Linux Samba HOWTO

David Wood (dwood@plugged.net.au) und Klaus-Dieter Schumacher (Klaus-Dieter.Schumacher@fernuni-hagen.de)
v1.0-2, 20. April 1997

Dieser Text beschreibt das Programm Samba, mit dem man einen Linux Rechner in ein Netzwerk von Windowsrechnern integrieren kann.

Inhaltsverzeichnis

1 Einleitung

1.1 Zur deutschen Übersetzung

Diese HOWTO wurde von Klaus-Dieter Schumacher ins Deutsche übersetzt. Senden Sie bitte Verbesserungsvorschläge zur deutschen Übersetzung an die folgende EMail-Adresse: `Klaus-Dieter.Schumacher@fernuni-hagen.de`. Die deutsche Version ist auf dem DLHP-Server zu bekommen:

```
http://www.tu-harburg.de/dlhp/
```

Hier sind auch weitere deutsche Übersetzungen zu haben. Für die Koordination des deutschen HOWTO-Projekts ist Marco Budde (`Budde@tu-harburg.de`) verantwortlich.

1.2 Copyright

1.3 Was ermöglicht Samba?

Das SMB-Protokoll (Server Message Block-Protokoll) wird von der Firma Microsoft für Windows 3.11, NT und 95/98 genutzt, um den gemeinsamen Zugriff auf Laufwerke und Drucker zu gestatten.

Mit der Sammlung der Samba-Tools von Andrew Tridgell besteht die Möglichkeit, daß sich UNIX-Rechner und damit auch Linux-Maschinen Platten und Drucker mit Windows-Rechnern teilen.

Insgesamt bietet Samba vier verschiedene Optionen an:

1. Linux-Laufwerke werden auch von Windows-Rechnern genutzt.
2. Windows-Laufwerke werden auch von Linux-Rechnern genutzt.
3. Ein an einer Linux-Maschine angeschlossener Drucker wird auch von Windows-Rechnern genutzt.
4. Ein an einem Windows-Rechner angeschlossener Drucker wird auch von Linux-Rechnern genutzt.

Alle vier Möglichkeiten werden nachfolgend beschrieben.

1.4 Hinweis

Die beschriebenen Prozeduren und Skripte arbeiten entweder für David Wood oder wurden David Wood zur Verfügung gestellt. Verschiedene Konfigurationen arbeiten eventuell nicht gemäß den hier vorgestellten Beschreibungen. Falls eine derartige Situation eintreten sollte, senden Sie bitte eine EMail mit entsprechenden Hinweisen an David Wood.

2 Weitere Informationen

Diese HOWTO versucht zu erläutern, wie der grundsätzliche SBM-Datei-Service und der Drucker-Service auf einem Linux-Rechner zu konfigurieren sind. Da es sich bei Samba um ein sehr komplexes und umfangreiches Programmpaket handelt, macht es keinen Sinn, hier die gesamte Samba-Dokumentation erneut vorzustellen.

Weitere Informationen sind in den nachfolgend genannten Dokumenten enthalten:

Samba-Dokumentation

Sie ist Bestandteil der Samba-Distribution, die sich u.a. auf

```
nimbus.anau.edu.au:/pub/trigde/samba/
```

befindet.

Linux Drucker HOWTO

Sie ist Bestandteil des deutschen Linux HOWTO Projektes.

Print2Win mini-HOWTO

Diese mini-HOWTO beschreibt, wie ein Drucker unter Windows95/98 bzw. Windows NT von Linux aus genutzt werden kann.

3 Installation

Die aktuellste Version des Quellcodes von Samba liegt auf:

```
nimbus.anu.edu.au:/pub/tridge/samba/
```

Bei fast allen modernen Linux-Distributionen besteht die Möglichkeit, Samba als Paket zu installieren, so daß man Samba nicht selbst kompilieren muß. Außerdem ist Samba dann bereits an die Gegebenheiten der entsprechenden Distribution angepaßt.

Die folgenden zwei Daemonen werden vom Samba-Paket benötigt.

smbd

Hierbei handelt es sich um den SMB-Daemonen.

nmbd

Ermöglicht den NetBIOS-Nameserver-Support für die Clients.

Beide werden im allgemeinen im Verzeichnis `/usr/sbin` installiert und werden entweder beim Booten durch die Start-Skripte oder mittels `inetd` gestartet. Beispielskripte finden sich in den Abschnitten 5, 6 und 8.

