

Handreichung  
zur Installation von Linux  
in Schulen

Linux –  
Systemsicherung und Klonen mit tar

Die meisten Imaging-Programme – sowohl für Windows wie für Linux – arbeiten auf Partitions-ebene oder kopieren ganze Festplatten. Die Systemsicherung und das Klonen von Linux-Arbeitsstationen mit dem Archivierungswerkzeug tar arbeitet dagegen auf Dateiebene. Dies ist in vielen Fällen der flexiblere Weg, da man unabhängig von der Festplatteneinteilung oder dem Dateisystem ist.

Diese Handreichung beschreibt die einzelnen Techniken, das Einrichten der Festplatte, den Umgang mit tar und dem Bootmanager Grub und liefert fertige Skripte, mit denen eine Linux-Arbeitsstation in wenigen Minuten installiert ist.

Für Schulungsumgebungen ist dies eine flexible Möglichkeit, verschiedene Linux-Installationen vorrätig zu halten und schnell zu installieren.

Voraussetzung ist ein Linux-System, das von einem USB-Stick gebootet werden kann. Dies ist in der Handreichung „Linux auf einem USB-Stick“ beschrieben.

**INHALT**

Systemsicherung und Klonen von Linux mit tar.....	3
Teil 1: Systemsicherung.....	4
Anpassen einer Linux-Installation zum Klonen .....	4
Packen der Linux-Installation in ein Tar-Archiv.....	6
Teil 2: Systemwiederherstellung bzw. Klonen .....	9
Einrichten der Festplatte.....	9
Systemwiederherstellung von Linux mit tar .....	10
Einrichten des Bootmanagers Grub 2 .....	10
Eventuelle Nacharbeiten am geklonten PC.....	12
Anhang 1: Skripte zur automatisierten Installation .....	13
Anhang 2: Erweiterte Skripte zur automatisierten Installation .....	15
Anhang 3: Umgang mit tar .....	18
Anhang 4: Umgang mit fdisk .....	22

**IMPRESSUM**

Akademie für Lehrerfortbildung und Personalführung, Dillingen (<http://alp.dillingen.de>)

Die Handreichung wurde im Rahmen des Projektes SCHULNETZ erarbeitet. Sie ist unter der Adresse <http://alp.dillingen.de/schulnetz/materialien> abrufbar.

Herausgeber: Akademie für Lehrerfortbildung und Personalführung  
Kardinal-von-Waldburg-Str. 6-7  
89407 Dillingen

Autor: Georg Schlagbauer, Akademie Dillingen

Technische Beratung: Christian Maushart, Bürgernetz Dillingen

E-Mail: [schlagbauer@alp.dillingen.de](mailto:schlagbauer@alp.dillingen.de)

Stand: November 2013



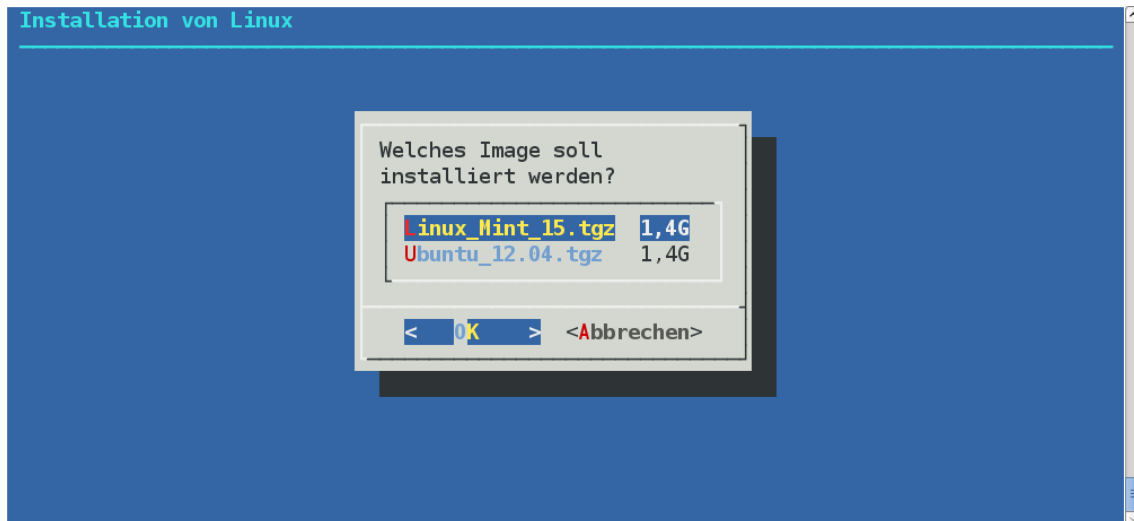
## SYSTEMSICHERUNG UND KLONEN VON LINUX MIT TAR

Um von einem Linux-System eine Systemsicherung anzufertigen oder um ein Linux-System zu klonen, ist nicht unbedingt ein Imaging-Programm nötig. In vielen Fällen ist es einfacher, schneller und zudem flexibler, wenn man nur die relevanten Dateien und Verzeichnisse sichert oder auf einen neuen Computer kopiert. Auf diese Art kann man z. B. von einem vorbereitem USB-Stick Linux in weniger als 5 Minuten auf einem neuen Computer installieren.

