



1.104.1-3 Geräte, Dateisysteme u. deren Integrität, FHS, Mounten und Unmounten 1.104.6 Harte und symbolische Links



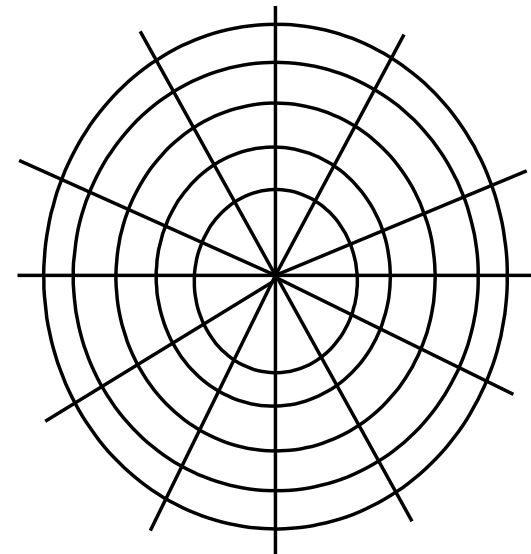
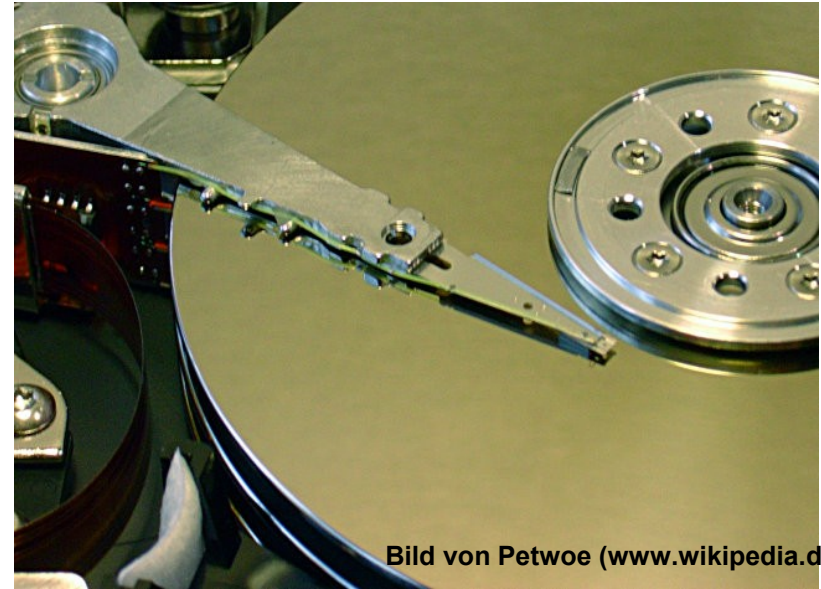
**Für LPIC1-101
ab 1.4.2009**

www.lpi.org



Copyright (©) 2006 by [W. Kicherer](#), 2009 by [A. Grupp](#). This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 2.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

- Festplattenaufbau CHS
 - Zylindern (C)
 - Köpfen (H)
 - Sektoren (S) à 512 Bytes
 - Heute normalerweise LBA (Logical Block Addressing)
 - Sektoren sind durchnummeriert
- Partitionierung
 - Durch die Partitionierung wird die Festplatte in kleinere, logische Teile aufgeteilt





- Wie alle Geräte unter Linux als „Device“ bezeichnet. Devices im Dateibaum unter **/dev**
- SCSI- und SATA-Festplatten mit „**sd**“ + Buchstaben
Bsp.: **/dev/sda**, **/dev/sdb**, usw.
- IDE-Festplatten mit „**hd**“ + fortlaufender Buchstaben (bei Verwendung des älteren IDE-Treibersystems)
Bsp.: **/dev/hda**, **/dev/hdb**, usw.
- Mit „libata-PATA“-Treiber (heute meist üblich) auch IDE-Platten mit „**sd**“ + Buchstaben



- Wahlfreier Zugriff auf einzelne Blöcke (Sektoren) möglich.
- Der verfügbare Platz muss in voneinander unabhängige logische Bereiche aufgeteilt werden (min. 1)
=> Partition(en)
- Zur Verwaltung der Partitionen wird ein gesonderter Bereich benötigt
=> Master Boot Record (MBR)