Die nachfolgenden Binärdateien werden üblicherweise im Verzeichnis `/usr/bin` installiert.

smbclient

Hierbei handelt es sich um den SMB-Client für UNIX-Rechner.

smbprint

Dies ist ein Skript, das das Drucken auf einem an einem SMB-Host angeschlossenen Drucker ermöglicht.

smbprint.sysv

Erfüllt die gleiche Funktion wie `smbprint`, allerdings auf Rechnern mit SVR4 UNIX.

smbstatus

Hiermit besteht die Möglichkeit, sich die aktuellen SMB-Verbindungen des lokalen Hosts anzeigen zu lassen.

smbrun

Hierbei handelt es sich um ein Skript, um die Ausführung von Anwendungen auf einem SMB-Host zu erleichtern.

Zusätzlich liegt mit dieser HOWTO ein mit `print` bezeichnetes Skript vor, das als brauchbares Front-End für das Skript `smbprint` dient.

Das Samba-Paket läßt sich einfach installieren. Es wird nur der Quellcode von einer der oben genannten Quellen übertragen und dann die README-Datei gelesen. Desweiteren ist die Datei `doc/INSTALL.txt` Bestandteil der Distribution. Sie enthält eine Anleitung, die Schritt für Schritt beschreibt, wie vorzugehen ist.

Danach werden die Daemonen in das Verzeichnis `/usr/bin`, die Binärdateien in das Verzeichnis `/usr/bin` und die Hilfedateien (man pages) in das Verzeichnis `/usr/man` kopiert.

Wenn das Samba-Paket kompiliert wird, ist im Makefile die Position der Konfigurationsdatei `smb.conf` anzugeben. Sie befindet sich üblicherweise im Verzeichnis `/etc`. Sie kann jedoch auch in jedes beliebige andere Verzeichnis kopiert werden.

Für diese Beschreibung wird angenommen, daß als Zuordnung für die Konfigurationsdatei `/etc/smb.conf`, als Zuordnung für die Log-Datei `log file = /var/log/samba-log.%m` und für das Lock-Verzeichnis `directory = /var/lock/samba` angegeben wurde.

Installation der Konfigurationsdatei `smb.conf`:

- Zunächst sollte in das Verzeichnis, in dem Samba enthalten ist, gegangen, dann in das Unterverzeichnis `examples/simple` gewechselt und die Datei `README` gelesen werden.
- Es ist sinnvoll, die Datei `smb.conf` aus diesem Verzeichnis nach `/etc` zu kopieren.

Es ist Vorsicht geboten. Wenn eine Distribution genutzt wird, bei der Samba bereits installiert ist, befindet sich eventuell bereits eine Samba-Konfigurationsdatei im Verzeichnis `/etc`. Diese Beispiel-Konfigurationsdatei sollte zunächst genutzt werden.

Wenn die Konfigurationsdatei nicht in das Verzeichnis `/etc` kopiert werden soll, dann ist ein symbolischer Link im Verzeichnis `/etc` auf das Verzeichnis anzulegen, indem sich die Datei `smb.conf` befindet. Also z.B.:

```
ln -s /path/to/smb.conf /etc/smb.conf
```

4 Das Starten der Daemonen

Die beiden SMB-Daemonen sind `/usr/sbin/smbd` und `/usr/sbin/nmbd`. Sie können beide mit dem Programm `inetd` oder als einzelne Prozesse gestartet werden. Wenn man den Rechner dauerhaft als Dateiserver konfigurieren will, dann sollten beide so von dem Programm `inetd` gestartet werden, daß sie automatisch neu gestartet werden, falls sie abstürzen sollten. Wenn man die SMB-Unterstützung nur gelegentlich nutzen will oder nur als Hilfe für die Systemadministration benötigt, dann können beide durch ein `/etc/rc.d/init.d` Skript oder auch unmittelbar durch die Eingabe von `smbd` und `nmbd` gestartet werden.

