NIS und NFS

Marcel Holtmann
Universität Bielefeld - Technische Fakultät
AG Rechnernetze und verteilte Systeme
marcel@rvs.uni-bielefeld.de

10. Juni 1999

Referenzen

1) NIS und NFS: Hal Stern
O'Reilly

2) Moderne Betriebssystem: Andrew S. Tanenbaum
Carl Hanser Verlag und Prentice-Hall International

3) Solaris System Administration: Thomas Handschuch
Springer Verlag

4) `http://metalab.unc.edu/LDP/HOWTO/NIS-HOWTO.html`
`http://metalab.unc.edu/LDP/HOWTO/NFS-HOWTO.html`

Probleme in verteilten Computerumgebungen

- Pflege der Konfigurationsdateien, wie z.B. Kennwort-, Gruppen- und Rechnerdatei
- Software muß jedem Rechner bereitgestellt werden
- Jeder Benutzer sollte auf jedem Computer immer das gleiche Arbeitsverzeichnis zur Verfügung haben

Die Konfigurationsdateien

- Die Datein `/etc/passwd`, `/etc/group`, `/etc/hosts` usw. müssen bei jeder Veränderung auf alle Rechner im Netzwerk verteilt werden
- **Nachteil:** Je größer das Netz um so mehr Verwaltungsarbeit
- **Lösung:** Zentrale Verwaltung der Dateien mit *NIS*

Installation von Software

- Softwarepakete oder -updates müssen immer auf allen Rechner installiert werden, auf denen die Programme benötigt werden
- **Nachteil:** Je größer das Netz um so mehr Verwaltungsarbeit
- **Lösung:** Installation der Software auf einem zentralen Server und dann das entsprechende Verzeichnis mit NFS ins lokale Dateisystem einbinden

Bereitstellung der Benutzerverzeichnisse

- Jedem Benutzer muß auf einer Workstation sein Arbeitsverzeichnis zur Verfügung stehen
- Alle möglichen Arbeitsverzeichnisse sollten auf jeder Maschine des Netzes gleichzeitig verfügbar sein
- **Nachteil:** Bei einer großen Anzahl von Benutzern kann man nicht jedes Homeverzeichnis mit NFS mounten
- **Lösung:** Dynamisches ein- und aushängen des Benutzerverzeichnisses mit Hilfe des *Automounters*