Beispiel eines MBR



Auf dem Weg:
Certified Linux



```
r-andreas:~ # hexdump -n 512 -C /dev/sda
```

```
00000000 eb 48 90 d0 bc 00 7c fb 50 07 50 1f fc be 1b 7c |.H....|.P.P....|
00000010 bf 1b 06 50 57 b9 e5 01 f3 a4 cb bd be 07 b1 04 |...PW.....|
00000020 38 6e 00 7c 09 75 13 83 c5 10 e2 f4 cd 18 8b f5 |8n.|.u.....|
00000030 83 c6 10 49 74 19 38 2c 74 f6 a0 b5 07 b4 03 02 |...It.8,t.....|
00000040 ff 00 00 20 01 00 00 00 00 02 fa 90 90 f6 c2 80 |... ..|
00000050 75 02 b2 80 ea 59 7c 00 00 31 c0 8e d8 8e d0 bc |u....Y|..1.....|
00000060 00 20 fb a0 40 7c 3c ff 74 02 88 c2 52 be 81 7d |. ..@|<.t...R..}|
00000070 e8 36 01 f6 c2 80 74 56 b4 41 bb aa 55 cd 13 5a |.6....tV.A..U..Z|
00000080 52 72 4b 81 fb 55 aa 75 45 a0 41 7c 84 c0 78 3e |RrK..U.uE.A|..x>|
00000090 75 05 83 e1 01 74 37 66 8b 4c 10 be 05 7c c6 44 |u....t7f.L...|.D|
000000a0 ff 01 66 8b 1e 44 7c c7 04 10 00 c7 44 02 01 00 |..f..D|.....D...|
000000b0 66 89 5c 08 c7 44 06 00 70 66 31 c0 89 44 04 66 |f.\..D..pf1..D.f|
000000c0 89 44 0c b4 42 cd 13 72 05 bb 00 70 eb 7d b4 08 |.D..B..r...p..}|
000000d0 cd 13 73 0a f6 c2 80 0f 84 e8 00 e9 8d 00 be 05 |..s.....|
000000e0 7c c6 44 ff 00 66 31 c0 88 f0 40 66 89 44 04 31 ||.D..f1...@f.D.1|
000000f0 d2 88 ca c1 e2 02 88 e8 88 f4 40 89 44 08 31 c0 |.....@.D.1.|
00000100 88 d0 c0 e8 02 66 89 04 66 a1 44 7c 66 31 d2 66 |.....f..f.D|f1.f|
00000110 f7 34 88 54 0a 66 31 d2 66 f7 74 04 88 54 0b 89 |.4.T.f1.f.t..T..|
00000120 44 0c 3b 44 08 7d 3c 8a 54 0d c0 e2 06 8a 4c 0a |D.;D.}<.T....L.|
00000130 fe c1 08 d1 8a 6c 0c 5a 8a 74 0b bb 00 70 8e c3 |.....l.Z.t...p..|
00000140 31 db b8 01 02 cd 13 72 2a 8c c3 8e 06 48 7c 60 |1.....r*....H|`|
00000150 1e b9 00 01 8e db 31 f6 31 ff fc f3 a5 1f 61 ff |.....1.1.....a.|
00000160 26 42 7c be 87 7d e8 40 00 eb 0e be 8c 7d e8 38 |&B|...}.@.....}.8|
00000170 00 eb 06 be 96 7d e8 30 00 be 9b 7d e8 2a 00 eb |.....}.0....}.*..|
00000180 fe 47 52 55 42 20 00 47 65 6f 6d 00 48 61 72 64 |.GRUB .Geom.Hard|
00000190 20 44 69 73 6b 00 52 65 61 64 00 20 45 72 72 6f | Disk.Read. Erro|
000001a0 72 00 bb 01 00 b4 0e cd 10 ac 3c 00 75 f4 c3 00 |r.....<.u...|
000001b0 00 00 00 00 00 00 00 00 15 c5 15 c5 00 00 80 01 |.....|
000001c0 01 00 07 fe ff ff 3f 00 00 00 27 79 1a 06 00 fe |.....?'y....|
000001d0 ff ff 07 fe ff ff 66 79 1a 06 bf fa 3f 06 00 fe |.....fy....'?...|
000001e0 ff ff 05 fe ff ff 25 74 5a 0c 9c 62 e8 18 00 00 |.....%tZ..b....|
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa |.....U.|
```

