

Linux, Mac und Windows in einem Netzwerk

Drei Welten prallen aufeinander: Ein Windows-Computer, ein Linux-System und ein Macintosh. Was auf dem Messestammtisch für heftige Diskussionen sorgt und Freundschaften spaltet, spielt hier keine Rolle. Über das Netzwerk finden alle Systeme zueinander. Dieser Beitrag zeigt, wie es geht.

In heterogenen Netzwerken gibt es keine Konkurrenz der Systeme, es gibt Kompatibilität und einträchtigen Datenaustausch. Das ist ideal für kleine Unternehmen, etwa Agenturen: Die Kreativen tummeln sich am Mac, die Programmierer nehmen Linux und die Verwalter den Windows-Rechner.

Grundlage aller Kommunikation ist die Netzwerk-Hardware. Die muss kompatibel sein. Sprich, wer einen I-Mac oder einen anderen aktuellen Mac sein eigen nennt, hat bereits einen kompatiblen Ethernet-Anschluss. Der Linux- und der Windows-Rechner brauchen je eine Netzwerkkarte. Schließlich benötigt der Netzwerker noch einen Hub oder Switch, und schon kann er sich an die Arbeit machen. Hub an die Steckdose, Netzwerkkabel an Hub und Computer stöpseln. Fertig.

Jetzt kommt die eigentliche Arbeit: Den Datenaustausch zwischen Mac, Linux und Windows software-seitig einrichten.

Datenaustausch zwischen Windows und Linux

Der Datenaustausch zwischen Linux und Windows hat einen Namen: Samba. Das ist ein Linux-Programm, das zwischen Windows-Netzen und Linux vermittelt. Die Software regelt Zugriffsrechte, bietet Netzwerkdrucker an oder fügt sich in eine Windows-Domäne ein.

Von Windows auf Linux zugreifen

Um von Windows auf einen Linux-Server zuzugreifen, müssen Sie einen Samba-Server installieren. Bei den meisten Linux-Distributionen ist Samba bereits enthalten. Wer sie nicht auf den CDs findet oder die topaktuelle Version braucht, sieht im Internet nach. Unter www.samba.org gibt es die Software zum Selbst-Kompilieren und als Binärversion. Ist Samba installiert, geht es an die Einrichtung des Servers. Hierfür müssen Sie als Benutzer root angemeldet sein. Die folgenden Schritte zeigen an Hand von Suse Linux, wie es geht.

Schritt 1: Benutzeraccounts einrichten

Zuerst müssen Sie für jeden Windows-Nutzer einen Account unter Linux einrichten, damit dieser überhaupt auf Samba und auf die frei gegebenen Verzeichnisse zugreifen kann. Das erledigen Sie mit

```
useradd benutzername
```

Danach vergeben Sie mit

passwd benutzername

ein Kennwort. Im Idealfall sollte das Passwort zum Einsatz kommen, das auch unter Windows gilt. Damit ersparen Sie sich und Ihren Kollegen die doppelte Kennworteingabe. Fügen Sie den neuen Benutzer der Benutzerliste von Samba hinzu:

smbpasswd -a benutzername

Dabei verlangt Samba wieder ein Kennwort. Falls keine Sicherheitsbedenken bestehen, verwenden Sie nochmals das zuvor schon vergebene Kennwort. Wiederholen Sie die hier beschriebenen Schritte für jeden Samba-Nutzer.

Wichtig: Die Benutzer müssen Sie auch dann einrichten, wenn Sie ein Laufwerk für alle Benutzer zum Schreiben und Lesen frei geben wollen. Denn die neueren Windows-Versionen wollen beim ersten Kontakt zu Samba ein verschlüsseltes Kennwort austauschen und verweigern den Login, falls es nichts zu tauschen gibt.

Schritt 2: Verzeichnisse anlegen

Als nächstes steht die Entscheidung an, welche Verzeichnisse frei gegeben werden. Für's erste reicht ein Ordner. Den legen Sie am besten unterhalb des home-Verzeichnisses an:

```
cd /home  
mkdir pub
```

Danach muss das Verzeichnis noch für die Gruppe user frei gegeben werden. Zu dieser Gruppe gehören auch die vorher eingetragenen Samba-Nutzer. Die Freigabe für die Benutzergruppe erledigt der Befehl

```
chgrp users pub
```

Zusätzlich gilt es noch, der Gruppe die richtigen Rechte einzuräumen. Dazu dient

```
chmod 770 pub
```

Damit ist das Verzeichnis für die Arbeit mit Samba vorbereitet.

