

# Samba - Linux im Windows - Netzwerk



Projekt im Fach Mehrplatzbetriebssysteme

Thomas Weindel, Janina Masser  
FS INF 02

# Inhaltsverzeichnis

<b><u>Inhalt</u></b>	<b><u>Seite</u></b>
1. Das Projekt	3
1.1 Die Aufgabenstellung	3
1.2 Die Vorgaben	3
1.3 Die Arbeitsmaterialien	3
2. Die Theorie	3
2.1 Samba – was ist das?	3
3. Die Umsetzung in der Gruppe	4
3.1 Benutzereinrichtung unter Linux	4
3.2 Gruppeneinrichtung unter Linux	5
3.3 Erstellen von Verzeichnissen	6
3.4 Rechtevergabe unter Linux	6
3.4.1 Aufbau der Rechte	7
3.4.2 Verzeichnis für Klassenarbeiten	7
3.5 Rechtevergabe unter Windows	8
3.6 Einrichten einer Domäne unter Windows	9
3.7 Roaming Profiles	9
4. Die Konfigurationsdatei	10
4.1 Die [global]-Sektion	11
4.2 Die [home]-Sektion	14
4.3 Die [profiles]-Sektion	14
4.4 Die [netlogon]-Sektion	15
4.4.1 Inhalt unserer Batchdatei für Lehrer	15
4.5 Die [lehrer]- und [schueler]-Sektion	16
4.6 Die smb.conf komplett	16
5. Hilfreiche Programme	18
5.1 smbstatus	18
5.2 testparm	18
6. Was haben wir gelernt?	19
7. Glossar	20
8. Quellen	22

## 1. Das Projekt

### 1.1 Die Aufgabenstellung

Unsere Aufgabenstellung war es, einen Samba-Server unter Linux einzurichten. Es sollen sinnvolle Verzeichnisse und Benutzer eingerichtet werden. Diesen Benutzern sollen sodann sinnvolle Rechte an bestimmten Verzeichnissen zugeordnet werden. Weiterhin soll auch die Möglichkeit bestehen, von einem Linux-Client sowie von einem Microsoft-Client aus auf den Server zuzugreifen.

### 1.2 Die Vorgaben

Benutzer:

- SchülerA,
- SchülerB,
- LehrerA,
- LehrerB

Verzeichnisse:

- Lehrer
- Schüler

Die Lehrer sollen Zugriffe auf die Verzeichnisse „lehrer“ und „schueler“ haben und von Schülern erstellte Verzeichnisse sperren können. Die Schüler sollen nur auf das Verzeichnis „schueler“ zugreifen können.

Wir haben uns - zusätzlich zur Aufgabenstellung - zum Ziel gesetzt einen Ordner „Klausuren“ im Schülerverzeichnis zu erstellen, in welchen die Schüler nur Dateien (z.B. Klassenarbeiten) hineinschieben aber den Ordnerinhalt nicht einsehen können. Dies ist sinnvoll, da bereits abgegebene Klassenarbeiten nicht mehr verändert werden können. Auch können die Lösungen der bereits abgegebenen Klassenarbeiten nicht von anderen Mitschülern eingesehen werden.

### 1.3 Die Arbeitsmaterialien

Für die Umsetzung des Projekts benötigten wir verschiedene Arbeitsmaterialien:

- einen PC, der als Linux-Server dient
- einen PC, der als Linux-Client dient
- einen PC, der als Windows-Client dient
- Linux-Software sowie das Samba-Paket
- Literatur

## 2. Die Theorie

### 2.1 Samba – was ist das?

Samba ist ein relativ ausgereiftes, standardkonformes und hochportables Softwarepaket, das auf den meisten Unix-Versionen problemlos installierbar ist. Samba dient als Datei- und Druckerserver für - über das Netzwerk angeschlossene - Windows-Rechner. Samba verwendet hierzu das von Microsoft benutzte SMB-Protokoll (Ser-

ver Message Block). Auf diese Weise müssen an den Windows-Rechnern keinerlei Veränderungen vorgenommen oder zusätzliche Software installiert werden.

Samba ist unter den Bedingungen der GNU (General Public License) für alle Zwecke kostenlos nutzbar. Damit entstehen beim Einsatz keinerlei Kosten.

Beispiele für verschiedene Dienste, die von Samba zu Verfügung gestellt werden:

- ⇒ Freigabe eines oder mehrerer Verzeichnisbäume
- ⇒ Freigabe von Druckern, die auf dem Server installiert sind, für Windows-Clients im Netzwerk
- ⇒ Authentifizierung von Clients, die sich an einer Windows-Domäne anmelden

Um Samba zu starten müssen zwei Prozesse aktiviert werden, `smbd` und `nmbd`. Der Server `smbd` bietet die eigentlichen Dateien- und Druckerdienste. Der Server `nmbd` ist für die Namensdienste zuständig. Die Befehle zum Starten der Prozesse lauten: „`rcnmb start`“ oder „`r smb start`“.

### 3. Die Umsetzung in der Gruppe

Um nun mit dem Server und den Clients arbeiten zu können, vernetzen wir die Rechner über einen Hub miteinander. Damit die Rechner sich nun auch im Netzwerk finden konnten, vergaben wir eine einheitliche Subnetmask sowie entsprechende IP-Adressen für jeden PC. Diese Schritte waren nötig, damit die Rechner miteinander kommunizieren und wir überhaupt mit der Arbeit am Server beginnen konnten. Um zu testen ob dies funktioniert kann man über den Ping-Befehl auf der Shell-Oberfläche bei Linux oder dem „Ausführen-Fenster“ unter Windows eine Verbindung zu einem zweiten Rechner aufbauen, indem man den Befehl: „ping“ 172.16.0.1 (entsprechende IP-Adresse des anderen Rechners) eingibt. Erhält man nun Antwort von diesem Rechner, können beide miteinander kommunizieren. Ist dies nicht der Fall, ist es empfehlenswert, noch einmal alle Kabel und Steckverbindungen sowie die eingegebene Subnetmask und IP-Adressen zu überprüfen. Erkennen sich die Rechner im Netz, kann mit der eigentlichen Arbeit begonnen werden.