512 Byte mit:

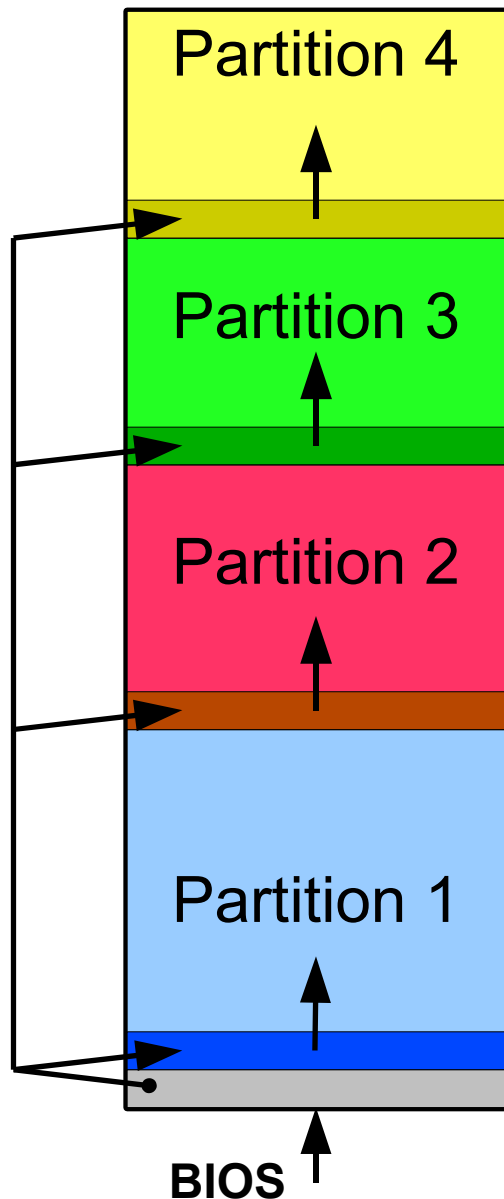
**446 Byte
Bootcode**

**4 Partitions-
einträgen
(07=NTFS,
05=Extend.)**

**und Magic
Word
0xAA55**

- Ebenfalls im MBR → Typ der Partition
- Hier Liste der Part.-Typen von „cfdisk“

01 FAT12	24 NEC DOS	81 Minix / old Linux	C1 DRDOS/sec (FAT-12)
02 XENIX root	39 Plan 9	82 Linux swap / Solaris	C4 DRDOS/sec (FAT-16 <
03 XENIX usr	3C PartitionMagic recov	83 Linux	C6 DRDOS/sec (FAT-16)
04 FAT16 <32M	40 Venix 80286	84 OS/2 hidden C: drive	C7 Syrinx
05 Extended	41 PPC PReP Boot	85 Linux extended	DA Non-FS data
06 FAT16	42 SFS	86 NTFS volume set	DB CP/M / CTOS / ...
07 HPFS/NTFS	4D QNX4.x	87 NTFS volume set	DE Dell Utility
08 AIX	4E QNX4.x 2nd part	88 Linux plaintext	DF BootIt
09 AIX bootable	4F QNX4.x 3rd part	8E Linux LVM	E1 DOS access
0A OS/2 Boot Manager	50 OnTrack DM	93 Amoeba	E3 DOS R/O
0B W95 FAT32	51 OnTrack DM6 Aux1	94 Amoeba BBT	E4 SpeedStor
0C W95 FAT32 (LBA)	52 CP/M	9F BSD/OS	EB BeOS fs
0E W95 FAT16 (LBA)	53 OnTrack DM6 Aux3	A0 IBM Thinkpad hiberna	EE EFI GPT
0F W95 Ext'd (LBA)	54 OnTrackDM6	A5 FreeBSD	EF EFI (FAT-12/16/32)
10 OPUS	55 EZ-Drive	A6 OpenBSD	F0 Linux/PA-RISC boot
11 Hidden FAT12	56 Golden Bow	A7 NeXTSTEP	F1 SpeedStor
12 Compaq diagnostics	5C Priam Edisk	A8 Darwin UFS	F4 SpeedStor
14 Hidden FAT16 <32M	61 SpeedStor	A9 NetBSD	F2 DOS secondary
16 Hidden FAT16	63 GNU HURD or SysV	AB Darwin boot	FD Linux raid autodetec
17 Hidden HPFS/NTFS	64 Novell Netware 286	B7 BSDI fs	FE LANstep
18 AST SmartSleep	65 Novell Netware 386	B8 BSDI swap	FF BBT
1B Hidden W95 FAT32	70 DiskSecure Multi-Boo	BB Boot Wizard hidden	
1C Hidden W95 FAT32 (LB	75 PC/IX	BE Solaris boot	
1E Hidden W95 FAT16 (LB	80 Old Minix	BF Solaris	

