

Praxis Linux-Administration

1. Veranstaltung

Carsten Gnörlich

Rechnerbetriebsgruppe
Technische Fakultät
Universität Bielefeld

28. April 2014

Übersicht

Themen für heute

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale
Kernel
Bootloader

Netzwerk
User
reboot
Tastaturlayout
Desktop

Szenario

Motivation

Laborumgebung

Maschine aufsetzen

Partitionieren

Partitionierung

Dateisystem anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Aufsetzen eines Servers

Situation im Serverraum

- ▶ Serverraum ist laut, kalt und eng
 - ▶ Server haben meist kein CD-ROM
- Aufsetzen im Serverraum ist sehr unbequem



Praxis Linux-
Administration

Carsten
Grönlisch

Szenario

Motivation

Laborumgebung

Maschine
aufsetzen

Partitionieren

Partitionierung

Dateisystem
anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Aufsetzen eines Servers

Typisches Vorgehen

Praxis Linux-
Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale

Kernel
Bootloader

Netzwerk
User

reboot
Tastaturlayout
Desktop

- ▶ Server in Schrank einbauen
 - ▶ elektrisch und netzwerkseitig anschließen
 - ▶ Netzwerk für DHCP-Boot vorbereiten
 - ▶ Maschine über USB-Stick booten
 - ▶ Aufsetzen und Konfiguration vom Büro aus machen (per ssh, serielle Konsole oder IPMI-Interface)
- Aufsetzen über Kommandozeile

Aufsetzen eines Servers

debootstrap – Debian-basiertes GNU/Linux aufsetzen

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale

Kernel
Bootloader

Netzwerk
User
reboot
Tastaturlayout
Desktop

- ▶ man arbeitet aus laufendem GNU/Linux heraus (z.B. gebooteter USB-Stick)
 - ▶ kann Debian oder Ubuntu aufsetzen
 - ▶ benötigte Softwarepakete werden aus dem Netz nachgeladen
 - ▶ per Script automatisierbar
 - ▶ Installation kann sehr knapp und zielgerichtet erfolgen (wir brauchen kein Office-Paket auf dem Server!)
- (-) man muß mehr Hand anlegen / konfigurieren als bei graphischen Installern
(Server brauchen typischerweise ohnehin spezielle Konfigurationen als Desktops, die im Installer so nicht vorgesehen sind)

Einführung in die Laborumgebung

Hardwareaufbau

Praxis Linux-Administration

Carsten Gnörlich

Szenario

Motivation

Laborumgebung

Maschine aufsetzen

Partitionieren

Partitionierung

Dateisystem anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

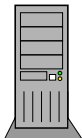
User

reboot

Tastaturlayout

Desktop

moehne
10.0.0.1



LINUX ADMIN 1



LINUX ADMIN 3



LINUX ADMIN 5



Netz: 10.0.0.0/24

LINUX ADMIN 2



LINUX ADMIN 4



LINUX ADMIN 6



- ▶ moehne dient als Server für
 - ▶ Debian-Pakete
 - ▶ Netzwerkdienste (DNS, DHCP)
- ▶ Nummer des Clients merken und beim nächsten Mal wieder nutzen

Booten:

- ▶ Memorystick einstecken
- ▶ Client einschalten, F12 fürs Boot-Menü
- ▶ im Boot-Menü Eintrag "USB Device" wählen
(am Ende des ersten Listenabschnittes)

Login:

- ▶ User *linux*, Paßwort *admin*
- ▶ `sudo befehl` oder
- ▶ `sudo -i` für dauerhafte Root-Shell

Haben wir eine IPv4-Adresse bekommen?

```
> sudo ifconfig
```

```
eth0 Link encap:Ethernet Hardware Adresse 00:1e:4f:9f:0c:78
      inet Adresse:10.0.0.164 Bcast:10.0.0.255 Maske:255.255.255.0
      inet6-Adresse: fe80::21e:4fff:fe9f:c78/64 Gültigkeitsbereich:Verbindung
      ...
```

```
lo Link encap:Lokale Schleife
     inet Adresse:127.0.0.1 Maske:255.0.0.0
     inet6-Adresse: ::1/128 Gültigkeitsbereich:Maschine
     ...
```

Funktioniert DNS?

```
> host moehne
```

```
moehne.localdomain has address 10.0.0.1
```

- ▶ zeigt auch, daß andere Maschinen im Netz 10.0.0/24 erreichbar sind.

