Projektbericht

Konzeption, Installation und Konfiguration eines Terminalservers

Vorname Name

Dokumentation der betrieblichen Projektarbeit im Zeitraum vom 02.04.2007 bis 27.04.2007 für [Betrieb]

1. Einleitung

1.1 Zielsetzung

Das Berufsschulzentrum für Elektrotechnik in Dresden möchte seinen Schülern die Möglichkeit anbieten, nicht mehr nur in Computerlaboren auf ihre Daten im LAN und das WAN zuzugreifen. Das hohe Bedürfnis, das Schulnetz für Recherche- und Projektarbeiten zu nutzen, steht in Konflikt mit den dauerausgelasteten und aufsichtsverpflichteten Kabinetten und soll durch die Einrichtung von mehreren Terminals befriedigt werden.
Daraus ergaben sich folgende Projektziele:

- Einrichtung einer kostengünstigen Terminalserverumgebung
- Zugriff auf eigene Daten mit schülereigenem Account ermöglichen
- Möglichkeit für Internetzugriff schaffen
- einfache Administration für Erweiterungen / Veränderungen entwickeln
- Umsetzung bis zum 27.04.2007

1.2 Projektumfang

Zur Planung und Realisierung der Anforderungen der Terminalserverumgebung mussten die möglichen Lösungen evaluiert, der Server installiert und konfiguriert sowie die Clients eingebunden und getestet werden. Die wesentlichen Aufgaben umfassten somit:

- Planung der zukünftigen Netz-, Ordner- und Rechtestruktur
- Auswahl eines Serverbetriebssystems
- Auswahl der Terminalserverapplikation
- Installation und Konfiguration des Servers
- Einrichtung der Anbindung an Fileserver und Internet
- Einbindung der Clients

Abweichungen zum Projektantrag

Nach der Antragstellung wurde in Rücksprache mit dem Projektbetreuer beschlossen, den Terminalserver redundant anzubieten, um den Dienst trotz Wartungsarbeiten weiterhin nutzen zu können. Die dafür benötigten Ressourcen waren vorhanden und die Zeitplanung wurde durch diesen zusätzlichen Schritt nur geringfügig verändert.

1.3 Projektumfeld

Das Berufsschulzentrum für Elektrotechnik (BSZET) in Dresden bietet sowohl im Bereich der Informationstechnik theoretisches Wissen für etwa 1300 Auszubildende, als auch Möglichkeiten für Fachschüler in Elektro- oder Maschinentechnik und ein technisches Gymnasium an. Es ist somit ein wichtiges Institut für Umschüler und Lehrlinge im Großraum Dresden und darüber hinaus.
Organisatorischer Ansprechpartner war Lehrer und Projektbetreuer XXXXX. Weiterhin wurden Schüler hinzugezogen, um die Nutzerfreundlichkeit des Endproduktes zu testen.
Das gesamte Projekt wurde in den Räumlichkeiten des BSZET durchgeführt.

2. Projektplanung

2.1 Ist-Analyse

Zum Zeitpunkt der Ist-Aufnahme stellt sich das Netzwerk wie folgt dar:
Ein vollständig ausgebautes und durch Vorgaben der Schulverwaltung abgesichertes LAN wird über das Novell e-Directory verwaltet, parallel dazu läuft Microsoft Windows' ADS. Der hauseigene DHCP-Server versorgt das Imaging-System Rembo mit IP-Adressen, die ansonsten statisch je Computer vergeben sind.
Nach Login mit schülerindividuellen Accounts werden Netzlaufwerke von einem Windows-Fileserver eingebunden, deren Schreib- / und Leserechte vom Administrator vorgegeben sind. Die Schüler werden dazu angehalten, jegliche Daten im Netzwerk zu speichern, da die Arbeitsstationen nach jedem Neustart mit dem Imaging-System aufgesetzt werden. Der Internetzugriff inklusive Content Filtering wird durch einen vorgeschalteten Proxy gesteuert.
Durch steten Zuwachs an Hardware befinden sich sehr viele PCs mit geringer Rechenleistung im Lager der Schule. Diverse Einzelbausteine erlauben die Erweiterung bestehender Systeme problemlos. Ausserdem sind bereits Rollwägen für die Terminalstände vorhanden.

2.2 Sollkonzept

Die Schüler sollen künftig verteilt aufgebaute Terminals nutzen können, um auch ausserhalb der Computerlabore und Stundenzeiten auf ihre eigenen Daten lesend und schreibend zugreifen zu können. Dabei ist besonders zu beachten, dass die vom Administrator vorgegebenen Rechte befolgt werden und das die Abmeldung die Netzwerkfreigaben sauber trennt. Durch die zentrale Speicherung muss nicht über Datensicherungsmaßnahmen nachgedacht werden.
Insgesamt muss der Missbrauch der nicht beaufsichtigten Computer vermieden werden, das heisst: es dürfen keine eigenen Programme installiert, keine vorhandenen gelöscht werden, keine Spiele oder ähnliche Programme gestartet werden. Weiterhin soll das Internet nur für Recherchezwecke im Rahmen der Nutzungsvorgaben einsetzbar sein.
Die Realisierung soll dabei möglichst kostengünstig und unter Zuhilfenahme bestehender Hard- und Software erfolgen; während der Umsetzung dürfen nur wenige, möglichst keine, Veränderungen am bestehenden Schulnetz durchgeführt werden.

2.3 Grundlagenwissen Terminalserver

Terminalserver dienen der Emulation mehrerer Terminals, die ihrerseits meist nur als Ein- und Ausgabeeinheit, sogenannte Thin Clients, agieren. Die Programme und Dienste werden über das Netzwerk bereitgestellt, auch ganze Betriebssysteme werden bei „diskless Clients", also Clients ohne Festplatte, rein über das Netzwerk genutzt.
Die Vorteile dieser Lösung sind die zentrale Administration und damit ein geringerer Wartungsaufwand, sowie kostengünstige Arbeitsstationen. Nachteilig ist die Zentralisierung und damit hohe Abhängigkeit vom Server, wie auch die erhöhten Initialkosten für einen leistungsstarken Terminalserver.