#### 3.1 Benutzereinrichtung unter Linux

Bevor sich ein Benutzer von einem Windows-Client an einem Samba-Server anmelden kann, muss dieser vorher als Linux- und Sambabeanutzer angelegt werden.

Ein Linuxbenutzer kann sowohl über das Yast-Kontrollzentrum als auch über die Shell-Oberfläche (ähnlich wie DOS bei Windows) eingerichtet werden. Wir haben in unserem Projekt nur mit der Shell-Oberfläche gearbeitet. So auch beim Einrichten des Linuxbenutzers.

Syntax: `useradd [Benutzer]`

Beispiel: `useradd SchuelerA`

Auch muss ein Passwort vergeben werden, was mit dem folgenden Befehl ausgeführt werden kann:

Syntax: `passwd [Benutzer]`  
Beispiel: `passwd SchuelerA`

Nun wird man aufgefordert das gewünschte Passwort (in unserem Fall identisch mit dem Benutzernamen) zweimal einzugeben. Somit ist nun der Linuxbenutzer „schuelerA“ angelegt. Um diesen jetzt in einen Samba-User umzuwandeln ist folgender Befehl notwendig:

Syntax: `smbpasswd [Parameter] [Benutzer]`  
Beispiel: `smbpasswd -a -U schuelerA`

*-a = add user (Benutzer hinzufügen)*  
*-U= usr; remote Benutzername*

Nun folgt die Abfrage des smb-Passwortes. Hier verwenden wir auch wieder den Namen des Linuxbenutzers als Passwort.

Dieser Vorgang kann nun für beliebig viele Benutzer wiederholt werden.

### 3.2 Gruppeneinrichtung unter Linux

In unserem Beispiel ist es wichtig, die verschiedenen Benutzer in Gruppen aufzuteilen, da wir Schüler und Lehrer als Benutzer haben und diese später nicht die gleichen Rechte an bestimmten Verzeichnissen erhalten sollen. Daher haben wir die Gruppe „schueler“ und die Gruppe „lehrer“ erstellt. Dies erreicht man mit dem Befehl

Syntax: `groupadd [Gruppenname]`  
Beispiel: `groupadd schueler`

Nun müssen die Benutzer auch der entsprechenden Gruppe zugewiesen werden. Hierzu ist folgender Befehl erforderlich:

Syntax: `Usermod [Parameter (siehe unten)] [Benutzer] [Gruppenname]`  
Beispiel: `usermod -g schuelerA schueler`

Jedem Benutzer kann eine primäre und eine sekundäre Gruppe zugewiesen werden. Dies erfolgt durch die zusätzlichen Parameter „-g“ (= primäre Gruppe) oder „-G“ (=sekundäre Gruppe).

In dem oben aufgeführten Beispiel wurde der Benutzer „schuelerA“ der Gruppe „schueler“ zugewiesen. In unserem Projekt haben wir die Benutzer wie folgt aufgeteilt:

- ⇒ Die Benutzer „schuelerA“ und „schuelerB“ sowie „lehrerA“ und „lehrerB“ sind Mitglied in der Gruppe „schueler“
- ⇒ Die Benutzer „lehrerA“ und „lehrerB“ sind Mitglied in der Gruppe „lehrer“

Da die Lehrer in einem späteren Schritt dieselben Rechte an einem Schülerverzeichnis erhalten sollten wie die Schüler, haben wir diese auch noch in die Gruppe „lehrer“ aufgenommen. Die Schüler sollten nicht auf das Lehrerverzeichnis zugreifen können. Daher sind sie auch kein Mitglied der Gruppe „lehrer“.

### 3.3 Erstellen von Verzeichnissen unter Linux

Im nächsten Schritt werden unter Linux die vorher schon überlegten Verzeichnisse angelegt. Dies wären in unserem Projekt die Verzeichnisse „lehrer“ und „schueler“. Hierzu muss der folgende Befehl eingegeben werden:

Syntax: `mkdir [Verzeichnisname]`

Beispiel: `mkdir lehrer`

Um später die Rechtevergabe zu erleichtern haben wir dem neu angelegten Verzeichnis „schueler“ die Gruppe „schueler“ zugewiesen. Dies geschieht mit dem folgenden Befehl:

Syntax: `chgrp [Gruppe] [/Verzeichnis, welchem die Gruppe zugeordnet werden soll]`

Beispiel: `chgrp schueler /schueler`

### 3.4 Rechtevergabe unter Linux

Die Freigabe von Dateien ermöglicht es, Dateien von verschiedenen Rechnern aus gemeinsam zu nutzen. Damit Dateien von Benutzern (nicht) gelesen, gelöscht oder bearbeitet werden können, müssen Rechte vergeben werden. Diese Rechte können Dateien und/oder Verzeichnissen zugeordnet werden.

In unserem Fall würde das folgendermaßen aussehen:

Wir haben bei Punkt 3.3 als Beispiel das Verzeichnis „lehrer“ erstellt. Nun wollen wir dem Eigentümer „owner“ sowie der Gruppe „group“ und den anderen Benutzern „other“, die Zugriff auf das Verzeichnis „lehrer“ erhalten sollen, Rechte zuweisen. Wir haben uns schon überlegt, dass die Gruppe „lehrer“ Vollzugriff auf das Verzeichnis erhalten soll. Die Schüler, welche dann zu den „allgemeinen Benutzern“ gehören, dürfen nicht auf das Verzeichnis zugreifen. Der Benutzer „root“ behält seine vollen Zugriffsrechte. Der Befehl für diese Rechtevergabe sieht wie folgt aus:

Syntax: `chmod [Rechte z.B. 777] [Verzeichnispfad]`

Beispiel: `chmod 777 /lehrer`

Im Verzeichnisbaum wird das Verzeichnis dann folgendermaßen dargestellt:

```
rwX r-x r-- lehrerA lehrer
```

rwX = Rechte für Owner

r-x = Rechte für Group

r-- = Rechte für Other

lehrerA = Owner  
 lehrer = Gruppe  
 lehrer = Verzeichnis

### 3.4.1 Aufbau der Rechte

Bei der Erstellung eines Verzeichnisses wird dieses automatisch mit Standardwerten befüllt, dies sind u.a. der aktuelle Nutzer und seine primäre Nutzergruppe. Eine Datei lässt sich vom Besitzer oder dem Administrator auch einem anderen Nutzer übertragen.

