Linux Samba HOWTO

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

Inhalt

1 Einleitung **1**

1.1 Zur deutschen Übersetzung 1

1.2 Copyright 2

1.3 Was ermöglicht Samba? 2

1.4 Hinweis 2

2 Weitere Informationen **2**

3 Installation **3**

4 Das Starten der Dämonen **4**

5 Grundsätzliche Konfiguration der Datei smb.conf **5**

6 Nutzen eines Linux-Laufwerks durch einen Windows-Rechner **6**

7 Gemeinsame Nutzung eines Windows-Laufwerkes **7**

8 Gemeinsame Nutzung eines Linux-Druckers **8**

9 Gemeinsame Nutzung eines Windows-Druckers **9**

1 Einleitung

Da dies eine Übersetzung ist, werden die Versionsnummern beibehalten, d.h. die deutsche Übersetzung beginnt mit der Version 1.0.

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 e-mail-Adresse: Klaus-Dieter.Schumacher@fernuni-hagen.de. Die deutsche Version ist auf ftp.heise.de:/pub/ix/Linux/docs/HOWTO und auf ftp.uni-stuttgart.de:/pub/systems/linux/local/doc zu bekommen. Hier werden auch weitere deutsche Übersetzungen zu haben sein. Für die Koordination des deutschen HOWTO-Projekts ist Marco Budde (Budde@tu-harburg.d400.de) verantwortlich.

1.2 Copyright

Dieses Dokument ist urheberrechtlich geschützt. Das Copyright für die englische *Samba-HOWTO*, auf der dieses Dokument basiert, liegt bei David Wood. Das Copyright für die deutsche Übersetzung liegt bei Klaus-Dieter Schumacher.

Das Dokument darf gemäß der GNU *General Public License* verbreitet werden. Insbesondere bedeutet dieses, daß der Text sowohl über elektronische wie auch physikalische Medien ohne die Zahlung von Lizenzgebühren verbreitet werden darf, solange dieser Copyright Hinweis nicht entfernt wird. Eine kommerzielle Verbreitung ist erlaubt und ausdrücklich erwünscht. Bei einer Publikation in Papierform ist das Deutsche Linux HOWTO Projekt hierüber zu zu informieren.

1.3 Was ermöglicht Samba?

Das SMB-Protokoll (Server Message Block-Protokoll) wird von der Firma Microsoft für Windows 3.11, NT und 95 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 e-mail 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 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 einigen Linux-Distributionen - z. B. Redhat, S.u.S.E- besteht die Möglichkeit, Samba als Paket zu installieren. Bei einigen anderen Distributionen sind nur die Binärdateien von Samba als integrierter Bestandteil vorhanden.

Die folgenden zwei Dämonen werden vom Samba-Paket benötigt.

```
smbd      Hierbei handelt es sich um den SMB-Daemonen.
nmbd      Ermoeglicht den NetBIOS- Nameserver-Support für die Klienten.
```

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

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

`smbclient`

Hierbei handelt es sich um den SMB-Klienten 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 Dämonen in das Verzeichnis `/usr/bin`, die Binärdateien in das Verzeichnis `/usr/bin` und die Hilfedateien (man pages) in das Verzeichnis `/usr/local/man` kopiert.

Wenn das Samba-Paket compiliert 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.

Achtung! Vorsicht ist 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 Dämonen

Die beiden SMB-Dämonen sind `/usr/sbin/smbd` und `/usr/sbin/nmbd`. Sie können beide mit dem Skript `inetd` oder als einzelne Prozesse gestartet werden. Wenn man den Rechner dauerhaft als Dateiserver konfigurieren will, dann sollten beide mit dem Skript `inetd` gestartet werden. Hierdurch werden sie automatisch erneut aufgerufen, falls sie abgestürzt sein sollten. Wenn man die SMB-Unterstützung nur gelegentlich nutzen will oder nur als Hilfe für die Systemadministration benötigt, dann können beide mit dem Aufruf des Skriptes `/etc/rc.d/init.d` oder auch unmittelbar durch die Eingabe von `smbd` und `nmbd` gestartet werden.