Um die Daemonen mit dem Programm `inetd` starten zu können, sind in die Konfigurationsdatei `/etc/inetd.conf` die nachfolgenden Zeilen einzutragen:

```
# SAMBA NetBIOS Dienste (für Datei- und Druckersharing)
netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd
netbios-ns  dgram  udp wait   root /usr/sbin/nmbd nmbd
```

Dann ist der inetd-Daemon erneut aufzurufen mit:

```
kill -HUP 1
```

Um die beiden Daemonen beim Hochfahren des Systems aufzurufen, ist das nachfolgende Skript als Datei `/etc/rc.d/init.d/smb` zu speichern und ein symbolischer Link auf die in den Kommentaren spezifizierten Dateien anzulegen:

```
#!/bin/sh

# /etc/rc.d/init.d/smb - Startet und beendet SMB Dienste.
#
# Die folgenden Dateien sollten symbolische Links auf
```

```
# diese Datei sein:
#
#   /etc/rc.d/rc1.d/K35smb  (Beenden bei einem Shutdown)
#   /etc/rc.d/rc3.d/S91smb  (Starten im Multiuser Modus)
#   /etc/rc.d/rc6.d/K35smb  (Beenden bei einem Reboot)
#
# Source Funktionsbibliothek.
. /etc/rc.d/init.d/functions

# Source Netzwerkkonfiguration.
. /etc/sysconfig/network

# Überprüfe, ob das Netzwerk läuft.
[ ${NETWORKING} = "no" ] && exit 0

# Wie wurde dieses Skript aufgerufen?
case "$1" in
  start)
    echo -n "Starte SMB Dienste: "
    daemon smbd -D
    daemon nmbd -D
    echo
    touch /var/lock/subsys/smb
    ;;
  stop)
    echo -n "Beende SMB Dienste: "
    killproc smbd
    killproc nmbd
    rm -f /var/lock/subsys/smb
    echo ""
    ;;
  *)
    echo "Syntax: smb {start|stop}"
    exit 1
esac
```

5 Grundsätzliche Konfiguration der Datei smb.conf

Die Samba-Konfiguration für Linux oder andere UNIX-Rechner wird ausschließlich durch die Datei `/etc/smb.conf` gesteuert. Diese Datei bestimmt, welche Ressourcen das System nach außen anbietet und welche Einschränkungen hierbei existieren.

In diesem Abschnitt wird das Mitnutzen von Linux-Laufwerken und Linux-Druckern durch Windows-Rechner beschrieben. Da die hier vorgestellte Konfigurationsdatei `smb.conf` nur als Einstieg gedacht ist, wurde sie sehr einfach gestaltet. In den folgenden Kapiteln werden dann weitere Möglichkeiten der gemeinsamen Nutzung von Resourcen beschrieben.

Jeder Abschnitt der Konfigurationsdatei `smb.conf` beginnt mit einem Abschnittstitel. Dies sind u.a.:

- global
- homes
- tmp

- printers
- public

Dabei sind den Abschnittstiteln folgende Bedeutungen zugeordnet:

global

In diesem Abschnitt werden Variablen definiert, die Samba für die Zuteilung aller Resourcen nutzt.

homes

In diesem Abschnitt wird einem Remote-User, also einen von einem anderen Rechner zugreifenden Benutzer, der Zugriff auf sein und nur sein Homeverzeichnis auf dem Linux-Rechner ermöglicht. Diese Maschine muß allerdings im lokalen Netz eingebunden sein. Voraussetzung ist allerdings, daß der Windows-Benutzer eine Zugangsberechtigung für die Linux-Maschine besitzt.

printers

Hiermit werden die in der `/etc/printcap` spezifizierten Drucker den Clients zur Verfügung gestellt.

Die folgende Beispieldatei `smb.conf` gestattet einem Remote-User den Zugriff auf sein Heimatverzeichnis auf dem Linux-Rechner und außerdem den Schreibzugriff auf ein temporäres Verzeichnis.

```
; /etc/smb.conf
;
; Achtung: Der Server muß nach Durchführen der Änderungen
;          in dieser Datei zunächst gestoppt und dann
;          erneut gestartet werden:
;           /etc/rc.d/init.d/smb stop
;           /etc/rc.d/init.d/smb start

[global]
; Die folgende Zeile ist zu entkommentieren,
; wenn Gästen der Zugriff erlaubt werden soll.
;  guest account  = nobody
   log file       = /var/log/samba-log.%m
   lock directory = /var/lock/samba
   share modes    = yes

[homes]
   comment     = Home Directories
   browseable  = no
   read only   = no
   create mode = 0750

[tmp]
   comment   = Temporary file space
   path      = /tmp
   read only = no
   public    = yes
```

6 Nutzen eines Linux-Laufwerks durch einen Windows-Rechner

Wie in der obigen einfachen `smb.conf`-Datei gezeigt wurde, läßt sich ein Linux-Laufwerk von Windows aus relativ einfach nutzen. Allerdings lassen sich in Samba die Einstellungen in großem Umfang verändern. Hierzu folgen nun einige Beispiele.

