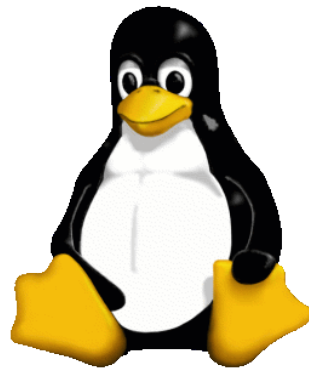


**Implementierung eines
Imaging-Systems
im PCLab5 der NTA-Isny**



Projektarbeit Internet-Technologie

NTA FH Isny
12. Info

David Mayr

Inhaltsverzeichnis

I	Einleitung.....	4
1	Was ist ein Image?.....	4
2	Was ist ein Imaging-System?.....	4
3	Images effizient mit Partimage erstellen.....	5
4	Ziel : PC-Labor 5 der NTA.....	5
5	Ausgangsbasis : Boot-Server / Diskless-Clients.....	5
II	Implementierung.....	5
1	Kernel: konfigurieren, compilieren und zum Boot vorbereiten.....	6
2	LTSP-Konfiguration.....	6
3	Das imagesys-Script.....	6
III	Fazit.....	8
IV	Quellenangaben.....	8
V	ANHANG.....	9
1	Die Datei imagesys (Shell-Script).....	9
2	Die Datei imagesys.config.....	11
3	Die Datei imagesys.functions.....	11

I Einleitung

1 Was ist ein Image?

Mit dem Begriff Image ist in diesem Zusammenhang ein Dateisystem-Image gemeint. Ein Dateisystem-Image ist ein direktes 1:1 Abbild eines Dateisystems – z.B. von einer Festplattenpartition, einer CDROM/DVD oder auch einer Diskette – das in einer Image-Datei abgespeichert ist.

Die einfachste Art unter Linux / UNIX ein Dateisystem-Image Byte für Byte anzufertigen, ist mit dem Befehl **cat** den Inhalt einer Quell-Geräte-datei durch das Umleitungszeichen **>** in eine Datei zu kopieren, z.B.:

```
cat /dev/hda1 > /tmp/partition-hda1.image
```

ein Image von der ersten Partition der Master-Festplatte am primären IDE-Controller erstellen

In einer etwas eleganteren Variante könnte man das Kommando **dd** verwenden, dem man bei Bedarf auch die gewünschte Blockgröße und Start-/End-Position des blockweisen Kopiervorgangs mitteilen könnte, z.B.:

```
dd if=/dev/fd0 of=/tmp/diskette.image bs=4k
```

ein Image von Diskette mit dd erstellen, dabei je 4KB-Blöcke lesen/schreiben

Bei der Erstellung eines Images ist generell zu beachten, dass das jeweilige Dateisystem entweder nicht oder nur im read-only Modus in Verwendung ist, da sonst Inkonsistenzen beim Image auftreten können.

Ein auf diese Art erstelltes Dateisystem-Image kann als Benutzer "root" einfach mit dem **mount**-Kommando und der Option "-o loop" in ein beliebiges vorhandenes Verzeichnis gemountet werden:

```
mount -o loop /tmp/diskette.image /mnt/
```

Disketten-Image in das Verzeichnis /mnt/ mithilfe der loop-Funktion mounten

Natürlich ist es so auch möglich, eine Sicherung von einem Dateisystem zu erstellen – vorausgesetzt, das Image wird an entsprechend geschützter Stelle¹ abgelegt. Für das Rücksichern können auch wieder die eben erwähnten Befehle **cat** oder **dd** verwendet werden – natürlich mit vertauschten Quell- und Ziel-Angaben.

Ein gewichtiger Nachteil dieser Images ist, dass sie immer genau dieselbe Größe wie die Quelle haben – egal wieviel von z.B. der Festplattenpartition belegt ist.

2 Was ist ein Imaging-System?

Mit dem Begriff Imaging-System ist hier ein System gemeint, mit dem es auf einfache Art und Weise möglich ist, die Dateisysteme in den Partitionen einer Festplatte in Images zu sichern und auf einem oder mehreren Computern wieder einzuspielen.

1 also nicht einfach in einer anderen Partition der selben Festplatte

3 Images effizient mit Partimage erstellen

Um besonders effizient Backups von Dateisystemen anzulegen empfiehlt sich das Programm **partimage**. Mithilfe dieses Programms werden innerhalb der gängigsten Dateisysteme² automatisch nur die belegten Blöcke aufgespürt und diese komprimiert in einem speziellen Image gespeichert. So braucht z.B. eine frische Windows2000-Installation mit MS-Office auf einer 40GB-Festplattenpartition als partimage-Datei nicht die vollen 40GB der Partition, sondern nur rund 0,7GB.

