

# Inhaltsverzeichnis

Was ist Samba?.....	2
Geschichte.....	2
Installation .....	3
1. Woher bekomme ich Samba ?.....	3
2. Die Installation des enthaltenen Samba- Paketes .....	3
3. Ein Samba Startskript .....	5
4. SWAT aktivieren .....	6
Eine einfache Basis- Konfiguration .....	6
Die Konfigurationsdatei testen .....	7
Der [homes] Abschnitt .....	8
Spezielle Konfigurationen .....	8
Zugriffsbeschränkungen .....	9
Samba als PDC.....	9
Nutzen von Windows Freigaben .....	12
1. smbclient.....	12
2. smbmount .....	13
Datensicherung mit smbclient.....	14
Drucken mit Samba .....	14
1. Zugriff auf Windows- Drucker .....	14
2. Freigabe von Linux- Druckern .....	14
Nützliche Zusatzoptionen .....	15
Verbindungsskripte .....	15
Admin Rechte .....	15
Verstecken von Dateien .....	15
Links folgen .....	15

## Was ist Samba?

Linux ist auf dem Vormarsch. Nicht nur springen immer mehr Firmen auf den Zug, auch traditionelle Linux-Firmen wie Red Hat oder SuSE sind am Expandieren. Und trotz des Vormarsches ist Microsoft vor allem auf dem Desktop noch nicht wirklich angegriffen.

Wer auf dem Desktop nicht auf Microsoft verzichten kann oder aus Firmengründen nicht umsteigen darf, wird sicherlich keine Lust haben, sich auch noch auf dem Server mit Microsoft-Produkten herumzuschlagen. Um dieses Problem zu beheben, kann Linux in der Kombination mit Samba eingesetzt werden, um einen SMB-Server aufzusetzen.

Samba ermöglicht es, einen MS-Windows-Server unter Linux und vielen anderen Unix-Systemen aufzubauen. Samba unterstützt eigentlich alle Clients, von MS-Clients für DOS bis Windows NT. Einzige Voraussetzung ist das gemeinsame Protokoll TCP/IP mit NBT (NetBios over TCP/IP). Besonders bei der Fehlersuche sollte nicht vergessen werden, dass Samba nur über TCP/IP funktioniert.

Samba vermittelt zwischen zwei Welten. Daher gibt es einige Problemstellen, an denen die Welten einfach nicht recht zusammenpassen wollen. Dazu gehören sowohl die Dateirechte als auch die Benutzer-Gruppen-Zuordnung.

## Geschichte

Der Beginn der Entwicklung wurde von Andrew Tridgell am Computer Science Lab an der Australia National University, Canberra, gelegt, als er versuchte, DEC's Pathwork (ein SMB-kompatibler Server) mit einer Sun zu verbinden. Da die Software nicht für Sun verfügbar war, begann er die Kommunikation zwischen dem Client und dem Server zu analysieren. Irgendwann hatte er das Protokoll soweit zerlegt, dass er beginnen konnte die Serversoftware zu entwickeln. Im Januar 1992 wurde "Server 0.1" veröffentlicht. Es gab auch einige Nachfragen und schon bald wurden "Server 0.5" und "Server 1.0" im Abstand von jeweils einem Monat veröffentlicht. Doch dann schlief die Entwicklung ein wenig ein, da Andrew keine Verwendung mehr für seinen "Server" hatte. Doch nachdem Andrew Linux entdeckt hatte und den PC seiner Frau an seinen Linux-Server anschließen wollte, portierte er seinen "Server" auf Linux und es lief auf Anhieb. Er fand dann auch bald die lange gesuchten Protokoll-Spezifikationen und begann daraufhin seinen Code zu verbessern. Danach wurde aus dem SMB-"Server" SaMBa.

Wer die ganze Geschichte lesen möchte, dem sei der Text `history` im Document-Verzeichnis zu empfehlen.

# Installation

## 1. Woher bekomme ich Samba ?

Quell- und Binärpakete gibt es auf unzähligen mirrors im Internet, die Samba-Hauptseite ist [www.samba.org](http://www.samba.org) von dort können sie einen mirror in ihrer Nähe wählen.