2.4 Zeitplanung

Prozess	geplant	benötigt
Ist-Analyse	0,5h	0,5h
Sollkonzept	0,5h	0,5h
Planung der Hardware und Netzwerkstruktur	2,0h	2,0h
Planung der Software, Dienste und Berechtigungen	4,0h	4,5h
Installation des Betriebssystems	2,5h	2,5h
Konfiguration der Dienste	2,0h	1,5h
Installation und Konfiguration des Terminalserverdienstes	4,0h	4,0h
Einrichtung der Benutzer	5,0h	5,0h
Einrichtung der Clients	0,5h	0,5h
Erstellen von Administrationsskripten	4,0h	4,0h
Testphase, Soll-Ist-Vergleich, Übergabe an Kunden	1,5h	1,5h
Handreichung für Terminalstände anfertigen	0,5h	0,5h
Erstellen der Projekt- und Kundendokumentation	8,0h	8,0h
Gesamt	**35,0h**	**35,0h**

2.5 Hardware

Als Server werden ungenutzte Maschinen mit 1400MHz (Intel Pentium III), 512MB DDR-RAM, 60GB IDE-Festplatte und 100Mbit/s-Netzwerkkarten von Intel gewählt. Bedingt durch die Architektur einer Terminalserver-Umgebung müssen besonders hohe Anforderungen an die Rechenleistung gestellt werden. Hierbei ist größerer Arbeitsspeicher gegenüber höherer CPU-Leistung zu bevorzugen, besonders bei mehreren parallel arbeitenden Nutzern. Daher wird die Leistung durch Einbau eines zweiten RAM-Bausteins auf 1024MB verdoppelt.
Für die Clients werden ausrangierte, unterschiedlich ausgestattete PCs genutzt, die durchschnittlich mit 500MHz und 128MB Arbeitsspeicher akzeptabel sind. Durch den diskless-Ansatz wird die Festplatte entbehrlich und kann anderweitig genutzt werden. Die Netzwerkkarte muss für eine ungestörte Arbeit mindestens 100Mbit/s leisten können und das PXE-Verfahren unterstützen.

2.6 Betriebssystem

Das Hauptaugenmerk bei der Wahl des Betriebssystems liegt auf dem Kostenfaktor und der Stabilität. Die Schule hat keine Lizenzen für Microsoft-Serverbetriebssysteme zur Verfügung. Der Aufbau auf Windows würde neben den Lizenzen für die einzelnen Clients und den Server zusätzliche Nutzer-Zugriffslizenzen benötigen[1]. Die Mehrkosten allein für 5 solcher CAL (Client Access License) liegen bei ca. 600€[2] und Produkte von Microsoft sind somit, auch in Hinblick auf zukünftige Erweiterungen, auszuschliessen.

---

[1] *siehe http://www.microsoft.com/germany/serverlizenzierung/produkte/windowsserver2003/terminalserver2003.mspx*

[2] *siehe http://www.heise.de/preisvergleich/?cat=swoff_ms&sort=artikel&bpmax=&asuch=2003+Terminalserver+User+CAL&filter=+Angebote+anzeigen+*

Es werden daher verschiedene, freie Linux-Distributionen unter folgenden Kriterien betrachtet: Erweiterbarkeit und Paketverwaltung, Stabilität, Ressourcenverbrauch, Aktualität. Dabei fiel die Wahl auf Debian Linux.
Die Entscheidungstabelle hierzu findet sich im Anhang Abschnitt 5.3.

Debian Linux

Debian legt besonderen Wert auf einen stabilen Betrieb, daher werden alle als „stable“ eingebundenen Pakete vor jedem Release so getestet, dass das System reibungslos funktioniert. Die Distribution ist gerade deswegen traditionell stark im Serverbereich. Der Nachteil davon ist, das die angebotenen Programme nicht mehr auf dem aktuellsten Stand sind. Dennoch überwiegen die Vorteile, wie auch beispielsweise die einfache Paketverwaltung, deutlich.

2.7 Dienste

In der Linux-Welt gibt es mehrere Projekte, die sich mit Terminalserverdiensten beschäftigen[3].
Besonders das „Linux Terminal Server Project“ (LTSP) sticht dabei hervor. Es bringt ein fertiges Dateisystem für Terminalclients und Konfigurationsautomatismen mit, arbeitet problemlos mit Debian zusammen und ist einfach anpassbar. Es ist weit verbreitet, sehr gut dokumentiert und daher auch die Wahl für dieses Projekt.
Weiterhin werden verschiedene, andere Dienste vorausgesetzt, um alle gewünschten Funktionen nutzen zu können:

- DHCP
- TFTP
- NFS
- OpenSSH
- Samba
- Heartbeat

2.8 Netzwerkstruktur

Da das bestehende Netz praktisch nur additiv geändert werden darf, fallen die Überlegungen für die zukünftige Netzwerkstruktur kurz aus. Die redundanten Terminalserver müssen nach aussen als ein Gerät auftreten und gleichzeitig individuell ansprechbar sein. Clients erhalten per DHCP eine reservierte Adresse aus einem hausüblichen Pool.
Die IP-Adressvergabe wird deshalb so geplant:

- Terminalserver1: `X.X.X.1 / 16`
- Terminalserver2: `X.X.X.2 / 16`
- gemeinsame, virtuelle Adresse: `X.X.X.3 / 16`
- DHCP-Pool: `X.X.X.X – X.X.X.X / 16`

Ein Netzplan dazu befindet sich im Anhang Abschnitt 5.2.

[3] *siehe http://www.linux-magazin.de/heft_abo/ausgaben/2004/09/einer_fuer_alle*

2.9 Verzeichnisstruktur und Berechtigungen

Zur Vermeidung von Vandalismus und ähnlicher ungewollter Nutzung werden die Nutzerrechte eingeloggter Terminalnutzer soweit wie nötig eingeschränkt. Es wird daher eine Verzeichnisstruktur angelegt, die nur die erforderlichen Berechtigungen enthält:

- `/home/userX/`
  - Home-Verzeichnis des jeweiligen Users, Besitzer userX
  - Rechte: volle Lese-/Schreib-/Ausführrechte für userX

- `/home/userX/Desktop/`
  - Verzeichnis für alle Programmverknüpfungen, Besitzer root
  - Rechte: nur Lese-/Ausführrechte für userX

Durch die explizite Übergabe des Desktop-Ordners an root wird verhindert, dass dort Dateien von einzelnen Nutzern angelegt oder gelöscht werden können.
Das Stammverzeichnis muss weiterhin dem User selber gehören, da a) die Desktop-Umgebung wichtige Dateioperationen darin durchführt und b) dort das Netzlaufwerk des Nutzers gemountet werden soll. Es wird jedoch später mit Automatismen dafür gesorgt, dass diese Ordner nicht unsinnig gefüllt werden.

3. Projektdurchführung

Hinweis: Sofern nicht anders angegeben wurden alle Konfigurationsschritte auf beiden Terminalservern identisch durchgeführt.

3.1 Installation des Betriebssystems

Nachdem der RAM durch einen zweiten 512MB-Riegel erweitert wurde, konnte mithilfe der heruntergeladenen und gebrannten DVD-Version das Debian-Grundsystem installiert werden. Die Partionierung sah hierbei wie folgt aus:

Bezeichnung	Dateisystem	Mountpoint	Größe	Typ
Auslagerung	/dev/sda1	-	1GB	swap
Systempartition	/dev/sda2	/	19GB	ext3

Die restlichen 40GB wurden für zukünftige Erweiterungen unbenutzt gelassen.
Das Programm fragt hierbei unter anderem nach dem Hostnamen und der IP-Adresse, die laut Planung eingetragen wurden.
Das Paketverwaltungstool APT kann direkt nach der Installation auf Wunsch nach Sicherheitsupdates suchen, was aber durch den schuleigenen Proxy verhindert wurde. Daher mussten vorher in der Datei `/etc/apt/apt.conf` folgende Zeilen eingetragen werden:

```
Acquire::http::Proxy "http://X.X.X.X:X";
Acquire::http::Pipeline-Depth "0";
```

Danach konnten die Updates problemlos automatisch aus dem Internet heruntergeladen und installiert werden.
Zuletzt wurde der Bootmanager GRUB in den MBR geschrieben und das System neugestartet. Dieser Prozess wurde auf beiden Servern bis auf die vergebenen Hostnamen und IP-Adressen identisch durchgeführt.