Am Ende dieser Handreichung sind Vorschläge für Skripte zusammengestellt, die diese Installation automatisiert erledigen. Diese Skripte können bei Bedarf an eigene Vorstellungen oder Rahmenbedingungen angepasst werden. Die wesentlichen Mechanismen sind im Detail beschrieben:

- Anpassen einer Linux-Installation zum Klonen
- Systemsicherung von Linux mit tar
- Partitionieren und Formatieren einer neuen Festplatte
- Systemwiederherstellung von Linux mit tar
- Installation des Bootmanagers Grub

Ziel könnte z. B. sein, einen bootfähigen USB-Stick zu erstellen, der über ein Menü anbietet, unterschiedliche vorbereitete Linux-Systeme auf einem Computer einzurichten. Man ist damit vor allem in Schulungsumgebungen sehr flexibel.



Das Erstellen des bootfähigen USB-Sticks ist in der Handreichung „Linux auf einem USB-Stick“ beschrieben. In dieser Handreichung geht es um das Erstellen von Images.

Die erweiterten Skripte (mit Fehlerrountinen und dialog als grafischer Oberfläche) sind, weil sie doch einige Seiten in Anspruch nehmen würden, nicht abgedruckt, sondern können unter <http://dozenten.alp.dillingen.de/schulnetz> heruntergeladen werden.

Ebenso sind in diesem Verzeichnis einige vorbereitete tar-Archive gesammelt.

## TEIL 1: SYSTEMSICHERUNG

Ziel der Systemsicherung ist es, ein Archiv (Linux.tgz) zu erhalten, das auf einen anderen Computer überspielt werden kann. Die Systemsicherung besteht aus zwei Teilen:

- Anpassen einer Linux-Installation zum Klonen
- Packen der Linux-Installation in ein Tar-Archiv

Falls nur an eine reine Systemsicherung (und nicht an das Klonen) gedacht ist, kann die Anpassung der Linux-Installation zum Klonen entfallen.

## ANPASSEN EINER LINUX-INSTALLATION ZUM KLONEN

Es sei vorausgesetzt, dass auf einem Computer ein fertiges Linux installiert ist, das man konservieren oder auf andere Computer bringen möchte. Diese Installation enthält alle gewünschten Programme und sollte auch bezüglich der Updates auf einem aktuellen Stand sein.

Am einfachsten ist es, wenn man bereits bei einer Neuinstallation auf die späteren Belange beim Klonen Rücksicht nehmen kann. Diese sind

- Gesamte Installation in eine Partition
- Evtl. Verzicht auf eine Swap-Partition

Im Folgenden sind einige Einstellungen gesammelt, die man ggf. vor dem Klonen noch beeinflussen sollte:

## NETZWERKEINSTELLUNGEN

Statische IP-Einstellungen sind beim Klonen ein Hindernis. Am einfachsten ist es, die Linux-Installation vor dem Klonen auf DHCP zu stellen.

## SWAP-PARTITION

Standardmäßig verwenden alle Linux-Installationen eine Swap-Partition. Bei Arbeitsplatzcomputern kann bei ausreichendem Arbeitsspeicher auf die Swap-Partition verzichtet werden. In den Skriptvorschlägen zum Klonen wird keine Swap-Partition eingerichtet.

## ÜBERPRÜFUNG, OB SWAP GENUTZT WIRD:

```
cat /proc/swaps
```



---

**DEAKTIVIEREN DER SWAP-PARTITION:**

```
swapoff -a
```

Eintrag aus `/etc/fstab` entfernen, ggf. noch zusätzlich Datei `/etc/initramfs-tools/conf.d/resume` entfernen bzw. Swap-Verzeichnis auskommentieren oder löschen

```
update-initramfs -u
```

Damit wird ein neues `initramfs` ohne Hinweis auf eine Swap-Partition angelegt.

**KONFIGURATIONSDATEI `/etc/fstab`**

Eine Partition in der Konfigurationsdatei `/etc/fstab` wird entweder durch die Gerätedatei (z. B. `/dev/sda1`) oder durch die UUID (Universally Unique Identifier z. B. `UUID=60c5...`) identifiziert. Vor dem Klonen sollte die UUID durch die Gerätedatei `/dev/sda1` ersetzt werden.

```
UUID=60c5... / ext3 defaults,errors=remount-ro 0 1  
ersetzen durch  
/dev/sda1 / ext3 defaults,errors=remount-ro 0 1
```

Dies ist für die meisten Installationen der richtige Eintrag und braucht deshalb nach dem Klonen nicht mehr geändert werden. In manchen Fällen, wenn die Installation nicht auf `/dev/sda1` erfolgt, ist es notwendig, nach dem Klonen die Gerätedatei wieder durch das korrekte Device oder die UUID zu ersetzen.

Hinweis: Die erweiterten Installationskripte im Anhang ersetzen den Eintrag `/dev/sda1` durch die korrekte UUID.

Der Eintrag für die Swap-Partition und weitere computerspezifische Einträge können eventuell entfernt werden.

**UDEV-KONFIGURATION UNTER `/ETC/UDEV/RULES.D`**

Im `udev`-Konfigurationsverzeichnis ist die Netzwerkkarte (`eth0`) einer bestimmten MAC-Adresse oder das CD-Laufwerk einer bestimmten SCSI-ID zugeordnet. Auf einer geänderten Hardware stimmen diese Einträge nicht mehr. Die `udev`-Hardware-Einträge (`70-persistent-cd.rules` und `70-persistent-net.rules`) werden – wenn sie nicht vorhanden sind – nach einem Neustart automatisch neu generiert und können deshalb vor dem Sichern entfernt werden.

---

**UDEV-EINTRÄGE**

```
/etc/udev/rules.d/70-persistent-cd.rules
```

```
/etc/udev/rules.d/70-persistent-net.rules
```



## LÖSCHEN DER UDEV-EINTRÄGE

```
rm /etc/udev/rules.d/*
```

Löschen aller udev-Einträge

Das Verzeichnis `/etc/udev/rules.d/` sollte nicht gelöscht werden.