Um für alle Windows-Anwender die gemeinsame Nutzung eines bestimmten Verzeichnisses zu ermöglichen, bietet es sich an, den Abschnitt `[tmp]` in der Datei `smb.conf` zu duplizieren und ihn dann, wie nachfolgend dargestellt ist, geeignet zu modifizieren. Die Konfigurationsdatei `smb.conf` wird also erweitert durch:

```
[public]
   comment   = Public Stuff
   path      = /home/public
   public    = yes
   writable  = yes
   printable = yes
```

Sollen alle Benutzer des oben genannte Verzeichnisses das Recht besitzen, lesend auf das Verzeichnis zuzugreifen, während nur die Mitglieder der Gruppe `staff` das Verzeichnis beschreiben dürfen, so ist der Eintrag zu ändern in:

```
[public]
   comment    = Public Stuff
   path       = /home/public
   public     = yes
   writable   = yes
   printable  = no
   write list = @staff
```

Weitere Hinweise zur gemeinsamen Nutzung der Festplatte können der Samba-Dokumentation oder den manual pages zu Samba entnommen werden.

7 Gemeinsame Nutzung eines Windows-Laufwerkes

Das SMB-Clientprogramm für UNIX-Rechner ist Bestandteil der Samba-Distribution. Es unterstützt eine FTP-ähnliche Schnittstelle von der Kommandozeile aus. Sie kann genutzt werden, um Dateien zwischen einem Windows-Server und einem Linux-Client auszutauschen.

Mit dem folgenden Aufruf läßt sich feststellen, welche gemeinsamen Resourcen auf einem Host zur Verfügung stehen:

```
/usr/sbin/smbclient -L host
```

Dabei ist `host` ein Platzhalter für den Namen des Rechners, den man betrachten will. Es wird eine Liste von Service-Namen zurückgeliefert, also Namen von Laufwerken oder Druckern, die für die gemeinsame Nutzung zur Verfügung stehen. Wenn der Server nach einem Paßwort fragt, dann verwenden Sie das Paßwort für den Gastzugang oder für Ihren persönlichen Zugang zu diesem Rechner. Beispielsweise:

```
smbclient -L zimmerman
```

Die Ausgabe sollte dann etwa so aussehen:

```
Server time is Sat Aug 10 15:58:27 1996
Timezone is UTC+10.0
Password:
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]
Server=[ZIMMERMAN] User=[] Workgroup=[WORKGROUP] Domain=[]

            Sharename      Type      Comment
```

```
            ---------      ----      -------
            ADMIN$         Disk      Remote Admin
            public         Disk      Public
            C$             Disk      Default share
            IPC$           IPC       Remote IPC
            OReilly        Printer   OReilly
            print$         Disk      Printer Drivers

This machine has a browse list:

            Server               Comment
            ---------            -------
            HOPPER               Samba 1.9.15p8
            KERNIGAN             Samba 1.9.15p8
            LOVELACE             Samba 1.9.15p8
            RITCHIE              Samba 1.9.15p8
            ZIMMERMAN
```

Die Suchliste zeigt gegebenenfalls weitere SMB-Server, die alle über das Netzwerk verfügbar sind, einschließlich ihrer Ressourcen an.

Um den Client zu nutzen, ist folgender Aufruf notwendig:

```
/usr/sbin/smbclient service <password>
```

Mit `service` ist dabei ein Rechner und der Resourcen-Name gemeint. Soll zum Beispiel ein Verzeichnis angesprochen werden, daß als `public` von einem Rechner mit dem Namen `zimmermann` zur Verfügung gestellt wird, so ist für `service` `\\zimmerman\public` anzugeben. Dabei sind die Restriktionen der Shell bei Backslashes zu beachten. In diesem Fall bedeutet es, daß der Aufruf lautet:

```
/usr/sbin/smbclient \\\\zimmerman\\public mypasswd
```

Dabei stellt `mypasswd` einen Platzhalter für die Zeichenkette des eigenen Paßwortes dar.

Der Client `smbclient` liefert dann z.B. folgende Rückmeldung:

```
Server time is Sat Aug 10 15:58:44 1996
Timezone is UTC+10.0
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]
smb: \>
```

Die Eingabe von `h` gibt eine Hilfe zu dem Samba Client aus:

```
smb: \> h
ls          dir         lcd         cd          pwd
get         mget        put         mput        rename
more        mask        del         rm          mkdir
md          rmdir       rd          prompt      recurse
translate   lowercase   print       printmode   queue
cancel      stat        quit        q           exit
newer       archive     tar         blocksize   tarmode
setmode     help        ?           !
smb: \>
```

Wenn man das Programm `ftp` nutzen kann, wird man die manual page zum `smbclient` nicht benötigen.