Schritt 3: Samba konfigurieren

In der aktuellen Suse-Distribution 7.3 verstecken sich die Konfigurationsdateien im Verzeichnis `/etc/samba`.

Editieren Sie die Datei `smb.conf`. Nach einigen Kommentarzeilen finden Sie den ersten wichtigen Eintrag unterhalb von `[global]`. Tragen Sie neben `workgroup` den Namen Ihrer Arbeitsgruppe ein. Wenn Sie wollen, tragen Sie zusätzlich diese Zeile ein:

```
server string = servername
```

An Stelle von `servername` verwenden Sie einen aussagekräftigen Namen für den neuen

Server, zum Beispiel "Datenaustausch" oder "public".

Einige Zeilen weiter unten finden Sie den Eintrag `interfaces`. Tragen Sie hier die IP-Adresse Ihres Linux-Rechners ein. Denken Sie daran, auch die Subnetz-Maske entsprechend anzupassen und vergessen Sie nicht, das Semikolon am Anfang der Zeile zu entfernen.

Nun gilt es nur noch, das in Schritt 2 eingerichtete Verzeichnis frei zu geben. Dazu gehen Sie ganz ans Ende der Datei und fügen einen neuen Eintrag ein:

```
[pub]
public =yes
writeable = yes
path = /home/pub
```

Die erste Zeile trägt in den eckigen Klammern den Namen des Verzeichnisses wie er im Netzwerk erscheint. Mit `public` darf jeder Nutzer darauf zugreifen, mit `writeable` sogar in den Ordner schreiben. Die letzte Zeile `path` enthält den Pfad des Verzeichnisses auf dem Linux-Rechner.

Wichtig: Zugriffsrechte wie `writeable` funktionieren nur, wenn der entsprechende User auch unter Linux Schreibzugriff auf dieses Directory hat.

Schritt 4: Samba testen

Wagen Sie den ersten Testlauf. Speichern Sie die Konfigurationsdatei und rufen Sie den Befehl `testparm` auf. Das Utility überprüft die `smb.conf` auf Syntax-Fehler. Logischen Fehlern wie falsch frei gegebenen Verzeichnisnamen oder Rechten allerdings kommt `testparm` nicht auf die Spur.

Verläuft `testparm` zufrieden stellend geht es an den ersten Start von Samba. Dazu geben Sie den Befehl

```
r smb start
```

ein. Läuft alles glatt, meldet sich Samba zum Dienst.

Um nachzusehen, ob die Verzeichnisse korrekt frei geschaltet sind, verwenden Sie den Befehl

```
smbclient -L servername
```

Statt `servername` geben Sie den Hostnamen Ihres Linux-Rechners ein. Die Frage nach dem Kennwort übergehen Sie mit einem `[Return]` und schon sehen Sie die für die Öffentlichkeit auf Ihrem Linux-Rechner frei gegebenen Verzeichnisse.

Jetzt ist es auch an der Zeit für einen ersten Blick von Windows aus auf den Samba-Server. Der sollte inzwischen in der Netzwerkumgebung und Ihrer Arbeitsgruppe zu sehen sein.

Funktioniert alles, starten Sie Yast. Unter Administration des Systems wählen Sie Konfigurationsdatei verändern und setzen dort den Wert `start_smb` auf `yes`.

Damit steht Samba bei jedem Linux-Neustart automatisch zur Verfügung.

Von Linux auf Windows zugreifen

Samba bietet sich nicht nur als Server an. Auch die Client-Dienste leisten viel. Das Utility für den Datenabruf von Windows heißt `smbmount`. Mit diesem Programm binden Sie ein frei gegebenes Windows-Verzeichnis im Linux-Dateisystem ein. Sinnvoll ist dies beispielsweise für Backup-Zwecke: Der Linux-Rechner kann damit Zeit gesteuert Backups von einzelnen Verzeichnissen anlegen oder Verzeichnisse auf verschiedenen Windows-Rechnern miteinander abgleichen. Ein Beispiel:

```
smbmount //server/dir /mnt/server/dir -o
fmask=0660,username=pcpro,password=test,uid=pcpro
```

An Stelle von `//server/dir` setzen Sie den Namen des Servers und des darauf frei gegebenen Verzeichnisses ein. `/mnt/server/dir` müssen Sie vorher erzeugen.

