeverzeichnis löschen
    `sudo /usr/sbin/userdel -r $login`;
    fehler("Benutzer löschen (userdel) [$login]") if ($? != 0);
    `sudo /usr/sbin/groupdel $login`;
    fehler("Benutzer löschen (groupdel) [$login]") if ($? != 0);

    $error .= "$login ";
  }

  $error .= "</b>\n";

  `sudo /usr/local/sbin/generate_apache_auth`;

  &print_page_data;
}

# ============================================================================
#                   FEHLER
# ============================================================================
sub fehler {
  my $text = shift;
```

```
  $error .= qq(<p><font color="red"><big><strong>Schwerwiegender Fehler:
               </strong></big> $text</font>);

  `sudo /usr/local/sbin/generate_apache_auth`;

  &print_page_data;
  exit 1;
}

# ============================================================================
#                    DO_SAVE_USER_DATA
# ============================================================================
sub do_save_user_data {
  if (defined($q->param("abort"))) {
    $error = "Abbruch!";
    &print_page_data;
    return;
  }

  my $login = $q->param("edit_user");

  # untaint
  $login =~ m/([\w\.\-]+)/;
  $login = $1;

  my $fullname;
  my $rights;
  my $quota;
  my $status;

  if (defined($q->param("fullname"))) {
    $fullname = $q->param("fullname");

    #untaint
    $fullname =~ m/([\w\.\ ,äöüÄÖÜ\-*#']+)/;
    $fullname = $1;

    `sudo /usr/sbin/usermod -c "$fullname" $login`;
    fehler("Benutzer ändern, Name (usermod) [$login]") if ($? != 0);
  }


  if (defined($q->param("rights"))) {
    # Rechte sammeln und überprüfen
    my @new_rights;
    push(@new_rights, "users");
    my @rights = $q->param("rights");
    foreach (@rights) {
      push(@new_rights, "schueler") if ($_ eq "schueler");
      push(@new_rights, "lehrer") if ($_ eq "lehrer");
      push(@new_rights, "internet") if ($_ eq "internet");
      push(@new_rights, "admin") if ($_ eq "admin");
      push(@new_rights, "homepage") if ($_ eq "homepage");
    }

    my $rights_string = join(",", @new_rights);

    my $shell = "/bin/false";
    if ($rights_string =~ m/admin/) { $shell = "/bin/bash"; }

    `sudo /usr/sbin/usermod -s $shell -G $rights_string,$login $login`;
    fehler("Benutzer ändern, Rechte (usermod) [$login]") if ($? != 0);
  }

  if (defined($q->param("lock"))) {
    $status = $q->param("lock");
    if ($status eq "gesperrt") {
      # sperre Benutzer $login
      `sudo /usr/bin/passwd -l $login`;
```

```
      fehler("Benutzer sperren (passwd) [$login]") if ($? != 0);
      `sudo /usr/bin/smbpasswd -d $login`;
      fehler("Benutzer sperren (smbpasswd) [$login]") if ($? != 0);
    } else {
      # Gebe Benutzer $login frei
      `sudo /usr/bin/passwd -u $login`;
      fehler("Benutzer freigeben (passwd) [$login]") if ($? != 0);
      `sudo /usr/bin/smbpasswd -e $login`;
      fehler("Benutzer freigeben (smbpasswd) [$login]") if ($? != 0);
    }
  }

  if (defined($q->param("quota"))) {
    # Quota
    $q->param("quota") =~ m/(\d+)/;
    my $quota = $1 * 1024; # Anzahl der KB-Blöcke


    `sudo /usr/sbin/setquota $login $quota $quota 0 0 /dev/md0`;
    fehler("Benutzer ändern, Quota (setquota) [$login]") if ($? != 0);
  }


  if (defined($q->param("passwort")) && $q->param("passwort") ne "") {
    my $password = $q->param("passwort");
    #untaint
    $password =~ m/([-\w#*,;:.+!$%&\/|?{\[()\]}]+)/;
    $password = $1;

    my $enc_pw = crypt($password,
        join("", (".", "/", 0..9, "a".."z", "A".."Z")[rand 64, rand 64]));

    `sudo /usr/sbin/usermod -p $enc_pw $login`;
    fehler("Benutzer ändern, Passwort (usermod) [$login]") if ($? != 0);

    open PIPE, "| sudo /usr/bin/smbpasswd -s $login > /dev/null"
        or fehler("Benutzer ändern, Passwort (pipe open) [$login]");
    print PIPE "$password\n$password\n"
        or fehler("Benutzer ändern, Passwort (pipe write) [$login]");
    close PIPE
        or fehler("Benutzer ändern, Passwort (pipe close) [$login]");
  }

  `sudo /usr/local/sbin/generate_apache_auth`;

  $error = "Benutzer $login geändert.";

  &print_page_data;
}
```

## A.8.1. Hilfsprogramm user-status3

Dateiname: `/usr/local/sbin/user-status3`.