Jede Datei bzw. Verzeichnisdatei besitzt zwölf voneinander unabhängige Schutzbits. Die wichtigsten neun beschreiben die Rechte der drei Nutzergruppen:

- ⇒ Eigentümer (Owner),
- ⇒ Gruppe (Group) und
- ⇒ Andere (Other)

Für alle Nutzergruppen kann der Zugriff lesend, schreibend oder ausführend gewährt werden. Kombinationen sind beliebig möglich. In der Dateiübersicht werden diese Rechte in der genannten Reihenfolge mit den Buchstaben r (readable), w (writable) und x (executable) gekennzeichnet. Ein nicht vergebenes Recht erhält einen Bindestrich „-“.

Die Rechte von Dateien/Verzeichnissen können auf verschiedene Weise geändert werden. Dies kann sowohl unter Linux als auch unter Windows erfolgen. Unter Linux gibt es die Möglichkeit die Dateirechte über die Konsole oder über die Grafische Oberfläche zu ändern.

Für die Rechtevergabe existiert ein einfacher Zahlencode von drei Ziffern. Die erste Ziffer bezieht sich auf den Besitzer der Datei/des Verzeichnisses, die zweite auf die Gruppe und die dritte auf alle anderen, also andere User, aber auch den Webserver.

	<b>Besitzer</b>	<b>Gruppe</b>	<b>Andere</b>
Lesen ( r )	4	4	4
Schreiben ( w )	2	2	2
Ausführen ( x )	1	1	1

Weist man einer Datei z.B. die Rechte 764 zu, so bedeutet dies, dass der Besitzer die Datei lesen, schreiben und ausführen darf ( $4+2+1=7$ ), die Gruppe lesen und schreiben ( $4+2=6$ ) darf und alle anderen nur lesen (4) dürfen.

### 3.4.2 Verzeichnis für Klassenarbeiten

Um darzustellen wofür man unseren Samba-Server einsetzen kann, haben wir im Verzeichnis „schueler“ einen Ordner „Klassenarbeiten“ erstellt. Ziel war es, dass die Lehrer Vollzugriff auf diesen Ordner erhalten. Die Schüler sollten jedoch in diesen

Ordner nur Daten abspeichern können. Das Einsehen in diesen Ordner soll den Schülern allerdings nicht erlaubt sein. So können sie, nachdem sie z.B. ihre Klassenarbeit in dem Ordner abgespeichert haben, keine Änderungen mehr vornehmen. Außerdem können die Schüler bereits abgegebene Arbeiten von sich selbst oder anderen Schülern nicht mehr einsehen.

Die Freigabe für diesen Ordner sieht wie folgt aus:

```
rwX rwX -w- lehrerA lehrer
```

- „lehrerA“, als Ersteller des Ordners, hat alle Rechte
- die Gruppe „lehrer“ hat alle Rechte
- die Schüler oder sonstige Benutzer haben nur Schreibrechte

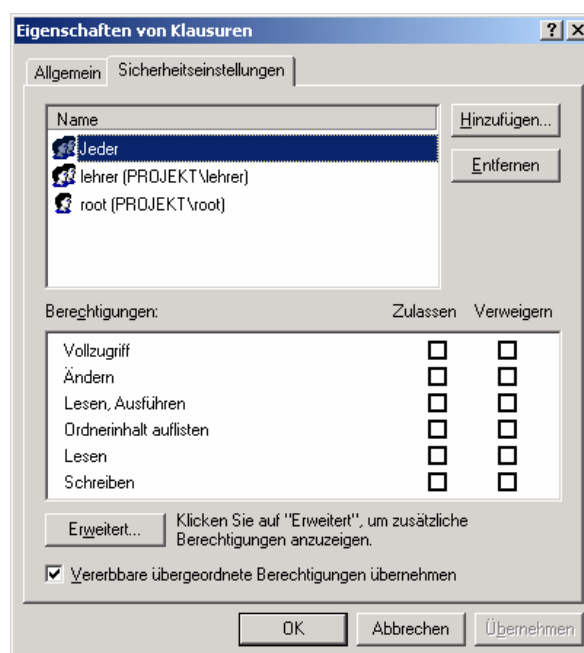
### 3.5 Rechtevergabe unter Windows

Um vom Windows-Client Rechte auf dem Samba-Server vergeben zu können, muss in der smb.conf des Linux-Servers folgender Befehl in die [global]-Sektion eingetragen werden:

```
admin users = lehrerA, lehrerB
```

Dadurch haben die Benutzer „lehrerA“ und „lehrerB“ die Administratorrechte (root) auf dem Samba-Server.

Um die Rechte eines einzelnen Ordners zu ändern, klickt man mit der rechten Maustaste auf das entsprechende Ordnersymbol. Nun hat man die Möglichkeit den Punkt „Eigenschaften“ auszuwählen. Ist man in dem Navigationsfenster „Eigenschaften“ erscheint ein Registerblatt „Sicherheitseinstellungen“. Hier können nun die Berechtigungen für Gruppen und Benutzer an dem jeweiligen Ordner vorgenommen werden.





