

Terminaldienste mit X2go



Ob nur im Nebenraum oder in New York: Überall steht die gewohnte Arbeitsumgebung zur Verfügung, die richtige Software, die passende Konfiguration. Der Anwender braucht sich nicht umgewöhnen, muss nichts nachinstallieren und seinem Arbeitsplatz keine Dokumente hintertragen. Möglich macht das X2go, eine beachtenswerte neue Terminalserver-Software für Linux. Heinz-M. Graesing

X2go wartet mit einer ganzen Reihe an Schmanckerln auf. Das erste fällt dem Anwender gleich ins Auge: X2go verwendet die NX-Bibliotheken, die dafür bekannt sind, die Bildschirmhalte im Vergleich zum Veteranen X11 durch trickreiches Caching und Datenkompression viel schneller zu übertragen. Dadurch lässt sich mit dem Terminalserver auch über schmalbandige Verbindungen noch flott arbeiten. In Sachen Authentifizierung setzt X2go auf Linux-Bordmittel: PAM und »sudo«, aber auch LDAP mit PAM-Modul lassen sich verwenden.

Entfernte Dateisysteme kann der Anwender sicher via SSHFS mounten, wofür nur Port 22 nötig ist. Weitere NX-Bibliotheken erlauben etwa den Nachrichtenversand an Clients oder die komplette LDAP-Verwaltung. Der auf der Grundlage von QT4 programmierte grafische Client ist sowohl als Desktop-Applikation wie auch als Login-Manager (à la XDM/KDN/GDM/LDM) verwendbar.

Schließlich deckt die Software viele Einsatzfälle vom einzelnen Arbeitsplatz bis zu großen Installationen mit sehr vielen Terminals, mehreren Servern und Load Balancing ab. Dabei macht sich eine PostgreSQL-Datenbank bezahlt, die im Hintergrund alle Sitzungsdaten verwaltet.

X2go ist allerdings noch recht jung, die Entwickler haben es erst im Sommer vergangenen Jahres vorgestellt und unter der GPL 2 freigegeben. Obwohl es sich bereits produktiv nutzen lässt, sind deshalb noch nicht alle geplanten Funktionen realisiert. So gibt es noch keine Unterstützung für lokale Drucker am Client. Auch die Dokumentation hinkt derzeit nach.

Remote Desktop

Der Terminalserver kann prinzipiell zwei Klassen von Clients bedienen: Remote Desktops und Thin Clients. Im Remote Desktop-Betrieb läuft ein NXProxy mit einem X-Server auf dem Client. Dadurch verhalten sich Programme auch bei einer schmalbandigen Verbindung fast wie lokale Anwendungen. Das ist anders als bei Protokollen wie VNC, die stets Pixeldaten anstelle von Zeichenbefehlen übertragen und deshalb sehr viel langsamer sind. Auch kommt es nicht zu den berüchtigten blinden Flecken auf dem Client-Display. Im LAN ist so kaum ein Unterschied zu lokalen Applikationen spürbar.

Manche Funktionen von X2go sind darüber hinaus besonders für große Installationen interessant. So ermöglicht die Anbindung entfernter

Filesysteme mit SSHFS einen Datenaustausch via Internet bei dem alle Fileattribute erhalten bleiben. So gemountete Dateisysteme zeigt KDE zudem auf dem Desktop an und ein normaler User mit entsprechenden Berechtigungen kann sie dort auch wieder entfernen.

Um das möglich zu machen, verwendet der Terminalserver das Kernelmodul FUSE (Filesystem in User Space). Fürs Weiterleiten von Audio-daten greift X2go auf die Netzwerkfunktionen von Arts zurück. Mit dem Wechsel auf KDE 4 wird das Projekt die künftige Multimedia-API Phonon unterstützen und von der Integration in QT4 profitieren. Weiter können X2go-Sitzungen während der Laufzeit Ihre Auflösung ändern oder im Fullscreen-Modus laufen.