Die meisten Linux- Distributoren bieten bereits eine vor kompilierte Sambaversion an welche dann auch sehr einfach zu Installieren ist, jedoch nicht immer alle Funktionen der Samba-Suite unterstützt.

## 2. Die Installation des enthaltenen Samba- Paketes

CD oder DVD mounten, z.B. nach /mnt

```
mount /dev/cdrom /mnt
```

dann in den Ordner mit den Paketen wechseln

```
cd /mnt/suse/i586/
```

und nun die Pakete: samba, samba-client und samba-common installieren

```
rpm -ihv samba
```

```
rpm -ihv samba-client
```

```
rpm -ihv samba-common
```

## 3. Installation der Source-Code Version

Samba Downloaden

```
wget http://us4.samba.org/samba/ftp/stable/samba-3.0.22.tar.gz
```

nun das tar.gz unzippen und dann auspacken, oder beides auf einmal!

```
gunzip samba-3.0.22.tar.gz
```

```
tar -xvf samba-3.0.22.tar
```

oder:

```
tar -xvzf samba-3.0.22.tar.gz
```

nun in den entstandenen Ordner wechseln

```
cd samba-3.0.22/source
```

Weiter wird nach der 3er Regel installiert configure, make und make install erstmal durch aufrufen des configure Befehls anzeigen lassen welche Optionen es gibt, dies wird mehr sein als auf einen Bildschirmseite darstellbar ist, deshalb die Ausgabe in einen Betrachter umleiten, z.B. less !

`configure help | less`

Nun sollte man sich diese Optionen genau ansehen und entscheiden welche davon man benötigt, in diesem Kurs werden wir nur wenige brauchen.

Hier einige Optionen und ihre Bedeutungen

<b>Option</b>	<b>Bedeutung</b>	<b>Standardwert</b>
<code>--prefix=verzeichnis</code>	Installiert Dateien im angegebenen Basisverzeichnis	<code>/usr/local/samba</code>
<code>--eprefix=verzeichnis</code>	Installiert Dateien im angegebenen Basisverzeichnis	<code>/usr/local/samba</code>
<code>--bindir=verzeichnis</code>	Benutzer ausführbare Dateien	<code>eprefix/bin</code>
<code>--sbindir=verzeichnis</code>	Administrator ausführbare Dateien	<code>eprefix/sbin</code>
<code>--libdir</code>	Installiert Programm-Bibliothek im Verzeichnis	<code>eprefix/lib</code>

Wir werden nur ein paar der Optionen verwenden:

```
./configure --with-smbwrapper --with-smbmount --with-msdfs \
--with-syslog --with-utmp 2>&1 | tee config.my.log
```

Der Befehl „tee“ wird hier verwendet um die Ausgabe auf dem Bildschirm da zu stellen und gleichzeitig in eine Datei zu schreiben.

Nun fahren wir mit dem „make“ Befehl fort:

```
make 2>&1 | tee make.log
```

Sollte auch dieser Befehl problemlos abgelaufen sein, kommen wir nun zum letzten Schritt dem „make install“

```
make install 2>&1 | tee install.log
```

Nun sollte Samba installiert sein, jedoch müssen wir noch die Pfade setzen um Komfortabel arbeiten zu können.

„`/usr/local/samba/sbin`“ sollte im root PATH enthalten sein:

```
echo export PATH=$PATH:/usr/local/samba/sbin >> /root/.profile
```

und „`/usr/local/samba/bin`“ sollte bei allen Benutzern im PATH sein:

```
echo export PATH=$PATH:/usr/local/samba/bin >> /etc/profile
```

nun noch der MANPATH damit die manpages auch gefunden werden:

```
echo export MANPATH=$MANPATH:/usr/local/samba/man
>> /etc/profile
```

nun müssen die Einstellungen auch auf die aktuelle Sitzung übernommen werden:

```
source /root/.profile ; source /etc/profile
```

