ThinClients in der IVV7

Oliver Baltz
<info@OliverBaltz.de>

23.1.2004
Erstauflage

25.11.2005
Dynamische Zuweisung der Startseiten
Anpassung des TFTP-Pfades
Wegfall des PXE-Servers

Inhaltsverzeichnis

1.1 Einleitung

Die IVV7 suchte eine effektive Lösung, die Rechercherechner verschiedener Fachbereiche einfach zu administrieren.
Wir entschieden uns für die "PXE-Boot"-Methode, bei der alle Clients über das TFTP-Protokoll - eine vereinfachte FTP-Methode - ihr gesamtes Betriebssystem über das Netzwerk in den Arbeitsspeicher laden.

Diese Methode ist:

- ✔ **kostengünstig** (für Server und Clients auf Linuxbasis fallen keine Lizenzgebühren an / alte Hardware kann verwendet werden)
- ✔ **ausfallsicherer** (da der Client völlig laufwerkslos ist, reduziert sich die Zahl der möglichen Fehlerquellen drastisch)
- ✔ **zeitsparend** (Änderungen am Betriebssystem müssen lediglich am zentralen Dateisystem vorgenommen werden)
- ✔ **einfach skalierbar** (weitere Clients können simpel in die *ThinClient*-Landschaft aufgenommen werden)

Der derzeitige *ThinClient* hat folgende **Features**:

- ✔ Debian Linux 2.4.32 (sarge-Zweig)
- ✔ Dateisystem: 90 MB (im RAM des Clients entpackt 230 MB)
- ✔ Mozilla Firefox 0.93-2
- ✔ Citrix MetaFrame Presentation Server Clients, Version 8
- ✔ Macromedia Flash Player 7
- ✔ HTTP-Umleitung über transparenten Proxy
- ✔ restriktive iptables-Firewall für eingehenden Datenverkehr

Für den *ThinClient* kann folgende Hardware verwendet werden:

- ✔ CPU >= 400 Mhz
- ✔ Arbeitsspeicher >= 256 MB
- ✔ PXE-fähige Netzwerkkarte
- ✔ keine Festplatte oder sonstige Laufwerke nötig

Eine Übersicht der derzeitigen *ThinClient*-Landschaft entnehmen Sie bitte der Anlage.

2 Installation

Die nachfolgende Installationsanleitung richtet sich an erfahrene Linuxadministratoren. Gute Kenntnisse im Umgang mit der Distribution *Debian* werden vorausgesetzt!

2.1.1 Erstellen eines Client-Dateisystems

Erstellen und Mounten eines 300MB großes EXT3-Dateisystems:

```
dd if=/dev/zero of=pxefs bs=1M count=300
mkfs.ext3 pxefs
mkdir /mnt/pxefs
mount -o loop pxefs /mnt/pxefs
```

Installieren eines schlanken und stabilen Debian-Betriebssystems im neu erstellten Dateisystem mittels *debootstrap*:

```
debootstrap sarge /mnt/pxefs
```

In die neue Betriebssystemumgebung wechseln:

```
chroot /mnt/pxefs
```

Die Installationsumgebung anpassen:

```
apt-setup
apt-get update
```

Die Datei */etc/inittab* sollte folgendermaßen aussehen:

```
id:2:initdefault:
si::sysinit:/bin/mount -t proc proc /proc
l2:2:wait:/etc/rc 2
z6:6:respawn:/sbin/sulogin
```

Erstellen eines Benutzers und einer Struktur für schreibbare Verzeichnisse:

```
adduser kiosk
mkdir /opt/home
```

2.1.2 Kompilieren eines *ThinClient*-Kernels

Der *ThinClient*-Kernel benötigt diverse zusätzliche Informationen, um mit RAM-Disks umgehen zu können. Wir haben uns für einen stabilen 2.4.32-Kernel entschieden. Nutzt man die *menuconfig*-Option, um die Kernelkonfiguration zu ändern, benötigt man zusätzlich folgende Kerneloptionen:

```
Block devices
   <*> RAM disk support
   (4096) Default RAM disk size
   [*] Initial RAM disk (initrd) support

File systems
   <*> Compressed ROM file system support
```

Nun kann der Kernel auf die Hardware des *ThinClients* angepasst werden.
Es empfiehlt es sich, den *ThinClient*-Kernel möglichst schlank zu halten, indem wirklich nur benötigte Treiber ausgewählt werden.