In den allerersten Anfängen war X2go einmal als Alternative zu Thin Clients von Sun gedacht, deshalb ist es vielleicht kein Zufall, dass es heute als ganz besonderes Bonbon auch das beherrscht, was das Sun-Marketing mit Blick auf hauseigene Clients Hot Desking oder Session Mobility nennt: Der Anwender kann Sitzungen anhalten und später an einem anderen Rechner an der derselben Stelle wieder aufnehmen. Schon allein deshalb ist X2go für alle eine gute Alternative, die sich den Komfort einer mobilen Session, Dateifreigaben und Sound wünschen. Möglicherweise macht es am Anfang ein wenig mehr Arbeit als ein kommerzielles Produkt, doch die macht sich bezahlt.

Thin Client-Betrieb

Auch bei der Unterstützung von Thin Clients kann X2go einige Stärken ausspielen. Startet der X2go-Client ohne Konfiguration, erscheint eine zweigeteilte Ansicht, die das Projekt Cardview nennt. Der rechte Bereich listet konfigurierte aber nicht aktive Sitzungen. Hier lassen sich Verbindungsdaten wie Username, Password, Server, Port oder Auflösung zuweisen.

In einem größeren Netz wäre es sicher keine gute Lösung, wenn die Benutzer diese Parameter jedes Mal aufs Neue fehlerträchtig eintippen sollten. Deshalb kennt der Client auch einen LDAP-Modus, der alle diese Daten aus dem zentralen Datenspeicher holt.

Zusätzlich kann man per Befehlszeile Toolbar und Menü entfernen um so ein Programm zu erhalten, dass sich als Loginmanager verwenden lässt. Ist alles richtig konfiguriert, bietet die Cardview nicht mehr die Sitzungsdaten, sondern die im LDAP hinterlegten Benutzer zur

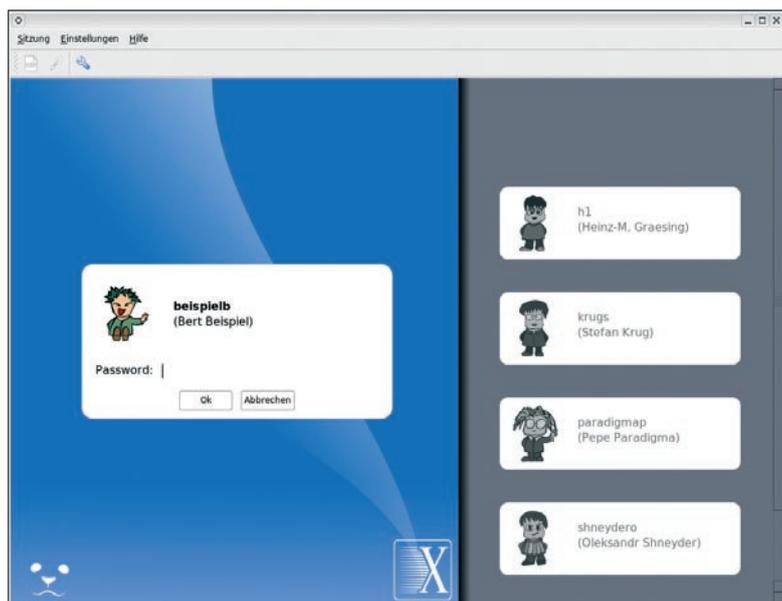


Abbildung 1: Alle im LDAP hinterlegten Benutzer erscheinen automatisch in einer Auswahl auf dem Anmeldeschirm.

Auswahl an (Abbildung 1). Der Client lässt sich so auf Minimaldistributionen, Live-CDs oder Netzwerkbootimages hinterlegen, da er keine weitere Konfiguration mehr benötigt. Für Netzwerkboot-Installationen bietet das Projekt eine Anzahl von Konfigurationsskripten an. Außerdem diskutieren die Entwickler auch über ein erweitertes grafisches Tool für die Netzwerkadministration, das DHCP-Zuweisungen und TFTP-Konfigurationen berücksichtigen soll.

Identität via Smartcard

Ein besonderes Highlight ist die Sitzungsauthentifizierung über USB-Stick oder Smartcard – wieder meint man das Vorbild Sun zu erkennen. X2go kann Benutzer anhand von Smartcards oder präparierten USB-Sticks erkennen und eine angehaltene Sitzung automatisch fortführen, sobald der Anwender eines dieser Authentifizierungsmedien (Abbildung 2) einsteckt.