```
#!/usr/bin/perl -wT
use User::pwent;

# Path setzen, wegen Taint-Mode
$ENV{"PATH"} = "/bin:/usr/bin";

# Parameter holen...
my $action = shift;
my $login = shift;
```

```
# $login untainten
if (defined($login)) {
  $login =~ m/([\w\.-]+)/;
  $login = $1;
}


if (defined($action) && defined($login)) {
     if ($action eq "-passwd") { &print_passwd(); }
  elsif ($action eq "-locked") { &print_locked(); }
  elsif ($action eq "-groups") { &print_groups(); }
  elsif ($action eq "-change") { &print_change(); }
  elsif ($action eq "-quota")  { &print_quota();  }
}

print "fehlerhafte Argumente...\n";
print "user-status action login\n";
print "action: {-passwd|-locked|-groups|-change|-quota}\n";
exit 1;




sub print_passwd {
  my $user = getpwnam($login);
  if (crypt("passwort", $user->passwd) eq $user->passwd) {
    print $login;
  } else {
    print "---secure";
  }
  exit 0;
}

sub print_locked {
  my $user = getpwnam($login);
  if (substr($user->passwd, 0, 1) eq "!") {
    print $login;
  } else {
    print "---unlocked";
  }
  exit 0;
}

sub print_groups {
  my $groups = `id -nG $login`;
  print $groups;
  exit 0;
}

sub print_change {
  open(SHADOW, "</etc/shadow");
  my @data = <SHADOW>;
  close(SHADOW);
  my @user = grep(/$login/, @data);
  my $line = shift @user;
  my $age = (split(/:/, $line))[2];

  $age = $age * 24 * 60 * 60;
  my $day;
  my $mon;
  my $year;
  (undef, undef, undef, $day, $mon, $year) = localtime($age);
  $mon += 1;
  $year += 1900;

  if ($day < 10) { $day = "0$day"; }
  if ($mon < 10) { $mon = "0$mon"; }
```

```
  print "$day.$mon.$year";
  exit 0;
}

sub print_quota {
  my $ret = `quota -v $login`;
  $ret =~ tr/ //s;
  my @zeilen = split /\n/, $ret;
  my @quota = split / /, $zeilen[2];
  $quota[2] =~ m/(\d+)/;
  my $quota_used = $1;
  $quota_used = int($quota_used/1024);
  $quota[3] =~ m/(\d+)/;
  my $quota_limit = $1;
  $quota_limit = int($quota_limit/1024);

  print "$quota_used/$quota_limit MB";
  exit 0;
}
```

# A.9. benutzer

Das Programm dient dazu, die Benutzerverzeichnisse (public-html) anzuzeigen.

Dateiname: `/usr/local/httpd/cgi-bin/benutzer`.

```
#!/usr/bin/perl -w
use CGI;
$html = new CGI;
$dir = "/home/public_html";
print $html->header, $html->start_html('Öffentliche Benutzer-Verzeichnisse');
print "Hier findet sich eine Liste der auf diesem Server abgelegten
HTML-Angebote von Sch&uuml;lern und Lehrern. Jeder, der ein eigenes
Benutzerkonto im Computerraum hat, kann ein eigenes HTML-Angebot erstellen und
zwar einfach im Verzeichnis <tt>public_html</tt> unterhalb des
Home-Verzeichnisses. Alle Dateien in diesem Verzeichnis mit seinen
Unterverzeichnissen sind &ouml;ffentlich.<P>";

setpwent;
while (my @user = getpwent) {
    $userdir ="$dir/$user[0]";
    if (-d $userdir) {
                opendir DIR, $userdir;
                rewinddir DIR;
                @dirlist = readdir DIR;
                if (@dirlist > 2) {
                    $text='';
                    if (-s "$userdir/index.html") {
                                open DATEI, "$userdir/index.html";
                                $text = join '',<DATEI>;
                                close DATEI;
                    } elsif (-s "$userdir/index.htm") {
                                open DATEI, "$userdir/index.htm";
                                $text = join '',<DATEI>;
                                close DATEI;
                    }
                    ($titel) =($text =~ m@<TITLE>(.*?)</TITLE>@is);
                    $titel = ($titel ? "; <i>Titel: $titel</i>":'');
                    $username = ($user[6] ? $user[6] : $user[0]);
                           print $html->a({href=>"/~$user[0]/"},$username.$titel),$html
                }
                closedir DIR;
        }
```

```
}
endpwent;
print '<hr><p><a href="/oldhomes/">Seiten ehemaliger Sch&uuml;ler</a>';
print $html->p,$html->a({href=>"/"},'Hier geht\'s zur&uuml;ck zum Anfang.'),
    $html->end_html;
```

# A.10. netzstatus.cgi

Dateiname: `/usr/local/httpd/htdocs/admin/netzstatus.cgi`.

```
#!/usr/bin/perl

print "Content-type: text/html\n\n";

my $uhrzeit = localtime();

print <<HTML;
<html>
<head>
  <title>Netzstatus</title>
</head>
<body>
<h1>Netzstatus</h1>

<i>$uhrzeit</i><p>

<table border="1">
<tr><th>Host</th><th>Beschreibung</th><th>Erreichbar?</th></tr>
HTML

my %hosts;
my %reachable;
$hosts{"192.168.0.1"} = "server";
$hosts{"192.168.0.136"} = "Access Point";
$hosts{"192.168.0.137"} = "PrintServer R23 hpdj930c";
$hosts{"192.168.0.138"} = "Drucker magicolor 2350 R23";
$hosts{"192.168.0.139"} = "Drucker OkiPage R23";
$hosts{"192.168.0.254"} = "Drucker HP LJ5 R104";

foreach (keys(%hosts)) {
  my $ip = $_;
  system("ping -nc1 $ip > /dev/null 2>&1");
  my $ret = $?;
  $reachable{$ip} = not $ret;
}

foreach (keys(%hosts)) {
  my $ip = $_;
  print qq(<tr><td>$ip</td><td>$hosts{$ip}</td>);
  print qq(<td bgcolor="green">Ja</td>) if ($reachable{$ip});
  print qq(<td bgcolor="red">Nein</td>) if (not $reachable{$ip});
  print qq(</tr>);
}

print <<HTML;
</table>
</body>
</html>
HTML
```

## A.10.1. Hilfsprogramm ping.pl

Dateiname: `/usr/local/sbin/ping.pl`.