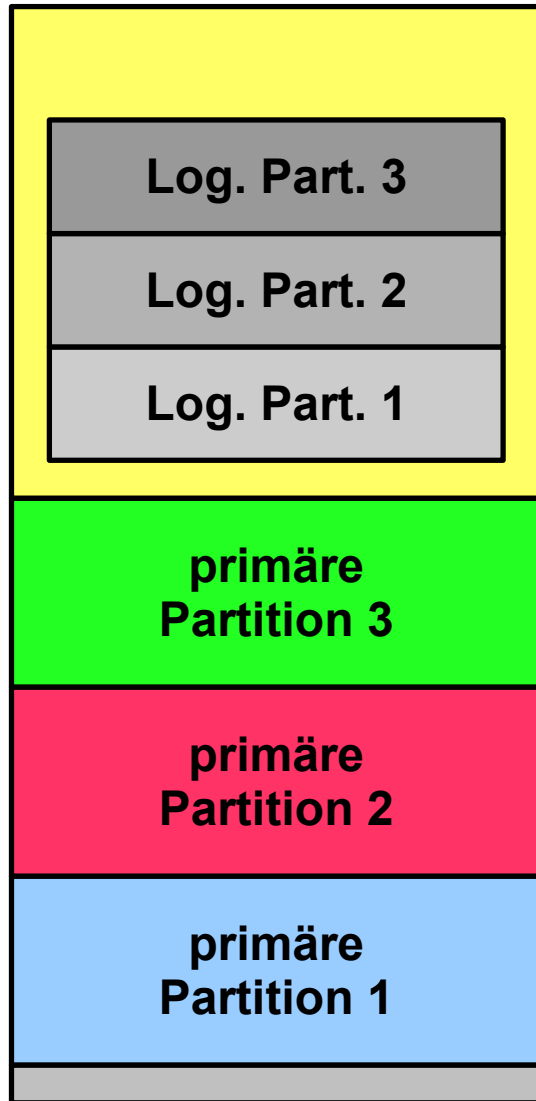


- Um mehrere Betriebssysteme booten zu können, steht als 2. Option für Boot-Code der Bootsektor der jeweiligen Partition zur Verfügung.
- Boot-Code im „Master Boot Record“ kann auch ein Bootmanager sein
⇒ Erlaubt gezielte Auswahl eines bestimmten Bootsektors
- Folge: Kaskadierung von Bootmanagern möglich!