8 Gemeinsame Nutzung eines Linux-Druckers

Um einen Linux-Drucker auch von Windows-Rechners aus nutzen zu können, muß zunächst sichergestellt sein, daß der Drucker bereits unter LINUX installiert und konfiguriert ist. Wenn das Drucken unter Linux möglich ist, dann läßt sich die gemeinsame Nutzung des Druckers relativ einfach einrichten.

Wie ein Drucker an einen Computer angeschlossen wird und wie Linux konfiguriert werden muß, um diesen nutzen zu können, beschreibt die *Linux Drucker HOWTO*.

Der folgende Abschnitt zur gemeinsamen Druckernutzung ist nur als Anregung zu verstehen. Falls weitergehende Informationen bezüglich der gemeinsamen Druckernutzung vorliegen, bitten wir, diese an `dwood@plugged.net.au` oder `Klaus-Dieter.Schumacher@FernUni-Hagen.de` zu senden. Wir werden uns um die Vervollständigung dieses Abschnitts kümmern.

Der `smb.conf`-Datei ist z.B. folgende Druckerkonfiguration hinzuzufügen:

```
[global]
   printing        = bsd
   printcap name   = /etc/printcap
   load printers   = yes
   log file        = /var/log/samba-log.%m
   lock directory = /var/lock/samba

[printers]
   comment      = All Printers
   security     = server
   path         = /var/spool/lpd/lp
   browseable   = no
   printable    = yes
   public       = yes
   writable     = no
   create mode = 0700

[ljet]
   security       = server
   path           = /var/spool/lpd/lp
   printer name   = lp
   writable       = yes
   public         = yes
   printable      = yes
   print command = lpr -r -h -P %p %s
```

Dabei muß sicher gestellt werden, daß der Drucker-Pfad, der oben unter `[ljet]` aufgeführt ist, zum Spool-Verzeichnis in der Datei `/etc/printcap` paßt.

Anmerkung

Unter Samba treten einige Probleme beim Zugriff auf an Linux-Rechnern angschlossenen Drcukern durch Windows-NT-Rechnern auf. Ein Problem unter NT ist, den gemeinsam zu nutzenden Drucker überhaupt zu sehen. Um dies zu beseitigen, lesen Sie bitte die Anmerkungen in der Datei `docs/WinNT.txt` der Samba-Distribution. Ein anderes Problem besteht beim Paßwort. Lesen Sie bitte hierzu die Kommentare in `docs/WinNT.txt`, um auch dieses Problem zu beseitigen.

9 Gemeinsame Nutzung eines Windows-Druckers

Um einen an einem Windows-Rechner angeschlossenen Drucker auch von Linux aus nutzen zu können, muß folgendermaßen vorgegangen werden:

1. In der Datei `/etc/printcap` müssen die passenden Einträge vorhanden sein. Diese müssen zur lokalen Verzeichnisstruktur, z.B. zum Spool-Verzeichnis, passen.
2. Das Skript `/usr/bin/smbprint` muß vorhanden sein. Es ist Bestandteil des Quellcodes von Samba. Dieses Skript liegt aber nicht allen binären Distributionen bei. Deshalb folgt unten die Vorstellung einer leicht geänderten Kopie des Skripts.
3. Sollen ASCII-Dateien nach Postscript konvertiert werden, dann muß `nenscript` oder ein vergleichbaren Produkt vorhanden sein. Bei `nenscript` handelt es sich um ein Postscript-Konvertierprogramm, das im allgemeinen im Verzeichnis `/usr/bin` installiert ist.

Der folgende `/etc/printcap` Eintrag ist für einen HP 5MP Drucker gedacht, der an einen WindowsNT-Rechner angeschlossen ist. Doch zunächst wird die Bedeutung der dort verwendeten Parameter erläutert. Weitergehende Informationen enthalten die *Linux Drucker HOWTO* und die manual pages zur Datei `printcap`.

```
cm - Kommentar
lp - Name des Gerätes, das für die Ausgabe geöffnet werden soll
sd - das sich auf dem lokalen Rechner befindende Spoolverzeichnis
     für den Drucker
af - die Datei für die Protokollierung der Druckerzugriffe
mx - die maximale Dateilaenge (Null bedeutet unbegrenzt)
if - Name des Skripts für den Eingabefilter
```

Und hier nun ein Auszug aus der `/etc/printcap` für den Drucker HP 5MP.

```
# /etc/printcap
#
# //zimmerman/oreilly über smbprint
#
lp:\
      :cm=HP 5MP Postscript OReilly an zimmerman:\
      :lp=/dev/lp1:\
      :sd=/var/spool/lpd/lp:\
      :af=/var/spool/lpd/lp/acct:\
      :mx#0:\
      :if=/usr/bin/smbprint:
```

Wichtig ist, daß man sich vergewissert, daß die Spool- und Accounting-Verzeichnisse existieren und für das Schreiben freigegeben sind.

