



## THINCLIENTS SELBST BAUEN

# Plattenlos glücklich

Die Netzwerktransparenz von X11 ermöglicht die Ausführung von Linux-Programmen und deren Anzeige auf verschiedenen Rechnern. Mit diesem Prinzip kann ein einzelner leistungsstarker PC Dutzende PCs mit Software versorgen. Die Einrichtung ist bei jeder Distribution mit Bordmitteln möglich.

VON **MATTIAS SCHLENKER**

Als Unix Mitte der 1980er eine grafische Oberfläche bekam, war die effiziente Nutzung vorhandener Ressourcen oberstes Gebot. Bereits vor Einführung des *X Window Systems* war es üblich, mit seriellen Terminals ohne eigene Logik auf die wenigen schrankgroßen Unix-Rechner zuzugreifen. Die Terminals reichten dabei lediglich die Tastatureingaben in die eine Richtung und die Ausgaben in die Gegenrichtung durch.

Als das *X Windows System* 1984 eingeführt wurde, stand ein ähnliches Konzept für die grafische Oberfläche an, bei dem auch Wert darauf gelegt wurde, Anwendungen von verschiedenen Rechnern auf einem Clientsystem nutzen zu können. Während die Terme „Client“ für das passive Terminal und „Server“ für den Rechner, der Rechenleistung anbietet, auch bei X11 gelten, ist es bei den Anwendungen umgekehrt: Beim *X Server* han-

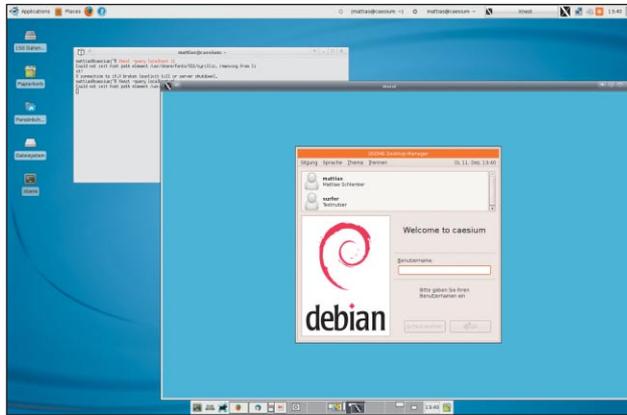
### Listing 1

```
#!/bin/sh

DIRS="/var/log var/tmp
/var/lib/xkb"

case $1 in
start)
  for i in $DIRS
  do
    mount $i
  done
;;
stop)
  for i in $DIRS
  do
    umount $i
  done
;;
esac
```

delt es sich um das Programm, dass die Anzeige übernimmt, er läuft in unserem Fall auf dem *Client-Rechner*, ein *X-Client* bezeichnet dagegen eine Applikation, die beim Server das Rendering in Auftrag gibt. Da die Verdrehung der Bezeichnungen zwar logisch ist, aber mitunter zu Verwirrungen führt, verwenden wir in diesem Artikel explizit „Client-Applikation“ und „Client-Rechner“.



Mit Xnest wird zunächst lokal die Konfiguration des Display-Managers getestet.

Anfang der 1990er waren dedizierte X-Terminals weit verbreitet. Die kompakten und lüfterlosen Rechner machten dank Auf- und Abwärtskompatibilität des X11-Produktes viele Wechsel der Unix-Server mit und blieben an Universitäten bis zur Jahrtausendwende im Einsatz. Mit der Verfügbarkeit billiger PCs und konkurrenzfähiger Linux-Distributionen setzte ein Paradigmenwechsel ein: Sowohl Workstations als auch Terminals wurden durch maßgeschneidert aus Standardkomponenten zusammengestellte PCs ersetzt. Als Terminal-Hardware können je nach Budget alte Rechner zum Einsatz kommen oder auf Basis von lüfterlosen Mainboards wie VIAs Epia-Serie Geräte auf die eigenen Anforderungen abgestimmt werden.

**Tests lokal**

Auf dem heimischen Linux-PC laufen X-Server und X-Clients eng verzahnt nebeneinander, so dass man sich keine Gedanken über die Trennung der Komponenten machen muss. Tests lassen sich mit ein paar kleinen Tricks

dennoch einfach durchführen: Mit ein wenig Kommandozeilenakrobatik simulieren Sie einen Thinclient.