Dabei spielt es keine Rolle, ob die Verbindung über LAN, WLAN oder über das Internet zustande kam. So kann der User eine Sitzung zum Beispiel an seinem Arbeitsplatz durch das Abstecken eines USB-Sticks anhalten und zuhause durch das Einstecken wiederaufnehmen. Eventuell vorhandene unterschiedliche Auflösungen gleicht der Client selbstständig aus. Da die Authentifizierung über USB-Stick nur eine etwa fünf MByte große CryptoFS-Partition benötigt,

kann der Rest als Datenspeicher dienen, der im Thin Client-Modus automatisch per SSHFS am Server gemountet wird.

Neben dem grafischen QT4-Client existiert auch ein Kommandozeilen-Client (Abbildung 3), der es ermöglicht, ohne großen Aufwand, Frontends für andere Toolkits umzusetzen. Auch ist die Verwendung aus Applikationen heraus denkbar, die eine X2go-Sitzung etablieren wollen.

Da die Clients Open Source sind, dürfte es keine Probleme in puncto Hardwarekompatibilität und Abhängigkeit von bestimmten Herstellern geben. Schon jetzt lassen sich auch andere Desktopsysteme als KDE über X2go veröffentlichen. Durch die Integration von nativen Clients auf diese Weise bleibt das Look&Feel der Desktopumgebung noch besser gewahrt.

Module fürs Kontrollzentrum

Auch das können vergleichbare Projekte kaum vorweisen: X2go verfügt über eine ganze Reihe von Administrationsprogrammen, die sich nahtlos in das KDE-Kontrollzentrum einfügen. So existiert ein Systemverwaltermodul, das Funktionen freigibt, die eine größere Zahl von Benutzern betreffen. Einige der Module taugen darüber hinaus sogar für die tägliche Netzwerkadministration, etwa des LDAP. Denn X2go verwendet keine eigenen LDAP-Schemata, so dass auch bestehende LDAP-Installationen verwendbar sind, vorausgesetzt die Benutzer existieren als Posix-Accounts.

Im einzelnen sind Module für die Benutzeradministration, Gruppenadministration (Abbildung 4), Namensauflösung und die benötigte Sitzungsverwaltung vorhanden. Alle Module besitzen eine Live-Suche und eine Checkbox zum Umschalten der Listviews in einen Treemodus. Das Aktivieren einer solchen Checkbox macht zum Beispiel sichtbar, auf welchem Server eine Sitzung läuft oder in welcher Gruppe sich ein Benutzer befindet. Von allen Modulen, die Benutzerdaten verwalten, ist es darüber hinaus möglich, den Usern eine Nachricht über das projekteigene Rich-Text-Messaging System zuzustellen.

Session-Sharing

Ein weiteres interessantes Feature der Kontrollzentrumsmodulen ist die Sitzungsüberwachung. Weil X2go zunächst für das schulische Umfeld entwickelt wurde, können der Root-Benutzer (oder ein Sudo-User) Sitzungen beobachten

und beeinflussen. Für die Überwachung greift das Projekt auf X11vnc zurück. Da die neuesten NX-Libraries einen Broadcast-Modus enthalten, wird derzeit geprüft, ob das Session-Sharing auch über die eigenen Funktionen von NX umsetzbar ist. Für die Darstellung der überwachten Sitzung nutzt X2go die KDE Remote Desktop Connection (KRDC).

Ein weiteres Produkt des ursprünglichen Entwicklungsziels Schuleinsatz ist X2gospyglass, mit dem sich mehrere X2go-Sitzungen gleichzeitig überwachen lassen. Ähnlich wie bereits etablierte Lösungen bietet es auch eine Classroom-View im Thumbnail-Format. Dabei lassen sich die Sitzungen so zusammenfassen, wie es der räumlichen Verteilung der Hardware entspricht. Es ist aber auch möglich, andere Auswahlkriterien wie etwa Posix-Gruppen zu nutzen. Um maximalen Platz für den Anzeigebereich zu schaffen, verzichtet X2gospyglass auf Toolbars oder Icons. Für die Konfiguration steht ein Menü zur Verfügung. Die Thumbnails passen sich in der Größe dem verfügbaren Platz an. Für die Benachrichtigung von Benutzern bietet X2go X2gomail. Ähnlich wie in den Kontrollzentrummodulen, hilft auch hier die Livesuche bei der Auswahl der Empfänger. X2gomail ist in die Module integriert und lässt sich auch von dort aus aufrufen.