3.2 Konfiguration der Dienste

DHCP

Die automatische Konfiguration der Clients erfolgt per DHCP. Das dazugehörige Paket wurde mit dem Befehl `apt-get install dhcp3-server` installiert. Danach konnten in der Konfigurationsdatei `/etc/dhcp3/dhcpd.conf` der Pool und globale Optionen eingetragen werden, siehe dazu auch Anhang Abschnitt 5.6. Damit der hauseigene DHCP-Server nicht gestört wird, werden mit der Option `deny unknown-clients` unbekannte Clients ignoriert.

OpenSSH

Aus Sicherheitsgründen wurde root der unmittelbare Fernzugriff per SSH verboten. Dazu musste in der Datei `/etc/ssh/sshd_config` die Zeile `PermitRootLogin no` eingetragen und der Dienst mit `/etc/init.d/ssh restart` neugestartet werden.

TFTP

Als TFTP-Daemon zum Dateiaustausch wurde tftpd-hpa gewählt, da er stabil und ordnungsgemäß funktioniert. Nach der Installation mit `apt-get install tftpd-hpa` wurde folgendes Skript in den Runlevel-Ordner geschrieben, damit der Dienst beim booten startet:

```
#!/bin/sh
start-stop-daemon -S --exec /usr/sbin/in.tftpd -- -ls /tftpboot
```

Der Dienst läuft somit im Hintergrund und kann alles im Ordner `/tftpboot` übertragen.

NFS

Das Betriebssystem der Clients wird per NFS (Network File System) gemountet. Der Dienst wird per Paketverwaltung mit `apt-get install nfs-user-server nfs-common` heruntergeladen und installiert. Die Konfiguration wird später mit LTSP automatisiert.

XDM

Der X Display Manager zeigt dem Nutzer einen grafischen Anmeldebildschirm. Er wird per GUI so konfiguriert, dass das Fernzugriffsprotokoll XDMCP aktiviert ist und TCP-Verbindung von anderen X-Servern beantwortet.

Heartbeat

Heartbeat dient zum Austausch von Statusmeldungen zwischen den beiden Servern. Es wird mit `apt-get install heartbeat` installiert. Danach wird die Haupt-Konfigurationsdatei `/etc/ha.d/ha.cf` angelegt, siehe dazu Anhang Abschnitt 5.5. Besonders wichtig ist hierbei die `/etc/ha.d/haresources`, sie legt die geteilten Ressourcen fest:

```
Terminalserver1    X.X.X.3      dhcp3-server
```

Dies bedeutet, dass die virtuelle IP-Adresse `X.X.X.3` eingerichtet und der DHCP-Server vom aktiven Knoten gestartet werden soll, vorzugsweise auf `Terminalserver1`.

Samba

Das SMB-Protokoll dient zum Datenaustausch zwischen Windows- und Linux-Welt. Die Implementation Samba wird mit `apt-get install samba smbfs` installiert, danach für das lokale Netz mit Arbeitsgruppe `XXX` konfiguriert.

3.3 Installation und Konfiguration des Terminalserverdienstes LTSP

Einer der Gründe für die Wahl von LTSP war die einfache Einrichtung. Mit dem Befehl `apt-get install ltsp-utils` wurden alle benötigten Verwaltungsprogramme installiert. Das GUI-gesteuerte Tool `ltspadmin` dient zur Installation weiterer Pakete. Ausserdem kann darin der Status aller verwendeten Dienste und Konfigurationsdateien abgefragt und teilweise automatisiert geändert werden.

Das Programm erstellte die für NFS nötige Konfigurationsdatei `/etc/exports` und `/etc/hosts` sowie `/etc/hosts.allow` für die Namensauflösung, siehe dazu Anhang Abschnitt 5.7 und 5.8. Ausserdem richtete es eine Grundkonfiguration für den eigentlichen Terminaldienst selber ein. In der Datei `/opt/ltsp/i386/etc/lts.conf` wurden globale Werte für die Clients festgelegt. Hier wurden unter anderem folgende wichtige Einträge vorgenommen:

```
X_MODE_0          = 800x600
XkbLayout         = de
XkbModel          = pc105
XkbVariant        = nodeadkeys
ALLOW_SHUTDOWN    = Y
```

Durch die vorgeschriebene Auflösung von 800x600 wird das Netzwerk geschont, da weniger Daten übertragen werden müssen. Die Einstellungen für die Tastatur (Xkb*) schalten das deutsche Standardlayout mit Umlauten ein. Die letzte Zeile sorgt dafür, dass die Clients per Befehl heruntergefahren werden können und nicht nur per Netzschalter, der physisch durch eine abgeschlossene Klappe nicht erreichbar ist. Die komplette Konfigurationsdatei befindet sich im Anhang Abschnitt 5.9.

Zu diesem Zeitpunkt wurde das System neu gestartet, um zu bestätigen, dass alle Dienste wie erwartet starten.

3.4 Einrichtung der Benutzer

Ordnerstruktur

Die Benutzer sollen anonym mit den Terminals arbeiten können, bis auf das eigene Netzlaufwerk zugegriffen wird. Daher wurden Standard-Benutzer angelegt, die per Auto-Login ohne Passwort angemeldet werden. Alle Benutzer müssen hierbei die gleichen Ordner, Dateien und Rechte haben. Um dies zu vereinfachen wurde das Prinzip des `skel`-Ordners genutzt: hierbei wird ein Grundgerüst / Gerippe (skeleton) für einen Benutzerordner gespeichert und bei jedem neuen User entsprechend übernommen. Zuerst wurde daher mit dem Befehl `useradd –m user1` ein Beispielnutzer angelegt. Nachdem mit diesem Account alle Einstellungen (Desktopverknüpfungen, Oberfläche) getätigt wurden, konnte dessen Ordner mit dem Befehl `rm –rf /etc/skel && cp –a /home/user1 /etc/skel` kopiert werden.

Desktop-Lockdown

Bei der Einrichtung des Desktops wurde auf eine zielgruppengerechte Gestaltung geachtet, Hintergrundbild sowie Icons wurden dem entsprechend optisch ansprechend und zugänglich gewählt.
Zusätzlich mussten globale Einschränkungen, das sogenannte Lockdown, mit den zur Verfügung stehenden Programmen[4] eingestellt werden. Hierbei war besonders zu beachten, dass die Benutzer keine Kontrolle mehr über das Aussehen ihrer Desktop-Umgebung haben sollten. Ausserdem musste der Proxy-Server der Schule vorgeschrieben werden. Dies stellt sicher, dass jeder Internetzugriff wie gewünscht gefiltert wird. Letztlich wurde mit dem Befehl `chmod –R 000 /usr/games` der Start von Spielen blockiert.