Falls nicht, mit `ping 10.0.0.1` prüfen, ob das Netz überhaupt erreichbar ist.

Maschine aufsetzen

Motivation

Praxis Linux-
Administration

Carsten
Grönlisch

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Bisher erreicht:

- ▶ lokale Maschine ist vom USB-Stick gebootet
- ▶ lokale Maschine ist im Netzwerk

Unser Ziel:

- ▶ Debian GNU/Linux auf der lokalen Festplatte¹ installieren
- ▶ Maschine danach von Festplatte rebooten

¹später: im RAID

Festplatte partitionieren

Festplatte finden

Praxis Linux-Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren

Partitionierung

Dateisystem
anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Erwartete Konfig der Maschinen:

- ▶ sda: Festplatte 80GB oder 160GB
- ▶ sdb: 8GB USB-Stick

Ist es wirklich so? Den USB-Stick zu plätten wäre doof ;-)

```
> sudo fdisk -l
```

```
Disk /dev/sda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders, total 156301488 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xb0bd80a3
```

```
Disk /dev/sda doesn't contain a valid partition table
```

```
Disk /dev/sdb: 8004 MB, 8004304896 bytes
19 heads, 5 sectors/track, 164562 cylinders, total 15633408 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000d11c5
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	2048	15632383	7815168	83	Linux

Exkurs: Massenspeicher

Block-Devices

Praxis Linux-Administration

Carsten Gnörlich

Szenario

Motivation
Laborumgebung

Maschine aufsetzen

Partitionieren

Partitionierung

Dateisystem anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

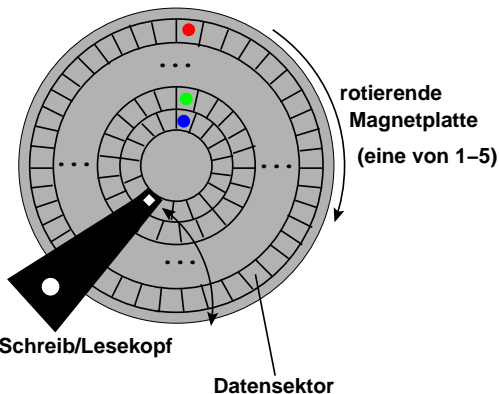
Netzwerk

User

reboot

Tastaturlayout

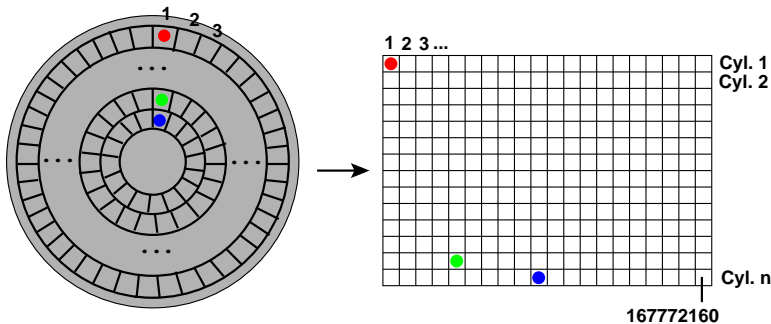
Desktop



- Festplatte besteht aus vielen Blöcken a 512 Bytes
- Blöcke sind durchnummeriert
- Festplatte wird in Blöcken (nicht Bytes) adressiert
- ▶ das nennt man ein "Block device" (Blockgerät)

Exkurs: Massenspeicher

Konzeptuelle Darstellung der Block-Device-Sektoren



Bem.:

- ▶ 167772160 Sektoren a 512 Bytes = 80GB Festplatte
- ▶ einige 10.000 Sektoren = 1 Zylinder (historisch bedingt!)

Praxis Linux-
Administration

Carsten
Grörllich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale

Kernel
Bootloader

Netzwerk
User

reboot
Tastaturlayout
Desktop

Massenspeicher

Block-Device als Rohgerät

Praxis Linux-Administration

Carsten Gnörlich

Szenario
Motivation
Laborumgebung

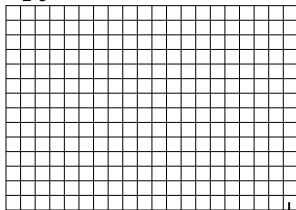
Maschine aufsetzen

Partitionieren
Partitionierung
Dateisystem anlegen

debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

/dev/sda

1 2 3 ...



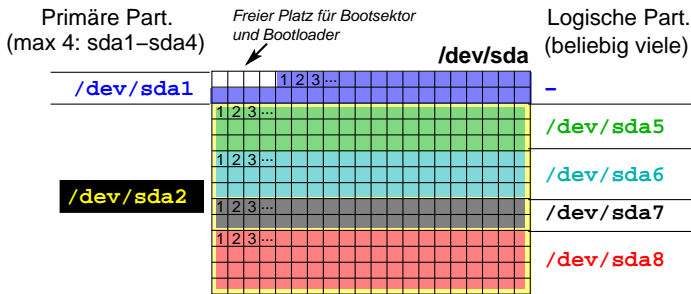
167772160

/dev/sda ist das *Rohgerät*