Anschließend wird der entsprechend konfigurierte Kernel kompiliert. Wichtig ist, dass der Installations-Pfad für die kompilierten Module in das zuvor angelegte *ThinClient*-Dateisystem zeigt.

```
make dep bzImage modules modules_install INSTALL_MOD_PATH=/mnt/pxefs/
```

Das fertige *bzImage* muss nun dem TFTP-Server als *ThinClient*-Kernel zur Verfügung gestellt werden.

```
cp arch/i386/boot/bzImage /var/lib/tftpboot/pxekernel
```

2.2 Konfiguration des *ThinClients*

Folgende Startskripte im Verzeichnis */etc/init.d* werden nacheinander gestartet:

- ➔ S01-make-rw-filesystem
- ➔ S02-start-network
- ➔ S03-firewall
- ➔ S04-keymap
- ➔ S05-define-homepage
- ➔ S06-start-kiosk
- ➔ S07-check-kiosk

2.2.1.1 /etc/init.d/S01-make-rw-filesystem

Dieses Startskript erstellt in der normalerweise nur lesbaren RAM-Disk schreibbar gemountete Verzeichnisse des Typs *tmpfs*.

```
#!/bin/sh
#S01-make-rw-filesystem

# Mount TMPFS
/bin/mount -t tmpfs -o size=32768K tmpfs /var
/bin/mount -t tmpfs -o size=32768K tmpfs /tmp
/bin/mount -t tmpfs -o size=32768K tmpfs /home/kiosk

# Create dirs in TMPFS
/bin/mkdir -p /var/lock
/bin/mkdir -p /var/log
/bin/mkdir -p /var/run
/bin/mkdir -p /var/tmp
/bin/mkdir -p /var/lib
/bin/mkdir -p /var/lib/xkb
/bin/mkdir -p /var/lib/dhcp

# Make /etc writeable
/bin/mkdir /tmp/etc
/bin/cp -dpR /etc/* /tmp/etc
/bin/mount -t tmpfs -o size=4096K tmpfs /etc
/bin/cp -dpR /tmp/etc/* /etc
/bin/rm -Rf /tmp/etc

# Create Mozilla`s TMPFS
/bin/mount -t tmpfs -o size=32768K tmpfs /usr/lib/mozilla-firefox
/bin/cp -dpR /opt/var/lib/mozilla-firefox /var/lib/mozilla-firefox
/bin/cp -dpR /opt/usr/lib/mozilla-firefox/* /usr/lib/mozilla-firefox
/bin/cp -dpR /opt/home/kiosk/.bash* /home/kiosk/
/bin/cp -dpR /opt/home/kiosk/.mozilla /home/kiosk/
/bin/cp -dpR /opt/home/kiosk/.fluxbox /home/kiosk/

/bin/chmod -R 755 /home/kiosk/
/bin/chown -R kiosk.kiosk /home/kiosk/
```

2.2.1.2 /etc/init.d/S02-start-network

Dieses Startskript setzt die Umgebungsvariable *HOSTNAME* und die vom DHCP-Server gelieferten Netzwerkparameter des *ThinClients*.

```
#!/bin/sh
#S02-start-network
HOSTNAME=`cat /etc/hostname`
/bin/hostname $HOSTNAME
# Bring up the local loopback
/sbin/ifconfig lo 127.0.0.1
/sbin/dhclient eth0
```