## SSH-HOST-KEYS IN `/etc/ssh`

Bei Arbeitsplatz-Computern in Schulungsumgebungen, für die diese Handreichung primär gedacht ist, muss auf diese Einstellung keine Rücksicht genommen werden.

Soll ein Serversystem geklont werden, ist es sinnvoll, auf eindeutige SSH-Keys zu achten, wenn auf die Server per ssh zugegriffen werden soll. Nach dem Klonen hätten sonst alle Server die gleichen Schlüssel, was ggf. ein Sicherheitsproblem darstellen kann.

Vor dem Klonen eines Serversystems kann man die host-keys entfernen oder in einer exclude-Datei angeben, dass diese nicht in das Archiv kopiert werden. Allerdings darf man dann nicht vergessen, nach dem Klonen den ssh-Dienst erneut zu installieren.

```
rm /etc/ssh/ssh_host_*
```

Löschen der Host-Keys

## PACKEN DER LINUX-INSTALLATION IN EIN TAR-ARCHIV

Wenn man noch keine Erfahrung im Umgang mit tar hat, kann es sinnvoll sein, zunächst im Anhang den „Umgang mit tar“ nachzulesen.

Ziel ist es, die gesamte Linux-Installation – das heißt, alle Dateien und Verzeichnisse – in ein tar-Archiv zu packen. Auszunehmen davon sind die Inhalte der Pseudo-Dateisysteme `/proc`, `/sys` und `/dev` sowie ggf. die Inhalte von `/media`, die nur Mountpunkte enthalten sollten.

<code>/proc</code>	Schnittstelle zum Kernel
<code>/sys</code>	Kernel-Informationen über Geräte und Treiber
<code>/dev</code>	Gerätedateien
<code>/media</code>	Mountpunkte für USB-Sticks, CDs, etc.

Gegebenenfalls kann es sinnvoll sein, weitere Dateien oder Verzeichnisse von der Sicherung auszuschließen:

<code>/tmp</code>	temporäre Dateien
<code>/usr/src</code>	Installationsquellen, z. B. Kernel-Header
<code>/etc/udev/rules.d</code>	UDEV-Dateien
<code>/etc/ssh/ssh_host_*</code>	SSH-Host-Keys



Das tar-Archiv kann aus dem laufenden Linux-System heraus erzeugt werden, obwohl dies nicht die empfohlene Methode ist. Beispiele:

```
tar -czf Linux.tgz -X exclude.txt --one-file-system --ignore-failed-read /
tar -czf Linux.tgz -X exclude.txt --ignore-failed-read /
```

(Die Pfade für das Archiv und die exclude-Datei müssen ggf. angegeben werden.)

#### BEISPIEL FÜR DIE DATEI EXCLUDE.TXT

```
/proc/*
/sys/*
/dev/*
/media/*
/tmp/*
/usr/src/*
/etc/udev/rules.d/*
/etc/ssh/ssh_host_*
```

Beim Sichern aus dem laufenden System heraus, können eventuell einzelne Dateien oder Verzeichnisse nicht gelesen werden (z. B. .gvfs). Mit der Option `--ignore-failed-read` werden diese übergangen, ohne dass der tar-Befehl abbricht.

#### SICHERUNG AUS EINEM LIVE-SYSTEM

Der empfohlene Weg, ein tar-Archiv zu erzeugen, ist, den Computer von einem Linux-Live-System zu starten, die zu sichernde Partition einzuhängen und daraus das tar-Archiv zu erzeugen. Problemlos funktioniert dies vor allem dann, wenn Linux in einem einzigen Dateisystem installiert wurde (ohne Aufteilung in verschiedene Partitionen). Ansonsten müsste man alle Partitionen manuell korrekt einhängen, um ein brauchbares Archiv zu erzeugen.

Im vorgestellten Skript wird zunächst ein Unterverzeichnis `/images` angelegt, in dem anschließend das erzeugte Archiv `Linux.tgz` abgelegt wird. Die zu sichernde Linux-Installation wird in `/dev/sda1` angenommen.

```
mkdir -p /images
mount /dev/sda1 /mnt
tar -czvf /images/Linux.tgz /mnt
```

Bei Bedarf kann man über eine exclude-Datei einzelne Dateien oder Verzeichnisse aus der Sicherung ausschließen.



```
mkdir -p /images
mount /dev/sda1 /mnt
tar -czvf /images/Linux.tgz -X /images/exclude.txt /mnt
```

---

#### BEISPIEL FÜR DIE DATEI EXCLUDE.TXT

```
/mnt/proc/*
/mnt/sys/*
/mnt/dev/*
/mnt/media/*
/mnt/tmp/*
/mnt/usr/src/*
/mnt/etc/udev/rules.d/*
```

Bei der Sicherung sollte man darauf achten, dass das erzeugte Archiv nicht auf dem zu sichernden Dateisystem liegt, sonst versucht man das Archiv mit zusichern und erhält eine entsprechende Fehlermeldung.

---

#### DAS ARCHIV Linux.tgz

Als Ergebnis der Sicherung erhält man das Archiv Linux.tgz, das auch den Mountpoint mnt enthält:

```
tar -tzf Linux.tgz

mnt/
mnt/etc/
mnt/etc/...
...
```





## TEIL 2: SYSTEMWIEDERHERSTELLUNG BZW. KLONEN

Ziel der Systemwiederherstellung ist es, aus einem Archiv (z. B. Linux.tgz) ein lauffähiges Linux-System zu erhalten. Dazu wird der Computer von einem Live-System (z. B. Live-USB-Stick) gestartet. Die Systemwiederherstellung besteht aus folgenden Tätigkeiten:

- Einrichten (Partitionieren und Formatieren) der Festplatte
- Systemwiederherstellung von Linux mit tar
- Einrichten des Bootmanagers Grub 2
- Eventuelle Nacharbeiten am geklonten PC

Falls nur an eine Systemwiederherstellung auf dem gleichen PC gedacht ist, auf dem die Sicherung erfolgte, können ggf. das Einrichten der Festplatte und das Einrichten des Bootmanagers entfallen.