```
#!/usr/bin/perl

use Net::Ping;

my $host = shift || "127.0.0.1";
chomp($host);

print $host;

my $p = Net::Ping->new("icmp");

if ($p->ping($host)) {
  print "\t1\n";
} else {
  print "\t0\n";
}

$p->close();
```

# A.11. smbstatus.cgi

Dateiname: `/usr/local/httpd/htdocs/admin/smbstatus.cgi`.

```
#!/usr/bin/perl -wT

print <<EOF;
Content-type: text/html

<html>
<head>
  <title>Samba Status</title>
</head>
<body>
<h1>Samba Status</h1>
<pre>
EOF

$ENV{"PATH"} = "/usr/bin/";
system("/usr/bin/smbstatus -b");

print <<EOF;
</pre>
</body>
</html>
EOF

exit 0;
```

# A.12. samba-logger.cgi

Dateiname: `/usr/local/httpd/htdocs/admin/samba-logger.cgi`.

```
#!/usr/bin/perl

# samba-logger.cgi
# 2002/03/20
# leichte änderung: 2004/07/19 für debian
# Andreas Dangel <a.dangel@gmx.de>

use CGI;

############################################################
################## MAIN PROGRAM ############################
############################################################

$q = new CGI;
print "Content-type: text/html\n\n";
print "<html>\n";
print "<head>\n";
print "    <title>samba-logger</title>\n";
print "</head>\n";
print "<body>\n";
print "<h1>samba-logger</h1>\n";

$url = $q->url();
print "<p>\n";
print "<a href=\"$url\">samba-logger.cgi</a>\n";
print "<p>\n";


if (defined($q->param('logfile'))) {
    $logfile = $q->param('logfile');
    @loglines = &openLogfile($logfile);
    &analyze_loglines(\@loglines);
} else {
    @SambaLogfiles = &getSambaLogfiles();
    print "<form method=\"post\">\n";
    foreach(@SambaLogfiles) {
        print "<input type=\"radio\" name=\"logfile\" value=\"$_\">$_<br>\n";
    }
    print "<input type=\"submit\" name=\"submit\">\n";
    print "</form>\n";
}

print "<p>\n";
print "<a href=\"$url\">samba-logger.cgi</a>\n";
print "<p>\n";

print "<hr>\n";
print "2002/03/20 - Andreas Dangel\n";

print "</body>\n";
print "</html>";

#print @SambaLogfiles;

#open(LOGFILE, "zcat $SambaLogfiles[1] |");
#@lines = <LOGFILE>;
#close(LOGFILE);
#print @lines;



############################################################
############################################################
sub getSambaLogfiles {
    @array = glob("/var/log/samba/log.smb*");
    my @array2 = grep(!/smbd/, @array);
    return @array2;
}
```

```
sub openLogfile {
    $file = shift;
    if ( $file =~ s/\.gz// ) {
        open(FHANDLE, "zcat $file |");
    } else {
        open(FHANDLE, "$file");
    }
    @lines = <FHANDLE>;
    close(FHANDLE);
    return @lines;
}

sub analyze_loglines {
    $samba_lines = shift;
    @input = @$samba_lines;
    #print @input;
    $zeilen = @input;
    @data = ();
    %logins = {};

for ($i = 0; $i < $zeilen; $i++) {
    $zeile1 = $input[$i];
    $i++;
    $zeile2 = $input[$i];

    if ($zeile1 =~ m/smbd\/service\.c\:make\_connection\(/) {
        if ($zeile2 =~ m/connect\ to\ service\ netlogon\ as\ user\ /) {
            $zeile1 =~ s/smbd\/service\.c\:make\_connection\(//;
            $zeile2 =~ s/connect\ to\ service\ netlogon\ as\ user\ //;

            $datum = substr($zeile1, 0, 24);
            $datum = substr($datum, 1, 19);
            #chomp($datum = `date -d \"$datum\" +\%s`);
            $adresse = substr($zeile2, index($zeile2, "("));
            $adresse = substr($adresse, 1, index($adresse, ")")-1);
            $rechner = substr($zeile2, 2, index($zeile2, "(")-3);
            #$adresse .= " ($rechner)";
            #$adresse = substr($zeile2, 2, 23);
            #$login = substr($zeile2, 25);
            #$login = substr($login, 0, index($login, "("));
            $login = substr($zeile2, index($zeile2, ")")+1);
            $login = substr($login, 0, index($login, "("));

            $logins{$adresse} = ["$adresse ($rechner)", $datum, $login];
        }
    }
    elsif ($zeile1 =~ m/smbd\/service\.c\:close\_cnum\(/) {
        if ($zeile2 =~ m/closed\ connection\ to\ service\ pub/) {
            $zeile1 =~ s/smbd\/service\.c\:close\_cnum\(//;
            $zeile2 =~ s/closed\ connection\ to\ service\ pub//;

            $datum = substr($zeile1, 0, 24);
            $datum = substr($datum, 1, 19);
            #chomp($datum = `date -d \"$datum\" +\%s`);
            #$adresse = substr($zeile2, 2, 23);
            $adresse = substr($zeile2, index($zeile2, "("));
            $adresse = substr($adresse, 1, index($adresse, ")")-1);


            if (exists($logins{$adresse})) {
                $ref = $logins{$adresse};
                push(@$ref, $datum);

                delete($logins{$adresse});

                push(@data, $ref);
             }
```

```
        }
    } else {
        $i--;
    }

}
    &sort(\@data);
    &output(\@data);
}

sub sort {
    $ref = shift;
    @data = @$ref;
    $lines = @data;
    #print STDERR "$lines\n";
    for ($i = 0; $i < $lines - 1; $i++) {
        #printf STDERR "%d of %d (%.1f%%)\r", $i, $lines-1, 100*$i/($lines-1);

        for ($k = 0; $k < $lines -1; $k++) {
            $ref_a = $data[$k];
            $ref_b = $data[$k + 1];