← MBR mit Bootcode od. Bootmanager

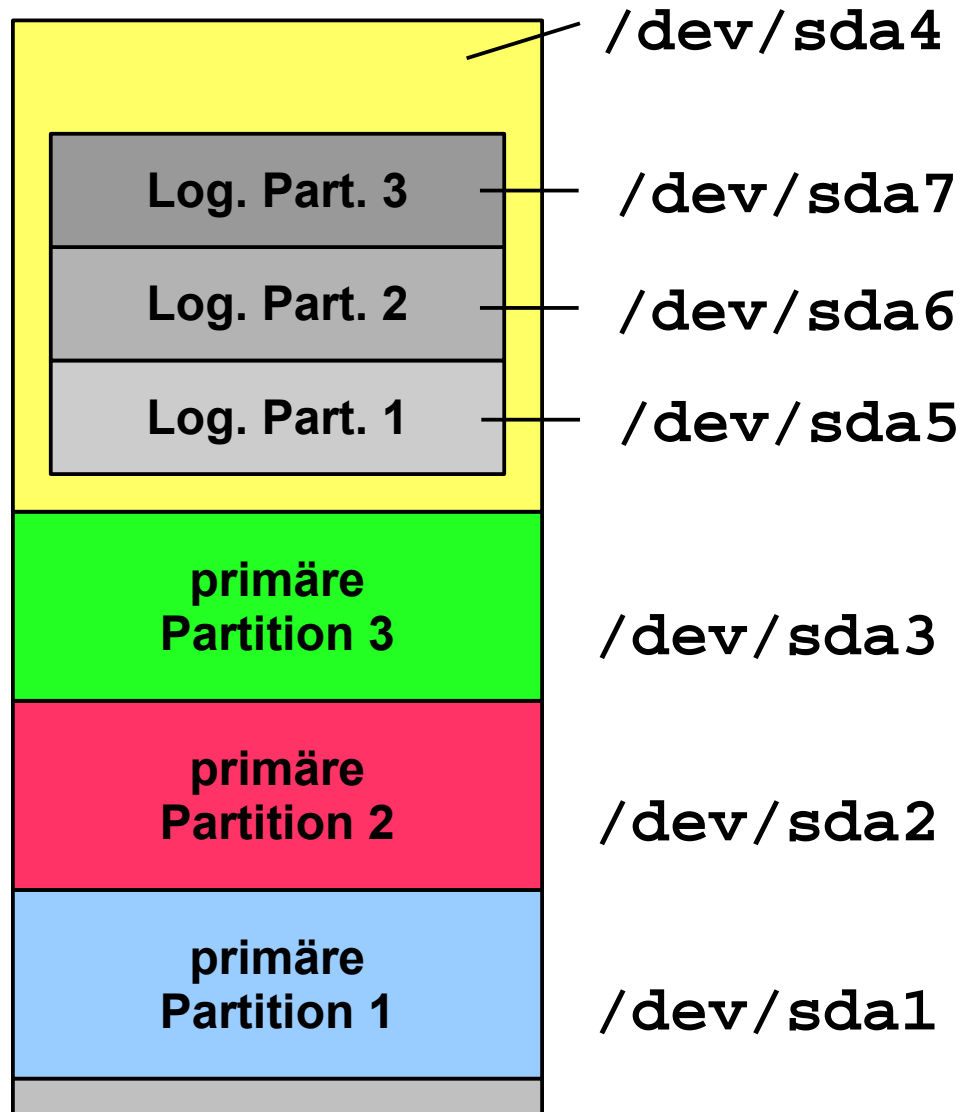


- Maximal 4 Partitionen im MBR verwaltbar:
 - Entweder 4 primäre Partitionen möglich, oder
 - ... drei primäre Partitionen und eine erweiterte Partition anlegbar.
- Erweiterte Partition dient als „Container“ für weitere Partitionen \Rightarrow „logische Partitionen“
- (E)IDE-Platten: max. 63 Partitionen
- SCSI-Platten: max. 15 Partitionen, z.B.
 - SCSI 8 Geräte (inkl. Controller!)
 - Wide SCSI 16 Geräte (inkl. Controller!)
 - Ultra3 SCSI 16 Geräte (inkl. Controller!)