3.5 Einbindung der Clients

Nachdem die PCs in den Rollwägen mit aller Peripherie und dem Netzwerk verbunden waren, wurde jeder einzelne gestartet und die MAC-Adresse notiert. Danach mussten diese in der DHCP-Konfigurationsdatei eingetragen um bedient zu werden. Durch den diskless-Ansatz waren keine weiteren Einstellungen nötig, da er nach der Bestätigung vom DHCP-Server alle weiteren Dateien erhält.

3.6 Erstellen von Administrationsskripten

Zur Vereinfachung der Administration wurden mehrere Shell-Skripte geschrieben, die entweder manuell oder periodisch aufgerufen werden können, unter anderem:

- `user_anlegen_loeschen.sh`: löscht alle userX und legt wieder 20 saubere Nutzer an (Anlage Abschnitt 5.10)
- `rechte_setzen.sh`: setzt alle benötigten Ordner-Rechte, wird automatisch beim Anlegen ausgeführt (Anlage Abschnitt 5.11)
- `ordner_leeren.sh`: leert den Benutzerordner, wird bei jedem Login ausgeführt (Anlage Abschnitt 5.12)
- `ordner_verbinden.sh`: mountet mit Samba das Netzlaufwerk des Nutzers nach Username/Passwort-Eingabe (Anlage Abschnitt 5.13)

[4] *siehe http://wiki.novell.com/index.php/Locking_Down_the_GNOME_Desktop*

4. Projektabschluss

4.1 Testphase

Server

Nachdem mit einem Neustart beider Server sichergestellt wurde, dass alle Dienste wie gewünscht starten, wurden verschiedene Hosts im Netzwerk erfolgreich gepingt. Danach wurde ein Knoten ausgeschaltet, um die Reaktion des anderen zu beobachten. Dieser übernahm wie erwartet nach rund 30 Sekunden die notwendigen Dienste.

Clients

Zunächst wurde ein normaler Arbeitsplatzrechner gestartet. Zwar kam seine DHCP-Anfrage beim Terminalserver an, dieser ignorierte ihn aber wie gewünscht. Danach wurde ein Terminalclient gestartet. Er wurde vom Server anhand seiner MAC-Adresse erkannt und mit einer IP-Adresse und den anderen Startoptionen versorgt. Alle beteiligten Dienste funktionierten wie erwartet, bis der Anmelde-Bildschirm erschien. Das Skript zum automatischen Login führte nach 5 Sekunden erfolgreich die interaktionslose Anmeldung durch. Der komplette Bootablauf eines Clients wird im Anhang Abschnitt 5.4 dargestellt. Im folgenden wurden die Rechte geprüft: es war weder möglich, etwas auf dem Desktop zu löschen oder anzulegen, noch irgendwelche festgeschriebenen Browser/Desktop-Einstellungen zu ändern oder Spiele zu starten. Danach wurden alle Desktopverknüpfungen in ihrer Funktion getestet. Die Anmeldung am schülereigenen Netzlaufwerk mit Lese- und Schreibrechten funktionierte ebenso tadellos wie das Abmelden und Ausschalten. Alle Tests verliefen ohne Probleme.

Funktionstest	Erfolgreich?
Server: Bootvorgang mit Diensten	Ja
Server: Netzwerkverbindung	Ja
Server: Dienste übernehmen bei Ausfall	Ja
Server: DHCP ignoriert unbekannte Clients	Ja
Client: Netzwerkkonfiguration per DHCP beziehen	Ja
Client: automatischer Login	Ja
Client: keine Rechte für nicht-authorisierte Aktionen	Ja
Client: keine Möglichkeit, Einstellungen zu ändern	Ja
Client: Zugriff auf schülereigene Daten mit Autorisierung	Ja
Client: gegebene Programme starten und nutzen	Ja

Neben den Funktionstests wurden mehrere Schüler an die Terminalstände geführt und nach einer kurzen Bediendauer nach ihrer Meinung gefragt. Dieser Usability-Test ergab überwiegend positive Resonanz .

4.2 Soll-Ist-Vergleich

Wie aus den Funktionstests und dem folgenden Soll-Ist-Vergleich ersichtlich, wurden alle Anforderungen erfüllt:

- Lese- und Schreibzugriff auf schülereigene Daten von Terminals aus möglich
- Rechte werden konsequent eingeschränkt
- Missbrauch und Vandalismus durch die Kombination Lockdown / Proxy-Server verhindert
- Lösung ist kostengünstig, keine Kosten durch zusätzliche Hard- / Software oder Lizenzen entstanden
- Schulnetz wird in keiner Hinsicht beeinträchtigt

4.3 Übergabe an Kunden

Die Einweisung in das System erfolgte praktisch an einem Terminalclient und an beiden Servern. Dabei wurden die Kundendokumentation, die Handreichung und die Debian-Installations-DVD überreicht.
Das Projekt wurde am 27.04.2007 termingerecht übergeben. Es entstanden dabei keine Änderungswünsche oder Beanstandungen.

4.4 Fazit und Ausblick

Aus dem Projekt entstand eine kostengünstige, wartungsfreundliche und leicht erweiterbare Terminalserver-Umgebung. Der Auftraggeber sowie die Schüler der Berufsschule sind mit der Lösung sehr zufrieden und nutzen sie rege.
Für die Zukunft könnte man über mehr Terminals und ein Load-Balancing zwischen den Servern nachdenken. Zusätzlich könnten gewisse Dienste im Schul-LAN in die Linux-Umgebung migriert werden, um weitere Kosten zu sparen.
Aus einem persönlichen Standpunkt betrachtet hat mir das Projekt viel gebracht. Nicht nur, dass ich jetzt die Wirkweise von per Netzwerk gebooteten PCs verstehe, auch konnte ich mich tiefergehend mit verschiedensten Diensten auf Linux beschäftigen. Ich hoffe, bin mir sogar sicher, dass sich das gesammelte Wissen in meiner beruflichen Laufbahn als nützlich erweisen wird.

5. Anhang

Hinweis: Die Konfigurations- und Skriptdateien wurden auf das wesentliche reduziert. Kommentare werden mit # eingeleitet, ein \ deutet auf einen Zeilenumbruch hin.