Mit `-o` beginnen die zusätzlichen Optionen:

- `fmask=660` weist allen Dateien im gemounteten Verzeichnis Lese- und Schreibrechte für den Benutzer und die Gruppe zu.
- `username=pcpro,password=test` sind der Benutzername und das Kennwort für den Zugriff auf die Windows-Freigabe.
- `uid=pcpro` weist den Verzeichnissen und Dateien den Benutzer `pcpro` als Eigentümer zu. Anderenfalls wäre immer `root` der Eigentümer.

Nach dem Mounten steht das Verzeichnis unter `/mnt/server/dir` zur Verfügung.

Die Anweisung `smbmount /mnt/server/dir` entfernt das Windows-Verzeichnis wieder aus dem Linux-Dateisystem.

Datenaustausch zwischen Mac und Windows NT/2000

Die Server-Betriebssysteme von Microsoft unterstützen Appletalk. So geht die Konfiguration unter Windows 2000 Server:

- Loggen Sie sich auf dem Server als Administrator ein.
- In den Netzwerkeigenschaften wählen Sie den Menüpunkt *Erweitert - Optionale Netzwerkkomponenten*.
- Markieren Sie *Weitere Dateidienste...* und klicken Sie auf Details.
- Im folgenden Fenster wählen Sie Dateidienste für Macintosh, klicken auf *Ok* und Weiter.
- Starten Sie die Computerverwaltung über *Start - Programme - Verwaltung - Computerverwaltung*.
- Klicken Sie doppelt auf *Freigegebene Ordner* und Freigaben. Per Rechtsklick im rechten Fenster starten Sie eine Neue Dateifreigabe. Suchen Sie dort den gewünschten Ordner aus und aktivieren Sie *Apple Macintosh unter Folgende Clients haben Zugriff*. Im folgenden Schritt des Assistenten legen Sie Nutzerberechtigungen fest, bevor Sie mit *Fertig* stellen den Ordner endgültig frei geben.

Datenaustausch zwischen Mac und Linux

Macintosh-Nutzer teilen sich in zwei Lager: Die mit OS X haben und die noch mit den älteren Betriebssystem-Versionen arbeiten.

OS X

Benutzer von Mac OS X haben es leicht - das hinter der grafischen Oberfläche steckende Unix-Betriebssystem sorgt für direkte Verbindung zu Windows. Der direkte Zugriff geschieht über die smb-Dienste. Hiermit greift der Mac sowohl auf Windows-, wie auch auf Samba-Volumes zu. Selbst stellt OS X noch keine Dienste zur Verfügung.

Mit den Menüeinträgen Gehe zu - Mit Server verbinden oder dem Shortcut *[Apfel] [K]* gelangen Sie zur Auswahl der Server im Netzwerk.

Für eine direkte Verbindung zu einem Ordner auf einem Windows-Rechner geben Sie diesen Befehl ein:

```
smb://rechnername/freigabe
```

Danach identifizieren Sie sich mit Usernamen und Kennwort auf dem Server und erhalten eine Verbindung zum Laufwerk auf dem Desktop.

Appletalk-Anbindung bringt Mac OS X auch mit. Das bedeutet: bei einem Upgrade auf das neue Mac-System müssen Sie nicht die komplette Netz-Infrastruktur umbauen. Über die Serverauswahl sehen Sie auch alle Appletalk-Rechner im Netz und können diese auf dem Desktop einbinden.

Mit Mac OS 9

Benutzer älterer Mac OS-Versionen benötigen eine Zwischenstation für den Datenaustausch mit Windows. Diese Zwischenstation heißt Linux. Ist auf dem Linux-Server bereits Samba installiert, packt man noch Appletalk dazu und schaltet für beide Welten dieselben Verzeichnisse frei. So geht's im Einzelnen:

- Richten Sie Samba so ein, wie im Abschnitt "Datenaustausch zwischen Windows und Linux" beschrieben.
- Starten Sie in Suse-Linux Yast und installieren Sie über aus der Serie n den Appletalk-Daemon. Sie finden ihn unter dem Namen *netatalk*.
- Nach der Installation finden Sie im Verzeichnis */etc/atalk* die Konfigurationsdatei *AppleVolumes.default*. Editieren Sie diese Datei und tragen Sie als letzte Zeile dies ein:

```
/home/pub "Offenes Verzeichnis" options:mswindows
```

Damit weisen Sie dem Verzeichnis */home/pub* die Bezeichnung "Offenes Verzeichnis" zu. Unter diesem Namen erscheint der Ordner später auf dem Mac-Schreibtisch.