### EINRICHTEN DER FESTPLATTE

Zum Einrichten der Festplatte verwendet man die beide Programme fdisk (Partitionieren) und mkfs (Formatieren). Eine kleine Einführung zum Umgang mit fdisk findet man im Anhang.

Auf einem Arbeitsplatzcomputer genügt eine einzige Partition für das Dateisystem. Auch auf eine Swap-Partition wurde im nachfolgenden Beispiel verzichtet.

### PARTITIONIEREN DER FESTPLATTE

Mit dem nachfolgenden Skript wird der MBR auf der Festplatte /dev/sda gelöscht (mit Nullen überschrieben), danach wird eine neue Partitionstabelle und eine primäre Partition erzeugt, die die Größe der gesamten Festplatte hat. (Alle vorhandenen Daten auf dieser Festplatte werden dabei gelöscht.)

```
dd if=/dev/zero of=/dev/sda bs=512 count=1
```

```
echo "  
n  
p  
1  
  
w  
" | fdisk /dev/sda > /dev/null 2>&1  
  
partprobe
```

Der Befehl partprobe informiert das Betriebssystem über Änderungen der Partitionstabelle. Gegebenenfalls muss das Programmpaket parted nachinstalliert werden. Falls vorher eine Partition gemounted war, kann auch ein Neustart des PC erforderlich werden.



## ANLEGEN DES DATEISYSTEMS

Das Anlegen eines neuen Dateisystems funktioniert mit mkfs. Je nach gewünschtem Dateisystem kommen die unterschiedlichen Programmvarianten mkfs.ext3, mkfs.ext4, mkfs.vfat in Frage

```
mkfs.ext3 /dev/sda1
```

```
mkfs.ext4 /dev/sda1
```

## SYSTEMWIEDERHERSTELLUNG VON LINUX MIT TAR

Bei der Systemwiederherstellung von Linux aus einem tar-Archiv sollte Linux von einem Live-System (Live-CD oder Live-USB-Stick) gestartet sein. Im folgenden Beispiel wird vorausgesetzt, dass die Sicherung unter /images/Linux.tgz zu finden ist und folgendes Aussehen hat:

```
tar -tzf /images/Linux.tgz
```

```
mnt/
```

```
mnt/etc/
```

```
mnt/etc/...
```

```
...
```

## ZURÜCKSPIELEN DES ARCHIVS

```
cd /
```

```
mount /dev/sda1 /mnt
```

```
tar -xzvf /images/Linux.tgz
```

## EINRICHTEN DES BOOTMANAGERS GRUB 2

Im Folgenden wird davon ausgegangen, dass der Computer von einem Live-System (USB-Stick) gebootet wurde und dass der Bootmanager Grub 2 im MBR der Festplatte /dev/sda installiert werden soll.

Des Weiteren wird davon ausgegangen, dass das neue Linux-System nach /dev/sda1 kopiert wurde und diese Partition in /mnt gemountet ist.

Am einfachsten ist es, wenn man vor der Installation von Grub das neue System in einer chroot-Umgebung startet. Damit man im neuen Dateisystem arbeiten kann, müssen die Pseudo-Dateisysteme /proc und /sys und die Gerätedateien unter /dev eingebunden werden:



```
mount -t proc none /mnt/proc
mount -t sysfs none /mnt/sys
mount -o bind /dev /mnt/dev
```

Diese Einbindung der Gerätedateien in die chroot-Umgebung funktioniert nur, wenn sowohl das Live-System als auch das installierte System die gleiche Architektur verwenden (64 bit – 64 bit oder 32 bit – 32 bit, aber nicht gemischt).

War die Einbindung der Gerätedateien erfolgreich, wird in der chroot-Umgebung der Bootmanager Grub 2 installiert.

```
chroot /mnt
grub-install /dev/sda
update-grub
exit
```

Nach dem Einrichten des Bootmanagers sollte das neue System von der Festplatte gestartet werden können.

#### WECHSEL IN EINE CHROOT-UMGEBUNG PER SKRIPT

Beim Einrichten des Bootmanagers wird in eine chroot-Umgebung gewechselt. In dieser chroot-Umgebung werden die beiden Befehle grub-install und update-grub aufgerufen. Danach wird die chroot-Umgebung mit exit wieder verlassen.

Obwohl dieser Vorgang sehr einfach ist, hat man bei der Ausführung der Befehle per Skript das Problem, dass nach dem Wechsel in die chroot-Umgebung das Skript nicht mehr zur Verfügung steht und man deshalb eingreifen müsste. Lösen lässt sich dies, indem man die Befehle der chroot-Umgebung in eine separate Datei schreibt (die in der chroot-Umgebung zugänglich ist) und diese Datei dem chroot-Befehl zur Ausführung mitgibt. Diese Datei (im Beispiel: temp-skript) lässt sich auch temporär per Skript erzeugen.

```
cat << XXX > /mnt/tmp/temp-skript
grub-install /dev/sda
update-grub
XXX
```

```
chmod a+x /mnt/tmp/temp-skript
chroot /mnt /tmp/temp-skript
rm /mnt/tmp/temp-skript
```



**EVENTUELLE NACHARBEITEN AM GEKLONTEN PC****COMPUTERNAME**

Windows-Computer bringen eine Meldung, wenn der gleiche Computername in Netz vorhanden ist. Bei Linux-Systemen spielt dies keine Rolle, solange man die Computer nicht mit dem Namen, sondern über die IP-Adresse anspricht. Will man dennoch einen eindeutigen Namen verwenden, übergibt man diesen in der Datei `/etc/hostname`

<code>/etc/hostname</code>	Nach einem Neustart wird aus dieser Datei der Name gelesen
<code>hostname neuername</code>	Setzen eines neuen Hostname im laufenden Betrieb

**SSH-DIENST**

Wurden (bei Serversystemen) die `ssh-host-keys` nicht ins Archiv kopiert, funktioniert der `ssh`-Zugriff auf den Server nicht. Dieser Dienst muss erneut installiert werden.