5.1 Glossar

ADS
: Active Directory Service: objektbasierter Verzeichnisdienst von Microsoft

DHCP
: Dynamic Host Configuration Protocol: Protokoll zum dynamischen Zuweisen diverser Konfigurationsparameter für Clients beim Bootvorgang

e-Directory
: Verzeichnisdienst (siehe ADS) von Novell

GUI
: Graphical User Interface: Oberfläche einer Software, die die Bedienung mit einem Zeigergerät wie der Maus erlaubt

Heartbeat
: Prüfsignal, um Statusmeldungen von überwachten PCs zu erhalten

IP
: Internet Protocol: Netzwerkprotokoll zum Aufbau von Verbindungen und Verschicken von Daten in einem Netzwerk

LAN
: Local Area Network: kleines Netzwerk, meist betriebs- / gebäudeintern

LTSP
: Linux Terminal Server Project: Terminalserver-Komplettlösung mit Linux

MBR
: Master Boot Record: Anfangsbereich einer Festplatte, der Informationen zum Booten und die Partitionierung enthält

NFS
: Network File System: Protokoll, das ohne Authentifizierung Daten im Netzwerk freigeben kann, bis hin zu kompletten Partitionen

Proxy
: Vermittlungsstelle zwischen zwei Netzen zur Kontrolle und Filterung oder Zwischenspeicherung von Daten

PXE
: Preboot Execution Environment: Umgebung, in der vor dem Bootvorgang betriebssystemunabhängige Anweisungen ausgeführt werden können

SMB
: Server Message Block: Protokoll ähnlich NFS, aber inklusive Authentifizierung

SSH
: Secure Shell: Protokoll zum Aufbau verschlüsselter Netzwerkverbindungen

Terminal(client)
: Computer zur Ein- und Ausgabe von Daten, die nichtlokal verarbeitet werden

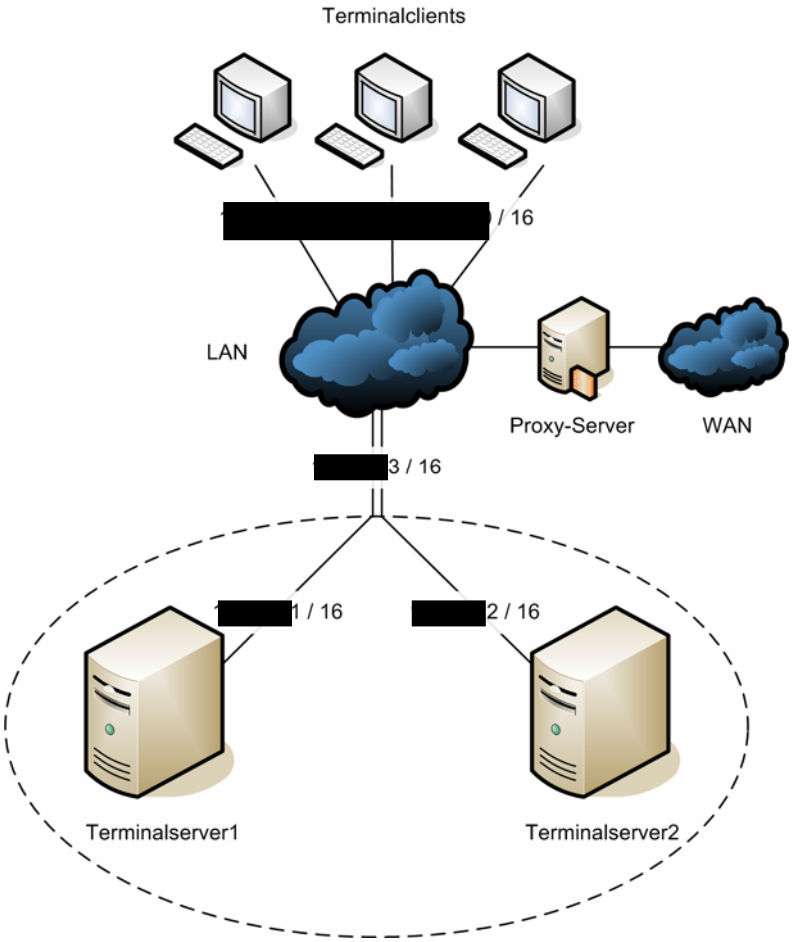
Terminalserver
: Computer zur Bereitstellung aller für Terminalclients benötigten Dienste

TFTP
: Trivial File Transfer Protocol: Dateitransferprotokoll, vereinfachtes FTP

WAN
: Wide Area Network: über mehrere Standorte erstrecktes Netzwerk

XDM
: X Display Manager: Programm zur Bereitstellung eines grafischen Anmeldebildschirms im X-Window-System

5.2 Soll-Netzplan



5.3 Entscheidungstabelle über Wahl der Linux-Distribution

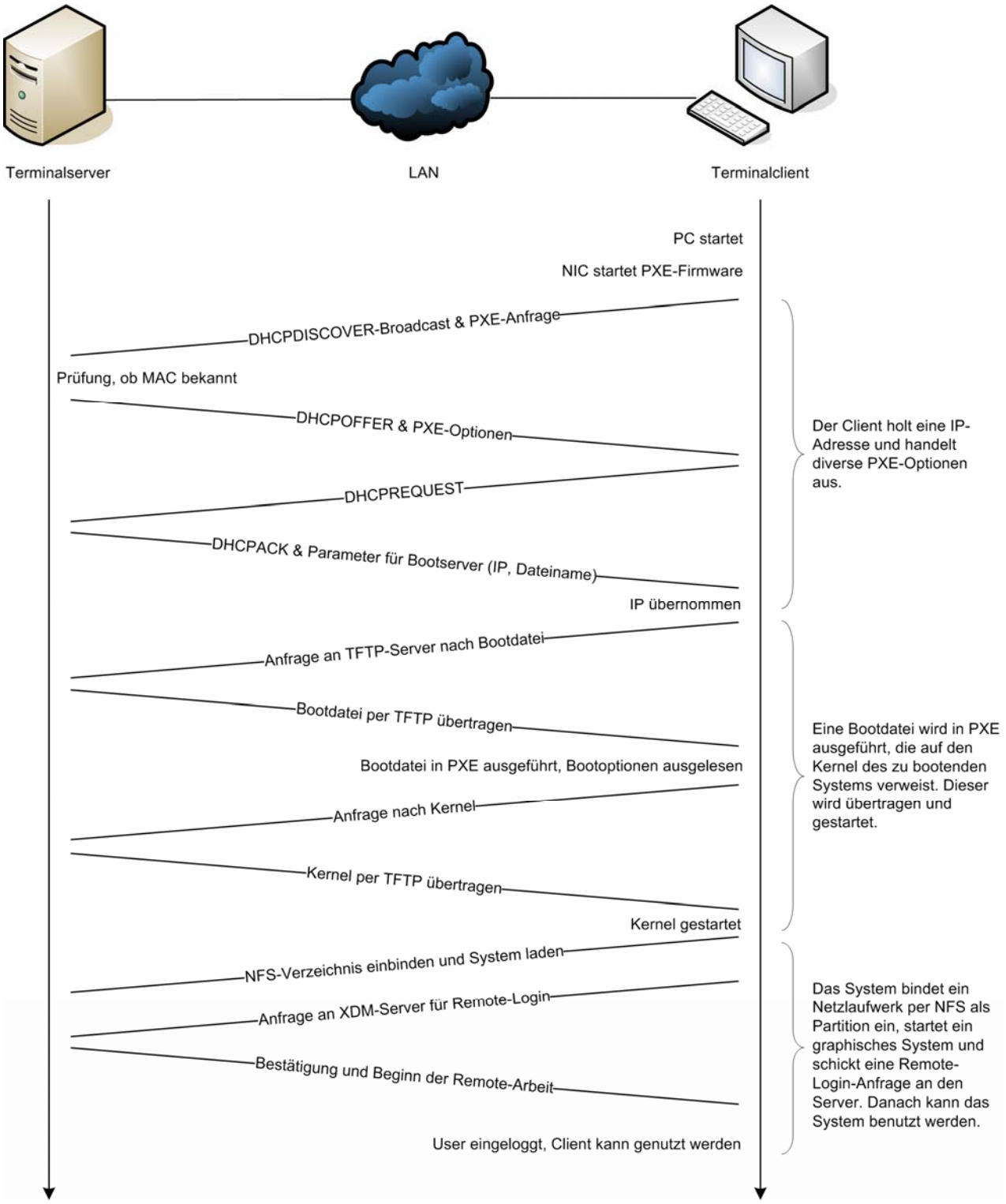
Die folgenden Bewertungen erfolgen aufgrund persönlicher Erfahrungen und einschlägiger Foren rund um das Thema Linux und seinen Distributionen.