Als Protokoll für entfernte Verbindungen dient *XDMCP*, das *X Display Manager Control Protocol*. Aktuelle Distributionen wie Ubuntu oder openSUSE bieten ein grafisches Konfigurationstool, in dem Sie XDMCP erlauben können. In Netzen mit mehreren Rechnern mit Serverfunktion sollten Sie zudem Chooser-Broadcasts aktivieren, um eine Auswahl der verfügbaren Server zu ermöglichen. Neben der Aktivierung von XDMCP ist es erforderlich, dass der Log-in-Manager die Rechner kennt, welche ein Anmeldefenster anfordern. Das geht über einen lokalen DNS-Server oder einfacher durch einen Eintrag der Form

```
192.168.1.23 client01
->client01.meinnetz.test
```

in der Datei */etc/hosts*. Ist die Konfiguration abgeschlossen, melden Sie sich ab und wechseln mit *Strg-Alt-F2* auf eine Textkonso-

le, wo Sie mit Administratorrechten den Displaymanager neu starten:

```
/etc/init.d/xdm restart
```

Je nach Distribution kann statt *xdm* auch *gdm* oder *kdm* erforderlich sein. Im neu gestarteten Displaymanager melden Sie sich ganz normal an. Jetzt fordern Sie mit dem Befehl

```
Xnest :1 -broadcast
```

ein Log-in bei allen Rechnern im Netz an, die XDMCP aktiviert haben. In diesem Fall erhalten Sie das Log-in-Fenster Ihres eigenen Rechners, meist in einer grafisch weniger aufwendigen Version. Wenn Sie sich hier einloggen, erfolgt die Kommunikation zwischen X-Server und den Client-Applikationen via TCP/IP über die lokale Linkverbindung.

**Zehn-Minuten-Thinclient**

Ist der erste Test erfolgreich verlaufen, können Sie, ohne zusätzliche Serverdienste zu benötigen, einen Thinclient aufsetzen, der von Festplatte bootet. Als Hardware genügen ein 200-MHz-Prozessor und 64 MByte RAM sowie eine 2 MByte-Grafikkarte. Praktisch keine Wünsche bleiben bei 300 MHz, 128 MByte und einer 8-MByte-Grafikkarte offen. Zwar lässt sich ein Minimalsystem mit X theoretisch auf 32 MByte Flash unterbringen, für den schnellen Test mit Standarddistributionen ist aber eine Festplatte mit zwei Gigabyte anzuraten. Installieren Sie eine beliebige Linux-Distribution zunächst als absolutes Minimalsystem und fügen nach der Installation via Paketmanager den X-Server hinzu. Sehr gut funktioniert dies mit Ubuntu und Debian, insbesondere bei Installation vom Netzwerk-ISO. Auf dem Testsystem verwendeten wir ein Ubuntu ohne *Preseed*-Angabe

**Fertigericht: Linux Terminal Server Project**

Ein Rundum-Sorglos-Paket für die Versorgung eines kleinen Netzes mit via PXE gestarteten Thin Clients bietet *LTSP*. *LTSP* enthält fertige Konfigurationen für den DHCP- und den TFTP- sowie den NFS-Server und bietet Scripte für den Zugriff auf lokale USB-Sticks und Soundkarten – gerade die beiden letzten Funktionen müssen bei „selbstgebauten“ Thinclients relativ aufwendig nachgebaut werden. Besonderes Schmankerl ist ein „Teachertool“, das via VNC in Schulungsumgebungen Zugriff auf die Desktops der Eleven bietet. Nachteil der *LTSP*-Lösung ist freilich die etwas eingeschränkte Flexibilität – insbesondere wenn die Terminals in ein bestehendes Netz mit bereits vorhandene NFS- und Bootservern eingebunden werden sollen. Die Quellen:

- https://help.ubuntu.com/community/UbuntuLTSP
- http://www.ltsp.org



Unter einem Dach: *LTSP* ist ein vorgefertigtes Thinclient-System, das allerdings schwerer anzupassen ist als das mit Standardtools aufgesetzte.