Es muß sichergestellt werden, daß die `if`-Zeile den richtigen Pfad für das unten angegebene Skript `smbprint` enthält und daß dabei auf das richtige Ausgabegerät verwiesen wird, also die spezielle `/dev`-Datei.

Ausdruck der modifizierten Datei `smbprint`:

```
#!/bin/sh -x

# Dieses Skript ist ein Filter, der vom lp-Daemon auf-
# gerufen wird. Es benutzt das Programm smbclient, um
```

```
# eine Datei auf dem angegebenen SMB-basierten Server
# auszudrucken. Man könnte z.B. einen printcap Eintrag
# wie folgenden haben:
#
# smb:\
#   :lp=/dev/null:\
#   :sd=/usr/spool/smb:\
#   :sh:\
#   :if=/usr/local/samba/smbprint:
#
# Dieses würde einen Unix Drucker mit dem Namen "smb"
# erzeugen, der über dieses Skript druckt. Hierfür muß
# das Spool-Verzeichnis /usr/spool/smb mit den passenden
# Rechten und Besitzern auf dem  eigenen System angelegt
# werden.

# In diesem Beispiel wird ein Windows PC mit dem Namen
# "lapland" verwendet, der einen Drucker unter dem
# Namen "printer" ohne Paßwort exportiert.

# Das Skript wurde von Michael Hamilton
# (hamiltom@ecnz.co.nz) so verändert, daß der Server,
# der Dienst und das Paßwort aus einer Datei
# /usr/var/spool/lpd/PRINTNAME/.config gelesen werden.
#
# Damit dieses funktioniert, muß der /etc/printcap
# Eintrag eine Accounting Datei (af=) enthalten:
#
#   cdcolour:\
#       :cm=CD IBM Colorjet on 6th:\
#       :sd=/var/spool/lpd/cdcolour:\
#       :af=/var/spool/lpd/cdcolour/acct:\
#       :if=/usr/local/etc/smbprint:\
#       :mx=0:\
#       :lp=/dev/null:
#
# Die /usr/var/spool/lpd/PRINTNAME/.config Datei sollte
# enthalten:
#   server=PC_SERVER
#   service=PR_SHARENAME
#   password="Paßwort"
#
# z.B.:
#   server=PAULS_PC
#   service=CJET_371
#   password=""

#
# Logfile fürs Debugging, kann bei Bedarf in  /dev/null
# geändert werden.
#
logfile=/tmp/smb-print.log
# logfile=/dev/null

#
```

```
# Der letzte Parameter für den Filter ist der Name der
# Accounting Datei.
#
spool_dir=/var/spool/lpd/lp
config_file=$spool_dir/.config

# Liest die folgenden Variablen aus der Konfigurations-
# datei:
#   server
#   service
#   password
#   user
eval `cat $config_file`

#
# Hilfestellung fürs Debugging; um Platz zu sparen,
# kann ">>" in ">" geändert werden:
#
echo "server $server, service $service" >> $logfile

(
# Hinweis: Eventuell möchten Sie die Zeile "echo translate"
# hinzufügen, um eine automatische CR/LF Umsetzung beim
# Drucken durchzuführen.
      echo translate
      echo "print -"
      cat
) | /usr/bin/smbclient
         "\\\\$server\\$service" $password -U $user \
                                 -N -P >> $logfile
```

Die meisten Linux-Distributionen enthalten `nenscript`, um ASCII-Dateien in PostScript zu konvertieren. Das folgende Perl-Skript gestaltet die Handhabung jedoch einfacher, in dem es eine einfache Schnittstelle zum Drucken unter Linux mit `smbprint` zur Verfügung stellt.

```
Syntax: print [-a|c|p] <dateiname>
        -a druckt <dateiname> als ASCII
        -c druckt <dateiname> formatiert als Source Code
        -p druckt <dateiname> als PostScript
         Wird kein Schalter gesetzt, so geht das Skript davon
         aus, daß es selbst den Dateityp ermitteln und
         die Datei dann entsprechend ausdrucken soll.
```

Wird `smbprint` genutzt, um ASCII-Dateien zu drucken, so werden zu lange Zeilen abgeschnitten. Falls es möglich ist, bricht das Skript dabei die zu langen Zeilen bei Leerzeichen und nicht in der Mitte eines Wortes um.