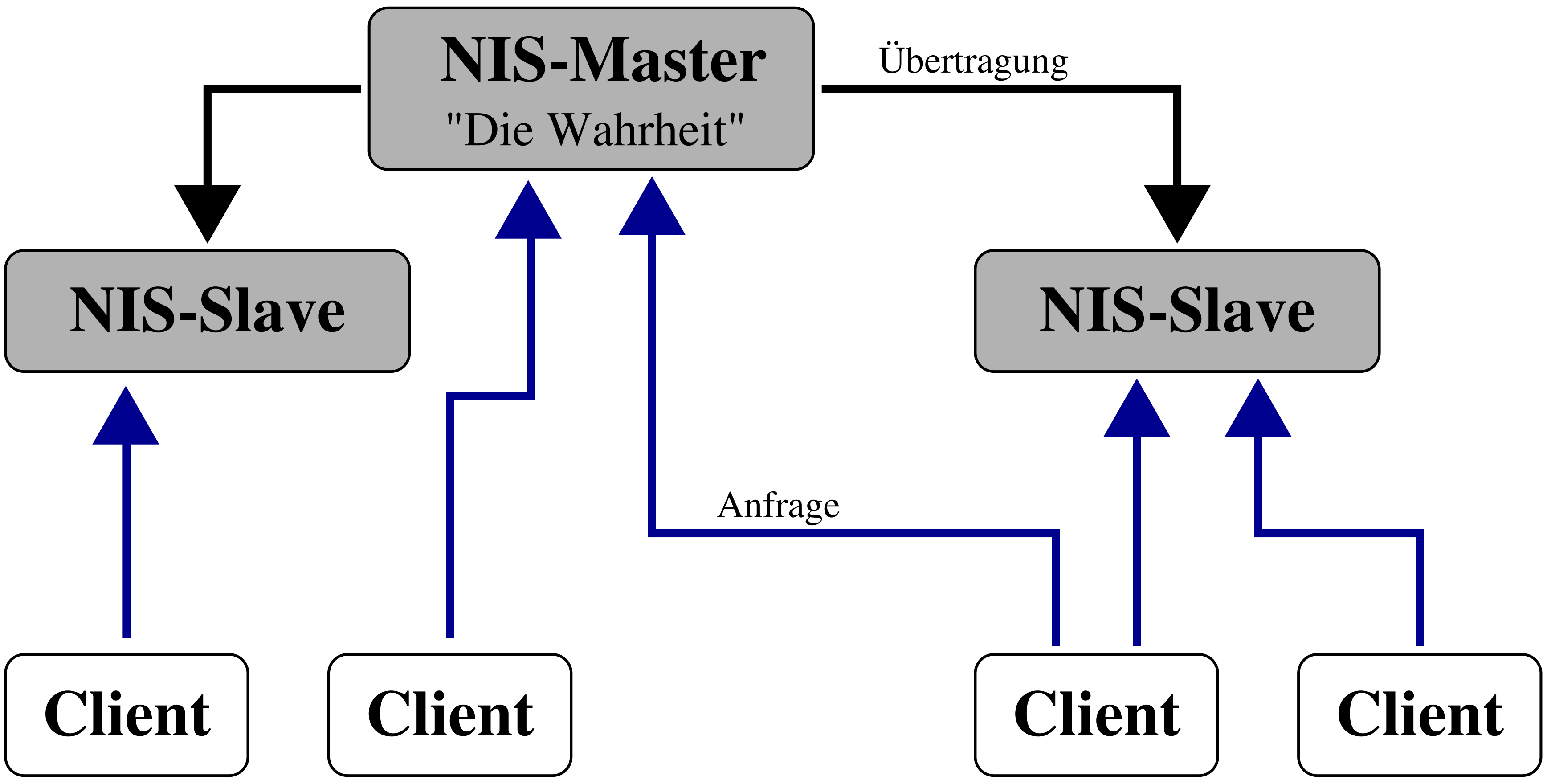
Network Information Service

- Alle Dateien, die keine systemspezifischen Daten enthalten, werden über NIS verwaltet. Die Dateien werden als NIS-$Maps$ bezeichnet
- NIS ist ein Client-Server-Modell
- Die Kommunikation erfolgt über RPC
- Alle Rechner, die auf die gleichen Daten zugreifen, befinden sich in einer Domäne
- Bei NIS gibt es einen Server, ein paar Stellvertreter und unendlich viele Clients

Welche Dateien werden von NIS verwaltet?

Datei	**Inhalt**
`/etc/hosts`	Rechnernamen und IP-Adressen
`/etc/passwd`	Benutzernamen, UID's und Kennwörter
`/etc/group`	Benutzergruppen
`/etc/services`	Netzportnummern und Dienstnamen
`/etc/netmasks`	Netzmasken
`/etc/networks`	Netzadressen
`/etc/protocols`	Netzprotokollnamen und -nummern
`/etc/aliases`	Aliasnamen und Postlisten für die elektronische Post
`/etc/rpc`	Programmnummern für Remote Procedure Call
`/etc/bootparams`	Informationen über plattenlose Knoten
`/etc/ethers`	Ethernet-Nummern (MAC-Adressen)
`/etc/netgroup`	Definition von Netzgruppen (wird von NIS benutzt)

Wie funktioniert NIS?



Programme eines NIS-Systems

domainname
Das Programm `domainname` weist dem NIS-System die Domäne zu

ypserv
Der Dämon `ypserv` ist der NIS-Server, der eine Bereitstellung der Maps erlaubt. Er wird sowohl auf NIS-Master- als auch auf NIS-Slaveservern gestartet

ypbind
Mit dem Dämon `ypbind` erfolgt die Kommunikation der Client-Rechner mit einem NIS-Server (sowohl Master- als auch Slaveserver)

Programme eines NIS-Systems

ypwhich

Das Dienstprogramm `ypwhich` zeigt die Bindungen einer Domäne an

```
> domainname
rvs.uni-bielefeld.de
> ypwhich
mother
```