## Swappen via Netzwerk

➤ Auf Desktopsystemen verwendet man in der Regel doppelt so viel Swap wie Arbeitsspeicher vorhanden ist. Diese Faustregel macht auch bei Systemen mit viel Speicher ausbau Sinn, weil dort lange nicht benötigte Speicherseiten mit Programmcode ausgelagert werden können und dafür Dateisysteminhalte gecached werden. Auf festplattenlosen Rechnern soll Swap lediglich dazu dienen, dass der knappe Speicher nicht vollläuft, beispielsweise wenn ein lokaler Drucker angesprochen wird oder der X-Server viele virtuelle Desktops cachen soll. Wir verwenden ein Verzeichnis `/exports/swap` auf dem Server, das schreibbar freigegeben wird und auf dem die Clients selbsttätig Swapfiles einrichten und mounten. Zudem wird im Script die Swappiness reduziert, damit nicht unnötig ausgelagert wird:

```
#!/bin/sh

HOSTNAME=$( hostname )
LASTLOOP=$( losetup -f )
SWAPEXPORT="192.168.1.42:/exports/swap"

case $1 in
  start)
    mkdir /tmp/swap
    mount „$SWAPEXPORT“ /tmp/swap
    if [ ! -f /tmp/swap/swapfile.${HOSTNAME} ]
    then
      echo 'Erstelle Swapfile -- Bitte Geduld!'
      dd bs=1048576 count=512 \
        if=/dev/zero \
        of=/tmp/swap/swapfile.${HOSTNAME}
    fi
    losetup ${LASTLOOP} \
      /tmp/swap/swapfile.${HOSTNAME}
    mkswap ${LASTLOOP}
    swapon ${LASTLOOP}
    echo 10 >/proc/sys/vm/swappiness
  ;;
  esac
```

in der Bootzeile, bei der das Paket `xserver-xorg` und die Schriften `xfonts-100dpi` `xfonts-75dpi` `xfonts-base` hinzugefügt wurden. Überzeugen Sie sich nach der Installation mit

```
dpkg -l | grep font
```

dass genügend Standardschriften installiert sind. Konfigurieren Sie den X-Server noch mit dem Befehl:

```
dpkg-reconfigure xserver-xorg
```

Von der Textkonsole können Sie nun den X-Server mit Log-in-Anforderung beim anderen Rechner starten. Die Angabe des Displays (oben :1) ist nicht notwendig, da auf diesem Rechner ja kein weiterer X-Server läuft:

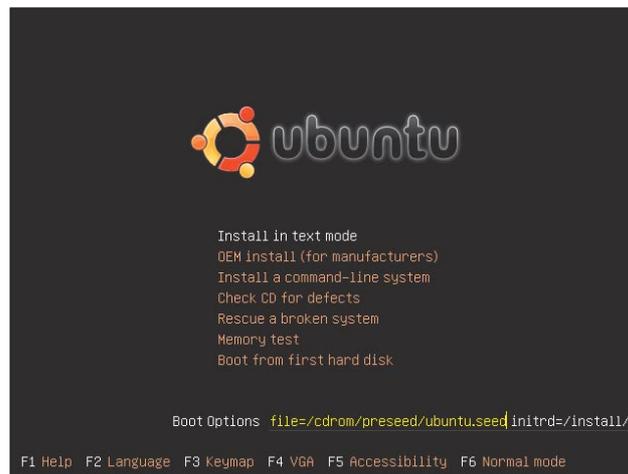
```
X -broadcast
```

Sollte der Broadcast fehlschlagen, probieren Sie eine direkte Anfrage mit der IP des Rechners, der XDMCP offeriert – und überprüfen Sie, ob Sie auf dem Server die IP-Adresse des Client-Systems richtig in Ihre `/etc/hosts` eingetragen haben. Klappt der Broadcast den-

noch nicht, versuchen Sie die direkte Anforderung unter Angabe der IP der Servermaschine:

```
X -query 192.168.1.42
```

Damit der X-Server bei jedem Systemstart mit



Lässt man bei der Alternate- oder Netzwerkinstallation das Seed-File weg, wird ein Minimalsystem installiert.

Anfrage startet, legen Sie am besten ein Script `/etc/init.d/xquery` mit dem folgenden Inhalt an:

```
#!/bin/sh

case $1 in
  start)
    /usr/bin/Xorg -broadcast &
    ;;
  stop)
    /usr/bin/killall -9 Xorg
    ;;
  esac
```

Dieses – zugegebenermaßen etwas krude – Script setzen Sie auf „ausführbar“ und verlinken es in den Runlevel-Verzeichnissen `/etc/rc2.d`, `/etc/rc3.d` und `/etc/rc5.d` unter dem Namen `S99xquery`. Nach einem Neustart erfolgt die automatische Anforderung des Log-ins am entfernten Rechner. Unpraktisch ist bei dieser Form des Thinclients der Shutdown. Sie müssen zunächst mit `Strg-Alt-F2` auf die Textkonsole wechseln, sich dort lokal anmelden und dann herunterfahren. Deutlich einfacher geht es, wenn Sie das Verhalten bei `Strg-Alt-Entf` von `Reboot` auf `Shutdown` verändern. Bei Debian müssen Sie hierfür die Datei `/etc/inittab` anpassen, bei Ubuntu die `/etc/event.d/control-alt-delete` – beispielsweise mit dem Befehl

```
/sbin/shutdown -h now
```

Fortan genügt für den Shutdown entweder `Strg-Alt-Entf` oder – wenn der Log-in-Manager bereits die Tastenkombination abfängt – `Strg-Alt-F1` gefolgt von `Strg-Alt-Entf`.