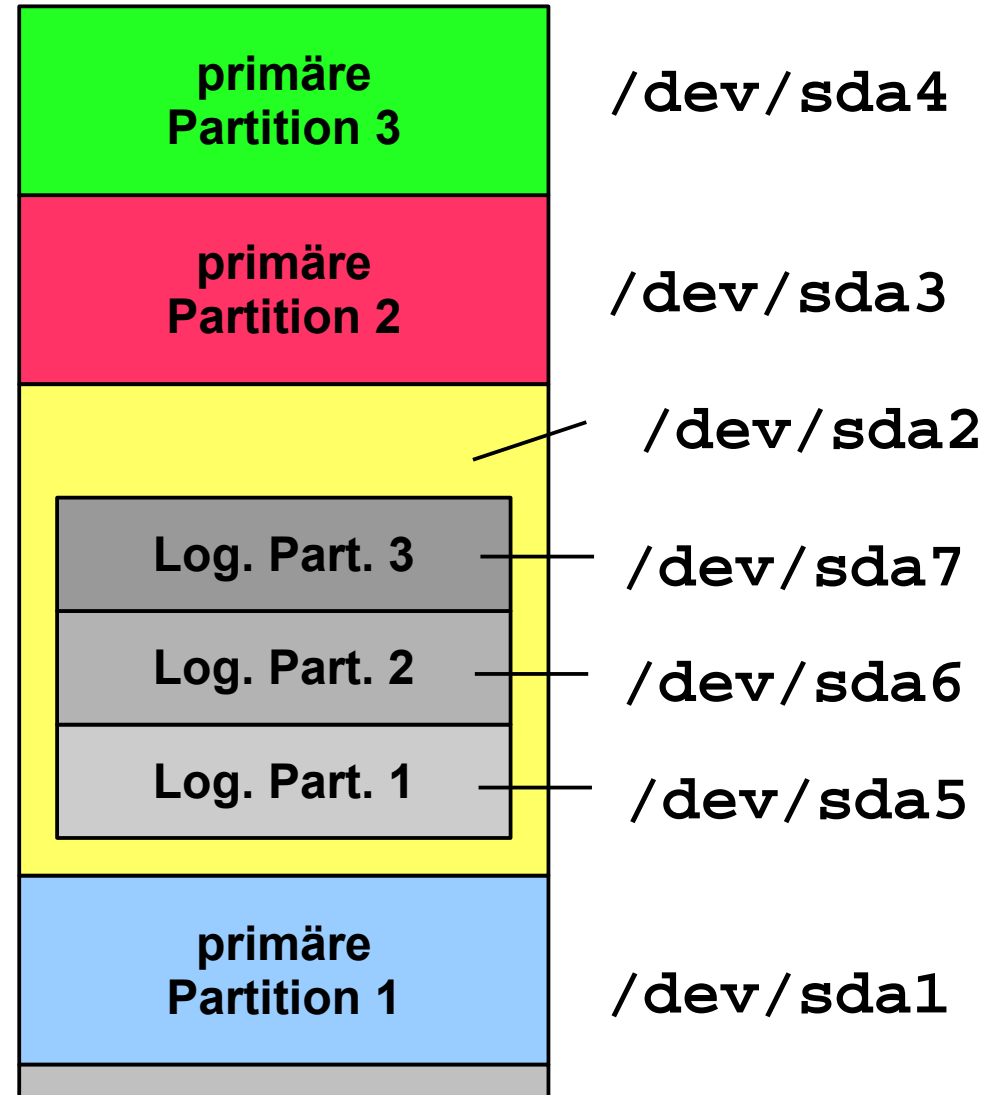


- Log. Partitionen befinden sich also alle innerhalb der einen möglichen erweiterten Partition.
- Bezeichnungen unter Linux:
 - Primäre Partitionen (bzw. erweiterte Partition) sind
`/dev/sda1 ... /dev/sda4`
 - Logische Partitionen starten immer erst ab `/dev/sda5` (auch wenn die Partition `/dev/sda4` usw. gar nicht existiert).

Bsp.: Devicenamen der Partitionen



Andere Variante:



Partitionierungs-“Altvater“: fdisk



Auf dem Weg:
Certified Linux



```
r-andreas:~ # fdisk /dev/sda
```

Die Anzahl der Zylinder für diese Platte ist auf 38913 gesetzt.
Daran ist nichts verkehrt, aber das ist größer als 1024 und kann
in bestimmten Konfigurationen Probleme hervorrufen mit:

- 1) Software, die zum Bootzeitpunkt läuft (z. B. ältere LILO-Versionen)
- 2) Boot- und Partitionierungssoftware anderer Betriebssysteme
(z. B. DOS FDISK, OS/2 FDISK)

```
Befehl (m für Hilfe): p
```

```
Platte /dev/sda: 320.0 GByte, 320072933376 Byte  
255 Köpfe, 63 Sektoren/Spuren, 38913 Zylinder  
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes  
Disk identifiziert: 0xc515c515
```

Gerät	boot.	Anfang	Ende	Blöcke	Id	System
/dev/sda1	*	1	6374	51199123+	7	HPFS/NTFS
/dev/sda2		6375	12901	52428127+	7	HPFS/NTFS
/dev/sda3		12902	38913	208941390	5	Erweiterte
/dev/sda5		12902	13554	5245191	82	Linux Swap / Solaris
/dev/sda6		13555	16818	26218048+	83	Linux
/dev/sda7		16819	25956	73400953+	83	Linux

```
Befehl (m für Hilfe): m
```

```
Befehl  Bedeutung  
a      (De)Aktivieren des bootfähig-Flags  
b      „bsd disklabel“ bearbeiten  
c      (De)Aktivieren des DOS Kompatibilitätsflags  
d      Eine Partition löschen  
l      Die bekannten Dateisystemtypen anzeigen  
m      Dieses Menü anzeigen  
n      Eine neue Partition anlegen  
....
```



Steuerung über Cursor-Tasten

cfdisk (util-linux-ng 2.13.1)

Festplatte: /dev/sda

Größe: 320072933376 Bytes, 320,0 GB

Köpfe: 255 Sektoren pro Spur: 63 Zylinder: 38913

Name	Flags	Part. Typ	Dateisystemtyp	[Bezeichner]	Größe (MB)
sda1	Boot	Primäre	NTFS	[^F]	52427,94
sda2		Primäre	NTFS	[^F]	53686,41
sda5		Logische	Linux swap / Solaris		5371,11
sda6		Logische	Linux ReiserFS		26847,32
sda7		Logische	Linux ext3		75162,61
		Logische	Freier Bereich		106574,96

[Bootbar] [Löschen] [Hilfe] [Maxim.] [Ausgabe]
[Ende] [Typ] [Einheit.] [Schreib.]