### 3. Ein Samba Startskript

Nun sollte ein Startskript erstellt werden um nicht jedes mal beide Dienste durch Eingabe des ganzen Pfades zu starten. Diese Skript sollte sich im Ordner /etc/init.d befinden, nennen wir es samba

```
vi /etc/init.d/samba
```

```
case "$1" in
start)
    [ -f /usr/local/samba/lib/smb.conf ] || exit 0
    echo " Samba wird gestartet ... "
    /usr/local/samba/sbin/smbd -D -s /usr/local/samba/lib/smb.conf
    /usr/local/samba/sbin/nmbd -D -s /usr/local/samba/lib/smb.conf
    ;;
stop)
    echo " Samba wird beendet ... "
    pkill smbd
    pkill nmbd
    ;;
restart)
    echo " Samba wird neu gestartet ... "
    $0 stop
    $0 start
    ;;
*)
    echo " Falsche Eingabe, versuchen Sie: $0 start | stop | restart "
    ;;
```

Dieses Skript sollte das Starten und beenden von Samba erleichtern!

Für die SuSE rcSkript- Fan's kann man nun noch einen Link erstellen:

```
ln -s /etc/init.d/samba /usr/sbin/rcsamba
```

nun kann Samba an jeder beliebigen stelle durch Aufruf des rcSkrits gestartet werden.

```
rcsamba start
```

## 4. SWAT aktivieren

Swat ist das (S)amba (W)eb (A)ministrations (T)ool. SWAT wird über den inetd oder den xinetd gestartet. Zuerst in der Datei /etc/services den Eintrag „swat 901/tcp“ hinzufügen oder das Kommentar Zeichen entfernen!

Wird auf dem System der initd verwendet muss in der Datei /etc/inetd.conf folgende Zeile eingetragen werden:

```
swat stream tcp nowait root /usr/local/samba/bin/swat swat
```

Wird der xinetd verwendet, so muss eine Datei namens swat im Verzeichnis /etc/xinetd.d erstellt werden, mit folgendem Inhalt:

```
service swat
{
    socket_type    = stream
    wait          = no
    protocol      = tcp
    user          = root
    server        = /usr/local/samba/bin/swat
    port          = 901
    disable       = no
}
```

nun müssen der inetd oder der xinetd neu gestartet werden:

```
/etc/init.d/(x)inetd restart
```

Nun sollte Sie SWAT über <http://localhost:901> mit einem Browser aufrufen können!

## Eine einfache Basis- Konfiguration

Die Konfiguration für den Samba- Server werden in der Datei „smb.conf“ vorgenommen, diese sollte sich im Ordner /usr/local/samba/lib befinden.

Die Datei ist in mehrere Abschnitte unterteilt. Jeder Abschnitt beginnt mit dem Abschnitts- Namen, der in [ ] eingefasst ist. Es gibt den Abschnitt [global] in dem globale Konfigurationen eingetragen werden. Weiterhin gibt es für jede Freigabe einen Abschnitt dessen Name der spätere Freigabename ist, z.B [Test]

```
[global]
```

```
workgroup = Schulung
```

```
[test]
```

```
comment = Bitte nur zu Testzwecken verwenden
```

```
path = /usr/local/samba/tmp
```

```
read only = no
```

```
guest ok = no
```

Diese noch sehr kurze Konfigurationsdatei weist den Samba-Server an den Ordner „/usr/local/samba/tmp“ auf dem Server als SMB-Freigabe namens „test“ anzubieten. Des Weiteren wird der Server zu einem Teil der Arbeitsgruppe „Schulung“

nun existiert dieser Ordner noch nicht, also werden wir ihn erstellen und die Rechte ändern, jetzt noch sehr großzügig, auf Sicherheit kommen wir später noch zu sprechen!

```
mkdir /usr/local/samba/tmp
```

```
chmod 777 /usr/local/samba/tmp
```

Da Samba die Benutzer anhand von Benutzernamen und Password identifizieren möchte müssen nun auch noch Samba- Benutzer angelegt werden, denn Samba benutzt eine andere Password- Verschlüsselung als UNIX um mit Windows Kompatibel zu sein.