Distribution / Kriterium	Mandrake	Fedora	Debian	openSuSE
Erweiterbarkeit / Paketverwaltung	+	+	++	+
Stabilität	0	+	++	+
Ressourcenverbrauch	-	0	0	-
Aktualität	+	+	-	++
Gesamt	**++**	**+++**	**++++**	**+++**

Legende: + = positiver Punkt, 0 = neutraler Punkt, - = negativer Punkt
mehr Punkte einer Richtung bedeuten stärkere Ausprägung dieser

Auswertung: Die Distribution Debian Linux ist bei diesen Kriterien die beste und zu wählen.

5.4 Schaubild: Bootvorgang eines Clients



5.5 ha.cf – Konfigurationsdatei von Heartbeat-Dienst

```
debugfile    /var/log/ha-debug
logfile      /var/log/ha-log
keepalive    5 # Zeit zwischen Statusprüfung
deadtime     30 # Zeit, nach der Knoten ohne Statusmeldung als tot gelten
initdead     120
udpport      694
bcast        eth0
auto_failback on # automatische Übernahme der Dienste anschalten
node         Terminalserver1
node         Terminalserver2
```

5.6. dhcpd.conf – Konfigurationsdatei von DHCP-Dienst

```
authoritative;
allow booting;
allow bootp;
# Boot-Optionen festlegen
option subnet-mask             X.X.X.X;
option broadcast-address       X.X.X.X;
option routers                 X.X.X.X;;
option domain-name-servers     X.X.X.X;;
option domain-name             "xxx";
next-server                    X.X.X.3;;
option root-path               “X.X.X.3:/opt/ltsp/i386";
filename                       "/lts/2.6.17.8-ltsp-1/pxelinux.0";

# Adresspool für bekannte Clients
subnet X.X.X.X netmask X.X.X.X {

      pool {
            range X.X.X.X X.X.X.X;
            deny unknown-clients;
      }
}
group {
    deny unknown-clients;
    # hier weitere Hosts nach diesem Schema eintragen
    host terminal1 {
            hardware ethernet X:X:X:X:X:X;
            fixed-address X.X.X.X;;
      }
}
```

5.7 exports – Zugriffskontrollliste für NFS-Dienst

```
# Betriebssystem als Read-Only anbieten
/opt/ltsp                 X.X.X.X/X.X.X.X(ro,no_root_squash,sync)
# Swap als Read-Write anbieten
/var/opt/ltsp/swapfiles   X.X.X.X/X.X.X.X(rw,no_root_squash,async)
```

5.8 hosts – feste Einträge für DNS-Namensauflösung

```
127.0.0.1    localhost.localdomain   localhost    Terminalserver1

X.X.X.1      Terminalserver1.xxx                  Terminalserver1
X.X.X.2      Terminalserver2.xxx                  Terminalserver2
```

5.9 lts.conf – Konfigurationsdatei für LTSP-Dienst

```
[Default]
      SERVER            = X.X.X.3
      XDM_SERVER        = X.X.X.3 # Terminalserver
      XSERVER           = auto
      X_MODE_0          = 800x600  # Auflösung
      X_MOUSE_PROTOCOL  = "IMPS/2" # Mausprotokoll
      X_MOUSE_DEVICE    = "/dev/psaux" # Anschlussport
      X_MOUSE_RESOLUTION = 400
      X_MOUSE_BUTTONS   = 3
      SCREEN_01         = startx
      XkbLayout         = de # Tastaturlayout
      XkbModel          = pc105
      XkbVariant        = nodeadkeys
      ALLOW_SHUTDOWN    = Y
```

5.10 user_anlegen_loeschen.sh – Skript zum Anlegen und Löschen mehrerer Benutzer

```
#!/bin/sh

# Alle Benutzer löschen
for aktuellerbenutzer in $(cat /etc/passwd | grep user | sed -n
's/:x:.*//p')
do
      echo 'Lösche Benutzer $aktuellerbenutzer'
            deluser --remove-all-files $aktuellerbenutzer
done

echo 'Lösche temporäre Dateien'
ls -al /tmp | grep users | sed -n "s/[^:]*//;s/:[0-9][0-9] //p" | xargs \
rm -rf {}

# Neue Benutzer anlegen im Stil user3 bis userX
for i in $(seq 3 23)
do
      echo -n 'Lege Benutzer user$i an... '
            useradd -m user$i
      echo 'Fertig.'

      echo '-n Lege Passwort fest... '
            echo user$i:xxx | chpasswd
      echo 'Fertig.'

      echo -n 'Setze Ordnerrechte... '
            sh /home/rechte_setzen.sh user$i
      echo 'Fertig.'
done
```

5.11 rechte_setzen.sh – Skript zum Setzen der korrekten Ordnerrechte für einen Benutzer

```
#!/bin/sh

# Übergabeparameter 1 soll der Username sein
if [ $# == 0 ]
then
      echo "Usage: rechte_setzen.sh username"
else
      aktuellerbenutzer=$1
      echo "Benutzer: $aktuellerbenutzer"

      echo "Setze root als Besitzer."
      # Root muss den Desktop-Ordner besitzen, um Schreibzugriffe zu
kontrollieren
      chown -R root:root /home/$aktuellerbenutzer/Desktop

      echo "Setze Lese-Rechte für Desktop-Inhalte."
      # Der Benutzer an sich darf nur Lese-Rechte besitzen
      chmod -R 744 /home/$aktuellerbenutzer/Desktop

      echo "Setze Sticky-Bit und Ausführ-Rechte für Desktop-Ordner."
      # Das Sticky-Bit erlaubt es, eigens angelegte Dateien zu löschen,
aber nur diese, wichtig für Systemprozesse
      chmod 1755 /home/$aktuellerbenutzer/Desktop
fi
```