4 Ziel : PC-Labor 5 der NTA

Im PC-Labor5 der NTA-Isny gibt es 22 PC-Systeme mit Wechselfestplatten, auf denen sich eine Installation von Windows 2000 und eine Installation von dem schon recht betagten SuSE Linux 8.2 befinden.

Für einige Aufgaben an den Computern wäre es wünschenswert, eine aktuellere Linux-Version und evtl. auch mehrere Instanzen für unterschiedliche Praktikas zu installieren.

Das Produkt dieser Projektarbeit soll dazu dienen, den Aufwand für eine Neuinstallation aller 22 Computer möglichst gering zu halten. Damit soll es möglich sein, innerhalb recht kurzer Zeit auf allem Computer den gleichen Ausgangszustand – bezogen von einem zuvor komplett wunschgemäß eingerichteten Referenzsystem – einzurichten.

5 Ausgangsbasis : Boot-Server / Diskless-Clients

Diese Arbeit basiert auf meinem Projekt des vorigen Semesters, in dem unter anderem ein Boot-Server für Festplattenlose Clients implementiert wurde. Festplattenlose Clients holen sowohl den Betriebssystem-Kernel als auch ihr root-Dateisystem von einem Server über das LAN. Dadurch starten diese Clients sehr schnell³ und man hat vollen Zugriff auf die evtl. trotzdem angeschlossene Festplatte, da sie ja für dieses System nicht in Gebrauch ist.

Der Boot-Server arbeitet mit einer angepassten Version 4.1.1 des LTSP⁴-Systems. Kurz gesagt sorgt dieses System dafür, dass *allen* Diskless-Clients *ein einziges* ca. 200 MB kleines root-Dateisystem im read-only Modus genügt. Die Clients erstellen beim Boot-Vorgang eine kleine RAM-Disk und speichern dort die ebenfalls während des Bootens automatisch erzeugte Hardware-Konfiguration ab. In einer einzigen zentralen Konfigurationsdatei auf dem Boot-Server lassen sich generelle, aber auch Rechner-spezifische Einstellungen vornehmen und die automatische Hardware-erkennung beeinflussen.

So lässt sich mit einem Eintrag in dieser zentralen Konfigurationsdatei z.B. ein zusätzliches Skript beim Bootvorgang der Clients aktivieren, das z.B. das Einspielen eines Images nach einer Rückfrage vornimmt.

2 u.a. ext2, ext3, reiserfs, fat, ntfs, hpfs, ufs, jfs, xfs, hfs, ...

3 Unter 15 Sekunden (!) ab dem Start des Netzwerk-Bootloaders bis zur graphischen Anmeldemaske (Athlon1300+)

4 LTSP steht für das **L**inux **T**erminal-**S**erver **P**roject, dessen Homepage unter www.ltsp.org zu finden ist.

II Implementierung

1 Kernel: konfigurieren, compilieren und zum Boot vorbereiten

Zunächst musste ein neuer Kernel für die Clients konfiguriert und compiliert werden, da bisher keine IDE-Unterstützung aktiviert war. Bei dieser Gelegenheit wurde der Kernel auf die Version 2.6.11 mit OpenMosix aktualisiert.

2 LTSP-Konfiguration

Bei der installierten LTSP-Software muss nur in der Datei `/opt/ltsp/i386/etc/lts.conf` die Zeile `SCREEN_02 = imagesys` eingetragen werden und unter `/opt/ltsp/i386/etc/screen.d/` das Script mit dem Namen **imagesys** angelegt werden, das im folgenden näher beschrieben wird. Dadurch wird beim Start eines Clients automatisch das Script **imagesys** gestartet, das nach einer Passwort-Abfrage die Aufgaben des Imaging-Systems erledigt.

Zusätzlich müssen noch **fdisk**, **dialog** und **partimage** statisch compiliert und dem LTSP root-Dateisystem unter `/opt/ltsp/i386/` hinzugefügt werden.

3 Das imagesys-Script

Das **imagesys**-Script wird am Client nach entsprechender LTSP-Konfiguration automatisch gestartet. Es fragt zuerst ein Passwort ab, das vor unerlaubten Zugriffen schützen soll.



Anschliessend wird per NFS das Image-Verzeichniss vom Server gemountet und alle verfügbaren IDE-Module geladen.