            $time_a = @$ref_a[1];
            #chomp($time_a = `date -d \"$time_a\" +%s`);
            $time_b = @$ref_b[1];
            #chomp($time_b = `date -d \"$time_b\" +%s`);


            if ($time_a gt $time_b) {
                #print "$time_a $time_b\n";

                $data[$k] = $ref_b;
                $data[$k + 1] = $ref_a;
            }

        }

        #exit;
    }
}

sub output {
    $ref = shift;
    @data = @$ref;

    print "<table border=1>\n";
    print "<tr><th>Anmeldung</th><th>Benutzer</th><th>IP-Adresse</th>
           <th>Abmeldung</th></tr>\n";

    foreach (@data) {
        $ref = $_;
        $newline  = "<tr><td>" . @$ref[1] . "</td>";
        $newline .=     "<td>" . @$ref[2] . "</td>";
        $newline .=     "<td>" . @$ref[0] . "</td>";
        $newline .=     "<td>" . @$ref[3] . "</td></tr>";

        print $newline;

    }
    print "</table>\n";
}
```

# Anhang B. GNU Free Documentation License

## B.1. English (original) version

Version 1.2, November 2002

```
Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose

specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the

public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and

contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their

copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

# B.2. Inoffizielle deutsche Übersetzung

This is an unofficial translation of the GNU Free Documentation License into German. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documentation that uses the GNU FDL--only the original English text of the GNU FDL does that. However, we hope that this translation

will help German speakers understand the GNU FDL better.

Dies ist eine inoffzielle deutsche Übersetzung der *GNU Free Documentation License*. Sie ist nicht von der Free Software Foundation herausgegeben und erläutert nicht die Bedingungen der GNU FDL -- Dies tut nur der original englische Text der GNU FDL. Dennoch hoffen wir, dass diese Übersetzung mit dazu beiträgt deutschsprachigen Personen das Verstehen der GNU FDL zu erleichtern.

Präambel

Der Zweck dieser Lizenz ist es, ein Handbuch, Textbuch oder ein anderes zweckdienliches und nützliches Dokument frei, im Sinne von Freiheit, zu machen; jedermann die Freiheit zu sichern, es zu kopieren und mit oder ohne Änderungen daran, sowohl kommerziell als auch nicht kommerziell weiter zu verbreiten.

Weiterhin sichert diese Lizenz einem Autor oder Verleger die Möglichkeit, Anerkennung für seine Arbeit zu erhalten ohne für Änderungen durch Andere verantwortlich gemacht zu werden.

Diese Lizenz ist eine Art des "copyleft", was bedeutet, daß von diesem Dokument abgeleitete Werke ihrerseits in derselben Weise frei sein müssen.

Dies vervollständigt die GNU General Public License, die eine "copyleft"-Lizenz ist, und für freie Software entworfen wurde.

Diese Lizenz wurde für Handbücher für freie Software entworfen, denn frei Software braucht freie Dokumentation: Ein freies Programm sollte von Handbüchern begleitet sein, die dieselben Freiheiten bieten, die auch die Software selbst bietet.

Diese Lizenz ist aber nicht auf Softwarehandbücher beschränkt; vielmehr kann sie für jede Art von textuellen Werken verwendet werden, unabhängig davon, was das Thema ist, oder ob es als gedrucktes Buch veröffentlicht wurde. Wir empfehlen diese Lizenz prinzipiell für Werke, die als Anleitungen oder Referenzen dienen sollen.

1. Anwendbarkeit und Definitionen

Diese Lizenz findet Anwendung auf jedes Handbuch oder andere Werk, unabhängig von dem Medium, auf dem es erscheint, das einen vom Rechteinhaber eingefügten Hinweis enthält, der besagt, daß das Werk unter den Bedingungen dieser Lizenz verbreitet werden darf.

Ein solcher Hinweis gewährt eine weltweit gültige, tantiemenfreie und zeitlich unbefristete Lizenz, die es gestattet das Werk, unter den hier festgelegten Bedingungen, zu nutzen.

Der Begriff *Dokument* wird im Folgenden für alle solche Handbücher und Werke verwendet.

Jede Person kann Lizenznehmer sein und wird im Folgenden mit *Sie* angesprochen.

Sie akzeptieren diese Lizenz, wenn Sie ein Dokument derart kopieren, verändern oder verteilen, daß Sie gemäß den Gesetzen zum Copyright die Erlaubnis benötigen.

Eine *modifizierte Version* des Dokumentes steht für jedes Werk, das das Dokument als Ganzes oder in Teilen enthält, sowohl auf Datenträger kopiert, als auch mit Änderungen und/oder in andere Sprachen übersetzt.

Ein *zweitrangiger Abschnitt* ist ein benannter Anhang oder eine Einleitung des Dokumentes, der sich ausschließlich mit dem Verhältnis des Autors oder Verlegers des Dokumentes zu dem eigentlichen Thema des Dokumentes (oder damit zusammenhängender Dinge) beschäftigt, und der nichts enthält, das direkt zu dem eigentlichen Thema gehört. (Wenn das Dokument beispielweise ein Buch über Mathematik ist, dann darf ein zweitrangiger Abschnitt nichts über Mathematik enthalten).

Dies kann eine historische Beziehung zu dem Thema, oder damit zusammenhängender Dinge, oder von gesetzlicher, gesellschaftlicher, philosophischer, ethischer oder politischer Art sein, die das Thema betreffen.

Die *unveränderlichen Abschnitte* sind benannte zweitrangige Abschnitte, deren Titel als unveränderlicher Abschnitt in dem Lizenhinweis, der das Dokument unter diese Lizenz stellt, aufgeführt sind.