Basis-Installation

Die Basisinstallation benötigt nicht mehr als etwa 15 Minuten. Voraussetzung ist momentan eine Debian Etch-Installation, die Unterstützung weiterer Distributionen ist angekündigt. Zur Installation des Servers und des Clients kann der Admin auf ein Online-Repository zurückgreifen. Dazu muss »/etc/apt/sources.list« folgende Zeile enthalten

```
# x2go Repository
deb http://x2go.obviously-nice.de/deb/ 2
testing main
```

Das Repository enthält Server, Client und Zusatzanwendungen. Ist es eingetragen, aktualisiert man die Apt-Datenbank mit »apt-get update«, und installiert anschließend den Client mit:

```
apt-get install x2goclient
```

Danach steht der X2goclient zur Verfügung. Über die Optionen, die es erlauben X2goclient als LDAP-Browser oder Vollbild-Loginmanager zu verwenden, informiert »x2goclient --help«.



Abbildung 2: Auf einer Smartcard oder einem USB-Stick kann der Anwender seine Sitzung mitnehmen. Steckt er die Karte an einem anderen Terminal ein, geht es dort weiter, wo er aufgehört hatte.

Für den Vollbild-Modus gibt es beispielsweise die Option »--maximize«. X2goserver lässt sich ebenfalls über das Repository beziehen. Die Installation startet nach:

```
apt-get install x2goserver sshfs 2
postgresql
```

Um festzulegen, welcher Systemgruppe es erlaubt ist, sich am X2goserver anzumelden, ist ein Eintrag in der Datei »/etc/sudoers« nötig. Verwendet man zum Hinzufügen der Zeile den Befehl »visudo«, prüft das Tool die Eingaben gleich auf Validität:

```
#Zusätzliche Zeile /etc/sudoers für 2
x2go (bestimmt loggingruppe)
%users ALL=(ALL) NOPASSWD: 2
/usr/bin/x2gopgwrapper
```

Das Beispiel erlaubt jedem Benutzer der Gruppe »users«, sich am X2go-System anzumelden. Weiterhin muss der Benutzer Teil der Gruppe »fuse« sein – nur dann kann er auch am Client, angeschlossene Dateisysteme verwenden.

Um die benötigte Datenbank anzulegen ruft man das Skript

```
/usr/lib/x2go/script/x2gocreatebase.sh
```

von einer Root-Konsole auf. Nachdem der Installateur den X2goserver mit »/etc/init.d/x2goserver restart« neu gestartet hat, sollte er

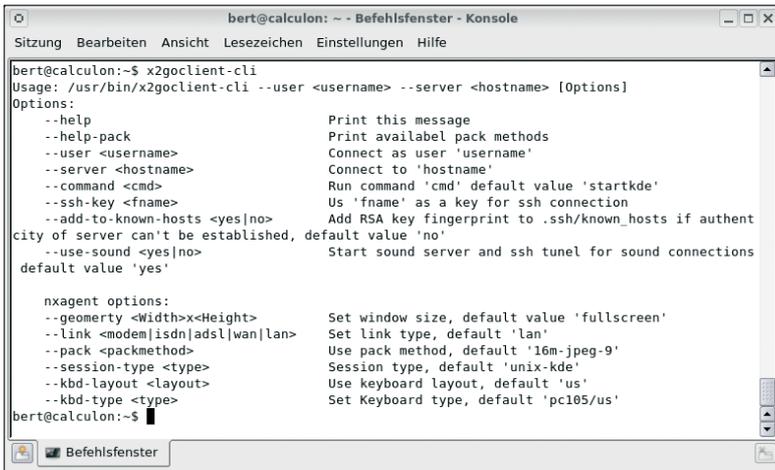


Abbildung 3: Der Kommandozeilenclient gestattet die Integration von X2go in andere Anwendungen.

sich durch »x2goclien« mit dem Server verbinden können.