### 3.6 Einrichten einer Domäne unter Windows

Warum muss unter Windows 2000 überhaupt eine Domäne eingerichtet werden? Sollte dies nicht der Fall sein, also ist die Domäne unter Windows nicht aktiviert, können sich nur die Benutzer anmelden, die auf dem entsprechenden Windows-Client ein Benutzerkonto haben. Meldet man sich aber an einer Domäne an, benötigt man das Benutzerkonto nicht, da man sich direkt am Samba-Server anmeldet. Der Server vergleicht nun den eingegebenen Benutzernamen und das Passwort mit den unter Samba angelegten Benutzern. Ist ein Benutzer nicht identisch kann er sich auch nicht an der Domäne anmelden. Dies alles hat den Vorteil, dass man z.B. in einer Schule an den Clients lediglich die Domäne einrichten muss und nicht alle verschiedenen Benutzergruppen. Dies geschieht alles nur **einmalig** auf dem Samba-Server. Ein weiterer großer Vorteil einer Domäne ist es, dass für jeden Benutzer der sich anmeldet ein individuelles Profil erzeugt wird (z.B. kann für jeden Lehrer der sich anmeldet automatisch das Lehrerverzeichnis gemappt werden).

Um nun unter Windows 2000 eine Domäne einzurichten, muss man den Benutzer „root“ als Samba- und Windows 2000 Benutzer anlegen, da dies als Administrator der Windows2000-Workstation nicht möglich ist. Hat man diesen Benutzer angelegt meldet man sich auf der Windows2000-Oberfläche als „root“ an. Nun klickt man mit der rechten Maustaste auf die Netzwerkkumgebung, dann auf Eigenschaften => Erweitert => Netzwerkidentifikation => Eigenschaften.

Nun erscheint das Navigationsfenster zum Einrichten einer Domäne. Hier klickt man nun im unteren Bereich des Fensters bei „Mitglied von...“ das Wort „Domäne“ an. Jetzt muss der Domäne-Name eingegeben werden. Dies muss derselbe sein wie der, der unter Linux in der smb.conf bei dem Parameter „workstation“ angegeben ist. Dies ist in unserem Fall „Projekt“. Im nächsten Schritt muss nun der Benutzername „root“ sowie das zugehörige Passwort eingegeben werden. Nun müsste die Domäne erfolgreich eingerichtet sein.

**WICHTIG!** Der Benutzer „root“ sollte nach der Einrichtung wieder aus dem Benutzerkonto unter Windows sowie als Sambabesutzer unter Linux gelöscht werden.

### 3.7 Roaming Profiles

Mit dem Einrichten einer Domäne haben wir die Grundlage für ein Roaming-Profil geschaffen. Die Benutzer melden sich direkt am Samba-Server an und erhalten von dort direkt ein Profil. Nachdem sich ein Benutzer von der Windows-Workstation abmeldet, wird sein Profil direkt auf dem Samba-Server gespeichert. Um zu vermeiden, dass bei der Abmeldung eine Fehlermeldung erscheint, müssen die Rechte für die Share, in welche die Profile gespeichert werden, für die Gruppe auf „schreibend“ gesetzt werden.

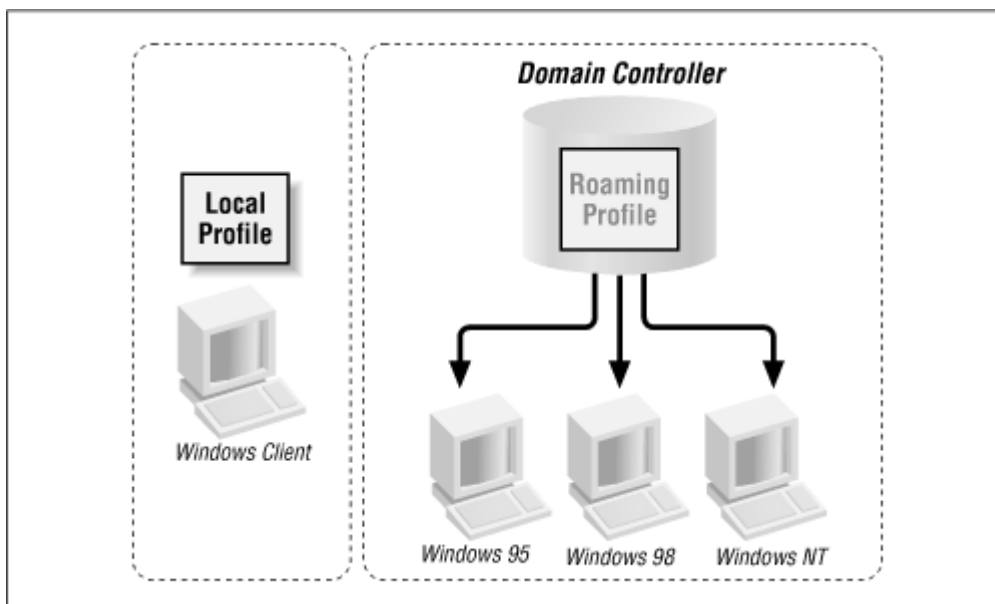
Unter Windows2000 kann jeder User sein eigenes Profil haben. Dies kann Folgendes beinhalten:

- das Erscheinungsbild des Desktops eines Users

- die in den Startmenüs erscheinenden Applikationen
- der Hintergrund

Wenn das Profil auf einer lokalen Festplatte gespeichert ist, wird es *local profile* genannt, da es beschreibt, wie eine Benutzer-Umgebung auf einer Workstation aussieht. Wenn das Profil andererseits auf einem Server gespeichert ist (in unserem Beispiel der Samba-Server), kann der Benutzer das gleiche Profil auf jede Client-Workstation herunterladen, die mit dem Server verbunden ist. Ein solches Profil wird *roaming profile* genannt, da der Benutzer auf allen, an den Server angeschlossenen Workstations dasselbe Profil benutzen kann. Das macht es besonders praktisch wenn jemand z.B. von seinem PC zu Hause und anschließend von seinem Laptop im Außendienst einloggen möchte.