Wenn ein Abschnitt nicht in die oben stehende Definition eines zweitrangigen Abschnittes passt, dann ist es

nicht erlaubt diesen Bereich als unveränderlichen Bereich zu kennzeichnen.

*Umschlagtexte* sind bestimmte, kurze Textstücke, die als *vorderer Umschlagtext* oder als *hinterer Umschlagtext* in der Notiz benannt werden, die besagt, dass das Dokument unter dieser Lizenz freigegeben ist.

Ein vorderer Umschlagtext kann bis zu 5 Worte enthalten, ein hinterer Umschlagtext bis zu 25 Worte.

Eine *transparente Kopie* des Dokumentes bezeichnet eine maschinenlesbare Kopie, dargestellt in einem Format, dessen Spezifikationen allgemein verfügbar sind, und das geeignet ist das Dokument auf einfache Weise mit einem allgemeinen Texteditor oder (für Bilder, die aus Pixeln bestehen) mit einem allgemeinen Bildberabeitungsprogramm oder (für Zeichnungen) mit einem häufig verfügbaren Zeichenprogramm zu überarbeiten, und das geeignet ist es als Eingabe für Textformatierer zu verwenden, oder als Eingabe für automatische Konvertierungsprogramme, die eine Reihe von unterschiedlichen Formaten erzeugen, die ihrerseits als Eingabe für Textformatierer verwendet werden können. Eine Kopie in ein anderes transparentes Dateiformat dessen Auszeichnung oder das fehlen der Auszeichnungen derart beschaffen sind, nachfolgende Modifikationen durch die Leser zu verhindern oder zu erschweren ist nicht transparent

Ein Bildformat ist nicht transparent, wenn es für eine wesentliche Menge von Text verwendet wird.

Eine Kopie, die nicht transparent ist, wird als *opak* bezeichnet.

Beispiele verwendbarer Formate für transparente Kopien schliessen einfachen ASCII-Text ohne Auszeichnungen, TeX-info Eingabe, LaTeX-Eingabeformat, SGML oder XML, sofern die verwendete DTD öffentlich verfügbar ist, sowie standardkonformes, einfaches HTML, Postscript oder PDF, die für Veränderungen durch Menschen entworfen sind, ein.

Beispiele für transparente Bildformate sind u.a. PNG, XCF und JPG.

Opake Formate sind unter anderen solche proprietären Formate, die nur von proprietären Textverarbeitungsprogramm gelesen und bearbeitet werden können, SGML oder XML deren DTD und/oder Verarbeitungswerkzeuge nicht allgemein verfügbar sind, und maschinengeneriertes HTML, PostScript oder PDF, das von manchen Textverarbeitungsprogrammen nur zu Ausgabezwecken erzeugt wird.

Mit *Titelseite* wird in einem gedruckten Buch die eigentliche Titelseite sowie die direkt darauf folgenden Seiten bezeichnet, die all das in lesbarer Form enthalten, was in dieser Lizenz gefordert ist, dass es auf der Titelseite erscheinen muss.

Für Werke, die in Formaten vorliegen, die keine Titelseiten haben, gilt als Titelseite der Text, der der auffälligsten Darstellung des Titels des Werkes direkt folgt, aber noch vor dem Inhalt des Werkes steht.

Ein Abschnitt mit dem Titel xyz bezeichnet einen benannten Unterbereich des Dokumentes, dessen Titel entweder genau xyz ist, oder der xyz in Anführungszeichen enthält, der einem Text folgt, der xyz in eine andere Sprache übersetzt. (Hier steht xyz für einen speziellen Abschnittsnamen, der im Folgenden erwähnt wird wie"Danksagung"(Acknowledgements), "Widmung"(Dedications), "Anmerkung"(Endorsement) oder "Historie"(History).).

*Den Titel erhalten* eines Abschnittes bedeutet, daß beim Modifizieren des Dokumentes dieser Abschnitt mit dem Titel xyz bleibt, wie es in dieser Definition festgelegt ist.

Das Dokument kann direkt hinter der Notiz, die besagt, dass das Dokument unter dieser Lizenz freigegeben ist, Garantieausschlüsse enthalten. Diese Garantieausschlüsse werden so behandelt, asl seien sie als Referenzen in diese Lizenz eingeschlossen, allerdings nur um Garantien auszuschliessen: Jede andere Implizierung, die dieser Ausschluss hat ist ungültig und keine Wirkung im Sinne dieser Lizenz.

2. Datenträgerkopien

Sie dürfen das Dokument auf jedem Medium sowohl kommerziell als auch nicht kommerziell kopieren und verbreiten, vorausgesetzt, daß diese Lizenz, die Copyright-Hinweise sowie der Lizenzhinweis, der besagt, daß diese Lizenz auf das Dokument anzuwenden ist, in allen Kopien reproduziert wird, und daß keine weiteren Bedingungen jeglicher Art zu denen dieser Lizenz hinzugefügt werden.

Sie dürfen in den Kopien, die Sie erstellen oder verbreiten, keinerlei technische Maßnahmen treffen um das Lesen oder das weitere Kopieren zu erschweren oder zu kontrollieren. Dennoch dürfen Sie Gegenleistungen für

Kopien akzeptieren. Wenn Sie eine ausreichend große Menge von Kopien verteilen, müssen Sie zusätzlich die bestimmungen von Ziffer 3 beachten.

Sie können ausserdem unter denselben Bedingungen, die oben angeführt sind, Kopien verleihen und sie können Kopien auch öffentlich bewerben.

3. Kopien in Stückzahlen

Wenn Sie gedruckte Kopien des Dokumentes (oder Kopien auf Medien, die üblicherweise gedruckte Umschläge haben), in einer Stückzahl von mehr als 100 veröffentlichen, und der Lizenzhinweis des Dokumentes Umschlagtexte verlangt, müssen die Kopien in Hüllen verpackt sein, die alle diese Umschlagtexte klar und lesbar enthalten. Die vorderen Umschlagtexte auf dem vorderen Umschlag, die hinteren Umschlagtexte auf dem hinteren Umschlag.