- kann man direkt nutzen
(in diesem Fall als 80GB Dateisystem)
- ist auf der Boot-Festplatte eher unüblich
- ▶ Partitionierung in kleine Pseudo-Block-Devices

Exkurs: Massenspeicher

Partitionierung macht aus einem Block-Device mehrere kleine Block-Devices



- ▶ `/dev/sda1`, `/dev/sda5`, ..., `/dev/sda8` sind ebenfalls wieder Block-Devices!

Bem: Unterscheidung primäre/logische Partitionen:

- ▶ "Unfall" aus historischen DOS-Zeiten!
- ▶ erledigt sich mit GPT und (U)EFI-Bios

Praxis Linux-Administration

Carsten Gnörlich

Szenario

Motivation
Laborumgebung

Maschine aufsetzen

Partitionieren
Partitionierung

Dateisystem anlegen
debootstrap
chroot
`/dev`, `/proc`, `/sys`

fstab
locale

Kernel
Bootloader

Netzwerk
User
reboot
Tastaturlayout
Desktop

Festplatte partitionieren

Partition für GNU/Linux-Installation erzeugen

Praxis Linux-
Administration

Carsten
Grönlisch

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Ziel: 8GB-Partition erstellen

- ▶ 8GB reichen sogar für kompletten Desktop mit Office

Beginn der Partitionierung von sda:

```
> sudo fdisk /dev/sda
```

```
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xcc5834ef.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
```

```
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
```

```
Command (m for help):
```

→ wir sind nun in einer Kommandozeilenumgebung von fdisk, um die Partitionierung von /dev/sda durchzuführen.

Festplatte partitionieren

Anlegen einer primären Partition

`fdisk` wird über Tastatureingaben wie unten gezeigt gesteuert; man kann die Defaults auch mit der Eingabetaste übernehmen:

- ▶ **n** (new - neue Partition anlegen)
- ▶ **p** (primäre Partition erzeugen)
- ▶ **1** (es gibt 1-4 primäre Part.; wir wollen die erste)
- ▶ **2048** (Voreinstellung für Start-Sektor übernehmen)
- ▶ **+8G** (die Partition soll 8 GByte groß werden)

```
Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-156301487, default 2048): 2048
Last sector, +sectors or +size[K,M,G] (2048-156301487, default 156301487): +8G
```


Festplatte partitionieren

Anzeigen der vorhandenen Partitionen

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Vertrauen ist gut, Kontrolle ist besser.
Man sollte das Ergebnis immer kontrollieren:

- ▶ **p** (print - Partitionstabelle ausgeben)

Command (m for help): p

```
Disk /dev/sda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders, total 156301488 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2b063ee1
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		2048	16779263	8388608	83	Linux

- ▶ Der End-Wert kann abweichen, wenn Ihr eine Festplatte mit anderer Größe habt.