Beispiel: Unterschied zwischen Local Profile und Roaming Profile



Quelle: [http://lug.krems.cc/docu/samba/ch06\\_06.html](http://lug.krems.cc/docu/samba/ch06_06.html)

Wie ein solches Roaming Profile in der Konfigurationsdatei von Samba erstellt wird, haben wir in Punkt 4.3 beschrieben.

#### 4. Die Konfigurationsdatei

Alle Informationen zum Samba-Server sind in einer zentralen Konfigurationsdatei gespeichert. Diese ist unter dem Pfad `/etc/samba/smb.conf` zu finden. Es handelt sich um eine einfache Textdatei, die in Sektionen unterteilt ist. Jede Sektion wird durch ihren Namen in eckigen Klammern eingeleitet und entspricht einer Freigabe. Es gibt vordefinierte Sektionen wie z.B. `[global]` und `[homes]`, es können aber auch Sektionen hinzugefügt werden, denen man einen beliebigen Namen, der noch nicht verwendet wird, zuweisen kann. Die erste Sektion `[global]` ist für die allgemeinen Servereinstellungen reserviert und muss stets vorhanden sein und genau diesen Namen haben.

Im Folgenden werden wir die verschiedenen Sektionen genauer erläutern:

## 4.1 Die [global]-Sektion

Wie bereits zu Beginn dieses Kapitels erwähnt, ist die [global]-Sektion schon vordefiniert. Man kann sie jedoch beliebig verändern und sie - je nachdem was man einrichten möchte - erweitern. Unsere [global]-Sektion sieht wie folgt aus:

```
[global]
workgroup = projekt
netbios name = filelux
server string = samba server %v
security = USER
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
allow trusted domains = no
min passwd length = 4
map to guest = Bad User
null passwords = no
password level = 0
username level = 0
unix password sync = no
restrict anonymous = no
large readwrite = no
debug timestamp = yes
domain logons = yes
domain master = yes
local master = yes
wins support = yes
os level = 65
preferred master = yes
ldap suffix = dc=example,dc=com
logon path = \\%L\profiles\%U
logon home = \\%L\profiles\%U
admin users = root, LehrerA, LehrerB

logon script = %U.log.bat
logon drive = z:

username map = /usr/local/samba/users.map
add machine script = /usr/sbin/useradd -c Machine-
d/var/lib/nobody-
ldap idmap suffix = ou=Idmap
ldap machine suffix = ou=Computers
```

Um die [global]-Sektion etwas transparenter zu machen möchten wir die Einstellungen erläutern:

**workgroup = projekt**

Um eine Anmeldung direkt mit dem Samba-Server zu ermöglichen muss der Arbeitsgruppen- bzw. Domänenname mit dem Workgroup- Parameter übereinstimmen.

**netbios name = filelux**

In der Netzwerkumgebung erscheint der Samba-Server mit diesem Namen.

```
server string = samba server %v
```

Dies wird als Kommentar in der Netzwerkumgebung gezeigt. %h wird durch den Servernamen und %v durch die Samba-Versionsnummer ersetzt.

```
security = USER
```

Alle Benutzer „user“ benötigen ein Passwort. Es sind verschiedene Sicherheitslevel möglich, die in der Konfigurationsdatei gesetzt werden können:

```
security = share
```

Hierbei müssen die User sich **nicht** am Server anmelden. Diese Einstellung zieht weitere Einstellungen bei den Freigaben (shares) nach sich und sollte vermieden werden.

```
security = user
```

Hier müssen sich die User mit einer gültigen username/password Kombination anmelden. Diese Art der Anmeldung ist zu bevorzugen, wenn die User private Verzeichnisse auf dem Server zugewiesen bekommen sollen (wichtig für E-mails). Das ist die Grundeinstellung.

```
security = server
```

Hier wird die gesamte Passwortverwaltung auf einem anderen Server abgewickelt.

```
encrypt passwords = yes
```

Hier kann festgelegt werden, ob die Passwörter verschlüsselt übertragen werden sollen. Wird hier „no“ eingetragen entsteht ein Sicherheitsrisiko, da die Möglichkeit besteht, dass die Passwörter von unbefugten eingesehen werden können.

```
smb passwd file = /etc/samba/smbpasswd
```

Dieser Befehl setzt den Pfad auf die smbpasswd-Datei

```
allow trusted domains = no
```

Wird verwendet, wenn der Parameter „security“ mit dem Wert „domain“ belegt ist. Erlaubt dem Administrator, seine Clients auf die Domain zu beschränken, in der er sich selbst befindet.

```
min passwd length = 4
```

Das Anmeldepasswort muss mindestens vier Zeichen lang sein.

```
map to guest = Bad User
```

Dieser Parameter kann nur gesetzt werden, wenn die Sicherheit auf „domain“, „Server“ oder „User“ gesetzt wurde. Hier kann festgelegt werden, wie der Samba-Server reagieren soll wenn sich ein Benutzer anmeldet, der auf der Unix Seite keinen Account hat. „Bad User“ bedeutet, dass ein Benutzer einen Gast-Zugang zum Server erhält, wenn er weder Unix- noch Sambabeneutzer ist.

```
null passwords = no
```

So werden keine Benutzer zugelassen, die ein „leeres“ Passwort haben. Bei „yes“ würden sie zugelassen werden.

```
password level = 0
```

Gibt an, mit welcher Genauigkeit die Passwörter überprüft werden sollen.

```
username level = 0
```

Gibt an, mit welcher Genauigkeit die Benutzernamen überprüft werden sollen.

```
unix password sync = no
```

Hier wird die Synchronisation des Unix-Passwortes mit dem verschlüsselten SMB-Passwort der smbpasswd-Datei eingestellt.

```
restrict anonymous = no
```