<code>aptitude reinstall openssh-server</code>	Installation des <code>ssh</code> -Servers
--	--

**KONFIGURATIONSDATEI `/etc/fstab`**

Bei der Vorbereitung eines Computers zum Klonen wurde empfohlen, die UUID durch die Gerätedatei `/dev/sda1` zu ersetzen. Wird das Image auf einen Arbeitsplatzcomputer (mit nur einer Festplatte) aufgespielt, ist dies die korrekte Einstellung. Eine Nacharbeit ist nicht erforderlich.

Bei Sondersituationen (z. B. mehrere Festplatten im PC, mehrere installierte Betriebssysteme auf dem PC, Installation auf einem USB-Stick) befindet sich die Linux-Installation nicht auf `/dev/sda1`, so dass nachträglich das korrekte Device oder die UUID (bei USB-Sticks) eingetragen werden muss.

In den im Anhang 2 vorgestellten erweiterten Skripten, wird in der Datei `/etc/fstab` die korrekte UUID eingetragen.

## ANHANG 1: SKRIPTE ZUR AUTOMATISIERTEN INSTALLATION

Im Folgenden sind zwei einfach gehaltene Shell-Skripte (startskript.sh und linux\_installieren.sh) dargestellt, die das wesentliche Vorgehen bei der automatisierten Installation von Linux verdeutlichen.

Die Skripte startskript.sh und linux\_installieren.sh können, um Tipparbeit zu ersparen, unter <http://dozenten.alp.dillingen.de/schulnetz> kopiert werden.

### ERZEUGEN DES ARCHIVS LINUX.TGZ

Im Folgenden wird davon ausgegangen, dass Linux auf /dev/sda1 installiert ist und die Installation wie im Kapitel „Anpassen einer Linux-Installation zum Klonen“ vorbereitet wurde.

Des Weiteren wird davon ausgegangen, dass der Computer von einem Linux-USB-Stick gebootet wurde und auf dem USB-Stick genügend Platz ist, um das erzeugte Archiv zu speichern.

```
mkdir -p /images
mount /dev/sda1 /mnt
tar -czf /images/Linux.tgz /mnt
```

### SHELL-SKRIPTE ZUR SYSTEMWIEDERHERSTELLUNG AUS DEM ARCHIV Linux.tgz

Im Folgenden wird davon ausgegangen, dass der Computer von einem Linux-USB-Stick gebootet wurde. Die Shell-Skripte sollten unter /root/skripte/ abgelegt sein, das Archiv ist unter /images/Linux.tgz zu finden und wurde wie oben beschrieben erzeugt.

Im Startskript erfolgt eine Sicherheitsabfrage, ob man wirklich die gesamte Festplatte löschen und Linux neu installieren möchte. Aus dem Startskript heraus wird das eigentliche Installationskript aufgerufen.

---

#### DAS STARTSKRIPT /root/skripte/startskript.sh

```
#!/bin/bash
clear
echo
echo
echo "Wenn Sie weitermachen wird die gesamte Festplatte"
echo "geloescht und Linux installiert."
echo "Wollen Sie wirklich weitermachen?"
read -p "(ja/nein) " ANTWORT
if [ "$ANTWORT" = "ja" ]; then
    sh /root/skripte/linux_installieren.sh /dev/sda /images/Linux.tgz
fi
```