Je nach Release der Debian-Installation kam es hin und wieder zu Problemen mit »/dev/fuse« kommen. Deshalb sollte man mit dem folgenden Befehl prüfen, ob das Device existiert: »stat /dev/fuse«. Fehlt das Device sind folgende zusätzliche Befehle notwendig, die in einer Root-Konsole auszuführen sind:

```
mknod /dev/fuse c 10 229
chown root:fuse /dev/fuse
chmod 660 /dev/fuse
```

Zur Verwaltung laufender Sitzungen eignet sich das KDE-Kontrollzentrumsmodul »x2gosession administration«. Weitere X2go-Module benötigen allerdings eine konfigurierte LDAP Installation.

LDAP-Installation

Jetzt geht es an die Konfiguration der Dienste. Zuerst ist die Samba-Konfigurationsdatei »/etc/samba/smb.conf« zu editieren und sicherzustellen, dass dort ein Eintrag nach dem Muster »workgroup = localdomain« existiert. Im Konfigurationsverzeichnis »/usr/share/x2goldaptools/config« lassen sich nun mit Hilfe von »./genconf« die nötigen Dienste einrichten. »./genconf« erwartet eine Eingabe nach folgendem Muster:

```
./genconf LDAP_URI organization country ?
domain netbiosname LDAPMASTER_URI
```

Auf die Konfiguration des LDAP-Servers folgt die der Zertifikate und Dienste. Dabei sollte man sehr genau vorgehen und sich am besten notie-

ren, was man in welches Feld eingibt. Als erstes wird im Verzeichnis »/usr/share/x2goldaptools/script/« das Skript »./makeCA« gestartet. Die Ausgabe »CA certificate filename« quittiert man mit [Enter]. Danach ist nach »Enter PEM passphrase« ein Passwort einzugeben, das später wieder benötigt wird. Für »Country Name« ist ein zweibuchstabiges Länderkürzel einzugeben, also beispielsweise »DE«. Hinter »State or Province Name« trägt der Admin ein Bundesland ein und bei »Locality Name« einen Städtenamen. »Organization Name« verlangt die Eingabe des Firmennamens und »Common Name« den eines Ansprechpartners, dessen E-Mail-Adresse nimmt das folgende Feld auf. Die folgenden beiden Abfragen nach dem »challenge password« und dem »optional company name« lässt man am besten frei. Auf die Aufforderung »Enter pass phrase for ./demoCA/private/./cakey.pem:« reagiert man mit dem oben bereits eingegebenen Passwort.

Anschließend signiert der Installateur mit »./makenewcert« das Serverzertifikat. Dabei sind noch einmal die gleichen Informationen einzugeben wie eben. Die Frage, ob signiert werden soll ist doppelt zu bejahen. Der LDAP-Server wird jetzt automatisch gestartet, die Kommunikation läuft ab sofort verschlüsselt ab. Danach rebootet man den LDAP-Server mit »/etc/init.d/slapd restart«.

Unmittelbar anschließend sollte der Anwender den LDAP-Baum füllen. Das Kommando »./initsystem« das aus »/usr/share/x2goldaptools/script/« starten muss, kopiert alle lokalen Benutzer, Benutzergruppen, Geräte und so weiter in den LDAP-Baum. Zugleich vergibt es die Passwörter für Samba vergeben.

Jetzt kann der Admin X2go in den LDAP Modus umschalten:

```
ldap.addserver Servername IP-Adresse
cp -R /usr/share/x2goldaptools/etc/* ?
/etc
```

Geschafft! Nun kann der Benutzer nach dem Aufruf von »apt-get install luma« mit »luma« durch den LDAP-Baum browsen.

Boot-Umgebung

Die vorangegangenen Erläuterungen setzten auf dem Client ein existierendes und konfiguriertes Betriebssystem voraus. Will man X2go in einer Thin Client Umgebung betreiben, in der die Clients über keine Festplatte (oder einen Flash-