Dabei werden die beim Erkennen der IDE-Geräte erzeugten Kernelmeldungen direkt auf dem Bildschirm ausgegeben, wie in folgendem Beispiel zu sehen ist.

```
[***] Stelle Verbindung zum Image-Server her: Portmap, NFS-Mount
[***] IDE-Module laden ...

Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
PIIX4: IDE controller at PCI slot 0000:00:07.1
PIIX4: chipset revision 1
PIIX4: not 100% native mode: will probe irqs later
   ide0: BM-DMA at 0x1070-0x1077, BIOS settings: hda:DMA, hdb:pio
hda: VMware Virtual IDE Hard Drive, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hda: max request size: 128KiB
hda: 2097152 sectors (1073 MB) w/32KiB Cache, CHS=2080/16/63, UDMA(33)
   /dev/ide/host0/bus0/target0/lun0: p1 p2

[***] Aktuelle Partitionstabelle:

Disk /dev/hda: 1073 MB, 1073741824 bytes
16 heads, 63 sectors/track, 2080 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1            1          1040        524128+   83  Linux
/dev/hda2          1041          2080        524160    83  Linux

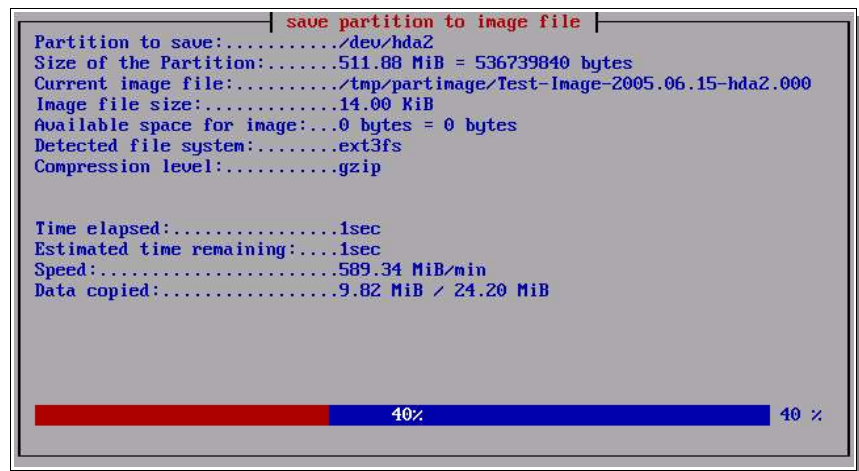
[***] Um jetzt zu starten, Enter druecken...
```

Anschliessend erscheint das Hauptmenü mit den folgenden Auswahlmöglichkeiten:

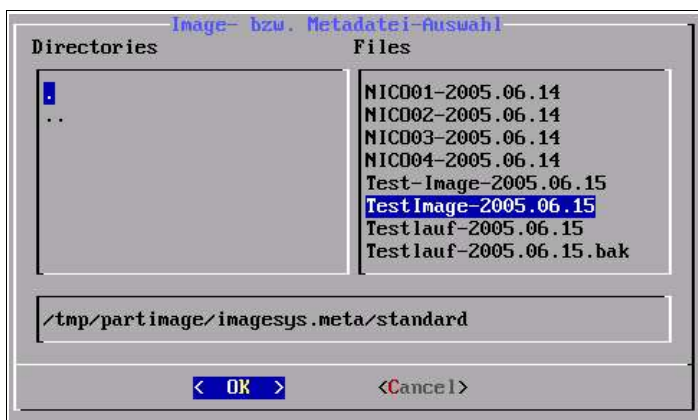


Der Menüpunkt "**Sicherung**" listet alle Partitionen der erkannten Festplatten auf und fragt nach einem Namen der Sicherung und den einzubeziehenden Partitionen.

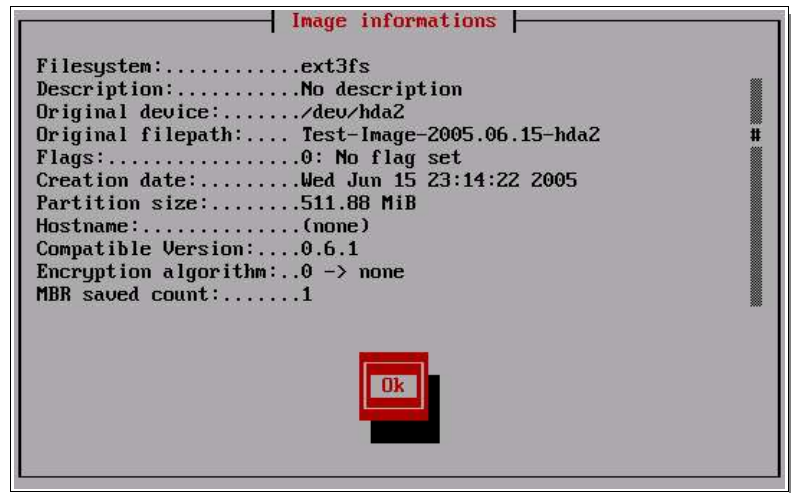
Sind die Angaben korrekt startet für jede ausgewählte Partition nacheinander ein Sicherungslauf von partimage.