Der Thinclient mit lokaler Festplatte hat viele Vorteile plattenloser Thinclients und kommt dank lokalem Swap sogar mit etwas weniger Arbeitsspeicher aus: Da ein X-Server mit vielen lokal gecachten Desktops und minimierten Fenstern schnell auf 200 MByte RAM kommt, ist bei plattenlosen Terminals der Arbeitsspeicher entsprechend zu wählen oder auf langsamen Netzwerk-Swap auszuweichen.

Dennoch ist eine Festplatte – eine alte Verschlossene besonders – ein Single Point of Failure, der nicht nur das Risiko eines Hardware-Defektes in sich birgt, sondern auch anfällig gegen Stromausfälle und das Fehlverhalten von

Nutzern ist. Gerade in Umgebungen, die einem großen Nutzerkreis zugänglich sind, also Rechnerräumen in Schulen und Universitäten sowie Internet-Cafés, sind deshalb festplattenlose Terminals zu bevorzugen.

### Ohne Platte aus dem Netz

Deutlich sicherer und durch die zentrale Verwaltung gut administrierbar sind deshalb festplattenlose Terminals. Statt eines lokalen Bootloaders auf Festplatte kommt der PXE-Bootmechanismus zum Einsatz, der seit ca. 1999 bei fast allen Onboard-Karten zu finden ist. Das Wurzeldateisystem wird vom Server per NFS gemountet. Im Prinzip kann jede Linux-Distribution für diesen Einsatzzweck angepasst werden. Wegen der relativ unkomplizierten Installationsmethode *debootstrap* raten wir jedoch zu Ubuntu oder Debian. *Debootstrap* kann auf jeder Distribution ausgeführt werden – Sie können mit dem richtigen *Debootstrap*-Paket unter Debian einen Ubuntu-Thinclient aufsetzen oder unter SUSE einen Debian-Thinclient.. Für unser Beispiel haben wir auf einem AMD64-Ubuntu ein i386-Ubuntu eingerichtet.

Als Erstes wird das Wurzelverzeichnis vorbereitet. Sie können ein beliebiges, per NFS exportierbares Verzeichnis wählen:

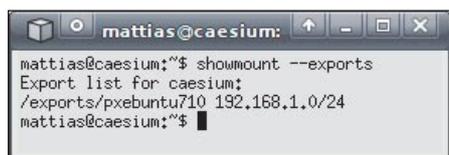
```
mkdir -p /exports/pxebuntu804
```

In dieses Verzeichnis wird nun mit *debootstrap* zunächst ein Ubuntu-Minimalsystem installiert:

```
debootstrap --arch i386 hardy
➔ /exports/pxebuntu804
➔ http://archive.ubuntu.com/ubuntu
```

Die Installation wird je nach Geschwindigkeit der Internetverbindung zwischen fünf Minuten und einer halben Stunde dauern. Zeit für einen Kaffee. Für die erste Stufe der Konfiguration mounten Sie */proc*, */dev* und */tmp* des Hostsystemes in das später exportierte Verzeichnis:

```
mount -t proc none
➔ /exports/pxebuntu804/proc
mount -o bind /tmp
➔ /exports/pxebuntu804/tmp
mount -o bind /dev
➔ /exports/pxebuntu804/dev
```



Das Kommando *showmount* zeigt die richtige Konfiguration des Display-Managers.

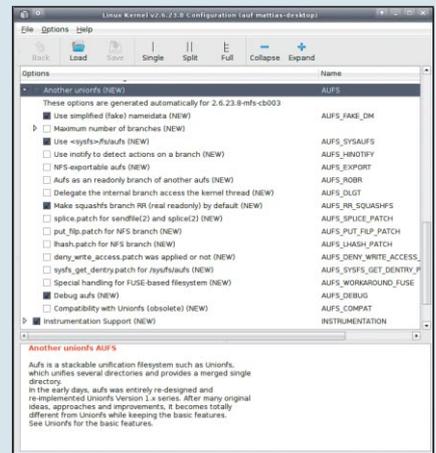
## Transparent gemountet

Was bei FreeBSD oder SunOS bereits vor Jahren Einzug in offizielle Kernel gehalten hat, ist bei Linux noch von der Aufnahme in Vanilla- und Distributionskernel entfernt: Dateisysteme, welche transparent über einen nur lesbar gemounteten Ast des Verzeichnisbaumes „drübergemountet“ werden können.