5.12 ordner_leeren.sh – Skript zum Entfernen aller ungewünschten Daten

```
#!/bin/sh

# Übergabeparameter 1 soll der Username sein
if [ $# == 0 ]
then
      echo "Usage: ordner_leeren.sh username"
else
      aktuellerbenutzer=$1
      echo $aktuellerbenutzer

      echo "User wird getrennt."
      # Prozess von User ausloggen
      /usr/bin/skill -KILL -u $aktuellerbenutzer
      sleep 2

      # Prüfen ob Laufwerk verbunden / User angemeldet
      verbunden=$(cat /proc/mounts | grep $aktuellerbenutzer | cut -f \
1,1 -d " ")
      if [ "$verbunden" == "//xxx/xxx" ]
      then

            # Verzeichnis-überwachende Prozesse beenden, unmounten
            echo "Alle Prozesse beenden."
            fuser -k /home/$aktuellerbenutzer/schule > /dev/null 2> \
/dev/null

            echo "Schulverzeichnis unmounten."
            smbumount /home/$aktuellerbenutzer/schule > /dev/null 2> \
/dev/null
      fi

      # Alle Benutzerdaten auflisten und alles rausfiltern, was nicht
gelöscht werden soll
      echo "Benutzerverzeichnis leeren."
      find /home/$aktuellerbenutzer/ -regex '[^.]*' | grep -v Desktop \
| grep -v "^/home/$aktuellerbenutzer/$" | sed -e "s/ /' '/" | xargs rm \
- rf {}

fi
```

5.13 ordner_verbinden.sh – Skript zum Einbinden des Schüler-Netzlaufwerks

```
#!/bin/sh

# Verzeichnis anlegen
mkdir ~/schule > /dev/null 2> /dev/null

# Prüfen ob Laufwerk verbunden / User angemeldet
verbunden=$(cat /proc/mounts | grep $USERNAME | cut -f 1,1 -d " ")
if [ "$verbunden" == "//xxx/xxx" ]
then
      # Verzeichnis bereits verbunden, also nur Filebrowser öffnen
      nautilus --browser ~/schule
else
      # User ist nicht angemeldet, also authentifizieren
      echo -n "Bitte geben Sie Ihren Benutzernamen ein (Stil: \
mustermannma): "
      read benutzername
      echo "Achtung: Passworteingabe erfolgt unsichtbar!"

      # Verzeichnis mit Samba-Tool mounten
      smbmount //xxx/xxx ~/schule -o username=$benutzername > \
/dev/null 2> /dev/null
      errorcode=$?

      if [ "$errorcode" != 0 ]
      then
            # Verzeichnis konnte nicht fehlerfrei geöffnet werden
            echo "Fehler! Bitte prüfen Sie Benutzername und Passwort"
            sleep 2
      else
            clear
            echo "Sie können nun auf Ihren eigenen Ordner zugreifen \
und haben volle Rechte. Es wird nun ein Dateibrowser geöffnet."
            echo "ACHTUNG: auch nachfolgende Benutzer haben somit \
Zugriff, solange sie sich nicht vom System abmelden!"
            sleep 8
            # Verzeichnis erfolgreich verbunden
            nautilus --browser ~/schule
      fi
fi
```

5.14 Kundendokumentation

Dokumentation der

Terminalserverumgebung des BSZET

Übersicht

Die Terminalserverumgebung des BSZET besteht aus zwei wesentlichen Komponenten: Zwei Terminalserver und die eigentlichen Clients.
Die Server sind hierbei identisch eingerichtet und treten nach aussen als eine Einheit auf, sodass einer der beiden Knoten ausfallen oder zur Wartung ausgeschaltet werden kann, ohne das gesamte System zu beeinträchtigen.
Die Clients werden mit einer Kombination verschiedener Dienste von den Servern mit einem kompletten Betriebssystem versorgt und benötigen beim Start keinerlei Nutzerinteraktion.

Im Folgenden werden die häufigsten Anwendungsszenarien und Fragen erläutert.

Wie stelle ich eine Verbindung zu den Servern her?

Es gibt prinzipiell zwei Möglichkeiten zur Administrierung der Server.
Zur grafischen Bedienung muss die entsprechende Peripherie angeschlossen werden. Danach kann in einer Konsole mit dem Befehl `skill –KILL –u $USERNAME` die aktuelle Sitzung beendet und per Anmeldebildschirm eine Administrator-Sitzung gestartet werden.
Zur Administration per Kommandozeile muss eine gesicherte Verbindung zum Server aufgebaut werden. Das freie Programm „PuTTy“ kann genau das. Bei der Abfrage nach Benutzername und Passwort muss der Benutzer `sshuser` genutzt werden. Danach kann mit dem Befehl `su –` zum Administrator gewechselt werden.

Welche IP-Adressen und Hostnamen haben meine Server?

Bezeichnung	IP-Adresse	Subnetzmaske
Terminalserver1	X.X.X.1	X.X.X.X
Terminalserver2	X.X.X.2	X.X.X.X
Virtuelle, gemeinsame Adresse	X.X.X.3	X.X.X.X
Pool für Terminalclients	X.X.X.X – X.X.X.X	X.X.X.X

Wie lauten die Passwörter für mein System?

Host	Benutzer	Passwort
Terminalserver1	root	xxx
Terminalserver2	root	xxx
Terminalserver1	sshuser	xxx
Terminalserver2	sshuser	xxx
Beide Terminalserver	user3 – user23	xxx

Ein Passwort kann, sobald als dieser Nutzer eingeloggt, mit dem Befehl `passwd` geändert werden.
Achtung: Diese Passwörter sind systemkritisch. Sie sollten regelmässig geändert und unter Verschluss gehalten werden.

Welche Funktion haben die einzelnen Skripte in /home/ und /root/?

- `autologin.sh`: Dieses Skript wird im Anmeldebildschirm ausgeführt. Es gibt einen derzeit nicht eingeloggten Benutzernamen aus, so dass sich das System anmelden kann.
- `herunterfahren.sh`: Es wird beim Klick auf das Icon „Computer herunterfahren“ gestartet. Es startet `herunterfahren_remote.sh` mit dem derzeitigen Benutzernamen
- `herunterfahren_remote.sh`: Es nimmt einen Benutzernamen entgegen und schaltet den Computer dieses Benutzers aus. Es kann nicht direkt von Benutzern ohne den Umweg über `herunterfahren.sh` gestartet werden.
- `inaktive_clients_herunterfahren.sh`: Es prüft die Aktivität derzeit verbundener Benutzer. Stellt es fest, dass der angesprochene Client nicht mehr erreichbar (weil ausgeschaltet) ist, beendet er die Sitzung sauber.
- `logout.sh`: Dieses Skript trennt alle Verbindungen zum Netzlaufwerk.
- `ordner_leeren.sh`: Es leert den Ordner eines Benutzers, so dass keine lokal angelegten Dateien gelöscht sind.
- `ordner_verbinden.sh`: Es stellt die Verbindung zum Netzlaufwerk auf `IS2` her und bindet es in `/home/userX/schule/` ein.
- `rechte_setzen.sh`: Es stellt alle benötigten Rechte für einen Benutzer ein.
- `user_anlegen_loeschen.sh`: Dieses Skript löscht alle vorhandenen normalen Benutzer und legt sie von `user3` bis `user23` neu an. Es vergibt ausserdem die exakten Rechte per `rechte_setzen.sh`. In diesem Skript kann das Standardpasswort und die Anzahl der neuen Benutzer festgelegt werden.
- `user_bereinigen.sh`: Mit diesem Skript werden alle Benutzer in ihren Ursprungszustand zurückgesetzt (Ordner leeren, Rechte setzen, temporäre Dateien löschen).