Erlaubt dem Administrator anonyme Benutzer vom Zugriff auf den Samba-Server auszuschließen.

```
debug timestamp = yes
```

DebugLog-Nachrichten besitzen alle eine Zeitmarke; hier kann deren Berücksichtigung ausgeschaltet werden.

```
domain logons = yes
```

Samba arbeitet als Domain-Controller und nimmt Einlogganforderungen von Clients entgegen.

```
domain master = yes
```

Der Daemon „nmbd“ kann in einem WAN die Funktion eines Domain Master Browsers übernehmen. In dieser Funktion kann der Daemon „nmbd“ die Browser-Funktion für eine Domain übernehmen, die mehrere Subnetze umfasst. Wenn der Parameter „domain master“ mit dem Wert „yes“ belegt ist, wird der nmbd-Daemon versuchen, diese Funktion im Netz zu übernehmen.

```
local master = yes
```

Gibt an, ob sich der Daemon „nmbd“ in einem Sub-Netz um die Rolle eines Local Master Browsers bewerben kann.

```
wins support = yes
```

Hier wird festgelegt, ob der Samba-Server als WINS-Server eingesetzt werden soll.

```
os level = 65
```

Dieser Parameter ist wichtig, wenn der Samba-Server im Netz mit mehreren Servern betrieben wird. Je höher der Wert gewählt wird (0-65) desto wahrscheinlicher ist es, dass er Master-Browser wird.

```
preferred master = yes
```

Masterbrowser der Domäne

```
logon path = \\%L\profiles\%U
```

Dieser Befehl gibt den Standort des Roaming profiles für den Benutzer an. Mit dieser Einstellung wird bei der ersten Anmeldung eines Benutzers an den PDC (**P**rietary **D**omain **C**ontroller) ein Unterverzeichnis mit dessen Namen in der Share „Profiles“ erstellt, in das bei der Abmeldung die benutzereigenen Profile gespeichert werden. %L ist der NetBiosName des Sambahservers. %U ist der aktuelle Benutzername

```
logon home = \\%L\profiles\%U
```

Dieser Befehl bewirkt, dass die Profile in die Heimverzeichnisse der Benutzer gespeichert werden.

```
admin users = root, LehrerA, LehrerB
```

Benutzer die hinter diesem Parameter stehen, erhalten die Rechte von "root". Dies ist vor allem wichtig, wenn eine Rechtevergabe vom Windows-Client auf dem Server erfolgen soll. Soll eine ganze Gruppe die „root“-Rechte bekommen wird dies durch „@“ gekennzeichnet (z.B. @lehrer).

```
logon script = %U.log.bat
```

Dieser Parameter legt fest, dass die Batchdatei .log.bat in Abhängigkeit von dem Benutzer, der sich an den Samba-Server anmeldet ausgeführt wird. Für jeden Benutzer muss im Verzeichnis „netlogon“ diese Batchdatei (benutzername.log.bat) angelegt werden.

```
logon drive = z:
```

Erlaubt die Angabe eines lokalen Pfades, mit dem das "home directory" verbunden wird; wird nur von WinNT-Workstation benutzt und ist nur sinnvoll, wenn der Samba-Server als logon-Server verwendet wird;

```
username map = /usr/local/samba/users.map
```

Abbildung einer Menge von Namen mit unterschiedlicher Schreibweise auf einen Namen vornehmen.

## 4.2 Die [home]-Sektion

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. Der Rechner muss allerdings im lokalen Netz eingebunden sein. Versucht also ein Windows-Benutzer von seinem Rechner aus zu einem dem Linux-Rechner eingestellten Netzlaufwerk eine Verbindung herzustellen, so wird er mit seinem persönlichen Heimatverzeichnis verbunden. Voraussetzung ist allerdings, dass der Windows-Benutzer eine Zugangsberechtigung für den Linux-Rechner besitzt.

Unsere [home]-Sektion sieht wie folgt aus:

```
[homes]
  comment = Heimatverzeichnis
  path = /home
  read only = no
  create mask = 0640
  directory mask = 0750
  browsable = no
```

Nun würden Benutzer, die auf beiden Systemen den gleichen Benutzernamen haben, ihr Home-Verzeichnis angezeigt bekommen. Diese Sektion ist also neben [global] eine weitere vordefinierte Sektion, deren Namen genauso lauten muss.

## 4.3 Die [profiles]-Sektion

Um ein sogenanntes Roaming-Profil zu erstellen sind zwei Schritte erforderlich:

1. Der Eintrag der [profiles]-Sektion in die smb.conf, die wie folgt aussieht:

```
[Profiles]
  path = /usr/local/samba/profiles
  read only = no
  create mask = 0700
  directory mask = 0770
  browseable = no
```

2. Der Eintrag in die [global]-Sektion (siehe auch Pkt. 4.1):

```
[global]
  logon path = \\%L\Profiles\%U
  logon home = \\%L\Proviles\%U
```

#### 4.4 Die [netlogon]-Sektion

Um virtuelle Laufwerke auf den Workstations zu mounten, wird die netlogon-Sektion sowie eine „Batchdatei“ für jeden Benutzer benötigt. In dieser Datei sind Informationen enthalten, welches Laufwerk gemountet werden soll und es wird ein Laufwerksbuchstabe zugewiesen. Die Batchdateien sind auf unserem Samba-Server unter „/usr/local/samba/netlogon“ abgespeichert.

Folgende Einträge in der smb.conf sind notwendig, um eine [netlogon]-Sektion zu erstellen:

```
[global]
  logon script = %U.log.bat
  logon drive = z:
```

```
[netlogon]
  path = /usr/local/samba/netlogon
  browseable = no
```

##### 4.4.1 Inhalt unserer Batchdatei für Lehrer

Für unsere zwei Gruppen „lehrer“ und „schueler“ haben wir entsprechende Laufwerke gemappt. Dadurch sehen die Lehrer die Laufwerke „lehrer“ und „schueler“ und können auf diese zugreifen. Die Schüler hingegen sehen nur das Laufwerk „schueler“ auf welches sie zugreifen können.