Beide Umschläge müssen Sie ausserdem klar und lesbar als den Herausgeber dieser Kopien benennen.

Der vordere Umschlag muss den gesamten Titel darstellen, mit allen Worten gleich auffällig und sichtbar. Sie können weiteres Material den Umschlägen hinzufügen.

Das Kopieren mit Änderungen, die auf Umschläge begrenzt sind, können, so lange der Titel des Dokuments erhalten bleibt, ansonsten als Datenträgerkopien behandelt werden.

Wenn der vorgeschriebene Text für einen der Umschläge zu umfangreich ist um lesbar zu bleiben, sollten Sie den ersten der aufgelisteten Texte auf den aktuellen Umschlag nehmen (so viel wie vernünftigerweise möglich ist) und den Rest auf direkt angrenzenden Seiten.

Wenn Sie mehr als 100 opake Kopien veröffentlichen oder verbreiten, müssen Sie entweder eine maschinenlesbare, transparente Kopie jeder opaken Kopie beilegen, oder mit bzw. in jeder opaken Kopie eine Computer-Netzwerk Adresse angeben, von wo die allgemeine, netzwerk benutzende Öffentlichkeit, Zugriff zum Download einer kompletten transparenten Kopie über öffentliche Standardnetzwerkprotokolle hat.

Wenn Sie sich für die letztere Möglichkeit entscheiden, müssen Sie mit Beginn der Verbreitung der opaken Kopien in Stückzahlen, zumutbare und vernünftige Schritte unternehmen, um sicher zu stellen, daß die transparenten Kopien mindestens ein Jahr nach der Auslieferung der letzten opaken Kopie (direkt oder über einen Agenten oder Händler) dieser Ausgabe an die Öffentlichkeit, an der genannten Adresse verfügbar bleiben.

Es ist erbeten, aber nicht gefordert, daß Sie ausreichend lange vor der Auslieferung einer grösseren Menge von Kopien, Kontakt mit den Autoren des Dokumentes aufnehmen, um jenen die Möglichkeit zu geben, Ihnen eine aktualisierte Version des Dokumentes zuzuleiten.

4. Modifikationen

Unter den obigen Bedingungen unter Ziffer 2 und 3 können Sie modifizierte Versionen kopieren und verbreiten, vorausgesetzt, daß Sie die modifizierte Version unter exakt dieser Lizenz herausgeben, wobei die modifizierte Version die Rolle des Dokumentes einnimmt, und dadurch die weitere Modifikation und Verbreitung an jeden Lizensieren, der eine Kopie davon besitzt.

Zusätzlich müssen Sie die folgenden Dinge in der modifizierten Version beachten:

1. Benutzen Sie auf der Titelseite (und auf Umschlägen, sofern vorhanden) einen Titel, der sich von dem Titel des Dokumentes und von früheren Versionen unterscheidet. (Die früheren Versionen sollten, wenn es welche gibt, in dem Abschnitt *Historie* aufgelistet werden.)

   Sie können denselben Titel wie den einer Vorgängerversion verwenden, wenn der ursprüngliche Herausgeber damit einverstanden ist.

2. Geben Sie auf der Titelseite eine oder mehrere Personen oder Einheiten, die als Autoren auftreten können, als für die Modifikationen verantwortliche Autoren der modifizierten Version, zusammen mit mindestens fünf der ursprünglichen Autoren der Ursprungsversion an (alle vorherige Autoren, wenn es weniger als fünf sind), es sei denn diese befreien Sie von dieser Notwendigkeit.

3. Geben Sie auf der Titelseite den Namen des Herausgebers als Herausgeber an.

4. Erhalten Sie alle Copyright-Vermerke des Dokumentes.

5. Setzen Sie einen passenden Copyright-Vermerk für Ihre Modifikationen direkt hinter die anderen Copyright-Vermerke.

6. Schliessen Sie direkt hinter den Copyright-Vermerken einen Lizenzhinweis ein, der die öffentliche Erlaubnis erteilt, die modifizierte Version unter den Bedingungen dieser Lizenz zu benutzen, wie es im Anhang weiter unten beschrieben ist.

7. Erhalten Sie im Copyright-Vermerk die komplette Liste der unveränderlichen Abschnitte und obligatorischen Umschlagtexte, die in dem Lizenzvermerk des Dokumentes aufgeführt sind.

8. Schliessen Sie eine unveränderte Kopie dieser Lizenz mit ein.

9. Erhalten Sie den Abschnitt "Historie". Erhalten Sie den Titel und fügen Sie einen Punkt hinzu der mindestens den Titel, das Jahr, die neuen Autoren und Herausgeber, wie sie auf der Titelseite aufgeführt sind, enthält. Sollte es keinen Abschnitt Historie geben, dann erstellen Sie einen, der Titel, Jahr, Autor und Herausgeber des Dokumentes, wie auf der Titelseite angegeben, enthält und fügen Sie einen Punkt hinzu, der die modifizierte Version wie oben dargestellt beschreibt.

10. Erhalten Sie die Netzwerkadresse, die angegeben wurde, um Zugang zu einer transparenten Kopie zu gewähren, sowie entsprechend angegebene Adressen früherer Versionen, auf denen das Dokument aufbaute. Diese Angaben können in den Abschnitt Historie verschoben werden. Sie können die Netzwerkadresse weglassen, wenn sie sich auf ein Werk bezieht, das mindestens 4 Jahre vor dem Dokument selbst veröffentlicht wurde, oder wenn der ursprüngliche Herausgeber der Version, auf die sich die Adresse bezieht, seine Erlaubnis erteilt.