(De)Aktivieren des bootfähig-flags der aktuellen Partition

DERZEIT NICHT LPIC-RELEVANT (obwohl meist vorhanden)

- Organisation von Verzeichnissen / Dateien
- Eigentlich unabhängig vom Partitionstyp!!!
- Vielzahl an Dateisystemen verfügbar (ext2, ext3, ReiserFS, xfs, vfat, ...)
- Derzeit gängigstes Linux-Dateisystem ist „ext3“
 - Entspricht „ext2“ + Journal-System (`tune2fs -j`)
 - Erstellung mit `mkfs` (binär identisch oder evtl. Hardlink zu `mkfs.ext2` bzw. 3 od. `mke2fs`)
 - Änderungen mit `tune2fs`
 - Default-Journal-Verhalten: ordered (Transaktion der Metadaten nach erfolgreichem schreiben der Daten)



Syntax:

mkfs -t *typ* *device*

Formatiert das Gerät *device*.

typ ext2, ext3, minix, xfs, vfat, ... (Abhängig von installierten Paketen)

device Gerätedatei (z.B. `/dev/hda3`)

Voraussetzungen

mkfs verwendet im Hintergrund **mkfs.ext2**, **mkfs.xfs**, Sie sind also Voraussetzung für die Arbeit von **mkfs**.

Optionen (abhängig vom Dateisystem)

- v Verbose
- c check, testet nach defekten Blöcken, diese werden gesperrt
- L Label, setzt ein Label für das Gerät (mke2fs)
- n Label, setzt ein Label für das Gerät (mkdosfs)



```
root@r-vm-ubu810:/sbin# ls -l mk*
-rwxr-xr-x 1 root root 26632 2008-07-01 18:23 mkdosfs
-rwxr-xr-x 1 root root 50976 2008-10-13 15:10 mke2fs
-rwxr-xr-x 1 root root 5484 2008-09-25 15:08 mkfs
-rwxr-xr-x 1 root root 9600 2008-09-25 15:08 mkfs.bfs
-rwxr-xr-x 1 root root 17852 2008-09-25 15:08 mkfs.cramfs
-rwxr-xr-x 1 root root 50976 2008-10-13 15:10 mkfs.ext2
-rwxr-xr-x 1 root root 50976 2008-10-13 15:10 mkfs.ext3
-rwxr-xr-x 1 root root 50976 2008-10-13 15:10 mkfs.ext4
-rwxr-xr-x 1 root root 50976 2008-10-13 15:10 mkfs.ext4dev
-rwxr-xr-x 1 root root 19000 2008-09-25 15:08 mkfs.minix
lrwxrwxrwx 1 root root 7 2009-04-17 15:39 mkfs.msdos -> mkdosfs
-rwxr-xr-x 1 root root 146460 2007-07-26 12:57 mkfs.reiserfs
lrwxrwxrwx 1 root root 7 2009-04-17 15:39 mkfs.vfat -> mkdosfs
-rwxr-xr-x 1 root root 333908 2008-05-03 08:41 mkfs.xfs
-rwxr-xr-x 1 root root 146460 2007-07-26 12:57 mkreiserfs
-rwxr-xr-x 1 root root 17952 2008-09-25 15:08 mkswap
root@r-vm-ubu810:/sbin#
```

- Hier Files redundant; bei anderen Distri's → Hardlinks
- Formatieren bzw. Anlegen des Dateisystems auf `/dev/sda3` durch den Befehl: `mkfs.ext3 /dev/sda3`
- Swap-Partionen mit `mkswap` initialisieren



```
user@r-vm-ubu810:~$ su -
Passwort:
root@r-vm-ubu810:~# mke2fs /dev/sda3
mke2fs 1.41.3 (12-Oct-2008)
Dateisystem-Label=
OS-Typ: Linux
Blockgröße=4096 (log=2)
Fragmentgröße=4096 (log=2)
66384 Inodes, 265072 Blöcke
13253 Blöcke (5.00%) reserviert für den Superuser
Erster Datenblock=0
Maximale Dateisystem-Blöcke=272629760
9 Blockgruppen
32768 Blöcke pro Gruppe, 32768 Fragmente pro Gruppe
7376 Inodes pro Gruppe
Superblock-Sicherungskopien gespeichert in den Blöcken:
    32768, 98304, 163840, 229376
```