Festplatte partitionieren

Partition als bootfähig markieren

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Manche BIOS-Versionen booten die Partition nur, wenn sie entsprechend markiert ist:

- ▶ **a** (activate - Partition bootfähig machen)
- ▶ **1** (erste Partition auswählen)
- ▶ **p** (print - Partitionstabelle ausgeben)

```
Command (m for help): a  
Partition number (1-4): 1
```

```
Command (m for help): p
```

```
Disk /dev/sda: 80.0 GB, 80026361856 bytes  
255 heads, 63 sectors/track, 9729 cylinders, total 156301488 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk identifier: 0x2b063ee1
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	16779263	8388608	83	Linux

- ▶ Der Stern unter "Boot" ist der Unterschied.

Festplatte partitionieren

Partitionstabelle auf Festplatte schreiben

Praxis Linux-
Administration

Carsten
Grönlisch

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

- ▶ **w** (write - Partitionstabelle schreiben)

```
Command (m for help): w  
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.  
Syncing disks.
```

- ▶ erst jetzt wurden Daten auf der Festplatte verändert.

Eine Linux-Installation benötigt im Wesentlichen:

- ▶ ein Dateisystem (in einer oder mehreren Partitionen)
- ▶ einen Bootloader an einer (vom BIOS) vordefinierten Stelle

Unser Vorgehen:

- ▶ Dateisystem anlegen
- ▶ Dateisystem mit Linux-Installation füllen
- ▶ einige Dinge im Dateisystem vorkonfigurieren
- ▶ Bootloader installieren
- ▶ neues System anbooten

Dateisystem anlegen

Leeres Dateisystem in neuer Partition anlegen

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

**Dateisystem
anlegen**

debootstrap
chroot
/dev, /proc, /sys

fstab
locale

Kernel
Bootloader

Netzwerk
User

reboot
Tastaturlayout
Desktop

```
> sudo mkfs.ext4 -L wheezy /dev/sda1
```

- ▶ Aufpassen, die richtige Partition anzugeben!
- ▶ Wir wählen ein ext4-Dateisystem. Dies ist momentan der Standard und empfohlen.
Jeder Dateisystemtyp hat sein eigenes mkfs (z.B. gibt es auch `mkfs.xfs` und `mkfs.msdos`).
- ▶ Das Dateisystem erhält einen Namen ("wheezy"), über den wir es gleich einbinden werden.

Dateisystem anlegen

Neues Dateisystem einbinden

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

**Dateisystem
anlegen**

debootstrap
chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

```
> sudo mount LABEL=wheezy /mnt
```

- ▶ /mnt ist ein leeres Verzeichnis zum temporären Einhängen von Dateisystemen.
- ▶ Der LABEL-Mechanismus hat den Vorteil, daß man den Gerätenamen unter /dev nicht kennen muß. Sehr praktisch auch, wenn man eine Festplatte hinzusteckt/entfernt und sich dadurch die Gerätenamen durcheinander würfeln.

Dateisystem anlegen

Neues Dateisystem kennzeichnen

Praxis Linux-Administration

Carsten Gnörlich

Szenario
Motivation
Laborumgebung

Maschine aufsetzen

Partitionieren
Partitionierung

Dateisystem anlegen

debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

```
> sudo touch /mnt/unsere-neues-linux
```

Wir installieren gleich ein neues GNU/Linux unter /mnt und werden darin arbeiten. Anhand der Datei `unsere-neues-linux` können wir erkennen, wann wir in der neuen Installation sind.