---

**DAS INSTALLATIONSSKRIPT linux\_installieren.sh**

```
#!/bin/bash
# Aufruf: Linux_installieren.sh <Festplatte> <Archiv>
# Vorsicht: Die Festplatte des Computers wird neu formatiert.
# Alle Daten auf der Festplatte sind damit geloescht.
#
Festplatte=$1
Archiv=$2
Partition=${Festplatte}1
if [ "$2" = "" ] ; then
    echo "Nicht genuegend Parameter"
    exit
fi

# Festplatte partitionieren
dd if=/dev/zero of=$Festplatte count=1
echo "
n
p
1

w
" | fdisk $Festplatte
partprobe || exit

# Dateisystem anlegen
mkfs.ext4 $Partition || exit

# Mounten
mount $Partition /mnt || exit

# tar-Archiv kopieren
cd / ; tar -xzvf $Archiv

# Grub einrichten
mount -t proc none /mnt/proc
mount -t sysfs none /mnt/sys
mount -o bind /dev /mnt/dev

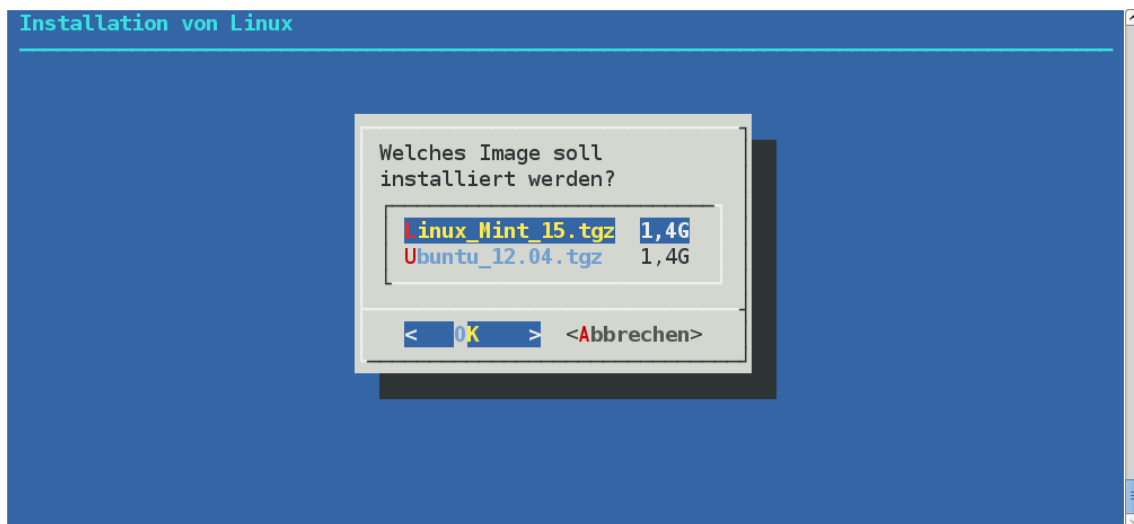
cat << XXX > /mnt/tmp/xx
grub-install $Festplatte
update-grub
XXX

chmod a+x /mnt/tmp/xx
chroot /mnt /tmp/xx
```



## ANHANG 2: ERWEITERTE SKRIPTE ZUR AUTOMATISIERTEN INSTALLATION

Im Unterschied zu den in Anhang 1 vorgestellten einfachen Skripten lassen die erweiterten Skripte eine Auswahl der Festplatte zu, auf der die Installation erfolgen soll und ermöglichen eine Auswahl des zu installierenden Archivs. Die Auswahl erfolgt jeweils über ein grafisch orientiertes Menü (mit dialog). Das Programm dialog muss dazu auf dem Live-USB-Stick installiert sein.



Die erweiterten Skripte können, unter <http://dozenten.alp.dillingen.de/schulnetz> kopiert werden.

### ERZEUGEN EINES ARCHIVS Linux.tgz

Archive können, wie in Anhang 1 beschrieben, erzeugt werden. Die Archive können einen beliebigen Namen (ohne Leerzeichen) mit der Endung `tgz` erhalten und sollten im Verzeichnis `/images` auf dem Live-Stick abgelegt werden.

### ERWEITERTE SHELL-SKRIPTE ZUR SYSTEMWIEDERHERSTELLUNG

Die Shell-Skripte können z. B. unter `/root/skripte/` abgelegt sein, die Archive befinden sich unter `/images/` mit der Endung `tgz`. Die Namen der Archive dürfen keine Leerzeichen enthalten.

Die Shell-Skripte bestehen aus folgenden Dateien:

```
startskript.sh
Festplatte_auswaehlen.sh
Datei_auswaehlen.sh
Linux_installieren.sh
```

Aus dem Startskript heraus werden die weiteren Skripte aufgerufen. Nachfolgend sind einige Techniken erläutert, die in den erweiterten Skripten verwendet werden:

## FESTPLATTEN UND USB-STICKS ERMITTELN

Zur Auswahl, auf welcher Festplatte oder auf welchem USB-Stick installiert werden soll, werden alle angeschlossenen Festplatten bzw. USB-Sticks ermittelt. Im Skript wird dazu der Befehl `fdisk` verwendet.

```
# fdisk -l

Disk /dev/sda: 53.7 GB, 53687091200 bytes
255 heads, 63 sectors/track, 6527 cylinders
...
Disk identifier: 0x000c6751
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            1         6079     48827392   83  Linux
/dev/sda2           6079         6527     3599360    82  Linux
...
Disk /dev/sdb: 274.9 GB, 274877906944 bytes
...
```

An der Ausgabe erkennt man, dass die Zeilen interessant sind, die mit „Disk /dev“ beginnen. Die Festplatte wird durch den zweiten Teilstring identifiziert:

```
devices=`fdisk -l 2>/dev/null | grep "^Disk /dev"
| awk '{print $2}' | cut -c1-8`
```

## GRÖÖE EINER FESTPLATTE ERMITTELN

Zur Ermittlung der Größe einer Festplatte eignet sich auch die obige Ausgabe mit `fdisk`, wobei nur der dritte und vierte Teilstring der jeweiligen Zeile von Interesse ist. Das Leerzeichen wird durch einen Unterstrich ersetzt, um nachfolgenden Routinen die Verarbeitung zu erleichtern. Das abschließende Komma wird entfernt.

```
# lw=/dev/sda
# size=`fdisk -l $lw 2>/dev/null | grep "^Disk /dev" `
# size=`echo $size | awk '{print $3"_"$4}' `
# size=`echo $size | sed 's/,,$//' `
```

## BOOTDEVICE FESTSTELLEN

Bei der Auswahl, auf welcher Festplatte installiert werden soll, sollen alle Festplatten (bzw. USB-Sticks) angeboten werden, außer dem Gerät, von dem der PC gebootet wurde. Im Skript wird dazu der Befehl „`df` /“ verwendet.