Für jeden Samba- Benutzer muss bereits ein gleichnamige UNIX- Benutzer existieren. Um einen Samba- Benutzer zu erstellen wird der Befehl „smbpasswd“ verwendet:

```
smbpasswd -a lxuser
```

nun können Sie für den Benutzer „lxuser“ ein Samba- Password vergeben, welches nicht mit dem UNIX- Password identisch sein muss.

Wenn alle diese Schritte erfolgreich erledigt wurden, sollte Sie jetzt von einem Windows- Rechner auf Ihre Freigabe Test zugreifen können.

## Die Konfigurationsdatei testen

Wenn sie die Konfigurationsdatei nicht mit SWAT sondern mit dem VI erstellt haben sollte sie diese vor der Verwendung auf Syntax- Fehler überprüfen. Extra hierfür gibt es ein Tool „testparm“

```
testparm /usr/local/samba/lib/smb.conf
```

Im ersten Teil der Ausgabe stehen Hinweise und Warnungen zu Ihrer Konfiguration, im zweiten Teil wird Ihre Konfiguration ausgegeben!

Sollte alles in Ordnung sein, können Sie ihren Samba- Server starten, mit dem Befehl:

```
smbclient -U% -L localhost
```

können Sie testen ob ihr Server läuft!

## Der [homes] Abschnitt

Der homes- Abschnitt ist ein besonderer Abschnitt der nur einer minimalen Konfiguration bedarf

```
[homes]
    comment = Home Verzeichnisse
    valid users = %S
    read only = No
    browseable = No
```

mit dieser sehr Kurzen Konfiguration ist es gegeben das jeder, der sich an dem Samba- Server anmeldet, Zugriff auf seinen Home- Ordner hat, jedoch nur auf seinen, alle anderen werden nicht angezeigt.

## Spezielle Konfigurationen

<b>Variable</b>	<b>Funktionen</b>
%L	Netbiosname des Samba- Servers
%u	Username des verbundenen Benutzers
%g	Gruppe des verbundenen Benutzers
%m	Netbiosname des Clients
%a	Client Architektur (Win95,Win98,Win2K,WinNT,WinXP)
%I	IP-Adresse des Clients
%T	Aktuelles Datum und Uhrzeit
%H	Home- Verzeichnis von %u
%D	Arbeitsgruppe oder Domain von %u

Mit Hilfe dieser Variablen, kann man Samba also Dynamisch konfigurieren, man kann zum Beispiel den Inhalt eines Ordners davon abhängig machen, welcher Benutzer sich gerade am Server anmeldet, hierfür ein Beispiel:

```
mkdir -p /testdaten/{m_hartig,testuser,lxuser}
chmod -R a+rxw /testdaten
```

In der smb.conf

```
[testfreigabe]
    path = /testdaten/%u
    read only = no
    browseable = yes
```

Nun sollte der Benutzer m\_hartig über die Freigabe „testfreigabe“ im Ordner „/testdaten/m\_hartig“ landen und der Benutzer lxuser im Ordner „/testdaten/lxuser“

## Zugriffsbeschränkungen

Es gibt mehrere Möglichkeiten fest zu legen welcher Benutzer auf welche Freigabe zugreifen darf. Die beste und sicherste Methode ist es, die Benutzer, Gruppen und Rechteverwaltung des UNIX-Systems zu nutzen, jedoch oft mit hohem Planungsaufwand verbunden. Deshalb kann man in der Samba-Konfigurationsdatei festlegen welcher Benutzer auf Freigabe zugreifen darf. Erläuterung an unserer ersten Freigabe

```
[test]
```

```
comment = Bitte nur zu Testzwecken verwenden
path = /usr/local/samba/tmp
read only = no
guest ok = no
valid users = m_hartig @users
```

Durch den Eintrag `valid users = m_hartig @users` weisen wir den Server an nur dem Benutzer „m\_hartig“ und allen Mitgliedern der Gruppe „users“ Zugriff auf diese Freigabe zu gewähren. Der Eintrag `guest ok = no` besagt das Personen die sich nicht mit Benutzernamen und Password am Server Identifiziert haben keinen Zugriff auf diese Freigabe Haben.