```
> ls -la /mnt
drwx----- 2 root root 16384 Mär 14 2013 lost+found
-rw-r--r--  1 root root      0 Apr 25 22:01 unsere-neues-linux
```

- ▶ jedes Dateisystem hat im Root-Verzeichnis einen `lost+found`-Ordner, der bei der `disaster-recovery` mittels `fsck` benutzt wird. Am besten läßt man ihn bis dahin in Ruhe.

Dateisystem anlegen

Dateien und Softwarepakete für ein minimales GNU/Linux-System nach /mnt kopieren

Praxis Linux-
Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

```
> sudo debootstrap --arch amd64 wheezy /mnt  
http://moehne/debian
```

Mindestens erforderliche Parameter:

- ▶ `--arch` - gewünschte Hardwarearchitektur.
Typischerweise `i386` oder `amd64` für 32/64bit-x86-Systeme.
- ▶ `Distribution` - Name der gewünschten GNU/Linux-Distribution.
z.B. `wheezy` für Debian Wheezy
oder `trusty` für Ubuntu Trusty Tahr.
- ▶ `Zielverzeichnis` - Verzeichnis zur Installation.
in unserem Fall `/mnt`.
- ▶ `Spiegelserver`, von dem die Softwarepakete geholt werden.
In der Laborumgebung `http://moehne/debian`; im
allgemeinen Fall `http://ftp.de.debian.org/debian`.

Dateisystem anlegen

Installiertes GNU/Linux anschauen

Praxis Linux-Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen

debootstrap

chroot
/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

```
> sudo debootstrap --arch amd64 wheezy /mnt http://moehne/debian
```

```
...
```

```
I: Base system installed successfully.
```

```
> ls -la /mnt
```

```
insgesamt 104
```

```
drwxr-xr-x 23 root root 4096 Apr 16 17:59 .
```

```
drwxr-xr-x 23 root root 4096 Apr 11 11:58 ..
```

```
drwxr-xr-x 2 root root 4096 Apr 16 17:59 bin
```

```
drwxr-xr-x 2 root root 4096 Jan 28 11:53 boot
```

```
...
```

```
-rw-r--r-- 1 root root 0 Apr 16 17:54 unser-neues-linux
```

```
drwxr-xr-x 10 root root 4096 Apr 16 17:57 usr
```

```
drwxr-xr-x 11 root root 4096 Apr 16 17:57 var
```

- ▶ Das neue GNU/Linux-System benötigt noch einige Konfigurationen und Zusatzpakete, bevor es booten kann.
- ▶ Wie arbeiten wir in dem neuen System, wenn es noch nicht booten kann?
(Hinweis: Der gleiche Weg funktioniert auch bei einer disaster recovery / mit einem rescue-System)

> `sudo chroot Verzeichnis Programm`

chroot führt ein Programm so aus, daß dessen Root-Verzeichnis nicht das normale /, sondern das angegebene Verzeichnis ist.

Für das Programm sieht es so aus, als wäre das angegebene Verzeichnis /.

Nutzen dieser Aktion:

- ▶ Einsperren von gefährdeten Programmen, z.B. apache, in ein eigenes Root-Verzeichnis
→ ein kompromittierter Apache kann so trotz Root-Rechten auf keine Dateien außerhalb des angegebenen Verzeichnisses zugreifen.
- ▶ Arbeiten in mit debootstrap erzeugten GNU/Linux-Installationen.

chroot

chroot kennenlernen

Praxis Linux-
Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Um mit `chroot` arbeiten zu können, müssen alle für das Programm wichtigen Komponenten in das Unterverzeichnis kopiert werden:

- ▶ das Programm selbst
- ▶ die benötigten Systembibliotheken
- ▶ ggf. weitere Hilfsprogramme, z.B. aus `/usr`
- ▶ ggf. Konfigurationsdateien aus `/etc`

chroot

chroot-Beispiel

Praxis Linux-
Administration

Carsten
Grörllich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Wir wollen eine chroot-Umgebung bauen, die nur eine `bash` und das `ls`-Kommando enthält.

Da wir für die nächsten Befehle Root-Rechte benötigen, holen wir uns zunächst eine Root-Shell:

```
> sudo -i
```

Die chroot-Umgebung soll im Verzeichnis `/mnt/x` liegen.
Wir legen `/mnt/x` sowie gleich auch `/mnt/x/bin` an.