2.2.1.3 /etc/init.d/S03-firewall

Ein iptables-Skript blockiert bis auf die IP eines Monitoring-Servers alle eingehenden Verbindungen. Außerdem veranlasst es den Client, sämtlichen HTTP-Traffic über den Proxy "ivv7proxy.uni-muenster.de" umzuleiten. Es handelt sich somit um einen für den *ThinClient* transparenten Proxy.

```
iptables -t nat -A OUTPUT -p tcp --dport 80 -j DNAT --to <PROXY-IP>:<PORT>
```

2.2.1.4 /etc/init.d/S04-keymap

Um ein korrektes deutsches Tastaturlayout zu erhalten, muss ein Keymap-Startskript existieren. Dazu muss ein symbolischer Link gesetzt werden:

```
ln -s /etc/init.d/keymap.sh /etc/rc2.d/S04-keymap
```

2.2.1.5 /etc/init.d/S05-define-homepage

Da das *ThinClient*-Dateisystem in mehreren Fachbereichen eingesetzt wird, die jedoch unterschiedliche Browser-Startseiten einsetzen möchten, werden die Startseiten auf Grund des PTR-DNS-Eintrags dynamisch in der Firefox-Konfiguration gesetzt. Alle Startseiten verweisen laut Skript auf die URL http://ivv7srv15.uni-muenster.de/pxestart-$HOSTID. Es werden alle Zahlen aus dem String des PTR-Eintrags entfernt, damit ganze *ThinClients*-Gruppen die gleiche Startseite wählen können.

Ein Besipiel: Ein *ThinClient* der Erziehungswissenschaft den PTR-Eintrag *ewbib01.uni-muenster.de*. Somit wird die Startseite http://ivv7srv15.uni-muenster.de/pxestart-ewbib der Firefox-Konfiguration angefügt.

```
# This script gets the hostname and defines the correct Browser-Homepage

case "$1" in
  start)

        HOMEPAGE_FILE="/home/kiosk/.mozilla/firefox/default.886/prefs.js"

        # get ip
        IP=`ifconfig  eth0  | grep Bcast | cut -d ":" -f 2 | cut -d " " -f 1`

        # get hostname
        HOSTNAME=`host $IP | grep Name | cut -d " " -f 2 | cut -d "." -f 1`

        # get first 3 bytes from hostname to define $HOSTID
        HOSTID=`echo $HOSTNAME | sed s/[0-9]//g`

        # convert upper chars to lower (case-sensitive webserver!)
        HOSTID=`echo $HOSTID | tr [:upper:] [:lower:]`

        # write missing line to mozilla-configuration
        echo user_pref\(\"browser.startup.homepage\", \"http://ivv7srv15.uni-muenster.de/pxestart-$HOSTID\"\)\; \
       >> $HOMEPAGE_FILE

exit 0
        ;;
  stop)
        ;;
  *)
        echo "Usage: /etc/init.d/S05-define-homepage {start|stop}"
        exit 1
esac

exit 0
```

2.2.1.6 /etc/init.d/S05-start-kiosk

Nachfolgend wird das Startskript beschrieben, welches den Grafikkartenhersteller ausliest und die entsprechend angepasste *XF86Config-4* lädt. Im Verzeichnis */etc/X11* liegen die Konfigurationsdateien im Format *XF86Config-4.<VGAHERSTELLER>*.
Der X-Server wird als Benutzer *kiosk* gestartet und hat somit stark eingeschränkte Systemrechte.

```
#!/bin/sh
#S05-start-kiosk

export http_proxy="http://ivv7proxy.uni-muenster.de:8080"
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
export LANG=de_DE@euro
VGA=`lspci | grep VGA | cut -d " " -f 5`

case "$1" in
  start)
      if [ "$VGA" == "Matrox" ]; then
            echo Whee! You do have a MATROX inside!
            su kiosk -c "startx /usr/bin/fluxbox -- -xf86config XF86Config-4.mga&"
      elif [ "$VGA" == "Silicon" ]; then
            echo Whee! You use a SIS VGA Adapter!
            su kiosk -c "startx /usr/bin/fluxbox -- -xf86config XF86Config-4.sis&"
      elif [ "$VGA" == "nVidia" ]; then
            echo Whee! You use a nVidia VGA Adapter!
            su kiosk -c "startx /usr/bin/fluxbox -- -xf86config XF86Config-4.nvidia&"
      elif [ "$VGA" == "ATI" ]; then
            echo Whee! You use a ATI VGA Adapter!
            su kiosk -c "startx /usr/bin/fluxbox -- -xf86config XF86Config-4.ati&"
      else
            echo Uuuhm. You use a $VGA-VGA and I am going to use VESA-VGA-Modules!
            su kiosk -c "startx /usr/bin/fluxbox -- -xf86config XF86Config-4.vesa&"
      fi
      sleep 5 #wait for fluxbox to come up
      su kiosk -c "export DISPLAY=":0"; exec /usr/bin/firefox -height 200 &"
exit 0
      ;;
  stop)
      ;;
  *)
      echo "Usage: /etc/init.d/kiosk {start|stop}"
      exit 1
esac

exit 0
```