Wenn an den Domänennamen kein Server gebunden ist gibt `ypwhich` eine Fehlermeldung aus

```
> ypwhich -d techfak.uni-bielefeld.de
ypbind: Domain techfak.uni-bielefeld.de not bound
```

Programme eines NIS-Systems

ypmatch

Das Programm `ypmatch` sucht einen Schlüssel in einer NIS-Map und gibt die Daten, die zu diesem Schlüssel gehören

```
> ypmatch mother hosts.byname
129.70.123.10   mother
```

ypcat

Mit `ypcat` kann man alle Werte einer NIS-Map ausgeben lassen

```
> ypcat passwd.byname
ladkin:Ye3FeskoSu1S2p3:10000:10:Peter B. Ladkin:/home/ladkin:/bin/tcsh
holtmann:ef6jgaMan9kadw7:100001:10:Marcel Holtmann:/home/holtmann:/bin/tcsh
...
```

Programme eines NIS-Systems

ypxfer

Mit dem Programm `ypxfer` werden NIS-Maps vom Master-Server auf den Stellvertreter kopiert

```
slave> cd /var/yp
slave> ypxfer -h master hosts.byname
```

ypxfer_1perhour / ypxfer_1perday

```
Skript ypxfer_1perhour:
 ypxfer passwd.byuid
 ypxfer passwd.byname
 ypxfer aliases.byname
 ypxfer mail.aliases
```

```
Skript ypxfer_1perday:
 ypxfer services.byname
 ypxfer protcols.byname

Eintrag in crontab
 0 * * * * ypxfer_1perhour
 0 0 * * * ypxfer_1perday
```

Programme eines NIS-Systems

yppasswd

Mit dem Dienstprogramm `yppasswd` kann man sein Paßwort in einem NIS-System ändern

```
> yppasswd
Changing NIS password for holtmann on mother.
Old password:
New password:
Retype new password:
NIS entry changed on mother.
```

yppasswdd

Auf dem NIS-Masterserver muß der `yppasswdd`-Dämon laufen. Beim Ändern des Paßwortes wird der Eintrag in der Datei `passwd` modifiziert, die Map neu erzeugt und die neue Map an die Slaveserver übertragen

Einrichten eines NIS-Masterservers

```
> echo rvs.uni-bielefeld.de > /etc/defaultdomain
> domainname ‘cat /etc/defaultdomain‘
> ypinit -m
In order for NIS to operate sucessfully, we have to construct a list
of the NIS servers.  Please continue to add the names for YP servers
in order of preference, one per line.  When you are done with the list,
type a <control D> or a return on a line by itself.
      next host to add:  mother
      next host to add:  sisyphos
      next host to add:  ^D
The current list of yp servers looks like this:

mother
sisyphos

Is this correct?  [y/n: y]  y

Installing the YP database will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
```

Einrichten eines NIS-Masterservers

```
Do you want this procedure to quit on non-fatal errors? [y/n: n]  n
OK, please remember to go back and redo manually whatever fails. If
you don't, some part of the system (perhaps the yp itself) won't work.
The yp domain directory is /var/yp/rvs.uni-bielefeld.de
There will be no further questions. The remainder of the procedure
should take 5 to 10 minutes.
Building /var/yp/rvs.uni-bielefeld.de/ypservers...
Running /var/yp /Makefile...
updated passwd
updated group
updated hosts
updated ethers
updated networks
updated rpc
updated services
updated protocols
updated netgroup
updated bootparams
updated aliases
```

Einrichten eines NIS-Masterservers

```
updated publickey
updated netid
updated netmasks
updated timezone
updated auto_master
updated auto_home
updated auto_direct
updated auto_vol
updated printers.conf

mother has been set up as a yp master server without any errors.

If there are running slave yp servers, run yppush now for any data bases
which have been changed.  If there are no running slaves, run ypinit on
those hosts which are to be slave servers.
> reboot
```