Um die Dämonen mit dem Skript `inetd` aufrufen zu können, sind in die Konfigurationsdatei `/etc/inetd.conf` die nachfolgenden Zeilen einzutragen:

```
# SAMBA NetBIOS services (for PC file and print sharing)
netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd
netbios-ns  dgram  udp wait   root /usr/sbin/nmbd nmbd
```

Dann ist der inetd-Dämon erneut aufzurufen mit:

```
kill -HUP 1
```

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

```
#!/bin/sh

#
# /etc/rc.d/init.d/smb - starts and stops SMB services.
#
# The following files should be synbolic links to this file:
# symlinks: /etc/rc.d/rc1.d/K35smb  (Kills SMB services on shutdown)
#           /etc/rc.d/rc3.d/S91smb  (Starts SMB services
#                                          in multiuser mode)
#           /etc/rc.d/rc6.d/K35smb  (Kills SMB services on reboot)
#
# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0
```

```
# See how we were called.
case "$1" in
  start)
     echo -n "Starting SMB services: "
     daemon smbd -D
     daemon nmbd -D
     echo
     touch /var/lock/subsys/smb
     ;;
  stop)
     echo -n "Shutting down SMB services: "
     killproc smbd
     killproc nmbd
     rm -f /var/lock/subsys/smb
     echo ""
     ;;
  *)
     echo "Usage: 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 Resourcen das System nach außen anbietet und welche Einschränkungen hierbei festgelegt sind.

In diesem Kapitel 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 Heimatverzeichnis auf dem Linux-Rechner ermöglicht. Diese Maschine muß allerdings im lokalen Netz eingebunden sein. Versucht also ein Windows-Benutzer mit dem Windows-Dateimanager, oder dem Windows-Explorer von seinem Windows-Rechner aus zu einem dem Linux-Rechner eingestellten Netzlaufwerk herzustellen, so wird er mit seinem persönlichen Heimatverzeichnis verbunden. 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 Klienten 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 Durchfuehren der Aenderungen
;          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 Gaesten 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 das Schreibrecht nur einer bestimmten Personengruppe vergeben werden wird, 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 `man-pages` zu Samba entnommen werden.

7 Gemeinsame Nutzung eines Windows-Laufwerkes

Das SMB-Klientenprogramm 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-Klienten 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 ggfs. weitere SMB-Server, die alle über das Netzwerk verfügbar sind, einschließlich ihrer Resourcen an.

Um den Klienten 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` anzugeben `\\zimmerman\public`. Dabei sind die Restriktionen der Shell beim Entfernen von Backslashes zu beachten. In diesem Fall bedeutet es, daß der Aufruf lautet:

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

Dabei stellt 'mypasswd' den Platzhalter für die Zeichenkette des eigenen Paßwortes dar.

Der Klient `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` listet als Hilfe alle möglichen Befehle für die Benutzung des Samba-Klienten auf:

```
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 ftp nutzen kann, werden nicht die `man-pages` für den `smbclient` benötigt.

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.

Bei der Installation eines lokalen Druckers an der Linux-Maschine bietet die *Linux Drucker HOWTO* geeignete Hilfestellung.

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 Kapitels 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. Es müssen die entsprechenden Einträge aus der Datei `/etc/printcap` vorliegen. Diese müssen zur lokalen Verzeichnisstruktur passen -also zum Spool-Verzeichnis u.s.w.-.
2. Das Skript `/usr/bin/smbprint` muß vorhanden sein. Es ist Bestandteil der 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 i. allg. im Verzeichnis `/usr/bin` installiert ist.

Anmerkung

Das weiter unten vorgestellte Perl-Skript vereinfacht das Drucken unter Samba. Es ermöglicht das Drucken von ASCII, Postscript und erzeugtem Postscript.

Der Eintrag in der `/etc/printcap` bezieht sich auf einen HP 5MP-Drucker, der an einen WindowsNT-Rechner angeschlossen ist. Doch zunächst wird die Bedeutung der dort verwendeten Abkürzungen erläutert. Weitergehende Informationen enthalten die *Linux Drucker HOWTO* und die `man-pages` für die Datei `printcap`.

```
cm - Kommentar
lp - Name des Geraetes, das für die Ausgabe geoeffnet werden soll
sd - das sich auf dem lokalen Rechner befindende Spoolverzeichnis
      fuer 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 via smbprint
#
lp:\
     :cm=HP 5MP Postscript OReilly on 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 Zugriffs-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.

`smbprint`

Dies ist ein Skript, das sich i.allg. im Verzeichnis `/usr/bin` befindet. Es stammt wohl von Andrew Tridgell, dem Programmierer von Samba. Das Skript wird mit der Quellcode-Distribution von Samba ausgeliefert. Es fehlt allerdings in einigen binären Distributionen. Deshalb wird es nachfolgend aufgeführt. Dabei muß allerdings beachtet werden, daß es einige kleine Änderungen enthält. Diese haben sich im Alltagseinsatz als brauchbar erwiesen.

Ausdruck der modifizierten Datei `smbprint`:

```
#!/bin/sh -x

# This script is an input filter for printcap printing on a unix machine.
# It uses the smbclient program to print the file to the specified
# smb-based server and service.
# For example you could have a printcap entry like this
#
# smb:lp=/dev/null:sd=/usr/spool/smb:sh:if=/usr/local/samba/smbprint
#
# which would create a unix printer called "smb" that will print via
# this script. You will need to create the spool directory /usr/spool/smb
# with appropriate permissions and ownerships for your system.

# Set these to the server and service you wish to print to
```

```
# In this example I have a WfWg PC called "lapland" that has a printer
# exported called "printer" with no password.

#
# Script further altered by hamiltom@ecnz.co.nz (Michael Hamilton)
# so that the server, service, and password can be read from
# a /usr/var/spool/lpd/PRINTNAME/.config file.
#
# In order for this to work the /etc/printcap entry must include an
# accounting file (af=...):
#
#   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:
#
# The /usr/var/spool/lpd/PRINTNAME/.config file should contain:
#   server=PC_SERVER
#   service=PR_SHARENAME
#   password="password"
#
# E.g.
#   server=PAULS_PC
#   service=CJET_371
#   password=""

#
# Debugging log file, change to /dev/null if you like.
#
logfile=/tmp/smb-print.log
# logfile=/dev/null

#
# The last parameter to the filter is the accounting file name.
#
spool_dir=/var/spool/lpd/lp
config_file=$spool_dir/.config

# Should read the following variables set in the config file:
#   server
#   service
#   password
#   user
eval `cat $config_file`

#
# Some debugging help, change the >> to > if you want to same space.
#
echo "server $server, service $service" >> $logfile

(
# NOTE You may wish to add the line `echo translate' if you want
# automatic CR/LF translation when printing.
      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 Postskript 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.

```
Usage: print [-a|c|p] {\(<\)}filename{\(>\)}
       -a prints <filename> as ASCII
       -c prints <filename> formatted as source code
       -p prints <filename> as 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 um und nicht in der Mitte eines Wortes.

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. Dabei sind Postskript-Dokumente sind schon geeignet formatiert, so daß sie unmittelbar weitergeleitet werden.

Und nun das Listing des Skriptes `print`:

```
#!/usr/bin/perl

# Script:   print
# Authors:  Brad Marshall, David Wood
#           Plugged In Communications
# Date:     960808
#
# Script to print to oreilly which is currently on zimmerman
# Purpose:  Takes files of various types as arguments and
# processes them appropriately for piping to a Samba print script.
#
# Currently supported file types:
#
# ASCII      - ensures that lines longer than $line_length characters
#              wrap on whitespace.
# Postscript - Takes no action.
# Code       - Formats in Postscript (using nenscript) to display
#              properly (landscape, font, etc).
#

# Set the maximum allowable length for each line of ASCII text.
$line_length = 76;

# Set the path and name of the Samba print script
$print_prog = "/usr/bin/smbprint";

# Set the path and name to nenscript (the ASCII-->Postscript converter)
$nenscript = "/usr/bin/nenscript";

unless ( -f $print_prog ) {
```

```
        die "Can't find $print_prog!";
}
unless ( -f $nenscript ) {
        die "Can't find $nenscript!";
}

\&ParseCmdLine(@ARGV);

# DBG
print "filetype is $filetype\n";

if ($filetype eq "ASCII") {
        \&wrap($line_length);
} elsif ($filetype eq "code") {
        \&codeformat;
} elsif ($filetype eq "ps") {
        \&createarray;
} else {
        print "Sorry..no known file type.\n";
        exit 0;
}
# Pipe the array to smbprint
open(PRINTER, "|$print_prog") || die "Can't open $print_prog: $!\n";
foreach $line (@newlines) {
        print PRINTER $line;
}
# Send an extra linefeed in case a file has an incomplete last line.
print PRINTER "\n";
close(PRINTER);
print "Completed\n";
exit 0;