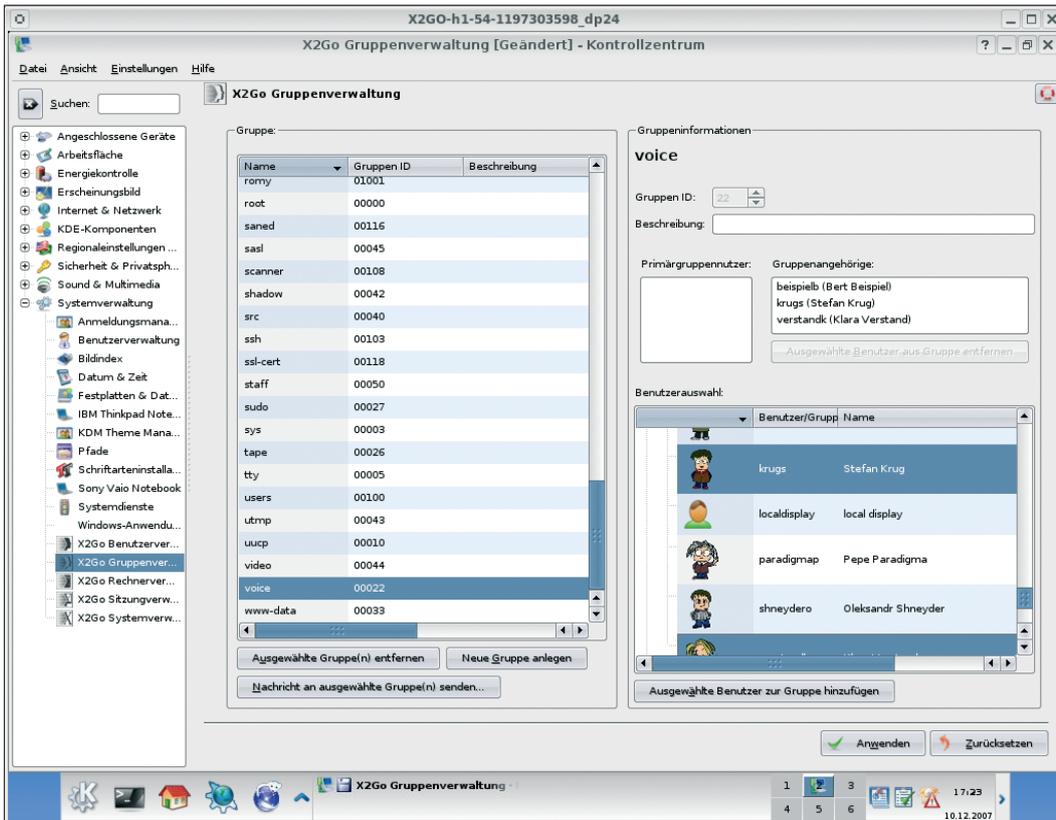


Abbildung 4: Die X2go-Gruppenverwaltung als KDE-Modul. Manche der Module taugen sogar für all-tägliche Administrationsaufgaben jenseits des Terminalserverbetriebs.

Speicher) verfügen, lässt sich auch via Netzwerk booten. Dafür sind aber zunächst erst ein paar weitere Pakete nachzuinstallieren. Das erledigt das Kommando:

```
#apt-get install dhcp atftpd ↵
nfs-kernel-server debootstrap
```

Der nun folgende Befehl installiert ein neues Debian-System in ein Verzeichnis, das der Anwender frei wählen kann. Im Beispiel ist das »/opt/x2goclient«:

```
#debootstrap --arch i386 etch ↵
/opt/x2gothinclient ↵
http://ftp.de.debian.org/debian/
```

Damit das neu kopierte System nach dem Start via »chroot« auch Rechnernamen in IP-Adressen auflösen kann, benötigt es eine Datei »resolv.conf«:

```
#cp /etc/resolv.conf ↵
/opt/x2gothinclient/etc/
```

Auch eine Netzwerkkonfiguration ist nötig. Sie entsteht in folgenden Schritten:

```
#echo "auto loiface lo inet loopback" >> ↵
/opt/x2gothinclient/etc/network/ ↵
interfaces
#echo x2gothinclient > ↵
/opt/x2gothinclient/etc/hostname
#echo "127.0.0.1 localhost" >> ↵
/opt/x2gothinclient/etc/hosts
#echo "192.168.0.1 x2goserver" >> ↵
/opt/x2gothinclient/etc/hosts
```

Ist mehr als ein X2goserver im Spiel, ist der letzte Schritt für jeden zu wiederholen. Um im neu installierten System auf das X2go-Repository zurückgreifen zu können, schreibt man es in »/etc/apt/sources.list«:

```
#echo "deb http://x2go.obviously-nice.de ↵
/deb/ testing main" >> ↵
/etc/apt/sources.list
```