Der Menüpunkt "**Ruecksicherung**" bietet einen Dialog zur Auswahl der vorhandenen Images an und startet dann die Ruecksicherung der ausgewählten Partitionen mit partimage.



Der Menüpunkt "**Image-Info**" liefert nach Auswahl einer Image-Datei Informationen über das ausgewählte Image.



III Fazit

Mit der Kombination aus LTSP-Bootserver, Partimage und einem relativ einfachen Shellsript lassen sich in kurzer Zeit ohne großen Aufwand viele PCs auf den selben Ausgangszustand bringen. Auch können so diejenigen PCs, die im Laufe der Zeit durch die Benutzung von vielen Studenten evtl. durch Fehlkonfiguration nicht mehr richtig funktionieren ohne große Fehlersuche gleich wieder vom Image zurückgesichert werden.

IV Quellenangaben

- Die Homepage des **Linux Terminal-Server Project**: <http://www.ltsp.org>
- Die Homepage von **PartImage**: <http://www.partimage.org>
- Die **dialog**-manpage online: <http://www.rt.com/man/dialog.1.html>

V ANHANG

1 Die Datei imagesys (Shell-Script)

```

1  #!/bin/bash
2
3  ### Dateien einbinden
4  . $(dirname $0)/imagesys.config
5  . $(dirname $0)/imagesys.functions
6
7  ### Bei STRG-C vor beenden noch aufräumen
8  trap '[ -e $TMP ] && rm $TMP' 2
9
10 ### Passwort abfragen und vergleichen
11 if [ "$(dpasswd "Code-Abfrage" "Bitte Sicherheitscode eingeben: ")" = "$PASS" ]
12 then :
13 else sleep 1 ; quit "Falscher Code. Beendet."
14 fi
15
16 ### Netzwerk
17 display "Stelle Verbindung zum Image-Server her: "
18 echo -n "Portmap, "
19 ps -ea f | grep -v grep | grep -q portmap || portmap
20 echo -e "NFS-Mount\n"
21 mkdir -p /tmp/partimage.
22 mount 2>&1 | grep -v "version older" | grep -q "on /tmp/partimage" || mount
23 server:/scratch/PARTIMAGE-BACKUP/ /tmp/partimage 2>&1 | grep -v "version older"
24 cd /tmp/partimage
25 [ -d $METADIR ] || mkdir -p $METADIR
26
27 ### IDE-Module laden
28 display "IDE-Module laden ... \n\n"
29 lsmod | grep -q ^"ide-disk" || modprobe ide-disk
30 for moduldatei in /lib/modules/$(uname -r)/kernel/drivers/ide/pci/*.ko
31 do modulname="$(basename $moduldatei .ko)"
32     lsmod | grep -q ^"$modulname" || modprobe $modulname
33 done
34 echo
35
36 ### Partitionen anzeigen
37 fdisklist="$(fdisk -l)"
38 display "Aktuelle Partitionstabelle: \n$fdisklist\n\n"
39 [ -z "$fdisklist" ] && displayerr "Keine Partitionen gefunden.\n\n"
40 display "Um jetzt zu starten, Enter druecken..."
41 read warte
42
43 ### Hauptmenu
44 case $(hauptmenu) in
45 s*)
46     echo -e "\n\n\n\n"