Weiterhin kann man auch anhand von Rechnernamen oder IP-Adressen festlegen wer Zugriff hat und wer nicht.

```
[test]
```

```
comment = Bitte nur zu Testzwecken verwenden
path = /usr/local/samba/tmp
read only = no
guest ok = no
valid users = m_hartig @users
host allow = 172.16.0.2
```

Nun dürfen nur noch die Benutzer „m\_hartig“ und die Gruppe „users“ vom Rechner mit der IP „172.16.0.2“ aus, auf diese Freigabe zugreifen!

## Samba als PDC

Kurze Erläuterung:

Ein Privat Domain Controller kann die Administration in einem großen Netzwerk erheblich erleichtern, sollte jedoch wohl überlegt sein und genau konfiguriert werden. Mit einem PDC gibt es die Möglichkeit, das jeder Benutzer sich an jedem Rechner im Netzwerk anmelden kann, und man eine Zentrale Benutzer und Gruppenverwaltung hat. In heterogenen Netzwerken, liegt der größte Vorteil, einen Samba- PDC zu verwenden darin, dass man auch einen NIS- Server oder LDAP nutzen kann um auch Linux- Clients mit dem selben Rechner zu verwalten, auch ist der Samba- PDC nicht an Lizenzen gebunden, dass heißt das sie beliebig viele Clients ohne zusätzliche kosten

anbinden können. Hier die Konfigurationsdatei, die später erläutert wird.

```
[global]
    netbios name = PDC
    workgroup = testdomain.de
    domain master = yes
    local master = yes
    preferred master = yes
    os level = 255
    domain logons = yes
    logon path = \\%L\profiles\%u
    logon script = logon.bat
    logon drive = X:
    time server = yes
    domain admin group = root
    wins support = yes

[netlogon]
    path = /testdaten/netlogon
    read only = yes
    browseable = no

[profiles]
    path = /testdaten/profiles
    read only = no
    browseable = no
    create mask = 0600
    directory mask = 0700

[homes]
    read only = no
    browseable = no
    guest ok = no
    map archive = yes
```

Nun die Erläuterung:

- „workgroup“      wird bei einem PDC als Domain- Name
- „domain master“ Der Server ist der Haupt- Domain- Controller
- „local master“    Der Server ist der Haupt- Domain- Controller für die lokale Domain
- „preferred master“    Der Server ist der bevorzugte PDC, bei mehreren

Domain's wird er als erster befragt

- „os level“ Der Server sollte den höchst möglich OS Level haben um zu gewährleisten, dass er Wahlen zum Domain Controller gewinnt
- „domain logons“ An dieser Domain darf sich angemeldet werden
- „logon path“ Hier befinden sich die Roaming- Profile
- „logon script“ Das hier angegebene Skript wird beim Anmelden an die Domäne auf dem Client ausgeführt und muss sich in der „netlogon“ Freigabe befinden
- „logon drive“ Das Home- Verzeichnis des Benutzers wird auf dem Clientrechner als Laufwerk X gemappt
- „time server“ Weist den Server an als Zeit- Server zu arbeiten
- „domain admin group“ Hier aufgeführte Benutzer können administrative Aufgaben ausführen z.B. Rechner zur Domain hinzufügen.
- „wins support“ Der Samba- Server arbeitet als WINS- Server, damit Clients wissen welcher Rechner für ihre Domain zuständig ist

Wenn das Programm „testparm“ nichts dagegen hat, können sie nun den Samba- Server starten. Wichtig ist, dass nur ein Server im Netzwerk für die selbe Domain zuständig ist! Deshalb sollte auch die „netbios aliases“ Option nicht verwendet werden!