Je nach Eintrag im Bootmanager grub erhält man mit `df` unterschiedliche Ausgaben (Eintrag in `/proc/mounts`):





```
# df /  
/dev/disk/by-uuid/60c5... ..
```

oder

```
# df /  
/dev/sda1 ..
```

Den Zusammenhang zwischen UUID und Gerätenamen stellt blkid her.

```
tempstring=`df / | grep "^/dev" | awk '{print $1}' | tail -c37`  
bootdevice=`blkid | grep $tempstring | cut -c1-8`
```

Die erste Zeile liefert das Bootdevice als UUID oder Gerätenamen. Die zweite Zeile liefert unabhängig davon den Gerätenamen, der auf die ersten 8 Zeichen reduziert wird (auf /dev/sda, /dev/sdb, etc.), um die Bootfestplatte zu erhalten.



## ANHANG 3: UMGANG MIT TAR

Dieses Kapitel gibt eine kleine Einführung in den Umgang mit tar.

tar (ursprünglich für tape archiver) eignet sich zum Archivieren von Daten. Zum Archivieren auf Bändern wird tar heute kaum mehr verwendet, tar eignet sich jedoch sehr gut, um viele Dateien und Verzeichnisse in eine Archivdatei zu packen, um diese aufzubewahren oder anderen Benutzern zur Verfügung zu stellen. Auch viele Linux-Programme werden in tar-Archiven ausgeliefert.

### BEISPIEL: ARCHIVIEREN ALLER HOME-VERZEICHNISSE

Im folgenden Beispiel werden alle Homeverzeichnisse (unter /home) in die Archivdatei archiv.tar gepackt. tar kann dabei mit einer alten Syntax (aus Unix-Zeiten) oder mit einer neueren Syntax (GNU-Linux, mit Minus-Zeichen bei den Optionen) verwendet werden.

```
tar cf archiv.tar /home    alte Syntax
```

```
tar -cf archiv.tar /home   neuere Syntax (nach GNU-Linux)
```

Die alte und die neue Syntax unterscheiden sich zum Teil auch in der Reihenfolge, wie die Optionen verwendet werden können. Im Folgenden wird nur die neuere Syntax verwendet.

### AKTIONEN VON TAR

- c (create) Archiv neu anlegen
- t (table) Inhalt eines Archivs anzeigen
- u (update) Neuere Dateien in das Archiv schreiben
- d (diff) Dateien mit einem Archiv vergleichen
- x (extract) Dateien aus dem Archiv zurückspielen

### ANGABE DES ARCHIVS

Als Archiv kann eine Datei, eine Gerätedatei (z. B. Bandlaufwerk) oder die Standardein- bzw. -ausgabe angegeben werden. Das Archiv wird mit der Option -f angegeben.

- f archiv.tar Angegebenes Archiv verwenden
- f - Standardeingabe bzw. Standardausgabe verwenden



## KOMPRIMIERUNG DES ARCHIVS

Archive werden häufig mit gzip oder bzip2 komprimiert. Es ist üblich, die Art der Komprimierung durch die Dateierweiterung deutlich zu machen.

- z Komprimierung mit gzip (Dateierweiterung .tar.gz oder .tgz)
- j Komprimierung mit bzip2 (Dateierweiterung .tar.bz2 oder .tbz)

```
tar -cf archiv.tar /home keine Komprimierung
tar -czf archiv.tgz /home Komprimierung mit gzip
tar -cjf archiv.tbz /home Komprimierung mit bzip
```

Bei der Komprimierung müssen die entsprechenden Programme (gzip oder bzip2) installiert sein. Die Update-Funktion (-u) zum Auffrischen eines Archivs funktioniert nicht bei komprimierten Archiven.

Die Komprimierung eines tar-Archivs kann auch getrennt von der Archivierung erfolgen. Die folgenden Befehle führen jeweils zum gleichen Ergebnis:

---

### EINPACKEN

```
tar -czf archiv.tar.gz /home

tar -cf archiv.tar /home
gzip archiv.tar

tar -cf - /home | gzip > archiv.tar.gz
```

---

### AUSPACKEN

```
tar -xzf archiv.tar.gz

gunzip archive.tar.gz
tar -xf archiv.tar

gunzip -c archive.tar.gz | tar -xf -
```

## PFADANGABEN IM ARCHIV

tar sichert relative Pfade. Auch bei einer absoluten Pfadangabe wird der führende / entfernt. Der relative Pfad geht vom Arbeitsverzeichnis aus. Dies ist das Verzeichnis, in dem tar aufgerufen wird (falls nicht durch die Option -C ein anderes Verzeichnis bestimmt wird).



Beispiel: Es wird das Verzeichnis /home/gs in ein tar-Archiv geschoben (In diesem Verzeichnis befinden sich 2 Dateien):

```
/#> tar -czvf gs.tgz home/gs
home/gs/
home/gs/datei1
home/gs/datei2
```

```
/home#> tar -czvf gs.tgz gs
gs/
gs/datei1
gs/datei2
```

```
/home/gs#> tar -czvf gs.tgz .
./
./datei1
./datei2
```

Das Beispiel zeigt die relativen Pfadangaben ausgehend vom Arbeitsverzeichnis. Beim Auspacken eines Archivs muss man sich deshalb im gleichen Arbeitsverzeichnis wie bei der Sicherung befinden, wenn die Daten an den ursprünglichen Ort zurückgeschrieben werden sollen.

## EINPACKEN VON DATEIEN UND VERZEICHNISSEN

Beim Einpacken können mehrere Dateien oder Verzeichnisse angegeben werden. Bei Verzeichnissen werden alle Dateien in diesem Verzeichnis gesichert.

```
/home#> tar -czvf home.tgz gs ms js
```

Es werden die Verzeichnisse gs, ms und js gesichert.