Syntax: net use [zu mappender laufwerksbuchstabe] \\[netbios name des Servers] [sharename]

Beispiele: net use m: \\filetux\lehrer  
net use n: \\filetux\schueler  
net time \\filetux /set /yes {Die eingestellte Uhrzeit des Servers wird als lokale Zeit auf die Workstations übertragen}

Bei den Schülern ist der Eintrag in der Batchdatei für das Laufwerk „m“ nicht vorhanden.

**Wichtig:**

Meldet sich nun ein Benutzer an der Workstation an, werden die virtuellen Laufwerke gemountet. Soll eines der virtuellen Laufwerke nun wieder entfernt werden, nutzt es nichts, den entsprechenden Parameter einfach zu löschen. Das virtuelle Laufwerk muss man unmounten, indem man als „root“ in die Batchdatei folgende Zeile hinzufügt:

```
net use [Laufwerksbuchstabe]:/del
```

#### 4.5 Die [lehrer]- und [schueler]-Sektion

Die [lehrer]-Sektion wurde von uns selbst erstellt. Der folgende Eintrag in die smb.conf ist notwendig um die in Punkt 3.4 erstellten Verzeichnisse auch anzeigen zu lassen. Ebenso ist es mit der [schueler]-Sektion.

```
[lehrer]
```

```
comment = Lehrerverzeichnis  
path = /sambashare/lehrer  
browseable = yes  
read only = no
```

```
[schueler]
```

```
comment = Schuelerverzeichnis  
path = /sambashare/schueler  
browseable = yes  
read only = no
```

#### 4.6 Die smb.conf komplett

```
[global]  
workgroup = projekt  
netbios name = filetux  
server string = samba server %v  
security = USER  
encrypt passwords = yes  
smb passwd file = /etc/samba/smbpasswd  
allow trusted domains = no  
min passwd length = 4  
map to guest = Bad User  
null passwords = no  
password level = 0  
username level = 0  
unix password sync = no  
restrict anonymous = no  
large readwrite = no  
debug timestamp = yes  
domain logons = yes  
domain master = yes  
local master = yes
```



```
wins support = yes
os level = 65
preferred master = yes
ldap suffix = dc=example,dc=com
logon path = \\%L\profiles\%U
logon home = \\%L\profiles\%U
admin users = root, LehrerA, LehrerB

logon script = %U.log.bat
logon drive = z:

username map = /usr/local/samba/users.map

[profiles]
path = /usr/local/samba/profiles
read only = no
create mask = 0700
directory mask = 0700
browseable = no

[netlogon]
path = /usr/local/samba/netlogon
browseable = no

[homes]
comment = Heimatverzeichnis
path = /home
read only = no
create mask = 0640
directory mask = 0750
browseable = no

[lehrer]
comment = Lehrerverzeichnis
path = /sambashare/lehrer
browseable = yes
read only = no

[schueler]
comment = Schuelerverzeichnis
path = /sambashare/schueler
browseable = yes
read only = no
```

## 5. Hilfreiche Programme

Es gibt einige Tools, mit denen man sich wichtige Informationen über den Samba – Server anzeigen lassen kann. Wir möchten im Folgenden einige Tools vorstellen die uns sehr geholfen haben.

### 5.1 smbstatus

Dieses Programm listet die aktuellen Samba – Verbindungen auf. Außerdem zeigt es die aktuelle Versionsnummer des Samba – Servers an.

```
Samba version 3.0.7-5-SUSE
PID      Username      Group        Machine
-----
 4421    lehrerA      lehrer      tom-85cgdtl0699 (192.168.178.25)

Service      pid      machine      Connected at
-----
profiles     4421    tom-85cgdtl0699  Sun Oct  9 15:22:32 2005
lehrer       4421    tom-85cgdtl0699  Sun Oct  9 15:22:37 2005
IPC$         4421    tom-85cgdtl0699  Sun Oct  9 15:22:36 2005
schueler     4421    tom-85cgdtl0699  Sun Oct  9 15:22:37 2005
IPC$         4421    tom-85cgdtl0699  Sun Oct  9 15:20:19 2005
netlogon     4421    tom-85cgdtl0699  Sun Oct  9 15:22:36 2005
profiles     4421    tom-85cgdtl0699  Sun Oct  9 15:22:37 2005
schueler     4421    tom-85cgdtl0699  Sun Oct  9 15:22:37 2005
IPC$         4421    tom-85cgdtl0699  Sun Oct  9 15:22:32 2005
lehrer       4421    tom-85cgdtl0699  Sun Oct  9 15:22:37 2005
Locked files:
Pid      DenyMode     Access      R/W         Oplock      Name
-----
4421     DENY_NONE    0x20089     RDONLY      EXCLUSIVE+BATC
/usr/local/samba/netlogon/lehrerA.log.bat  Sun Oct  9 15:22:37 2005
```

In unserem Beispiel ist zu sehen, dass der Benutzer „lehrerA“ mit der Standardgruppe „lehrer“ am Sonntag den 9.10.05 um 15:22 Uhr von der Windows2000 Workstation „tom-85cgdtl0699“ und der IP-Adresse: 192.168.178.25 auf die share „schueler“ und „lehrer“ zugegriffen hat.

Hier noch einige sinnvolle Parameter zum Befehl „smbstatus“:

- V     Gibt die Versionsnummer des Programms aus.
- S     Bewirkt, dass „smbstatus“ nur Freigaben auflistet.
- u     Wählt nur die Information aus, die relevant für den Benutzernamen ist.
- h     Gibt eine Zusammenfassung der Kommandozeilenoptionen aus.