Mit *options:mswindows* zwingen Sie dem Ordner Windows-Konventionen auf. Dabei wird auch die Standard-Codepage *iso8859-1* eingebunden. Das garantiert passende Umlaute. In der Literatur findet sich diese Codepage oft auch noch explizit mit angegeben:

```
options:mswindows,codepage:maccode.iso8859-1
```

Das allerdings führte auf mehreren Testsystemen zu Problemen mit den Umlauten.

Um Verzeichnisse nur für das Lesen frei zu geben, fügen Sie den Parametern noch ein *ro* hinzu. Komplette sähe die Konfigurationszeile in diesem Fall so aus:

```
/home/pub "Offenes Verzeichnis" options:ro,mswindows
```

- Starten Sie die den Dämon mit */etc/init.d/atalk start*
- Danach können Sie die Verbindung ausprobieren. Öffnen Sie auf dem Mac über das Apfelmenü den Punkt *Auswahl*. Klicken Sie auf *AppleShare* und danach auf den Linux-Server und Ok. Nach der Eingabe des Usernamen und Kennwort für den Linux-Rechner bietet Ihnen der Rechner die frei gegeben Netzwerkverzeichnisse an. Die von Ihnen gewählten Verzeichnisse erscheinen auf der Schreibtischoberfläche des Mac.
- Wenn alles geklappt hat, starten Sie die Appletalk-Dienste für Linux automatisch beim Hochfahren. Hier führt der Weg wieder über Yast, Administration des Systems und Konfigurationsdatei verändern. Der zuständige Parameter ist *start_atalk*.

Datenaustausch zwischen Office für Mac und Office XP

Die beiden Office-Versionen für Mac und PC verstehen sich recht gut -- so lange Sie Dokumente austauschen. Bei Formatvorlagen klappt auch noch alles. Makros jedoch lassen sich nicht immer 1:1 umsetzen. Sobald es an das Dateisystem geht, wird es heikel. So gibt beispielsweise die Eigenschaft *ActiveDirectory.Path* unter Windows den kompletten Pfad des Verzeichnisses aus, in dem eine Datei gespeichert ist. Mit Mac:Word 2001 auf einem Mac OS 9 allerdings funktioniert das nicht. Zwar gibt die Eigenschaft *brav* den Pfad wieder. Aber um zum Beispiel *ChangeFileOpenDirectory ActiveDocument.Path* zu verwenden muss man dem Ganzen noch einen Doppelpunkt hinzu fügen:

ChangeFileOpenDirectory ActiveDocument.Path & ":"

Wer also Makros auf beiden Rechnerwelten verwenden will, sollte Zeit für die Anpassung der Makros einplanen.

Mehr Glück hat man mit Dokumentvorlagen von Windows-Word. Die lassen sich einfach umwandeln:

1. Die Dokumentvorlage auf den Mac kopieren.
2. Die Vorlage als Dokument öffnen und als Vorlage wieder speichern. Die Vorlage landet dann im Ordner Eigene Vorlagen.
3. Wenn Sie die *normal.dot* von Windows verwenden wollen, verlassen Sie Mac-Word, löschen die ursprüngliche Normal-Vorlage und ersetzen Sie mit der konvertierten Version. Wichtig: eine Dateiendung *.dot* ist nicht notwendig.

Korrekte Umlaute mit Samba und Windows

Samba versteht, mit Umlauten umzugehen. Das ist besonders dann sinnvoll, wenn Windows-Dateien mit Linux weiter bearbeitet, etwa auf eine CD gebrannt werden sollen. Auch für den Datenaustausch mit einem Mac sollte die Konvertierung stimmen. Um die Umlaut-Umsetzung einzuschalten, tippen Sie folgende zwei Zeilen in die *smb.conf*

```
character set = ISO8859-15
client code page = 850
```

Besitzer von SUSE 7.3, die erstmals Samba installieren, finden den entsprechenden Eintrag schon in der *smb.conf*.
Nach einem Samba-Neustart mit *rcsmb restart* funktionieren die Umlaute korrekt.