Nun lässt sich das System über »chroot« starten. »#chroot /opt/x2gothinclient/«. Alle folgenden Befehle müssen in dieser Chroot-Umgebung laufen. Um die Installation startfähig zu machen, fehlen noch der Kernel und weitere Programme einer Linux Basisinstallation: ▶

```
#apt-get update
#apt-get install locales
#dpkg-reconfigure locales
#apt-get install kernel-image-2.6-686 ↗
syslinux
```

Zum Abschluss muss die Installation für das X2go-System konfiguriert werden. Installieren Sie hierzu das Paket »x2gothinclientsystem« (Dabei sollte sich der Installateur)immer noch in der chroot Umgebung befinden:

```
#apt-get install x2gothinclientsystem
#cd /usr/share/x2gothinclientsystem/↗
script/
#./x2gothinclient_install.sh
#exit
```

Das Netzwerkbootssystem ist damit einsatzbereit. Jetzt gilt es nur noch, die im folgenden beschriebenen Infrastrukturdienste zu konfigurieren.

Netzwerk-Konfiguration

Das betrifft zuerst einen FTP-Server, für den in »/etc/default/atftpd« eine Konfiguration nach folgendem Beispiel zu hinterlegen ist:

```
USE_INETD=false
OPTIONS="--daemon --port 69
--tftpd-timeout 300 --retry-timeout 5
--mcast-port 1758 --mcast-addr
239.239.239.0-255
--mcast-ttl 1 --maxthread 100
--verbose=5 /tftpboot"
```

Die erste Zeile bewirkt, dass nicht der Inetd-Superserver der den FTP-Daemon »atftpd« startet und stoppt, sondern das Init-Skript »/etc/init.d/atftpd« Das angegebene Verzeichnis »/tftpboot« muss existieren, gegebenenfalls legt man es vorher an.

Nun folgt die PXE-Bootkonfiguration:

```
#cp /opt/x2gothinclient/boot/vmlinuz-* ↗
/tftpboot/
#cp /opt/x2gothinclient/boot/initrd.img-* ↗
/tftpboot/
#ln -s /tftpboot/vmlinuz-*/tftpboot/ ↗
vmlinuz
#ln -s /tftpboot/initrd.img-*/tftpboot/ ↗
initrd.img
#cp /opt/x2gothinclient/usr/lib/syslinux/↗
pxelinux.0 /tftpboot/
```

Anschließend füllt der Admin »/tftpboot/pxelinux.cfg/default« nach folgendem Vorbild:

```
label linux
kernel vmlinuz
append root=/dev/nfs nfsroot=↗
192.168.0.1:/opt/x2gothinclient ro
initrd=initrd.img ip=dhcp
```

Der DHCP-Server wird wie im folgenden Beispiel über die Datei »/etc/default/dhcp/dhcp.conf« eingerichtet:

```
group
{
    subnet 192.168.0.0 netmask 255.255.255.0
    {
        default-lease-time 720000;
        max-lease-time 1440000;
        authoritative;
        range 192.168.0.10 192.168.0.100;
        filename "/pxelinux.0";
        next-server 192.168.0.1;
    }
}
```

Ganz zum Schluss exportiert der Admin »/opt/x2gothinclient« via NFS

```
/opt/x2gothinclient 192.168.0.0/24 ↗
(ro, async, no_root_squash)
```

und schreibt die Informationen über den oder die X2go-Server ins LDAP:

```
#ldap.addserver Name IP
```

Fazit

X2go ist schon jetzt eine beachtliche Terminalserver-Alternative für den Produktivbetrieb mit etlichen interessanten Eigenschaften wie Smartcard-Authentifizierung und mobile Sessions.

Auf der Agenda der nächsten Zeit stehen weitere interessante Features wie etwa Seamless Windows, die es ermöglichen sollen, dass sich Anwendungen vom Terminalserver mit lokalen Applikationen einen Desktop teilen. Auch eine Live-CD plant das Projekt für die nähere Zukunft. Neben den derzeit verwendbaren Betriebssystemen Debian Etch (für Server und Clients) sowie Windows XP (für Clients) sind noch weitere Plattformen geplant. Auch sollen zusätzliche grafische Tools die Administration weiter erleichtern. (jcb) ■■■

Infos

(1) X2go: (<http://x2go.berlios.de>)