11. Erhalten Sie für alle Abschnitt, die als *Danksagungen* (Acknowledgements) oder *Widmungen* (Dedications) überschrieben sind, den Titel sowie die Substanz und den Ton aller vom Geber gemachten Danksagungen und/oder Widmungen in diesem Abschnitt.

12. Erhalten Sie alle unveränderlichen Abschnitte unverändert, sowohl im Titel als auch im Text. Abschnittsnummern oder dergleichen gelten hierbei nicht als Teil des Titels.

13. Löschen Sie alle Abschnitte, die als *Anmerkungen* (Endorsements) überschrieben sind. Ein solchen Abschnitt sollte nicht in der modifizierten Version enthalten sein.

14. Benennen Sie keinen Abschnitt in *Anmerkungen* um, oder in einen Namen, der in Konflikt mit einem unveränderlichen Abschnitt gerät.

15. Erhalten Sie alle Garantieausschlüsse.

Wenn die modifizierte Version neue Vorspannabschnitte oder Anhänge enthält, die zweitrangige Abschnitte sein können, und die kein vom Dokument kopiertes Material enthalten, können Sie, nach Ihrem Belieben, einige oder alle diese Abschnitte als unveränderliche Abschnitte in die Lizenzanmerkung der modifizierten Version aufnehmen. Diese Titel müssen sich von allen anderen Titeln unterscheiden.

Sie können einen Abschnitt *Anmerkungen* anfügen, sofern dieser nichts als Bemerkungen, verschiedener Stellen, zu der modifizierten Version enthält.

Beispielsweise Publikumsreaktionen oder eine Mitteilung, daß der Text von einer Organisation als maßgebliche Definition eines Standards geprüft wurde.

Sie können einen Teil mit bis zu fünf Worten als vorderen Umschlagtext und einen mit bis zu 25 Worten als hinteren Umschlagtext an das Ende der Liste mit den Umschlagtexten der modifizierten Version hinzufügen.

Nur je ein Teil für den vorderen Umschlagtext und den hinteren Umschlagtext können von jeder Einheit hinzugefügt (oder durch entsprechende Anordnung erstellt) werden.

Wenn das Dokument bereits einen Umschlagtext für denselben Umschlag enthält, das von Ihnen oder der Einheit, in deren Namen Sie tätig sind, bereits früher eingefügt wurde, dürfen Sie keine neue hinzufügen. Sie können aber den alten ersetzen, wenn sie die ausdrückliche Genehmigung des Herausgebers haben, der den früheren Text eingefügt hat.

Der/die Autor(en) und Herausgeber des Dokumentes geben duch diese Lizenz weder implizit noch explizit die Erlaubnis ihren Namen für Werbung in den Anmerkungen der modifizierten Version zu benutzen.

5. Dokumente Kombinieren

Sie können mehrere Dokumente, die unter dieser Lizenz freigegeben sind, unter den Bedingungen unter Ziffer 4 für modifizierte Versionen miteinander kombinieren, vorausgesetzt, daß in der Kombination alle unveränderlichen Abschnitte aller Originaldokumente, enthalten sind, und daß Sie diese alle in der Liste der unveränderlichen Abschnitte der Lizenzanmerkung des kombinierten Dokumentes aufführen, sowie alle Garantieausschlüsse erhalten.

Das kombinierte Werk braucht nur eine Kopie dieser Lizenz zu enthalten, und mehrere identische unveränderliche Abschnitte können durch eine einzelne Kopie ersetzt werden.

Wenn es mehrere unveränderliche Abschnitte mit unterschiedlichem Inhalt aber gleichem Namen gibt, machen Sie den Namen eindeutig, indem Sie am Ende des Titels, in Anführungszeichen, den Namen des original Autors oder Herausgebers, falls bekannt, oder andernfalls eine eindeutige Nummer anhängen.

Machen Sie dasselbe mit den Titeln der Abschnitte in der Liste der unveränderlichen Abschnitte im Lizenzhinweis des kombinierten Werkes.

In der Kombination müssen Sie alle Abschnitte mit dem Titel *Historie* in den unterschiedlichen Dokumenten zu einem einzelnen Abschnit *Historie* zusammenführen; entsprechend verfahren Sie mit den Abschnitten *Danksagungen* und *Widmungen* . Sie müssen alle Abschnitte mit dem Titel *Anmerkungen* löschen.

6. Sammlungen von Dokumenten

Sie können eine Sammlung von Dokumenten erstellen, bestehend aus diesem Dokument und weiteren, unter dieser Lizenz stehenden Dokumenten, wobei Sie die einzelnen Kopien dieser Lizenz in den verschiedenen Dokumenten durch eine einzelne Kopie, die in der Sammlung enthalten ist, ersetzen, vorausgesetzt, Sie befolgen in allen andern Punkten, für jedes der Dokumente, die Regeln für Datenträgerkopien.

Sie können ein einzelnes Dokument aus einer solchen Sammlung herausziehen und einzeln unter dieser Lizenz verbreiten, vorausgesetzt, Sie fügen eine Kopie dieser Lizenz in das extrahierte Dokument ein, und befolgen ansonsten die Bedingungen dieser Lizenz für Datenträgerkopien.

7. Aggregation mit unabhängigen Werken

Eine Zusammenstellung des Werkes, oder von Ableitungen davon, mit anderen, separaten und unabhängigen Dokumenten oder Werken, in oder auf demselben Band eines Speicher- oder Verbreitungsmediums, wird dann eine Aggregation genannt, wenn die Copyrights der Zusammenstellung nicht dazu verwendet werden die Rechte der Benutzer, die für die einzelnen Werke gewährt werden, stärker zu beschränken als dies durch die Lizenzen der einzelnen Werke geschieht.