Einrichten eines NIS-Slaveserver

```
> echo rvs.uni-bielefeld.de > /etc/defaultdomain
> domainname `cat /etc/defaultdomain`
> ypinit -s mother

Do you want this procedure to quit on non-fatal errors? [y/n: n]  n
OK, please remember to go back and redo manually whatever fails. If
you don't, some part of the system (perhaps the yp itself) won't work.
The yp domain directory is /var/yp/rvs.uni-bielefeld.de
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the data bases from mother.
Transferring netmasks.byaddr...
Transferring netid.byname...
Transferring publickey.byname...
Transferring mail.byaddr...
Transferring mail.aliases...
```

Einrichten eines NIS-Slaveserver

```
Transferring netgroup.byhost...
Transferring netgroup.byuser...
Transferring ethers.byname...
Transferring ethers.byaddr...
Transferring hosts.byaddr...
Transferring hosts.byname...
Transferring group.bygid...
Transferring group.byname...
Transferring passwd.byuid...
Transferring passwd.byname...
Transferring ypservers...

sisyphos's nis data base has been set up
without any errors.
> reboot
```

Einrichten eines NIS-Clients

```
> echo rvs.uni-bielefeld.de > /etc/defaultdomain
> domainname ‘cat /etc/defaultdomain‘
> ypinit -c

In order for NIS to operate sucessfully, we have to construct a list of the
NIS servers.  Please continue to add the names for YP servers in order of
preference, one per line.  When you are done with the list, type a <control D>
or a return on a line by itself.
        next host to add:  mother
        next host to add:  sisyphos
        next host to add:  ^D
The current list of yp servers looks like this:

mother
sisyphos

Is this correct?  [y/n: y]  y
> reboot
```

Generierung der NIS-Maps

- Die NIS-Maps werden mit Hilfe des Programms `makedbm` erzeugt

- In dem Verzeichnis `/var/yp` erfolgt die Generierung der Maps mit Hilfe von `make`

- Als Basis für die Maps werden die Standardkonfigurationsdateien des Systems benutzt. Es ist möglich auch andere Dateien anzugeben

- **WICHTIG!**
  Die Map `passwd` sollte nicht aus der Systemdatei erzeugt werden, da man sonst das Root-Passwort über NIS jedem bereitstellt

Generierung der NIS-Maps

```
> cd /var/yp
> make
updated passwd
pushed passwd

updated hosts
pushed hosts

updated netid
pushed netid
```

Das Makefile von NIS

```
DIR =/etc
YPDBDIR=/var/yp

all: passwd group hosts ethers networks rpc services protocols \
     netgroup bootparams aliases publickey netid netmasks timezone

hosts.time: $(ETCDIR)/hosts
    @($(MULTI) $(B) -l $(DIR)/hosts);
    @($(STDHOSTS) $(DIR)/hosts $(CHKPIPE))| \
    (awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$1, $$0 }' $(CHKPIPE)) | \
    $(MAKEDBM) $(B) - $(YPDBDIR)/$(DOM)/hosts.byaddr;
    @touch hosts.time;
    @echo "updated hosts";
    @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) hosts.byname; fi
    @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) hosts.byaddr; fi
    @if [ ! $(NOPUSH) ]; then echo "pushed hosts"; fi

passwd: passwd.time
group: group.time
hosts: hosts.time
```

Die NIS-Maps im Detail

- NIS benutzt zur Speicherung der Maps das DBM-Format, welches mit Hilfe des Programmes `makedbm` erzeugt wird
- Jede NIS-Map wird mit einem Schlüssel (Key) gespeichert, über den die Suche erfolgt (z.B. `passwd.byname` bzw. `passwd.byuid`)
- Ein vollständiger Mapname ist *Map*.*Key*
- Für die Standard-Maps gibt es aber Alias-Namen, wie z.B. `passwd` anstelle von `passwd.byname`
- Die erzeugten DBM-Dateien werden unter `/var/yp/`*domainname* im Dateisystem abgelegt

Aktivierung von NIS

- In der Datei `/etc/nsswitch.conf` steht die Konfiguration des Systems für die Benutzung des NIS-Services durch die Systemroutinen

```
# the following two lines obviate the "+" entry in /etc/passwd
passwd:     files nis
group:      files nis

# consult /etc "files" only if nis is down.
hosts:      nis dns xfn [NOTFOUND=return] files
ethers:     nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files

netgroup:   nis

automount:  files nis
aliases:    files nis

# for efficient getservbyname() avoid nis
services:   files
```