Welcher Dienst hat welche Funktion?

Dienst	Funktion
DHCP	Vergibt dynamisch IP-Adressen
SSH	Ermöglicht verschlüsselte Fernwartung
TFTP	Überträgt benötigte Daten beim Bootvorgang
NFS	Stellt benötigte Dateien bereit
Samba	Ermöglicht den Datenaustausch mit Windows-Netzlaufwerken
XDM/GDM	Stellt einen Anmeldebildschirm bereit
Heartbeat	Verwaltet die Clusterknoten und startet bestimmte Dienste

Wie starte ich die einzelnen Dienste neu?

Dienst	Befehl
DHCP	Siehe Heartbeat-Neustart
SSH	`/etc/init.d/ssh restart`
TFTP	`killall in.tftpd && /usr/sbin/in.tftpd -ls /tftpboot`
NFS	`/etc/init.d/nfs-common restart`
Samba	`/etc/init.d/samba restart`
XDM/GDM	`/etc/init.d/gdm restart`
Heartbeat	`/etc/init.d/heartbeat restart`

Welche wichtigen Logdateien gibt es und wie kann ich sie auswerten?

Zum Anzeigen einer Logdatei kann der Befehl `cat Dateiname | more` benutzt werden. Um die Ausgabe einer Logdatei zu verfolgen, kann der Befehl `tail –f Dateiname` benutzt werden.

Die wichtigsten Logdateien sind folgende:

- `/var/log/syslog`: Log für das gesamte System
- `/var/log/ha-log`: Log-Informationen für den Heartbeat-Dienst
- `/var/log/ha-debug`: detailierte Debug-Informationen für Heartbeat
- `/var/log/auth.log`: Historie über eingeloggte Benutzer
- `/var/log/samba/log.smbmount`: Informationen über die verbundenen Netzlaufwerke

Wie füge ich neue Clients in das System ein?

Nachdem der Client hardwareseitig zum Netzwerk verbunden wurde, muss seine MAC-Adresse ausgelesen werden. Diese steht entweder auf der Netzwerkkarte oder beim Bootvorgang. Zusätzlich muss sichergestellt werden, dass im BIOS des Clients das erste Bootmedium das Netzwerk ist.

Danach muss dieser Client dem DHCP-Dienst der Server bekannt gemacht werden. Dazu muss die Datei `/etc/dhcp3/dhcpd.conf` im Bereich `group` editiert werden:

```
group {
    deny unknown-clients;
    # hier weitere Hosts nach diesem Schema eintragen
    host terminal1 {
            hardware ethernet X:X:X:X:X:X;
            fixed-address X.X.X.X;
      }
}
```

Der neue Client muss nach diesem Schema hinzugefügt werden:

```
    host terminalX { #Name des Clients
            hardware ethernet XX:XX:XX:XX:XX:XX; #MAC-Adresse
            fixed-address X.X.X.XXX; #feste IP-Adresse
      }
```

Die IP-Adresse sollte hierbei aus dem Bereich `X.X.X.X – X.X.X.X` stammen. Diese Änderung muss auf allen Terminalservern durchgeführt werden.

Nach dem Speichern der Datei muss zuletzt der DHCP-Dienst neugestartet werden.

Wie füge ich neue Benutzer in das System ein?

Ein neuer Benutzer kann mit folgender Befehlsfolge erstellt werden:

```
useradd -m username
echo username:xxx | chpasswd
sh /home/rechte_setzen.sh username
```

Dabei ist zu beachten, dass das Schema `userX` eingehalten wird, wobei X für eine Zahl ab 24 steht. Es muss ausserdem sichergestellt werden, dass X sich nahtlos an die bestehenden User anpasst. Sind beispielsweise die Benutzer `user3` bis `user30` im System erstellt, muss der nächste Benutzer `user31` heissen.

Troubleshooting

Problem	Ursache	Lösung
Client erhält keine Antwort vom DHCP-Server	Netzwerkverbindung unterbrochen	Kabel prüfen
	DHCP-Dienst nicht aktiv	DHCP-Dienst neustarten
	Client nicht in DHCP-Liste eingetragen	dhcpd.conf auf richtige Einträge (korrekte MAC) prüfen
Client erhält keine Dateien	TFTP-Dienst nicht aktiv	TFTP-Dienst neustarten
Grauer Bildschirm mit großem X	Client hat keinen Kontakt mit dem X-Server	Prüfen, ob `X.X.X.3` erreichbar ist. Falls nicht, Heartbeat-Dienst neustarten
Netzlaufwerke funktionieren nicht	Keine Verbindung zu IS2-Fileserver	IS2 prüfen, Netzwerk-verbindungen prüfen
	Username:Passwort falsch	Benutzeraccount prüfen
Kein Internetzugang	Keine Verbindung zum Proxy	Erreichbarkeit von `X.X.X.X` prüfen

5.15 Handreichung für Terminalstände

Diese Kurzanleitung wird auf jedem Terminalstand angebracht und dient als Einweisung in das System.

Bedienungsanleitung Terminals

Starten des Rechners

Den Computer bitte ganz normal starten. Er meldet sich dann automatisch am System an, hier bitte **nichts** eingeben.

Netzlaufwerk verbinden

- stellt eine Verbindung zum eigenen Schülerordner her

Hier können wie gewohnt die eigenen Dateien gespeichert und editiert werden.

Persönlichen Ordner öffnen

- öffnet den persönlichen, **temporären** Ordner

Bitte **nichts** lokal auf dem Terminalclient speichern, sondern in das Netzlaufwerk. Lokal gespeicherte Daten sind nach einem Neustart unwiederbringlich gelöscht!

Abmelden

- trennt die Verbindung zum persönlichen Ordner

Aus Datenschutzgründen sollte dieser Button grundsätzlich beim Verlassen des Rechners betätigt werden, andernfalls hat jeder Zugriff auf deinen Schülerordner!

Herunterfahren

- trennt alle Verbindungen und fährt den Client runter

5.16 Verwendete Quellen

Neben den in den Fußnoten angegebenen Quellen wurden zusätzlich folgende benutzt:

- *http://www.ltsp.org/*
- *http://www.bszet.de/schulgeschichte.php?ansicht=1*
- *http://de.wikipedia.org/wiki/Terminalserver*
- *http://de.wikipedia.org/wiki/Sticky_bit*
- *http://www.aims.ac.za/pipermail/aims-tech/2006-September/000759.html*
- Man-Pages der einzelnen Linux-Dienste

Alle Internet-Quellen beziehen sich auf den Stand vom 27.04.2007.