```
# cd /mnt  
# mkdir -p x/bin  
# cd x  
# ls -la
```

chroot

chroot-Beispiel

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Die chroot-Umgebung soll die `bash` und das `ls`-Kommando enthalten:

```
# cp /bin/bash /bin/ls bin
```

Welche Systembibliotheken diese Programme benötigen, findet man mit `ldd` heraus:

```
# ldd /bin/bash
linux-vdso.so.1 => (0x0000ffff41fff000)
libtinfo.so.5 => /lib/x86_64-linux-gnu/libtinfo.so.5 (0x00007f454cc67000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f454ca63000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f454c6d7000)
/lib64/ld-linux-x86-64.so.2 (0x00007f454cea1000)
```

Man könnte diese Bibliotheken jetzt einzeln in die chroot-Umgebung kopieren (analog für das Programm `ls`).

chroot

chroot-Beispiel

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Um die Sache abzukürzen, kopieren wir einfach alle Systembibliotheken:

```
# cp -r /lib .  
# cp -r /lib64 .
```

Außerdem markieren wir unsere chroot-Umgebung:

```
# touch dies-ist-mnt-x
```

chroot

chroot-Beispiel

Praxis Linux-Administration

Carsten Gnörlich

Szenario
Motivation
Laborumgebung

Maschine aufsetzen

Partitionieren
Partitionierung
Dateisystem anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Wir starten jetzt eine `bash` in der `chroot`-Umgebung und probieren sie aus:

```
# chroot /mnt/x bin/bash -i
bash-4.2# cd /
bash-4.2# ls -la
total 20
drwxr-xr-x  5 0 0 4096 Apr 17 12:00 .
drwxr-xr-x  5 0 0 4096 Apr 17 12:00 ..
drwxr-xr-x  2 0 0 4096 Apr 17 11:59 bin
-rw-r--r--  1 0 0     0 Apr 17 12:00 dies-ist-mnt-x
drwxr-xr-x 16 0 0 4096 Apr 17 12:00 lib
drwxr-xr-x  2 0 0 4096 Apr 17 12:00 lib64
```

- ▶ `cd` ist eine eingebaute Funktion der `bash` und muß daher nicht als Programm vorhanden sein.

chroot

chroot-Beispiel

Praxis Linux-
Administration

Carsten
Grönlisch

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

- ▶ Versucht nun aus der chroot auszubrechen, also auf das übrige Dateisystem zuzugreifen!
- ▶ Mit Strg-D beendet Ihr die `bash` in der chroot-Umgebung
→ Rückfall in die normale Root-Shell
- ▶ Nicht vergessen, aufzuräumen:
`rm -rf /mnt/x`

chroot

chroot-Beispiel

Praxis Linux-Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

- ▶ Versucht nun aus der chroot auszubrechen, also auf das übrige Dateisystem zuzugreifen!
- ▶ Mit Strg-D beendet Ihr die `bash` in der chroot-Umgebung
→ Rückfall in die normale Root-Shell
- ▶ Nicht vergessen, aufzuräumen:
`rm -rf /mnt/x`

Arbeiten an der Installation vorbereiten

/dev und andere Kernel-Dateisysteme

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Bevor wir mit `chroot` in unsere Installation in `/mnt` wechseln, sind noch einige Sachen vorzubereiten:

```
# cd /mnt
# ls -la /dev
drwxr-xr-x 16 root root          3300 Apr 26 20:17 .
drwxr-xr-x 24 root root          4096 Apr 25 22:01 ..
crw-rw---T  1 root video       10, 175 Apr 26 20:17 agpgart
crw-----T  1 root root        10, 235 Apr 26 20:17 autofs
drwxr-xr-x  2 root root          300 Apr 26 20:17 block
...
# ls -la /mnt/dev
drwxr-xr-x  2 root root 4096 Dez 14 2012 .
drwxr-xr-x 24 root root 4096 Apr 25 22:01 ..
```

Ebenso sind `/mnt/proc` und `/mnt/sys` leer.

- ▶ `/dev`, `/proc`, `/sys` füllt der laufende Kernel.
- ▶ man darf sie nicht durch Kopieren von Dateien erzeugen.

Arbeiten an der Installation vorbereiten

/dev und andere Kernel-Dateisysteme

Praxis Linux-Administration

Carsten Gnörlich

Szenario

Motivation
Laborumgebung

Maschine aufsetzen

Partitionieren
Partitionierung

Dateisystem anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Erzeugen von /dev, /proc und /sys in /mnt:

```
# mount -t devtmpfs dev /mnt/dev
# mount -t proc proc /mnt/proc
# mount -t sysfs sysfs /mnt/sys
# ls -la /mnt/dev
drwxr-xr-x 16 root root          3300 Apr 26 20:17 .
drwxr-xr-x 24 root root          4096 Apr 25 22:01 ..
crw-rw----T  1 root video      10, 175 Apr 26 20:17 agpgart
crw-----T   1 root root       10, 235 Apr 26 20:17 autofs
drwxr-xr-x  2 root root           300 Apr 26 20:17 block
...
```

- ▶ ohne diese Kernel-Dateisysteme gäbe es in der chroot gleich Probleme.

Arbeiten an der Installation vorbereiten

Einige andere Dateien kopieren

Wir müssen einige Dateien in `/mnt` anlegen. Es ist einfacher, diese aus dem laufenden GNU/Linux zu übernehmen und anzupassen:

```
# cp /etc/apt/sources.list /mnt/etc/apt
# cp /etc/fstab /mnt/etc
```

Installation vervollständigen

chroot nach /mnt

Praxis Linux-
Administration

Carsten
Grölich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Wir können nun mit `chroot` nach `/mnt` wechseln. Alle Konfigurationen, die wir nun vornehmen, betreffen die neue GNU/Linux-Installation und nicht das laufende System.

```
# chroot /mnt # wird kein Programm angegeben, startet eine bash
# cd /
# ls -la
drwxr-xr-x 24 root root 4096 Apr 25 22:01 .
drwxr-xr-x 24 root root 4096 Apr 25 22:01 ..
drwxr-xr-x  2 root root 4096 Dez 12 14:22 bin
...
-rw-r--r--  1 root root    0 Apr 17 12:00 user-neues-linux
...
# ls /proc
...
```

Installation vervollständigen

/etc/fstab konfigurieren

Die neue Installation verhält sich fehlerhaft, wenn die benötigten Dateisysteme nicht in der /etc/fstab eingetragen sind. Also modifizieren wir die eben kopierte /etc/fstab wie folgt:

```
# vi /etc/fstab

# /etc/fstab: static file system information.
...
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
LABEL=wheezy / ext4 errors=remount-ro 0 1
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/sr1 /media/cdrom1 udf,iso9660 user,noauto 0 0
```

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Installation vervollständigen

locale nachinstallieren und konfigurieren

Praxis Linux-
Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale

Kernel
Bootloader

Netzwerk
User

reboot
Tastaturlayout

Desktop

Solange die “locale” (= Ländereinstellungen) nicht gesetzt ist, wird es massenweise Warnungen geben. Also aktualisieren wir die Paketlisten und installieren die locale nach:

```
# apt-get update
# apt-get install locales
# dpkg-reconfigure locales
```

`dpkg-reconfigure locales` startet einen Dialog zur Auswahl der gewünschten locales. Empfohlen ist die Auswahl der folgenden beiden:

- ▶ `de_DE.UTF-8`
- ▶ `en_US.UTF-8`

Als Default sollte man die festlegen, die den eigenen Gewohnheiten entspricht.

Installation vervollständigen

Kernel installieren

Das minimale System verfügt noch über keinen Kernel, da es Szenarien gibt, in denen dieser “von außen” zur Verfügung gestellt wird (z.B. durch `chroot` oder in speziell konfigurierten VMs).

Wir installieren den 64bit-Kernel:

```
# apt-get install linux-image-amd64
```

Praxis Linux-
Administration

Carsten
Grölich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Installation vervollständigen

Bootloader nachinstallieren

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Die neue Installation benötigt außerdem einen Bootloader. Wir installieren den *Grub2*:

```
# apt-get install grub2
```

Es erscheint ein Dialog, der nach dem Gerät fragt, in das der Bootloader installiert werden soll. Wir wählen `/dev/sda`.

(hier hätte es sich jetzt gerächt wenn wir `/dev` nicht gemountet hätten!)

Installation vervollständigen

Netzwerk konfigurieren

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader

Netzwerk

User
reboot
Tastaturlayout
Desktop

Wenn die neue Installation nach dem Booten das Netzwerkinterface nicht hochfährt, können wir uns per `ssh` nicht anmelden und müssen in den Keller laufen.

Also ergänzen wir:

```
# vi /etc/network/interfaces
...
auto eth0
iface eth0 inet dhcp
```

Installation vervollständigen

User anlegen

Praxis Linux-
Administration

Carsten
Grörllich

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen

debootstrap
chroot

/dev, /proc, /sys
fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Wir müssen zumindest den User `root` aktivieren, indem wir ihm ein Paßwort geben. Außerdem legen wir einen weiteren User `linux` an und geben ihm `sudo`-Rechte:

```
# passwd root
```

```
# addgroup --gid 1000 linux
```

```
# adduser --uid 1000 --gid 1000 linux
```

(dem Dialog folgen)

```
# apt-get install sudo
```

```
# addgroup linux sudo
```

Installation testen

System das erste Mal starten

Wir können das neue System nun starten:

- ▶ `chroot` mit Strg-D verlassen
- ▶ `sudo reboot`
- ▶ wenn das System im BIOS ist, USB-Stick abziehen
- ▶ nach dem Neustart F12 für BIOS-Bootmenu drücken
- ▶ im Bootmenü die SATA-Festplatte (SATA0-ST...) wählen

Wenn wir vorher alles richtig gemacht haben, wird das neue System starten und wir können uns als User `linux` mit dem vergebenen Paßwort anmelden.

Installation testen

Tastaturlayout einstellen

Praxis Linux-
Administration

Carsten
Grönlisch

Szenario

Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung

Dateisystem
anlegen

debootstrap

chroot

/dev, /proc, /sys

fstab

locale

Kernel

Bootloader

Netzwerk

User

reboot

Tastaturlayout

Desktop

Wir haben nun ein amerikanisches Tastaturlayout. Dies haben wir in der chroot nicht bemerkt, da diese das deutsche Layout von dem System auf dem Memorystick übernommen hat.

- ▶ `sudo apt-get install console-data`
 - ▶ im Dialog die Tastaturbelegung aus der Liste für diese Architektur wählen:
 - ▶ qwertz
 - ▶ German
- ▶ `sudo apt-get install keyboard-configuration`
 - ▶ Deutsch

Installation testen

graphische Oberfläche installieren

Praxis Linux-
Administration

Carsten
Grölich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys

fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Auf einem Server, der nur per `ssh` erreichbar ist, hat eine graphische Oberfläche keinen Sinn. Da wir jedoch einen Bildschirm angeschlossen haben, können wir es uns bequemer machen:

- ▶ `sudo apt-get install xfce4 lightdm`
- ▶ `sudo service lightdm start`

Ende der heutigen Vorlesung

Praxis Linux-
Administration

Carsten
Gnörlich

Szenario
Motivation
Laborumgebung

Maschine
aufsetzen

Partitionieren
Partitionierung
Dateisystem
anlegen
debootstrap
chroot
/dev, /proc, /sys
fstab
locale
Kernel
Bootloader
Netzwerk
User
reboot
Tastaturlayout
Desktop

Vielen Dank fürs Mitmachen!

Bis nächste Woche!