NIS und der Domain Name Service

- Mit dem NIS-Service kann man nicht nur die Rechnernamen oder IP-Nummern der eigenen Maschinen herausbekommen, sondern auch von allen anderen im Internet eingetragenen Rechnern

```
> ypmatch www.mit.edu hosts
18.181.0.31     DANDELION-PATCH.MIT.EDU
```

- Wenn eine Anfrage an die Maps `hosts.byname` und `hosts.byaddr` gemacht wird und kein gültiger Schlüssel gefunden wird, fragt der NIS-Server einen DNS-Server an und übergibt dessen Resultat

Network File System

- NFS ist ein Dateisystem, welches die Grenzen einer Festplatte auf das Intranet ausdehnt

- Für einen Rechner ist es transparent, ob die Verzeichnisse von einer lokalen Festplatte oder über NFS gemountet werden

- Fast jedes `mount`-Kommando der Unix-Derivate ist so erweitert worden, das man Dateisysteme über TCP/IP-Netze mounten kann

- NFS ist ein Client-Server-Modell, dessen Kommunikation über RPC erfolgt

Die Client Seite von NFS

1. Man benötigt einen Unix-Kernel, der das NFS-Protokoll unterstützt

2. Das `mount`-Kommando sollte Unterstützung für NFS besitzen

```
> mount mother:/usr/local /usr/local
```

Oder in der Datei `/etc/fstab`:

```
/dev/hda3           /             ext2     defaults      1  1
mother:/usr/local   /usr/local    nfs      rw,bg,hard    0  0
```

Die Server von NFS

nfsd

Der NFS-Dämon stellt die Implementierung des NFS-Protokolls auf der Serverseite bereit. Es laufen normalerweise immer mehrere Instanzen des Dämon, damit mehrere Anfragen gleichzeitig beantwortet werden können

mountd

Der `mountd` behandelt die Anfragen zum Einhängen der Dateisysteme (das „mounten“). Das *mount*-Protokoll ist nicht Bestandteil von NFS

lockd / statd

Das Sperren von Dateien (*locking*) wird mit Hilfe der Dämon `lockd` und `statd` koordiniert. Die beiden Dämonen müssen auf dem Server und Client laufen, sind aber für die Grundfunktionalität von NFS nicht notwendig

Einrichten eines NFS-Servers

- Es müssen die entsprechenden Dämonen auf dem Server gestartet werden
- Verzeichnisse, die über NFS gemountet werden sollen, müssen explizit freigegeben (*exportiert*) werden
- Normalerweise werden die Dämonen vom Boot-Skript gestartet, sobald Verzeichnisse beim booten als exportierbar gekennzeichnet wurden

Exportieren von Dateisystemen

- Die Verzeichnisse bzw. Dateisysteme, die exportiert werden sollen, werden in der Datei `/etc/exports` eingetragen

```
/usr/local     -ro,access=judhistar:sisyphos:mercurio
/home/ladkin   -rw=sisyphos:mercurio,root=judhistar
/ftp           -ro
```

Exportieren von Dateisystemen

- Das eigentliche Exportieren erfolgt mit dem Programm `exportfs`, welches die Daten aus `/etc/exports` erhält
- Beim Systemstart werden diese Programme alle automatisch ausgeführt

```
if [ -f /etc/exports ]; then
  exportfs -a
  nfsd 8
  echo -n ' nfsd'
  mountd -n
fi
```

Exportieren von Dateisystemen

- Unter dem Unix-Derivat System V (z.B. Solaris) ist der Exportmechanismus durch das *Sharing* ersetzt worden

- Das Programm `exportfs` wird durch das Programm `share` ersetzt

- Die Datei für die zu exportierenden Dateisysteme wurde in `/etc/dfs/dfstab` untergebracht

- Das mounten von Dateisystemen wird beim Systemstart von der Datei `/etc/vfstab` geregelt