2.2.1.7 /etc/X11/XF86Config-4

Hier ein Beispiel einer Konfigurationsdatei des X-Servers. Speziell für die *ThinClients* wurde sie so angepasst, dass man sowohl USB-, als auch PS/2-Mäuse mit dieser X-Server-Konfiguration benutzen kann.

```
Section "InputDevice"
      Identifier     "PS2-Mouse"
      Driver         "mouse"
      Option         "Device"              "/dev/psaux"
      Option         "Protocol"            "ImPS/2"
      Option         "Emulate3Buttons"     "true"
      Option         "ZAxisMapping"        "4 5"
EndSection
Section "InputDevice"
      Identifier     "USB-Mouse"
      Driver         "mouse"
      Option         "Device"              "/dev/input/mice"
      Option         "Protocol"            "ImPS/2"
      Option         "Emulate3Buttons"     "true"
      Option         "ZAxisMapping"        "4 5"
EndSection
```

Anpassen des *ServerLayouts* mit den geänderten *InputDevices*:

```
Section "ServerLayout"
      Identifier     "Default Layout"
      Screen         "Default Screen"
      InputDevice    "Generic Keyboard"
      InputDevice    "USB-Mouse" "CorePointer"
      InputDevice    "PS2-Mouse" "SendCoreEvents"
EndSection
```

2.2.1.8 /etc/init.d/S06-check-kiosk

Es soll sichergestellt werden, dass für den Benutzer zu jeder Zeit ein geöffnetes Browserfenster zur Verfügung steht. Sobald der Browser *Firefox* gestartet ist, wird eine Lock-Datei im Homeverzeichnis des Users angelegt (*home/kiosk/.mozilla/firefox/default.886/lock)*. Auch der X-Server legt nach erfolgreichem Start eine Lock-Datei an (*/tmp/.X0-lock)*. Alle 5 Sekunden wird in diesem Skript die Existenz der beiden Dateien geprüft und gegebenenfalls der Browser und/oder X-Server neu gestartet.
Es ist wichtig, dass dieses Skript zu allerletzt aufgerufen wird!

```
#!/bin/sh
#S06-check-kiosk

case "$1" in
  start)
    sleep 60
      while true
      do
        FLUXBOX="/tmp/.X0-lock"
        FIREFOX="/home/kiosk/.mozilla/firefox/default.886/lock"

        if [ ! -e $FLUXBOX ]
        then
          echo Restarting Fluxbox...
          /etc/init.d/S05-start-kiosk start
          sleep 60

        elif [ ! -h $FIREFOX ]
        then
          echo Restarting Firefox...
          su kiosk -c "export DISPLAY=":0"; exec /usr/bin/firefox&"
          sleep 10
        fi
        sleep 5
      done
      ;;
  *)
    echo "Usage: `basename $0` {start}"
    exit 1
    ;;
esac
exit 0
```

2.2.2 *Mozilla Firefox*

Der Browser *Mozilla Firefox* kann wie gewohnt installiert werden:

```
apt-get install mozilla-firefox
```

Alle zentralen Bowsereinstellungen können in der Datei

/home/kiosk/.mozilla/firefox/default.886/prefs.js

vorgenommen werden. Hier findet man z. B. die URL der Startseite. Dabei sollte man besonders auf die korrekt gesetzten Rechte dieser Dateien achten. Der User *kiosk* muss Lese- und Schreibrechte besitzen.