```



```

45 S*).
46     echo -e "\n\n\n\n"
47     display "Bitte Basisname fuer das Image eingeben: " ; read input
48     display "Partitionen:\n`fdisk -l`\n\n"
49     display "Welche Partitionen sichern? (z.B. hda1 hda2 hda4): " ; read partitionen
50     BASISNAME="$input-$(date +%Y.%m.%d)"
51     [ -e $METADIR/$BASISNAME ] && cp $METADIR/$BASISNAME $METADIR/$BASISNAME.bak
52     echo -n "" >$METADIR/$BASISNAME
53     for part in $partitionen
54     do
55         if [ -e /dev/$part ]
56         then IMGNAME="$BASISNAME-$part"
57             display "Sichere Partition /dev/$part nach /tmp/partimage/$IMGNAME... \n"
58             ; sleep 1
59             if partimage -z1 -d -f3 -b save /dev/$part /tmp/partimage/$IMGNAME
60             then echo "$IMGNAME.000:$part" >> $METADIR/$BASISNAME
61             else displayerr "FEHLER bei PARTIMAGE, Partition /dev/$part >
62             /tmp/partimage/$IMGNAME ..."
63             fi
64         else displayerr "Geraetedatei /dev/$part nicht vorhanden." ; sleep 3
65         fi
66     done
67     ;;
68     ###
69 R*)
70     while [ ! -e "$datei" ]
71     do
72         dialog --backtitle "$BACKTITLE" --title "Image- bzw. Metadatei-Auswahl"
73         --fselect $METADIR/standard 8 60 2>$TMP
74         datei="$(cat $TMP)"
75         [ -r "$datei" ] || displayerr "\n\n"$datei"\ ist keine Datei.\n\n"
76     done
77     echo ; display "Zuerst evtl. MBR einspielen ... \n\n" ; sleep 2
78     partimage restmbr "$(head -nl $datei | cut -d: -f1)"
79     dyesno "Images einspielen?" "Sind Sie sicher, dass folgende Images eingespielt
80     werden sollen?\n\n$(cat $datei | sed 's:/ / nach \\/dev\\/g')"
```

```

89 .      datei="$(cat $TMP)"
90 .      [ -r "$datei" ] || displayerr "\n\n\"$datei\" ist keine Datei.\n\n"
91 .      done
92 .      for part in $(cat $datei)
93 .      do
94 .          display "INFO zu $(echo $part | cut -d: -f1) ... \n" ; sleep 2
95 .          partimage imginfo $(echo $part | cut -d: -f1) || sleep 3
96 .      done
97 .      ;;
98 *)
99 .      quit "Abbruch.  Eingabetaste zum neustart..."
100 .      ;;
101 .      esac
102 .
103 .      displayerr "ENDE"
104 .      [ -e $TMP ] && rm $TMP
105 .      sleep 1
  
```

2 Die Datei imagesys.config

```

# Pseudo-Passwort ("Sicherheitscode")
PASS=" "

# Dialog: Haup-Titel
BACKTITLE="IMAGING-System  NTA FH Isny  :::  D.Mayr@LunaBOX.de  :::  [$(hostname)]"

# Temp/Datei
TMP=/tmp/$(basename $0).tmp

# Verz. mit Metainfos
METADIR=/tmp/partimage/$(basename $0).meta
  
```

3 Die Datei *imagesys.functions*

```
#####
# Funktionen:
display() {
|   echo -en "${gruen}[***] ${normal}${1}"
}
displayerr() {
    echo -en "${rot}[!!!] ${gelb}${1}${normal}"
}
dtextbox() {
    dialog --backtitle "$BACKTITLE" --title "$1" --textbox "$2" 0 0
}
dmsgbox() {
    dialog --backtitle "$BACKTITLE" --title "$1" --msgbox "$2" 0 0
}
dmsgbox3s() {
    dialog --timeout 3 --backtitle "$BACKTITLE" --title "$1" --msgbox "$2" 0 0
}
dmsgbox5s() {
    dialog --timeout 5 --backtitle "$BACKTITLE" --title "$1" --msgbox "$2" 0 0
}
dyesno() {
    dialog --backtitle "$BACKTITLE" --title "$1" --yesno "$2" 0 0
}
dpasswd() {
    dialog --stdout --backtitle "$BACKTITLE" --title "$1" --passwordbox "$2" 0 0
}
dinput() {
    dialog --stdout --backtitle "$BACKTITLE" --title "$1" --inputbox "$2" 0 0
}
quit() {
    dmsgbox5s "Ende." "$@"
    exit
}
hauptmenu() {
    dialog --stdout --backtitle "$BACKTITLE" --title "Haupt-Auswahl:" \
    --radiolist "Bitte auswaehlen:" 10 70 3 \
    . "Sicherung" "Sicherung der Festplattenpartitionen" off \
    . "Ruecksicherung" "Ruecksicherung der Festplattenpartitionen" on \
    . "Image-Info" "Information ueber Image" off
}
#####
# FARBEN:
rot='\033[01;32m'
gelb='\033[01;33m'
blau='\033[01;34m'
gruen='\033[01;32m'
weiss='\033[01;29m'
normal='\033[00m'
#####
```