Nun muss es für jeden Rechner der Mitglied in der Domain sein soll einen so genannten Maschinen Account geben, diese werden ähnlich wie Benutzeraccounts verwaltet, werden jedoch niemals dazu genutzt um sich als Benutzer am Server an zu melden, deshalb sollten Sie kein Home- Verzeichnis und keinen Standardshell haben! Wichtig ist das Maschinen Account's mit einem „\$“ Zeichen enden! Ein Beispiel für einen Rechner mit dem Namen „test-rechner“ :

```
useradd -d /dev/null -s /bin/false -c RECHNERZUGANG test-rechner$  
smbpasswd -a test-rechner$
```

Nun sollte es möglich sein den Rechner zur Domain hinzuzufügen.

Das logon Skript:

die Datei logon.bat sollte mit einem Windows Editor erstellt werden, da Linux einen andere Formatierung verwendet. In dieser Datei können normale Windows Befehle enthalten sein z.B.:

```
net use T: \\PDC\test
```

um die Freigabe „test“ auf dem Rechner „PDC“ als Laufwerk T zu mappen

```
net time \\PDC /set /yes
```

um Die Uhrzeit auf dem Client mit der des Servers zu synchronisieren

# Nutzen von Windows Freigaben

## 1. smbclient

Mit dem „smbclient“ kann man sich Freigaben auf einem Windows- oder Samba- Server anzeigen lassen z.B.

```
smbclient -L 192.168.0.2
```

hier die Ausgabe:

```
Domain=[ITC] OS=[Unix] Server=[Samba 3.0.10]
```

<i>Sharename</i>	<i>Type</i>	<i>Comment</i>
-----	----	-----
<i>Software</i>	<i>Disk</i>	<i>für alle Betriebssysteme</i>
<i>IPC\$</i>	<i>IPC</i>	<i>IPC Service (sokrates)</i>
<i>ADMIN\$</i>	<i>IPC</i>	<i>IPC Service (sokrates)</i>

```
Anonymous login successful
```

```
Domain=[ITC] OS=[Unix] Server=[Samba 3.0.10]
```

<i>Server</i>	<i>Comment</i>
-----	-----
<i>SOKRATES</i>	<i>sokrates</i>
<i>WINSERVER</i>	
<i>Workgroup</i>	<i>Master</i>
-----	-----
<i>ITC</i>	<i>WINSERVER</i>

Im ersten Abschnitt sehen wir die Freigaben auf dem Server, im zweiten Abschnitt sehen wir Informationen über die Arbeitsgruppe.

Nun kann man eine Verbindung zu der Freigabe herstellen, in diesem Beispiel wird davon ausgegangen, dass man sich am Server Authentifizieren muss!

```
smbclient //192.168.0.2/Software -U m_hartig%123
```

mit der Option „-U“ gibt man Benutzernamen und Passwort durch ein „%“ Zeichen getrennt an. Mit der Option „-A“ könnte man eine Datei angeben, in der, der Benutzername und das Passwort enthalten sind, angeben!

```
smbclient //192.168.0.2/Software -A /path/zur/datei
```

Inhalt der Datei:

```
username = m_hartig
```

```
password = 123
```

Wenn Sie nun am Windows- oder Samba- Server angemeldet sind, funktioniert der „smbclient“ ähnlich wie ein „FTP- Client“, Sie können „help“ eingeben um eine List möglicher Befehle zu bekommen. Mit „get“ können Sie Dateien downloaden, mit „put“ uploaden und mit „mget“ mehrerer Dateien downloaden!