### 5.2 testparm

Dieses Programm gibt Aufschluss darüber, ob die Syntax der Konfigurationsdatei richtig ist. Fehler werden mit „unknown parameter“ ausgegeben. So lassen sich Feh-

ler schon frühzeitig feststellen und man kann sich bei der Fehlersuche Zeit sparen. Hier als Beispiel unsere smb.conf aufgerufen:

```
Load smb config files from /etc/samba/smb.conf
Processing section "[profiles]"
Processing section "[netlogon]"
Processing section "[homes]"
Processing section "[lehrer]"
Processing section "[schueler]"
Loaded services file OK.
Server role: ROLE_DOMAIN_PDC
Press enter to see a dump of your service definition
```

In unserem Beispiel ist zu sehen, dass das Programm keine Fehler gefunden hat. Die Konfigurationsdatei unseres Samba-Servers ist also syntaktisch richtig aufgebaut.

## 6. Was haben wir gelernt?

Zur Vorbereitung auf die Projekte der Klasse haben wir bereits im Unterricht die Grundlagen der Textkonsole (Shell) von Linux gelernt. Dies war der Grundstein für das weitere Vorgehen. So war es uns bereits zu Beginn möglich Ordner, Verzeichnisse, Gruppen und Benutzer anzulegen und auch wieder zu löschen. Die Vorgabe von Rechten an Ordnern und Verzeichnissen war ein weiterer wichtiger Schritt als Vorbereitung zum eigentlichen Projekt.

Die im Unterricht erlernten Grundlagen konnten durch unser Projekt verinnerlicht und weiterentwickelt werden. Wir lernten, wie die Konfigurationsdatei von Samba aufgebaut ist, welche Befehle für unsere Aufgabe wichtig sind und wie man unter Windows2000 die Rechte auf dem Samba-Server steuern kann.

## 7. Glossar

Domäne	Computergruppe, die innerhalb eines Netzwerks miteinander verbunden ist. Zwei eindeutig definierte, adressierbare Bereiche des Internets.
Subnetmask	Die Hauptaufgabe der Subnetmask ist es, die Netzwerk- und Hostteile zu trennen. Eine Subnetmask ist genau so lang wie eine IP Adresse.
IP-Adresse	Das Internet Protokoll (IP) sieht für jeden an ein IP-Netzwerk angeschlossenen Rechner eine eindeutige Adresse, die sogenannte IP-Adresse (oft auch einfach abgekürzt als IP, engl. ip address) vor. Sie besteht also aus 4 Zahlen und wird durch Punkte getrennt. Damit lassen sich ca. 4,22 Milliarden eindeutige Adressen vergeben, wobei gewisse Adressblöcke reserviert sind.
PDC	PDC steht für <b>P</b> rietary <b>D</b> omain <b>C</b> ontroller, der die Verzeichnisse der Domänen, d.h. die Subnetze, verwaltet und kontrolliert Benutzeranfragen (z.B. darf sich der Benutzer einloggen).
Shell	Unter "Shell" ("Schale") versteht man die Benutzeroberfläche eines zeichenbasierten Betriebssystems, über die der Nutzer mit dem Rechner kommuniziert. Vorrangige Aufgabe ist das Übernehmen von Befehlen und die Ein- und Ausgabe von Meldungen des Systems.
SMB	Server Message Block, Protokoll zur Kommunikation zwischen Requester- und Serverkomponente bei den Netzwerkbetriebssystemen von Microsoft und IBM.
Linux	Linux ist ein frei verfügbares Multitasking und Multiuser Betriebssystem für eine Vielzahl von Plattformen. Es wurde erfunden von dem damals 21-jährigen Linus Torvalds 1991 begonnen und wird seither von einer Vielzahl an Entwicklern aus aller Welt weiterentwickelt.
testparm	Mit diesem Programm kann man die smb.conf auf syntaktische Korrektheit prüfen. Das Programm liest die Konfigurationsdatei ein und gibt Fehlermeldungen aus, sofern es unbekannte Parameter findet.
smbpasswd	Wird zur Pflege der verschlüsselten Paßwörter auf Serverseite verwendet.
NetBIOS	Sobald Windowsrechner Dateisysteme austauschen, sich gegenseitig in der Netzwerkumgebung sehen oder Drucker freigeben, funktioniert die Kommunikation über NetBIOS. Was ist NetBIOS? Hierzu gibt es unterschiedliche Meinungen. Fragt man IBM, ist

NetBIOS ein Protokoll, viele andere bezeichnen NetBIOS als reine Softwareschnittstelle zur Kommunikation von Rechnern. Mit dieser Schnittstelle werden Programmen unterschiedliche Dienste zur Kommunikation zur Verfügung gestellt. NetBIOS wurde entworfen, um in kleinen, lokalen Netzen Kommunikation zu ermöglichen. Beim Entwurf von NetBIOS wurde zunächst darauf geachtet, die Dinge sehr einfach zu halten, um sie in kleinen lokalen Netzen anwendbar zu machen.

## GNU

General Public License. Die meisten Software-Lizenzen sind daraufhin entworfen worden, den Benutzern die Freiheit zu nehmen, die Software weiter zu geben und zu verändern. Die GNU soll diese Freiheit garantieren. Sie soll sicherstellen, dass die Software für alle Benutzer frei ist.

## 8. Quellen

<http://home.arcor.de/thomas.litsch/s-samba.htm>

<http://www.64-bit.de/dokumentationen/netwerk/b/001/25741-inh.htm#TopOfPage>

<http://nafoku.de/t/unix.htm>

<http://www.hostsharing.net/dokumentation/einstieg-bei-hostsharing/die-wichtigsten-unix-kommandos.html>

[http://www.soziologie.uni-halle.de/unger/scripts/workshop\\_internet/scr\\_unix\\_dateisystem.html#a7](http://www.soziologie.uni-halle.de/unger/scripts/workshop_internet/scr_unix_dateisystem.html#a7)

<http://www.semibyte.de/sambaref/>

<http://www.SerNet.DE/>

<http://samba.SerNet.DE/>

<http://www.scheib.info/downloads.htm>