Wenn das Werk in einer Aggregation vorhanden ist, so gilt diese Lizenz nicht für die anderen Werke dieser Aggregation, die keine Ableitung des Dokumentes sind.

Wenn die Bestimmungen für die Umschlagtexte aus Ziffer 3 Anwendung finden, und wenn das Dokument weniger als die Hälfte der gesammten Aggregation ausmacht, dann können die Umschlagtexte auf Seiten gesetzt werden, die das Dokument innerhalb der Aggregation umschliessen, oder auf das elektronische Äquivalent eines Umschlages, wenn das Dokument in elektronischer Form vorliegt.

Andernfalls müssen sie auf gedruckten Umschlägen erscheinen, die das gesamte Werk umschliessen.

8. Übersetzung

Übersetzungen werden als eine Art von Modifikationen betrachtet. Damit können Sie eine Übersetzung des Dokumentes unter den Bestimmungen von Ziffer 4 verbreiten.

Um die unveränderlichen Abschnitte durch eine Übersetzung zu ersetzen, benötigen Sie die spezielle Erlaubnis des Copyright-Inhabers. Sie können allerdings Übersetzungen von einigen oder allen unveränderlichen Abschnitten zu den original Versionen der unveränderlichen Abschnitte hinzufügen.

Sie können eine Übersetzung dieser Lizenz und allen Lizenzhinweisen im Dokument sowie allen

Garantieausschlüssen hinzufügen, vorausgesetzt, daß Sie ebenso die originale englische Version dieser Lizenz und aller Hinweise und Ausschlüsse beifügen.

Sollten die Übersetzung und die Originalversion dieser Lizenz oder eines Hinweises oder Ausschlusses voneinander abweichen, so hat die Originalversion vorrang.

Wenn ein Abschnitt des Dokumentes als Danksagung, Widmungen oder Historie überschrieben ist, so erfordert die Forderung (Ziffer 4) den Titel dieses Abschnittes zuerhalten, die Änderung des aktuellen Titels.

9. Abschlussbestimmungen

Sie dürfen dieses Dokument nicht kopieren, verändern, unterlizensieren oder verteilen mit der Ausnahme, daß Sie es ausdrücklich unter dieser Lizenz tun.

Jedweder andere Versuch zu kopieren, zu modifizieren, unter zu lizensieren oder zu verbreiten ist unzulässig und führt automatisch zum Entzug der durch diese Lizenz gewährten Rechte. Dennoch verlieren jene Parteien, die von ihnen Kopien oder Rechte unter dieser Lizen erhalten haben, nicht Ihre Rechte, so lange sie sich in völliger Übereinstimmung mit der Lizenz befinden.

10. Spätere Überarbeitungen dieser Lizenz

Die *Free Software Foundation* kann von Zeit zu Zeit neue, überarbeitete Versionen der *GNU Free Dokumentation License* veröffentlichen. Diese neuen Versionen werden im Geiste gleich bleiben, können sich aber in Details unterscheiden um neuen Problemen oder Besorgnissen gerecht zu werden.

Siehe: http://www.gnu.org/copyleft/

Jede Version dieser Lizenz erhält eine eigene Versionsnummer.

Wenn das Dokument bestimmt, daß eine bestimmt numerierte Version *oder jede spätere Version* dafür gilt, haben Sie die Wahl den Bestimmungen dieser speziell benannten Version zu folgen, oder jeder Version, die später von der *Free Software Foundation* , nicht als Entwurf, veröffentlicht wurde.

Anhang:

Wie Sie diese Lizenz für Ihre Dokumente verwenden können

Um diese Lizenz in einem Dokument zu verwenden, das sie selbst geschrieben haben, schliessen Sie eine Kopie dieser Lizenz (eine englische Kopie des Originals anm. des Übersetzers) in Ihr Dokument mit ein, und setzen Sie den folgenden Copyright- und Lizenzhinweis gleich hinter die Titelseite:

Copyright (c) YEAR YOUR NAME

Persmission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation;

with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

Es folgt eine Übersetzung des oben stehenden Hinweises, der nur zur Klarheit hier angegeben ist ! (anm.: des Übersetzers)

Copyright Jahr Ihr Name

Kopieren, Verbreiten und/oder Modifizieren ist unter den Bedingungen der

*GNU Free Documentation License* , Version 1.2 oder einer späteren Version, veröffentlicht von der *Free Software Foundation* , erlaubt.

Es gibt keine unveränderlichen Abschnitte,

keinen vorderen Umschlagtext und keinen hinteren Umschlagtext

Eine Kopie des Lizenztextes ist unter dem Titel

*GNU Free Documentation License* enthalten.

(Ende der Übersetzung des Lizenzhinweistextes)

Wenn Sie unveränderlichen Abschnitte, vordere und hintere Umschlagtexte haben, ersetzen Sie die Zeile: "Es gibt keine..... Umschlagtext" durch die Folgende:

Mit den unveränderlichen Abschnitten:

„ Liste dem den Titeln der unveränderlichen Abschnitte “ mit dem vorderen Umschlagtext:

„ vorderer Umschlagtext “ und dem hinteren Umschlagtext:

„ hinterer Umschlagtext “

Wenn Sie unveränderliche Abschnitte, aber keine Umschlagtexte oder irgend eine andere Kombination der drei Bereiche haben, mischen Sie die verschiedenen Alternativen, daß sie zu Ihren Anforderungen passen.

Wenn Ihr Dokument nicht-triviale Codebeispiele enthält empfehlen wir diese Beispiele parrallel unter einer freien Softwarelizenz Ihrer Wahl, beispielsweise der *GNU General Public License* zu lizensieren, um ihren Gebrauch in freier Software zu erlauben.