## 2. smbmount

Man kann Windows- Freigaben aber auch ins Dateisystem einbinden, das ist wahrscheinlich die beste Variante, da es für den Benutzer vollkommen Transzparent ist, und sich alles so verhält als wäre es ein lokales Dateisystem. Ich werde mich hier auf das selbe Beispiel wie beim „smbclient“ beziehen. Natürlich brauchen wir auch hier wieder einen Benutzernamen und ein Passwort. Weiterhin muß es im Dateisystem einen leeren Ordner geben, der als Mountpoint verwendet werden kann.

```
mkdir /smb
```

und nun können wir die Verbindung herstellen:

```
smbmount //192.168.0.2/Software /smb -o username= m_hartig , \
password= 123
```

nun sollte über den Ordner „/smb“ der Inhalt der „Software“ Freigabe des Rechners „192.168.0.2“ erreichbar sein. Mit der Option -o kann man nun Optionen angeben z.B. „ro“ um die Freigabe, schreibgeschützt zu mounten. Oder die Option „credentials=/path/zur/datei“ kann verwendet werden um wieder eine Datei anzugeben in der Benutzernamen und Passwort enthalten sind, hier kann die selbe Datei wie für den „smbclient“ verwendet werden. Weitere Optionen:

<b>Option</b>	<b>Funktion</b>
username	Benutzername zur Authentifizierung
password	Password zur Authentifizierung
credentials	Angabe der Datei die Benutzernamen Passwort enthält
uid	Benutzer ID die auf des gmountete Dateisystem angewendet wird
gid	Gruppen ID die auf des gmountete Dateisystem angewendet wird
fmask	auf Dateien anzuwendende Zugriffsrechte
dmask	auf Ordner anzuwendende Zugriffsrechte
guest	unterdrückt die Kennwortabfrage
ro	mountet schreibgeschützt
rw	mountet les-/schreibbar

## Datensicherung mit smbclient

Der „smbclient“ ist auch in der Lage Datensicherungen von Windows-Freigaben zu erstellen. Zuerst eine Verbindung zur Freigabe aufbauen:

```
smbclient //192.168.0.2/Software -U m_hartig%123
```

dann in der „smbclient“ Sitzung den Backup- Modus setzen:

```
tarmode full hidden system
```

dann mit dem Tar Befehl das Archiv erzeugen.

```
tar c test.tar
```

nun sollte sich das Tar- Archiv „test.tar“ in dem Ordner befinden von dem aus Sie den „smbclient“ gestartet haben.

## Drucken mit Samba

### 1. Zugriff auf Windows- Drucker

Da Samba ins System mit eingebunden ist, können Sie Windows Drucker nun über YaST oder das KDE- Kontrollzentrum einbinden.

### 2. Freigabe von Linux- Druckern

Mit dem Abschnitt [printers] kann man alle im System installierten Drucker auf sehr einfache weise freigeben

```
[printers]
```

```
printable = yes
```

```
printing = CUPS
```

```
printcap name = /etc/printcap
```

```
path = /var/spool/lpd/tmp
```

der angegebene Path muss existieren und samba muss Schreibzugriff darauf haben. Dieser Ordner wird für die Ablage von Spool-Dateien verwendet.

Einen speziellen Drucker freigeben:

```
[HP-Laser]
```

```
printable = yes
```

```
printing = CUPS
```

```
printcap name = /etc/printcap
```

```
path = /var/spool/lpd/tmp
```

```
printer = HP-Laser-Drucker
```

Über die Option „printer“ muss der Name des Druckers auf dem UNIX-System angegeben werden, nun kann man den Zugriff auf diesen Drucker genauso wie bei Verzeichnis- Freigaben einrichten.

# Nützliche Zusatzoptionen

## **Verbindungsskripte**

mit der Option

```
postexec = /path/zum/skript
```

wird ein Skript vor dem Verbindungsaufbau ausgeführt, mit

```
preexec = /path/zum/skript
```

wird ein Skript nach Beendigung der Verbindung ausgeführt

## **Admin Rechte**

mit der Option

```
admin user = lxuser
```

bekommt der Benutzer „lxuser“ root- Rechte auf die Freigabe

## **Verstecken von Dateien**

die Option

```
hide files = /*.*/*
```

versteckt alle Dateien die mit einem „.“ beginnen, mit der Option

```
veto files = /*b/*
```

wird der Zugriff auf alle Dateien die mit einem „b“ Beginnen verweigert, diese werden auch nicht angezeigt

## **Links folgen**

Option

```
follow symlinks = yes
```