Von Live-CDs wie Ubuntu, Knoppix und sidux kennt man UnionFS und AUFS (Another Union FS). Alle können dort gute Dienste leisten, wo *tmpfs* – gegebenenfalls in Kombination mit *rsync* – von den Möglichkeiten her an seine Grenzen stößt.

Eine weitere Alternative mit Transparenz auf Blockebene statt auf Dateiebene ist die Kombination von *Cowloop* (Copy-on-Write-Loop) mit Network Block Devices.

Leider ist in allen Fällen das manuelle Patchen des Kernels erforderlich, um Union-Dateisysteme nutzen zu können, was Updates erschwert und gegebenenfalls die Anpassung der *Initrd*-Skripte erfordert. Da Thinclients jedoch deutlich seltener Updates erfordern als die

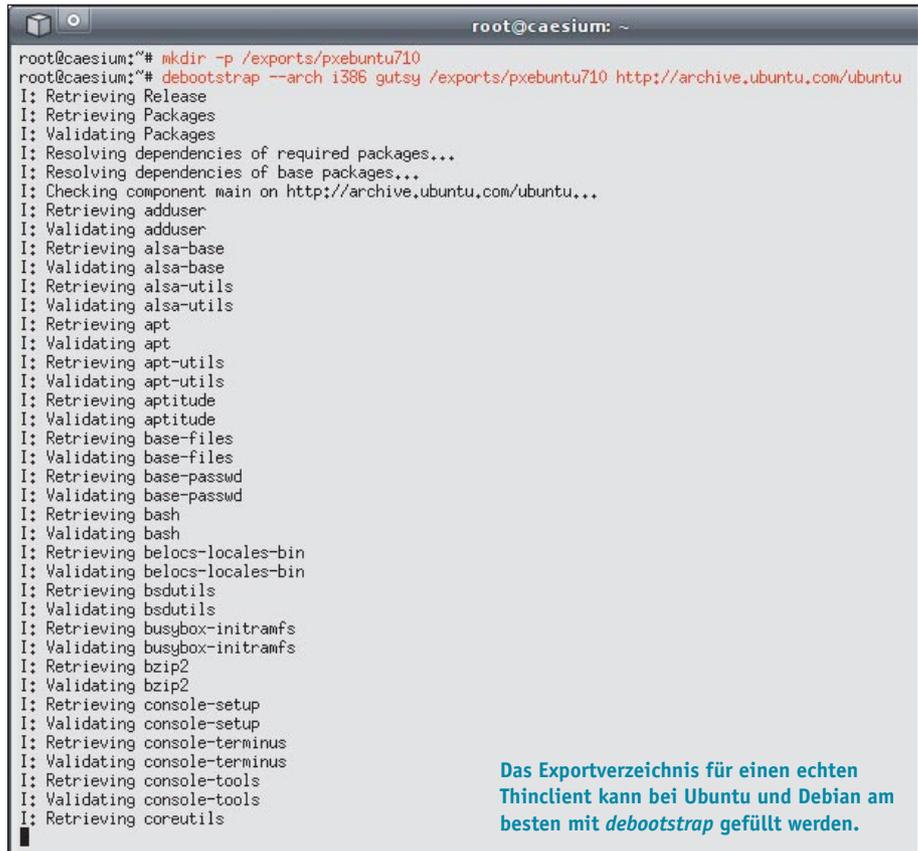


AUFS rüstet ein transluzentes Dateisystem nach: Auf ein Read-Only-Medium kann so scheinbar geschrieben werden.

Applikationsserver, kann sich die Erstellung stark angepasster Lösungen in Umgebungen mit vielen Clients schnell rentieren.

Weitere Infos zu den genannten Lösungen finden Sie unter

- ➔ <http://de.wikipedia.org/wiki/UnionFS>
➔ <http://aufs.sourceforge.net>
➔ <http://nbd.sourceforge.net>
➔ [www.atcomputing.nl/Tools/cowloop](http://www.atcomputing.nl/Tools/cowloop)



Das Exportverzeichnis für einen echten Thinclient kann bei Ubuntu und Debian am besten mit *debootstrap* gefüllt werden.