# ------------------------------------------------- #
#        Everything below here is a subroutine      #
# ------------------------------------------------- #
sub ParseCmdLine {
        # Parses the command line, finding out what file type the file is

        # Gets $arg and $file to be the arguments (if the exists)
        # and the filename
        if ($#_ < 0) {
                \&usage;
        }
        # DBG
#       foreach $element (@_) {
#               print "*$element* \n";
#       }

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

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

```
        }

        # Defining the file type
        unless ($cmd) {
                # We have no arguments

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

                # Process $file for what type is it and return $filetype
        } else {
                # We have what type it is in $arg
                if ($cmd =~ /^-p$/) {
                        $filetype = "ps";
                } elsif ($cmd =~ /^-c$/) {
                        $filetype = "code";
                } elsif ($cmd =~ /^-a$/) {
                        $filetype = "ASCII"
                }
        }
}

sub usage {
        print "
Usage: print [-a|c|p] <filename{\(>\)}
       -a prints <filename> as ASCII
       -c prints <filename> formatted as source code
       -p prints <filename> as Postscript
        If no switch is given, print attempts to
        guess the file type and print appropriately.\n
";
        exit(0);
}

sub wrap {
        # Create an array of file lines, where each line is < the
        # number of characters specified, and wrapped only on whitespace

        # Get the number of characters to limit the line to.
        $limit = pop(@_);

        # DBG
        #print "Entering subroutine wrap\n";
        #print "The line length limit is $limit\n";

        # Read in the file, parse and put into an array.
        open(FILE, "<$file") || die "Can't open $file: $!\n";
        while(<FILE>) {
                $line = $_;

                # DBG
```

```
                #print "The line is:\n$line\n";

                # Wrap the line if it is over the limit.
                while ( length($line) > $limit ) {

                        # DBG
                        #print "Wrapping...";

                        # Get the first $limit +1 characters.
                        $part = substr($line,0,$limit +1);

                        # DBG
                        #print "The partial line is:\n$part\n";

                        # Check to see if the last character is a space.
                        $last_char = substr($part,-1, 1);
                        if ( " " eq $last_char ) {
                            # If it is, print the rest.

                            # DBG
                            #print "The last character was a space\n";

                            substr($line,0,$limit + 1) = "";
                            substr($part,-1,1) = "";
                            push(@newlines,"$part\n");
                        } else {
                             # If it is not, find the last space in the
                             # sub-line and print up to there.

                            # DBG
                            #print "The last character was not a space\n";

                             # Remove the character past $limit
                             substr($part,-1,1) = "";
                             # Reverse the line to make it easy to find
                             # the last space.
                             $revpart = reverse($part);
                             $index = index($revpart," ");
                             if ( $index > 0 ) {
                               substr($line,0,$limit-$index) = "";
                               push(@newlines,substr($part,0,$limit-$index)
                                    . "\n");
                             } else {
                               # There was no space in the line, so
                               # print it up to $limit.
                               substr($line,0,$limit) = "";
                               push(@newlines,substr($part,0,$limit)
                                    . "\n");
                             }
                        }
                }
                push(@newlines,$line);
        }
        close(FILE);
}

sub codeformat {
```

```
        # Call subroutine wrap then filter through nenscript
        \&wrap($line_length);

        # Pipe the results through nenscript to create a Postscript
        # file that adheres to some decent format for printing
        # source code (landscape, Courier font, line numbers).
        # Print this to a temporary file first.
        $tmpfile = "/tmp/nenscript$$";
        open(FILE, "|$nenscript -2G -i$file -N -p$tmpfile -r") ||
                die "Can't open nenscript: $!\n";
        foreach $line (@newlines) {
                print FILE $line;
        }
        close(FILE);

        # Read the temporary file back into an array so it can be
        # passed to the Samba print script.
        @newlines = ("");
        open(FILE, "<$tmpfile") || die "Can't open $file: $!\n";
        while(<FILE&>) {
                push(@newlines,$_);
        }
        close(FILE);
        system("rm $tmpfile");
}

sub createarray {
        # Create the array for postscript
        open(FILE, "<$file") || die "Can't open $file: $!\n";
        while(<FILE>) {
                push(@newlines,$_);
        }
        close(FILE);
}
```