Samba: Rechte erzwingen

Mit Samba geschriebene Dateien haben auf dem Linux-Rechner seltsame Attribute: Meist sind sie als ausführbar markiert. Das ist lästig. Und wenn der Linux-Server auch über eine Shell zugänglich ist, stellen diese Attribute ein Sicherheitsrisiko dar: Unter Umständen kommen so Nutzer an die Daten, die sie via Samba nicht lesen dürfen. Für eine sinnvollere Attribut-Vergabe sorgen Sie mit dem Eintrag

create mode = 610

in der jeweiligen Verzeichnisfreigabe in der smb.conf. In diesem Beispiel geben Sie dem Eigentümer einer Datei das Recht, diese zu lesen und zu modifizieren und den Gruppenmitgliedern das Leserecht für die Dateien. Andere Nutzer dürfen nichts. Sollen auch andere Nutzer die Dateien modifizieren dürfen, dann verwenden Sie

create mode = 660

Die Attribute setzt Samba, sobald eine Datei auf den Server kopiert wird. Zusätzlich sollten Sie die Attribute von Verzeichnissen kontrollieren. Dafür zuständig ist

directory mode = 770

In diesem Beispiel ist jedes neu erzeugte Verzeichnis für den Eigentümer und für die Mitglieder der Gruppe für Lesen und Schreiben zugänglich.

NTFS - LESEN und SCHREIBEN

Files von NTFS Partitionen lesen und schreiben

@Autor: [André Stiehm](#)

Als erstes mal ein Dankeschön an Holger Wirth, der mich bei diesem Unterfangen begleitet hat - und es zu Ende führte, als ich bereits entnervt meine letzte NTFS-Partition in FAT32 konvertiert hatte :-))

captive-NTFS ist die bislang einzigste freie Möglichkeit, auf eine NTFS-Partition - egal ob Windows NT, 2000 oder XP - vollen Schreibzugriff zu erhalten. Das Projekt von [Jan Kratochvíl](#) nutzt dazu den Windowstreiber. Um das ganze zum laufen zu kriegen, müssen folgende Schritte durchgeführt werden:

1) lufs

Von [der Sourceforge-Seite](#) das aktuelle [lufs-Paket](#) herunterladen. Danach die hier aufgeführten Schritte durcharbeiten:

```
tar zxvf lufs-0.9.7.tar.gz
cd lufs-0.9.7
```

Dann überprüfen, ob ein File mit Namen *modversions.h* unter */usr/src/linux/include/linux* vorhanden ist. Wenn das nicht der Fall ist, wie zum Beispiel bei SuSE 9.1, ein leeres File anlegen:

```
touch /usr/src/linux/include/linux/modversions.h
```

Danach

```
./configure --with-kheaders=/usr/src/linux/include
```

Achtung! Fedora Core 2 hat einen Bug - vor dem *make* muss auf Zeile 111 von *./kernel/Linux/2.6/proc.c* ein *,0* angehängt werden, damit's aufgeht. Die Zeile muss also folgendermassen aussehen:

```
if((res = sock_create(PF_UNIX, SOCK_STREAM, 0, s,0)) < 0){
```

danach kann's weitergehen wie gewohnt:

```
make
make install
```

falls nicht vorhanden, ein Directory *extra* erstellen, ansonsten beim uebernaechsten Punkt weitermachen:

```
mkdir /lib/modules/<kernel-version>/extra/
cp ./kernel/Linux/2.6/lufs.ko /lib/modules/<kernel-version>/extra/
depmod -a
```

Alles okay? Dann weiter zum naechsten Schritt!

2) captive NTFS

Von der [Downloadseite](#) kann entweder ein RPM (zum Beispiel fuer SuSE 9.1) oder der Source-Tarball heruntergeladen werden. Installiert wird entweder per:

```
rpm -ivh captive-static-1.1.5-0.i386.rpm
```

oder etwas komplizierter

```
tar zxvf captive-static-1.1.5.tar.gz
cd captive-static-1.1.5
./install
```

Ab hier gelten die Schritte wieder fuer alle - egal, ob von RPM oder Source installiert wurde:

```
captive-install-acquire
```

Koennen die Windows-Treiber (*ntoskrnl.exe*) durch das Tool nicht gefunden werden, kann man von [hier](#) ein gezipptes Package mit einem XP Bootscreen herunterladen und per *unzip* irgendwohin entpacken. Danach das GUI nochmals suchen lassen. Die Eintraege in der *fstab* werden automatisch angelegt, falls dies aus irgendeinem Grund nicht der Fall sein sollte, kann dies auch per Hand oder mit Hilfe des Befehls *captive-install-fstab* nachtraeglich gemacht werden.

Von Hand kann eine Partition folgendermassen gebootet werden:

```
mount -t captive-ntfs /dev/hda1 /mnt/windows/c/
```

Der Eintrag in der */etc/fstab* muss etwa folgendermassen aussehen:

```
/dev/hda1          /mnt/windows/c    captive-ntfs defaults 0 0
```