Für das Flash-Plugin gibt es in den *contrib*-Sources ein Debianpaket.

```
apt-get install flashplugin-nonfree
```

2.2.3 Abschließende Änderungen am Dateisystem

Nach Abschluss aller Änderungen am Dateisystem müssen die Verzeichnisse */usr, /var* und */home* nach */opt* verschoben werden, damit das Startskript */etc/init.d/S01-make-rw-filesystem* diese später zurück auf schreibbare Verzeichnisse der RAM-Disk verschieben kann.

```
mv /home/kiosk /opt/home/
mv /usr /opt/
mv /var /opt/
```

Um ein schlankes Dateisystem zu erhalten, empfiehlt es sich, den Inhalt des Verzeichnisses */var* zu entfernen. Man benötigt die Inhalte später noch, um z.B. nachträglich Pakete installieren zu können.

2.2.4 Nachträgliches Bearbeiten des *ThinClient*-Dateisystems

Um nachträglich Änderungen am aktuellen Dateisystem vornehmen zu können, muss man das komprimierte zunächst mounten, um anschließend in einer *chroot*-Umgebung an einer Kopie arbeiten zu können:

```
mkdir /mnt/pxefs
mount -o loop pxefs /mnt/pxefs
cp -r –preserve=all /mnt/pxefs /opt/
umount /mnt/pxefs
chroot /opt/pxefs
```

2.2.5 Die Netzwerkkarte des *ThinClients*

Der ThinClient funktioniert in Kombination mit den derzeitigen DHCP-, und TFTP-Servern zuverlässig in der PXE-Version 2.0 und dem Build 083. Im Betrieb sind derzeit die Karten *Intel Pro 100* und *3COM 3C905CX-TX-M*.

Intel Pro 100-Netzwerkkarten lassen sich problemlos flashen. Unter der URL *http://downloadfinder.intel.com/scripts-df/license_agr.asp?url=/6100/eng/pxe20-pdk.exe* wird der *Intel Boot Agent* zum Download angeboten. Kopiert man die Dateien aus dem Verzeichnis *bin* namens *e100_m.nic* und *futil.exe* auf eine Windows-Bootdiskette, kann man mit dem Kommando *futil e100_m.nic* das EPROM der Netzwerkkarte leicht flashen.

2.3 NIC_online

Durch das Eintragen der *ThinClients* in der *NIC_online*-Anwendungsumgebung „PXE-IVV07" werden ihnen zusätzliche DHCP-Parameter mitgegeben, die die IP des nächsten PXE-Servers mitteilen.
Zum Heranbooten an den TFTP-Server werden die 2 Parameter „Server host name" (Option 66) und „Boot filename" (Option 67) benötigt.

Zuständig für das Erstellen einer Anwendungsumgebung ist das ZIV.

2.4 TFTP-Server

Der TFTP-Server stellt über das TFTP-Protokoll dem anfragenden *ThinClient* den Bootloader *pxelinux* bereit. Der Bootloader muss zuvor via *apt* installiert und in das TFTP-Pub kopiert werden:

```
apt-get install syslinux
cp /usr/lib/syslinux/pxelinux.0 /var/lib/tftpboot/
```

Die Konfigurationsdatei des Bootloaders findet sich im Verzeichnis */var/lib/tftpboot/pxelinux.cfg*. In derDatei *default* finden sich Informationen über den Ort von Dateisystem und Kernel für den *ThinClient*. Weiterhin wird hier definiert, dass die Files für den *ThinClient* in einer RAM-Disk abgelegt werden sollen. Eine erste Bootmessage, die der Client beim Starten anzeigen soll, kann hier ebenfalls definiert werden.