Bei der Angabe mehrere Dateien oder Verzeichnissen sollte man nicht zwischen absoluten und relativen Pfaden wechseln. Dies würde zu Inkonsistenzen im Archiv führen.

```
/home#> tar -czvf home.tgz gs ms /home/js (ungünstig)
```

tar sichert Besitzverhältnisse, Zeitstempel und Zugriffsrechte mit.

## INHALT VON ARCHIVEN DURCHSUCHEN

Wird nach einer bestimmten Datei gesucht, kann die Ausgabe mit grep eingeschränkt werden.

```
/home#> tar -tzvf home.tgz | grep datei1
```



## AUSPACKEN VON DATEIEN UND VERZEICHNISSEN IN ARCHIVEN

Wird beim Auspacken eines Archivs kein Datei- oder Verzeichnisname angegeben, so wird das gesamte Archiv ausgepackt, ansonsten nur die angegebenen Dateien und Verzeichnisse.

```
/home#> tar -xzvf home.tgz gs/datei1
```

Es wird nur die Datei `gs/datei1` wiederhergestellt.

## SYMBOLISCHE LINKS

Symbolische Links werden als symbolische Links gespeichert, falls nicht angegeben wird, dass den symbolischen Verknüpfungen gefolgt werden soll.

`-h, --dereference` symbolischen Verknüpfungen folgen

## HARDLINKS

Hardlinks werden erkannt und im tar-Archiv markiert, falls sich diese im gleichen tar-Archiv befinden.

```
> tar -tzvf gs.tgz
```

```
drwxr-xr-x gs/gs 0 2012-02-26 10:22 gs/
-rw-r--r-- gs/gs 18 2012-02-26 09:45 gs/datei1
hrw-r--r-- gs/gs 0 2012-02-26 09:45 gs/datei2 Verknüpfung zu gs/datei1
```

In der Ausgabe sieht man, dass `datei2` als Hardlink zu `datei1` markiert ist. Dies macht jedoch Probleme, falls man versuchen würde `datei2` separat wieder herzustellen.

Mit der Option `--hard-dereference` kann tar veranlasst werden, Hardlinks als separate Dateien zu speichern.

`--hard-dereference` Hardlinks als separate Dateien auflösen

## WEITERE OPTIONEN

<code>-v</code>	Ausführliche Anzeige
<code>-p</code>	Dateizugriffsrechte erhalten (Voreinstellung für root)
<code>-X Datei</code>	aufgelistete Dateien oder Verzeichnisse auslassen
<code>--one-file-system</code>	bleibt innerhalb des Dateisystems
<code>--ignore-failed-read</code>	bricht nicht ab, wenn einzelne Dateien nicht gelesen werden können



## ANHANG 4: UMGANG MIT FDISK

fdisk ist ein interaktives Programm auf Kommandozeile. Das Löschen oder neu anlegen von Partitionen ist damit relativ einfach, wenn man sich mit der grundlegenden Syntax vertraut gemacht hat.

Ältere Versionen von fdisk (bis 2.17) starten in einem DOS-kompatiblen Modus, der mit der Option `-c` deaktiviert werden kann. Mit der Option `-u` wird die Anzeige auf Sektoren umgestellt. Bei neueren Versionen von fdisk kann auf die Optionen `-u` und `-c` verzichtet werden.

In den folgenden Beispielen wird davon ausgegangen, dass die zu partitionierende Festplatte mit `/dev/sda` angesprochen wird.

```
fdisk /dev/sda          Aufruf von fdisk
fdisk -c -u /dev/sda   Aufruf von fdisk bei älteren Versionen (bis 2.17)
```

fdisk lässt zwischen dem Master Boot Record (MBR) und dem Beginn der ersten Partition etwas Platz. Im DOS-kompatiblen Modus beginnt die erste Partition bei Sektor 63, ansonsten bei Sektor 2048. Der freie Platz zwischen dem MBR und der ersten Partition ist für den Grub-Bootmanager notwendig.

```
fdisk -l              Anzeige der Partitionierung aller Festplatten und USB-Laufwerke
fdisk /dev/sda        Bearbeiten der Festplatte /dev/sda
```

## LÖSCHEN EINER FESTPLATTE

Zum Löschen (kein physikalisches Löschen) einer Festplatte gibt es mehrere Möglichkeiten:

- Mit fdisk alle Partitionen löschen (Option `d`, `delete`).
- Mit fdisk eine neue leere DOS-Partitionstabelle anlegen (Option `o`)
- Mit dd den MBR löschen (`dd if=/dev/zero of=/dev/sda bs=512 count=1`)

Die Möglichkeit mit dd ist die radikalste, da sie einfach den MBR mit Nullen überschreibt.



## SKRIPTGESTEUERTER UMGANG MIT FDISK

Obwohl fdisk als interaktives Programm gedacht ist, kann man es auch in Skripten verwenden, wenn fdisk über eine Pipe die erforderlichen Befehle mitgegeben werden.

### LEERE DOS-PARTITIONSTABELLE ANLEGEN:

```
echo "  
o  
w  
" | fdisk /dev/sda > /dev/null 2>&1
```

Wenn die Bildschirmausgaben nicht interessieren, können sie (wie angegeben) nach /dev/null umgeleitet werden.

### LINUX-PARTITION /dev/sda1 MIT 20 GB ANLEGEN:

```
echo "  
n  
p  
1  
  
+20G  
w  
" | fdisk
```

Wichtig ist die Leerzeile in diesem Skript (wie bei der interaktiven Eingaben, wenn der default-Wert akzeptiert wird).

### SWAP-PARTITION /dev/sda2 MIT 1 GB ANLEGEN:

```
echo "  
n  
p  
2  
  
+1G  
t  
2  
82  
w  
" | fdisk
```