Regeln für den Export

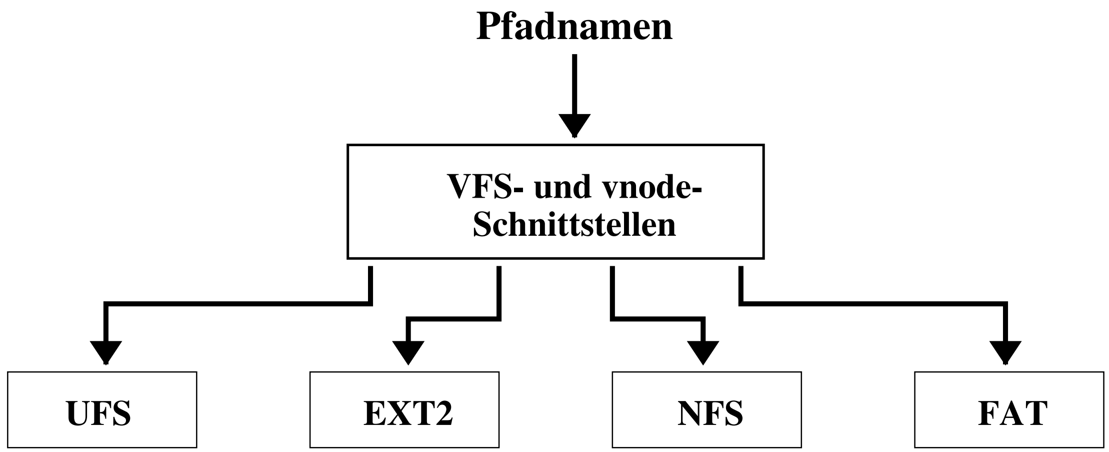
1. Es können nur lokale Dateisysteme exportiert werden

2. Unterverzeichnisse können nur dann exportiert werden, wenn sie auf einem anderen physikalischen Gerät liegen

3. Dateisysteme, die auf unterschiedlichen physikalischen Geräten liegen, müssen getrennt exportiert werden, auch wenn sie hierarchisch gemountet werden

Das virtuelle Dateisystem

- Unter modernen Unix-Systemen sind alle Dateisysteme mit Hilfe von VFS implementiert. VFS ist das Vaterobjekt, welches alle abstrakten Methoden für den Zugriff auf Dateisysteme definiert



- VFS benutzt dabei die virtuellen Knoten (*vnode*), welche sich von dem unter Unix-Dateisystemen bekannten *inodes* ableiten

Der Automounter

- Der Automounter ist ein Dienstprogramm, das NFS-Dateisysteme automatisch einhängt, wenn sie benutzt werden und sie wieder aushängt, wenn sie nicht mehr benötigt werden

- Der Automounter kann NIS zur Verwaltung der NFS-Konfigurationsdateien benutzen

- Die Datei `/etc/fstab` muß nicht aktualisiert werden

- Die physikalischen Devices befinden sich in der `/etc/fstab`; alle Netzwerkdateisysteme werden vom Automounter verwaltet

Die Automounter-Maps

- Der Automounter benutzt die Datei /etc/auto_master zur Konfiguration der NFS-Mounts

```
# Master map for automounter
#
+auto_master
/xfn                -xfn
/net                -hosts              -nosuid,nobrowse
/home               auto_home           -nobrowse
/vol                auto_vol
/-                  auto_direct
```

Die direkte Map

- Der Automounter kann **eine** direkte Map für Mount-Punkte enthalten

- Eine direkte Map entspricht einer über NIS verteilte `fstab` in der die Einträge für die NFS-Dateisystem enthalten ist

- In der Master-Map wird die direkte Map mit `/-` eingeleitet

- Die Einträge in der `direct`-Map folgen dann folgenden Aufbau

```
# Mount-Punkt   Filesystem auf dem Server    Optionen
#
/var/mail       mailhost:/var/mail           -rw
```

Die indirekten Maps

- Alle Anderen sind *indirekte* Maps
- In den indirekten Maps werden Unterverzeichnisse spezifiziert an deren Stelle das importierte NFS-Dateisystem eingehängt
- Der Mountpunkt in der Mastermap stellt dabei das Elternverzeichnis für die Verzeichnisse der indirekten Map da
- Mit den indirekten Maps werden Dateisysteme von verschiedenen Partitionen oder Serveren unter einem gemeinsamen Verzeichnis zusammengefaßt (z.B. die Homeverzeichnisse)