2.4.1 Installierte Hard- und Software

- OS: Debian Linux
- CPU: Intel Pentium III
- RAM: 256 MB
- Debian-Pakete: tftpd-hpa (v0.39-1)

2.4.2 Konfiguration

Der TFTP-Dämon wird über den *Internet Super Server* gestartet. In der "/etc/inetd.conf" wurde folgende Zeile hinzugefügt:

```
tftp  dgram udp wait root  /usr/sbin/in.tftpd /usr/sbin/in.tftpd -s /var/lib/tftpboot
```

Nun wird der *Internet Super Server* neugestartet:

```
/etc/init.d/inetd restart
```

Die Datei */var/lib/tftpboot/X86PC/pxelinux/pxelinux.cfg/default* wurde folgendermaßen angepasst:

```
DISPLAY ivv7.msg
LABEL linux
kernel pxekernel
append initrd=pxefs root=/dev/ram0 ramdisk_blocksize=4096 ramdisk=153600
```

Eine Bootmessage kann in der Datei /*var/lib/tftpboot/pxelinux.cfg/ivv7.msg* definiert werden.

2.5 Proxy-Server

Der *Squid-HTTP-Proxy* überprüft sämtlichen Internet-Datenverkehr auf Port 80. Es handelt sich um einen transparenten Proxy, da er nicht gezielt in der Browserkonfiguration angegeben werden muss. Der URL-Filter basiert auf einer Textdatei namens whitelist.txt. Diese Whitelist akzeptiert durch Zeilenumbrüche getrennte URLs in folgenden Formaten:

```
www.uni-muenster.de
.uni-muenster.de
www.
www.uni-muenster.de/ivv7
```

Whitelist kann von Hand oder auch per Skript gepflegt werden. Die IVV7 pflegt diese Liste über ein PHP-Skript, welches alle 5 Minuten aus einer MySQL-Datenbank die URLS ausliest und bei Änderungen eine neue *whitelist.txt* generiert.

2.5.1 Installierte Hard- und Software

- OS: Debian Linux
- CPU: Intel Pentium III
- RAM: 512 MB
- Debian-Pakete: squid (v2.56)

2.5.2 Konfiguration

Nachfolgend die Konfigurationsdatei */etc/squid/squid.conf*:

```
#/etc/squid/squid.conf
#System settings
http_port 8080
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
visible_hostname ivv7proxy.uni-muenster.de
coredump_dir /var/spool/squid

#be transparent
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on

refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern .               20%     4320

#Access Control Lists
acl all src 0.0.0.0/0.0.0.0
acl localhost src 127.0.0.1/255.255.255.255
acl whitelist url_regex -i "/etc/squid/whitelist.txt"
```

Unser nachfolgendes Skript liest aus einer MySQL-Datenbank die URL-Whitelist aus.
Eine zentrale IVV7-Administrationsoberfläche pflegt auch diese Datenbank. Mit dem nachfolgenden Skript lässt sich der Inhalt einer MySQL-Datenbank in eine Datei schreiben.

```
#!/usr/bin/php4
<?php

//DATABASE
mysql_connect("MYSQL-SERVER","USERNAME","PASSWORD") or die ("Mysql-Connecect failed");
mysql_select_db("DATABASE") or die ("Database could not be opened");

//VARIABLES
$FILENAME = '/etc/squid/whitelist.txt';

$handle = fopen($FILENAME, 'w');
$query=mysql_query("SELECT allowedurl FROM whitelist");
while ($url=mysql_fetch_array($query))
{
   fwrite($handle,"$url[0]\n");
}
fclose($handle);
?>
```

3 Schlusswort

Möchten Sie ebenfalls auf *ThinClients* setzen, fragen Sie uns ruhig um Rat. Wir sind Ihnen bei der Umsetzung gerne behilflich.

Abschließend möchten wir uns besonders bei Herrn Ketteler-Eising vom ZIV bedanken, der durch seine unermüdliche Konfigurationsarbeit am universitären DHCP-Server maßgeblich zum Erfolg des *ThinClient*-Projekts beigetragen hat.