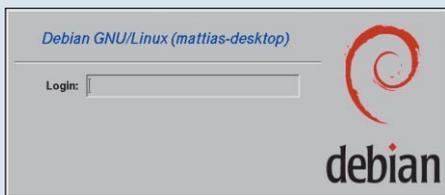
## XDM statt KDM oder GDM

☛ **Klappt die XDMCP-Konfiguration von XDM oder KDM trotz offener Firewall nicht, ist es am besten, zum alterwürdigen XDM zu greifen. Der ist in wenigen Schritten konfiguriert. Die Konfiguration befindet sich in der Regel in `/usr/X11R6/lib/X11/xdm` oder `/etc/X11/xdm`:**

**1** In der Datei `xdm-config` ist die letzte Zeile auszukommentieren, entgegen dem heutzutage üblichen Schema erledigen Sie dies mit einem Ausrufezeichen:

```
!DisplayManager.requestPort: 0
```

**2** Die `Xaccess` regelt den Zugriff. Um Chooser und direkte Anfragen zu erlauben, reicht es, wenn diese aus zwei Zeilen besteht:



Mit sprödem Charme aber schnell konfiguriert: Der mittlerweile 20 Jahre alte Log-in-Manager XDM

```
*
* CHOOSER BROADCAST
```

**3** Soll kein lokales Log-in gestartet werden, kann die `Xservers` entsprechend angepasst werden:

```
#:0 local /usr/bin/X ...
```

Nach dem Neustart des XDM sollte nun ein Log-in – mit dem etwas schlichteren Anmeldefenster – möglich sein.

Wechseln Sie nun mit `chroot` in das vorbereitete Verzeichnis und installieren Sie zusätzliche Software und den Kernel nach:

```
chroot /exports/pxebuntu804
apt-get install ssh \
xserver-xorg linux-image \
nfs-common portmap rsync \
xfonts-100dpi \
xfonts-75dpi xfonts-base \
xfonts-encodings \
xfonts-scalable
```

Nun sollten Sie ein Root-Passwort setzen und Passwort-Hashing aktivieren:

```
shadowconfig on
passwd
```

Damit die `initrd` ein via NFS gemountetes Root-Dateisystem zulässt, muss die Datei `/etc/initramfs-tools/initramfs.conf` angepasst werden:

```
MODULES=netboot
BOOT=nfs
```

Die Ramdisk entsteht nun mit dem folgenden Befehl – beachten Sie, dass Name und Versionsnummer vom installierten Kernel abhängen:

```
mkinitramfs -o /boot/initrd.img-2.6.22-
↳14-generic
```

Passen Sie noch die `/etc/fstab` des späteren Thinclients an:

```
proc /proc proc defaults 0 0
/dev/nfs / nfs defaults 1 1
none /tmp tmpfs defaults 0 0
none /var/run tmpfs defaults 0 0
```

```
none /var/lock tmpfs defaults 0 0
none /var/tmp tmpfs noauto 0 0
none /var/log tmpfs noauto 0 0
none /var/lib/xkb tmpfs noauto 0 0
```

Auf die Zeilen mit dem Schlüsselwort `noauto` gehen wir später ein. Konfigurieren Sie das Netzwerk der Terminals in der `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback
iface eth0 inet manual
iface eth1 inet manual
iface eth2 inet manual
```

Löschen Sie noch die Datei `/etc/hostname` damit der Rechnername per DHCP gesetzt werden kann. Wegen einer fehlenden Datei `/etc/mtab` konnten auf dem Testsystem die Verzeichnisse `/var/tmp`, `/var/log` und `/var/lib/xkb` nicht gemountet werden. Wir behalten uns deshalb mit einem kleinen Startscript, das als `/etc/rcS.d/S04fixmounts` verlinkt wurde – siehe Listing

1. Passen Sie zum Schluss noch die Tastaturbelegung an:

```
dpkg-reconfigure
↳console-setup
```

### Konfiguration des Servers

Auf dem Server muss der NFS-Kernelserver und das Paket `nfs-common` mit zu-

sätzlichen NFS-Werkzeugen installiert sein. Passen Sie die `/etc/exports` des Serversystems wie folgt an:

```
/exports/pxebuntu804
↳192.168.1.0/24(ro,no_root_squash)
```

Starten Sie anschließend den NFS-Server neu

```
/etc/init.d/nfs-kernel-server start
```

und testen Sie, ob Sie von einem anderen Rechner das Verzeichnis `/exports/pxebuntu710` mounten können. Die Konfiguration des DHCP- und PXE-Daemons erklärt der Artikel aus der letzten Ausgabe von `linuxlife`, den Sie auf der Heft-DVD finden. Sind beide Dienste so konfiguriert, dass der Start aus dem Netzwerk mit `memtest` funktioniert, kopieren Sie Kernel und Ramdisk aus `/exports/pxebuntu710/boot` in das TFTP-Verzeichnis und tragen beide in der Datei `pxelinux.cfg/default` ein. Da `pxelinux` manchmal mit langen Dateinamen Probleme macht, ist es ratsam, die Dateien umzubenennen:

```
DEFAULT menu.c32
PROMPT 0
TIMEOUT 500
KBDMAP german.kbd
```

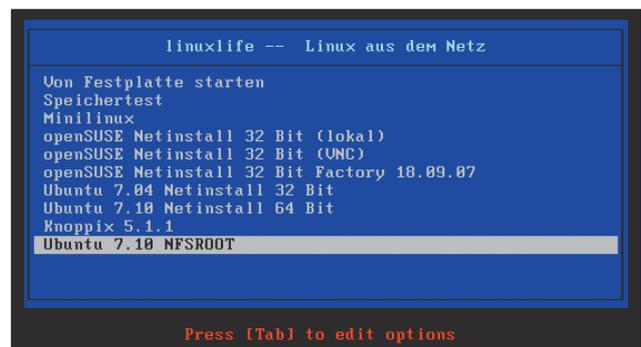
```
MENU TITLE PXE Bootmenu
```

```
LABEL ubunet
MENU LABEL Ubuntu 8.04 Thinclient
KERNEL ubu804.bzi
APPEND initrd=ubu804.dsk root=/dev/nfs
↳nfsroot=192.168.1.42:/data/pxeboot/
↳ubuntu804 ip=dhcp ro
```

Der Thinclient bootet bereits, startet aber im Textmodus, schließlich ist die Grafikkarte noch nicht konfiguriert.

### X für alle

Ein potenzielles Problem für Thinclients ist, dass sich verschiedene Rechner mit grundverschiedener Hardware das Verzeichnis `/etc/` mit Konfigurationsdateien teilen müs-



**PXELINUX als Bootloader: Die Konfiguration des DHCP- und TFTP-Daemons erklärt ein PDF aus der letzten Ausgabe auf der Heft-DVD.**

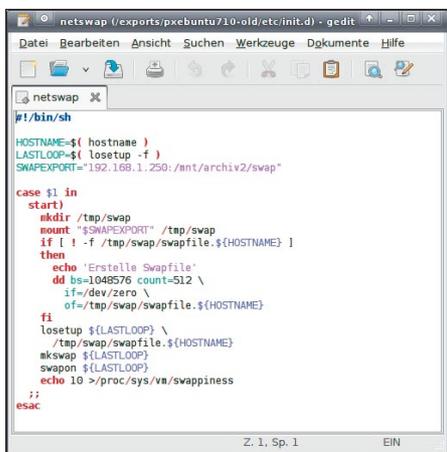
sen. Stark hardwareabhängige Konfigurationen können jedoch vom Hostnamen abhängig gemacht werden. Fehlt die Datei `/etc/X11/xorg.conf`, sucht der X-Server nach einer Datei

```
/usr/lib/X11/xorg.conf.hostname
```

Damit dieser Mechanismus funktioniert, weisen Sie zunächst im DHCP-Server dem anhand der MAC-Adresse der Netzwerkkarte eindeutig identifizierbaren Client eine immer gleiche IP-Adresse und einen Hostnamen zu:

```
host client02 {
    hardware ethernet 00:0c:29:8d:15:bc;
    fixed-address 192.168.1.25;
    option host-name „client02“;
}
```

Nach Neustart des DHCP-Servers und Reboot des Clients, ist auf Letzterem nach dem Login der Hostname (*none*) durch *client02* ersetzt. Um Xorg zu konfigurieren, melden Sie sich als Root am Client an. Da die Konfigurationsdatei `/etc/X11/xorg.conf` nur lesbar ist, benötigen Sie einen kleinen Trick, um Änderungen vornehmen zu können. Zunächst wird das Konfigurationsver-



**Feinarbeit leisten beim plattenlosen System kleine Boot-Skripte, hier für die Aktivierung einer per NFS bereitgestellten Swap-Datei.**

zeichnis ins temporäre Verzeichnis kopiert und dann über die alte gemountet:

```
rsync -avHP /etc/X11/ /tmp/X11/
mount -o bind /tmp/X11/ /etc/X11/
```

Jetzt konfigurieren Sie den Xserver und testen ihn:

```
dpkg-reconfigure xserver-xorg
Xorg -broadcast
```

Verlief der Test erfolgreich, kopieren Sie die Datei `/etc/X11/xorg.conf` unter dem Namen `/usr/lib/X11/xorg.conf.client02` in das exportierte Dateisystem und löschen Sie die `/etc/X11/xorg.conf`. Da Ubuntu 8.04 keine Auflösungseinstellung mehr per *dpkg-reconfigure* vornimmt, müssen Sie unter Umständen Frequenzen und Auflösung von Hand eintragen. Sie können dann die oben erwähnte *S99xquery* aktivieren.

### Fazit

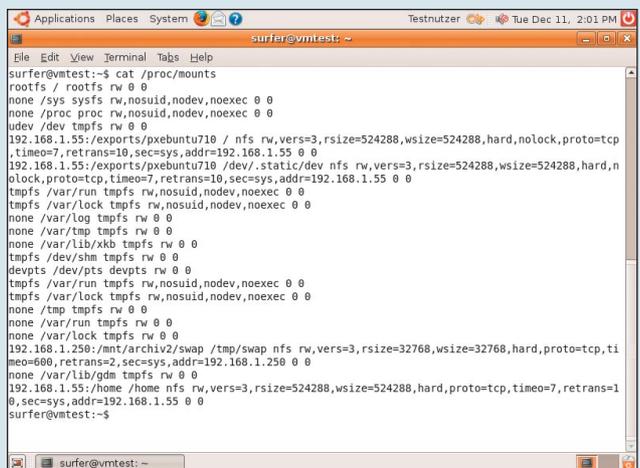
Festplattenlose Rechner unter Linux sind keine Zauberei sondern normales Handwerk, das seit jeher unter Unix vorgesehen ist. Während die Einrichtung des Grundsystems so simpel ist, dass sie sich auch lohnt, um den alten 667er-Pentium per PXE zum Thinclient zu machen, bedeutet die Versorgung eines größeren Netzes mit dünnen und halbfetten Clients einen etwas größeren Aufwand, der aber mittelfristig durch die zentrale Wartbarkeit mehr als wettgemacht wird. **jkn**

## Auf Halbfettstufe

Es gibt Einsatzbereiche, wo typische Thinclients nicht ideal geeignet sind. Typischerweise ist das der Fall, wenn lokale Rechen- oder Grafikleistung benötigt wird: Mögen vierzig Office-User auf einem aktuellen Zweikernsystem mit vier Gigabyte RAM kein großen Problem darstellen, dürfte es bereits bei zehn Programmierern, die ständig ihr Projekt durchkompilieren, bereits hinsichtlich der Rechenleistung eng werden. Ein guter Kompromiss besteht bei derartigen Umgebungen darin, zwar das gesamte System aus dem Netz zu booten, aber den Displaymanager und damit den Desktop lokal zu starten. Ein weiterer Vorteil „halbfetter Clients“ ist die problemlose Nutzung lokaler Hardware: Soundkarten, DVD-Brenner und USB-Sticks können direkt angesprochen werden, und Videos ruckeln nicht. Die Einrichtung entspricht dabei weitgehend der des Thinclients, als zusätzliche Pakete kommen die typischen Desktop-Applikationen zum Einsatz, beispielsweise *ubuntu-desktop* ohne *network-manager*. Die Heimatverzeichnisse legt man auf ein separates NFS-Exportverzeichnis, das schreibbar freigegeben wird. Für die Nutzerauthentifizierung existieren zwei Möglichkeiten: In Umgebungen mit nicht allzu vielen Clients und vertrauenswürdigen

Nutzern werden `/etc/passwd`, `/etc/shadow` und `/etc/group` mit im exportierten Verzeichnis freigegeben – hier ist allerdings darauf zu achten, dass numerische Nutzer- und Gruppen-IDs denen des exportierten `/home` entsprechen. In großen Umgebungen empfiehlt sich die Nutzung eines separaten NIS- oder LDAP-Servers. Einen spürbaren Performance-Gewinn kann die Verwendung lokaler Swap-Partitionen bedeuten: Im Arbeitsspeicher kann der Kernel dann Platz für gecachte Dateien schaffen, was ein schnelleres Ansprechen bedeutet. Denkbar sind auch lokale Partitionen für temporäre Dateien, bei der Umsetzung sollten Sie jedoch darauf achten, die Unterschiede in der erforderlichen Konfiguration gering zu halten, beispielsweise mit einem *rc-Script*, das die erste gefundene Linux-Partition unabhängig vom Festplattentyp unter

`/tmp` mountet. Für Kiosksysteme, bei denen sich mehrere PCs einen Nutzer und damit ein Heimatverzeichnis teilen, kann das bereits vorgestellte *aufs* (*Another UnionFS*) Schreibzugriffe ins RAM umlenken. Soll der Kernel nicht gepatcht werden und sind die im Heimatverzeichnis vorgehaltenen Datenmengen gering, ist ein als *tmpfs* gemountetes Heimatverzeichnis, auf das beim Start mittels *rc-Script* via *rsync* die Daten kopiert werden, die bessere Wahl.



Ubuntu und Debian lassen alleine mit Bordmitteln halbfette Clients zu, die zwar ohne eigene Platte auskommen, aber Programme lokal ausführen.