Die Formatierung des Quellcodes erfolgt mit `nenscript`. Es wird eine ASCII-Datei genommen und zweispaltig einschließlich Kopfzeile formatiert. Diese Kopfzeile enthält u.a. das Datum und den Dateinamen. Die Zeilen werden numeriert. Wenn man dieses Skript als Beispiel verwendet, lassen sich auch andere Formatierungen gestalten.

PostScript-Dokumente sind bereits passend formatiert, so daß sie direkt weitergeleitet werden können.

Und nun das Listing des Skriptes `print`:

```
#!/usr/bin/perl
```

```
# Skript:   print
# Autoren:  Brad Marshall, David Wood
#           Plugged In Communications
# Datum:    960808
#
# Skript, um auf "oreilly" zu drucken, welcher zur Zeit
# an "zimmerman" angeschlossen ist. Das Skript nimmt
# verschiedene Typen von Dateien als Argumente an und
# bereitet sie entsprechend auf, um sie an ein Samba
# Druckskript zu übergeben.
#
# Zur Zeit werden folgende Formate unterstützt:
#
# ASCII      - Stellt sich, daß Zeilen, die länger als
#              $line_length sind, bei einem Leerzeichen
#              umgebrochen werden.
# Postscript - Keine Veränderung notwendig.
# Code       - Wird mittels nenscript als Postscript
#              formatiert, um richtig dargestellt zu
#              werden (Hochformat, Font, etc.).
#

# Setzt die maximal erlaubte Länge einer Zeile bei
# ASCII Text.
$line_length = 76;

# Setzt den Pfad und Namen des Druckskriptes von Samba
$print_prog = "/usr/bin/smbprint";

# Setzt den Pfad und Namen von nenscript
# (dem ASCII nach PS Konverter)
$nenscript = "/usr/bin/nenscript";

unless ( -f $print_prog ) {
   die "Kann $print_prog nicht finden!";
}
unless ( -f $nenscript ) {
   die "Kann $nenscript nicht finden!";
}

&ParseCmdLine(@ARGV);

# DBG
print "Dateityp ist $filetype\n";

if ($filetype eq "ASCII") {
   &wrap($line_length);
} elsif ($filetype eq "code") {
   &codeformat;
} elsif ($filetype eq "ps") {
   &createarray;
} else {
   print "Unbekannter Dateityp.\n";
   exit 0;
```

```
}
# Daten an smbprint übergeben.
open(PRINTER, "|$print_prog") ||
   die "Kann  $print_prog nicht öffnen: $!\n";
foreach $line (@newlines) {
   print PRINTER $line;
}
# Ein zusätzlichen Zeilenvorschub senden, wenn die Datei
# eine unvollständige letzte Zeile enthält.
print PRINTER "\n";
close(PRINTER);
print "Fertig\n";
exit 0;

# -------------------------------------------------- #
#       Alles weiter unten sind Unterfunktionen       #
# -------------------------------------------------- #
sub ParseCmdLine {
   # Kommandozeile parsen und den Dateityp der Datei
   # herausfinden

   # Läßt $arg und $file die Argumente, falls sie
   # existieren, und den Dateinamen werden.
   if ($#_ < 0) {
      &usage;
   }
   # DBG
#    foreach $element (@_) {
#       print "*$element* \n";
#    }

   $arg = shift(@_);
   if ($arg =~ /\-./) {
      $cmd = $arg;
      # DBG
#     print "\$cmd gefunden.\n";

      $file = shift(@_);
   } else {
      $file = $arg;
   }

   # Den Dateityp definieren.
   unless ($cmd) {
      # Wir haben keine Argumente.

      if ($file =~ /\.ps$/) {
         $filetype = "ps";
      } elsif ($file =~ /\.java$|\.c$|\.h$|\.pl$|\.sh$|\.csh$|\.m4$|\.inc$|\.html$|\.htm$/) {
         $filetype = "code";
      } else {
         $filetype = "ASCII";
      }

   } else {
```

```
        # Der Typ ist in $arg angegeben.
        if ($cmd =~ /^-p$/) {
           $filetype = "ps";
        } elsif ($cmd =~ /^-c$/) {
           $filetype = "code";
        } elsif ($cmd =~ /^-a$/) {
           $filetype = "ASCII"
        }
     }
  }

  sub usage {
     print "
  Syntax: print [-a|c|p] <dateiname>
          -a druckt <dateiname> als ASCII
          -c druckt <dateiname> formatiert als Source Code
          -p druckt <dateiname> als Postscript