Die Map für die Homeverzeichnisse

- Mit der Map `auto_home` werden die Mountpunkte der Homeverzeichnisse angegeben und in dem Dateisystem unter `/home` eingebunden

```
# Home directory map for automounter
#
+auto_home
ladkin        mother:/export/disk1/ladkin
holtmann      mother:/export/disk2/&
heiko         sisyphos:/export/disk1/&
```

```
> cd /home
> ls
heiko    holtmann     ladkin
```

Ein `/usr/local` für alle

Problem:

Wir haben drei unterschiedliche Hardware-Plattformen und drei verschiedene Betriebssysteme

- Solaris (SunOS) auf UltraSPARC Maschienen
- Solaris (SunOS) auf x86-PC's
- Linux auf x86-PC's
- Irix auf SGI-Mips

Wir wollen:

Alle Programme unter `/usr/local` auf allen Platformen zur Verfügung stellen

Der erste Versuch

- Auf dem Server werden vier Verzeichnisse auf einer physikalischen Partitionen angelegt (für jede Platform eine)

```
> mkdir /export/disk1/local.sparc.SunOS
> mkdir /export/disk1/local.i386.SunOS
> mkdir /export/disk1/local.i386.Linux
> mkdir /export/disk1/local.mips.Irix
```

- Die Verzeichnisse werden nun in die `/etc/fstab` auf **jedem** Client-Rechner eingetragen

```
mother:/export/disk1/local.sparc.SunOS   /usr/local   nfs   ...
```

Die Nachteile

1. Sobald man die Daten auf eine andere Partition oder Server schieben möchte (z.B. wegen Platzmangel) muß man auf **allen** Clients den Eintrag in der `fstab` ändern

2. Dateien wie Info-Files oder Man-Pages werden für jede Platform abgespeichert. Es reicht aber diese allgemein zur Verfügung zu stellen und dann von allen Plattformen nutzen zu lassen

Der zweite Versuch

- Wir erzeugen eine NIS-Map `auto_master` und `auto_direct` und benutzen den Automounter zum einbinden der Dateisysteme

```
# Master map for automounter
#
/home           auto_home
/-              auto_direct
```

```
# Direct mounting map for automounter
#
/usr/local    mother:/export/disk1/local.$CPU.$OS
```

Der nächste Versuch

- Folgende Verzeichnisse werden angelegt

```
> mkdir /export/disk1/bin.sparc.SunOS
> mkdir /export/disk1/bin.i386.SunOS
> mkdir /export/disk1/bin.i386.Linux
> mkdir /export/disk1/bin.mips.Irix
> mkdir /export/disk1/lib.sparc.SunOS
> mkdir /export/disk1/lib.i386.SunOS
> mkdir /export/disk1/lib.i386.Linux
> mkdir /export/disk1/lib.mips.Irix
> mkdir /export/disk2/include
> mkdir /export/disk2/share
> mkdir /export/disk2/man
```

Der nächste Versuch

- Das Verzeichnis `/usr/local` wird unter die Kontrolle des Automounters gestellt

```
# Master map for automounter
#
/home          auto_home
/usr/local     auto_local
```

Der nächste Versuch

- Das Verzeichnis `/usr/local` wird unter die Kontrolle des Automounters gestellt

```
# Local map for automounter
#
bin       mother:/export/disk1/&.$CPU.$OS
lib       mother:/export/disk1/&.$CPU.$OS
include   mother:/export/disk2/&
share     mother:/export/disk2/&
man       mother:/export/disk2/&
src       sisyphos:/export/disk42/&
```

Das Endergebnis

- Mit Hilfe von NIS, NFS und dem Automounter kann eine gemeinsame Umgebung in einem Computernetz von sehr unterschiedlichen Platformen und Betriebssystemen geschaffen werden

- Der Endbenutzer findet auf jedem System seine persönliche Umgebung wieder und kann immer die gleichen Programme benutzen

- Ein sauber geplantes Netz muß nicht schwer zu verwalten sein

- **Als oberstes Ziel gilt die Benutzerfreundlichkeit**