           Wird kein Schalter gesetzt, so geht das Skript davon
           aus, daß es selbst den Dateityp ermitteln und
           die Datei dann entsprechend ausdrucken soll.
  ";
     exit(0);
  }

  sub wrap {
     # Erzeuge ein Array von Zeilen, wo jede Zeile weniger
     # Zeichen enthält als das angegebene Limit. Zu lange
     # Zeilen werden an Leerzeichen umgebrochen.

     # Limit der Zeichenanzahl pro Zeile besorgen
     $limit = pop(@_);

     # DBG
     #print "Zeilenumbruch Routine\n";
     #print "Das Limit für die Zeilenlänge ist $limit\n";

     # Datei lesen, parsen und in einem Array speichern.
     open(FILE, "<$file") ||
        die "Kann $file nicht öffnen: $!\n";
     while(<FILE>) {
        $line = $_;

        # DBG
        #print "Die Zeile ist:\n$line\n";

        # Zeile umbrechen, wenn sie über dem Limit ist.
        while ( length($line) > $limit ) {

           # DBG
           #print "Breche um...";

           # Die ersten $limit+1 Zeichen besorgen.
           $part = substr($line,0,$limit +1);

           # DBG
```

```
            #print "Der Teile der Zeile ist:\n$part\n";

            # Überprüfen, ob der letzte Buchstabe ein
            # Leerzeichen ist.
            $last_char = substr($part,-1, 1);
            if ( " " eq $last_char ) {
               # Wenn ja, gib den Rest aus.

               # DBG
               #print "Der letzte Buchstabe war ein Leerzeichen\n";

               substr($line,0,$limit + 1) = "";
               substr($part,-1,1) = "";
               push(@newlines,"$part\n");
            } else {
               # Wenn nein, finde das letzte Leerzeichen in
               # dem Teil der Zeile und gib die Zeile bis
               # dort aus.

               # DBG
               #print "Der letzte Buchstaben war kein Leerzeichen\n";

               # Entferne die Buchstaben nach $limit
               substr($part,-1,1) = "";
               # Kehre die Zeile um, um es leicht zu machen,
               # das letzte Leerzeichen zu finden.
               $revpart = reverse($part);
               $index = index($revpart," ");
               if ( $index > 0 ) {
                  substr($line,0,$limit-$index) = "";
                  push(@newlines,substr($part,0,$limit-$index)
                       . "\n");
               } else {
                  # Da es kein Leerzeichen in der Zeile gab,
                  # wird diese nur bis zum Limit ausgegeben.
                  substr($line,0,$limit) = "";
                  push(@newlines,substr($part,0,$limit)
                       . "\n");
               }
            }
         }
         push(@newlines,$line);
      }
      close(FILE);
   }

   sub codeformat {
      # Rufe die Funktion zum Zeilenumbruch auf.
      &wrap($line_length);

      # Schicke das Ergebnis durch nenscript, um eine
      # Postscript Datei zu erzeugen, die einige Ein-
      # stellung speziell für Source Code (Hochformat,
      # Courier Font, Zeilennummern) enthält. Drucke
      # diese zuerst in eine temporäre Datei.
```

```
   $tmpfile = "/tmp/nenscript$$";
   open(FILE, "|$nenscript -2G -i$file -N -p$tmpfile -r") ||
        die "Kann nenscript nicht öffnen: $!\n";
   foreach $line (@newlines) {
      print FILE $line;
   }
   close(FILE);

   # Lese die temporäre Datei wieder in ein Array ein,
   # so daß sie an das Druckskript von Samba übergeben
   # werden kann.
   @newlines = ("");
   open(FILE, "<$tmpfile")
      || die "Kann $file nicht öffnen: $!\n";
   while(<FILE&>) {
      push(@newlines,$_);
   }
   close(FILE);
   system("rm $tmpfile");
}

sub createarray {
   # Erzeuge das Array für Postscript
   open(FILE, "<$file")
      || die "Kann $file nicht öffnen: $!\n";
   while(<FILE>) {
      push(@newlines,$_);
   }
   close(FILE);
}
```