Schreibe Inode-Tabellen: erledigt

Schreibe Superblöcke und Dateisystem-Accountinginformationen: erledigt

Das Dateisystem wird automatisch nach jeweils 20 Einhäng-Vorgängen bzw. alle 180 Tage überprüft, je nachdem, was zuerst eintritt. Veränderbar mit `tune2fs -c` oder `-t`.

```
root@r-vm-ubu810:~# tune2fs -j /dev/sda3
tune2fs 1.41.3 (12-Oct-2008)
```

Erstelle Journal-Inode: erledigt

Das Dateisystem wird automatisch nach jeweils 20 Einhäng-Vorgängen bzw. alle 180 Tage überprüft, je nachdem, was zuerst eintritt. Veränderbar mit `tune2fs -c` oder `-t`.

```
root@r-vm-ubu810:~# mount -t ext3 /dev/sda3 /mnt
```

- **mkfs.xfs** zum Anlegen des Dateisystems
- **fsck.xfs** nur zur Kompatibilität vorhanden. Gibt nur Hinweis auf **xfs_check/xfs_repair** aus.
- **xfs_check** reine Diagnose, Reparatur nicht gemounteter Systeme mit **xfs_repair**
- **xfs_info** gibt Geometrie des Dateisystems aus (entspricht **xfs_growfs -n**)
- **xfs_metadump** extrahiert Metadaten des Filesystems in Datei. Rein für Debugging bzw. Analyse im Fehlerfall. Sichert keine Daten! Restaurierung mit **xfs_mdrestore**
- Datendump mit **xfs_dump**, **xfs_restore**
- **xfs_metadump** u. **xfs_info** explizite für LPIC1-101 ab 1.4.2009 angegeben!??



`mkswap device`

Formatiert das Gerät *device* als Swap-Gerät
(*device* kann auch eine Datei sein, welche z.B. mit
`dd bs=1024 if=/dev/zero of=/var/tmp/swapfile
count=5120` erzeugt wurde).

`swapon [option]`
`swapon [-p prio] device`

Startet die Verwendung des Swap-Geräts.

Optionen

- a alle swap-Geräte starten (→ `/etc/fstab`)
- s Anzeige der Belegung und der Priorität
- p *prio* Priorität festlegen

- Linux kennt keine Laufwerke C:\, D:\, ...
- Es gibt im laufenden Betrieb nur **einen** Verzeichnis- und Dateibaum
- Dateibaum startet immer bei „ / “ - dem root-Verzeichnis (da dort nur der Systemadministrator „root“ Schreibrecht hat)
- Die Dateisysteme aus anderen Partitionen werden in den Hauptbaum eingehängt / gemountet
- Auch der Hauptbaum muss auf „ / “ gemountet werden
- Befehl „mount“

Dateibaum erstellen

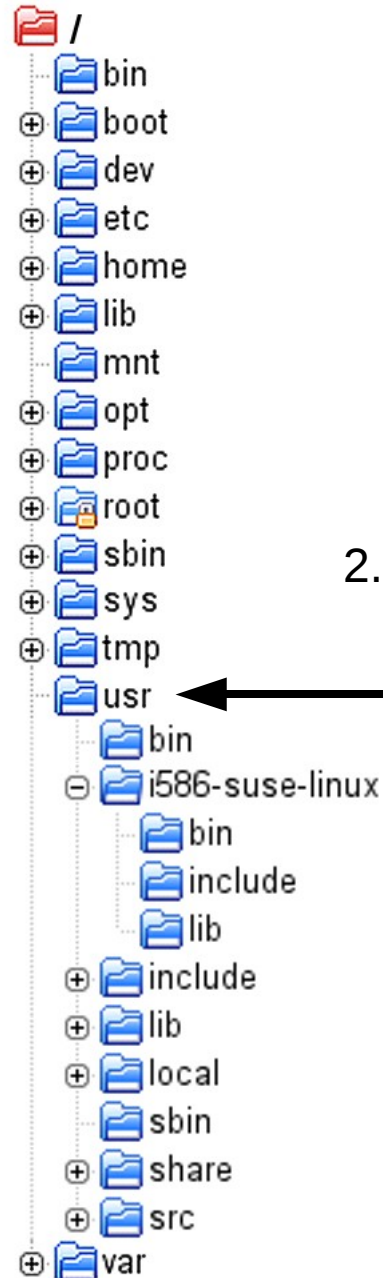


Auf dem Weg:
Certified Linux



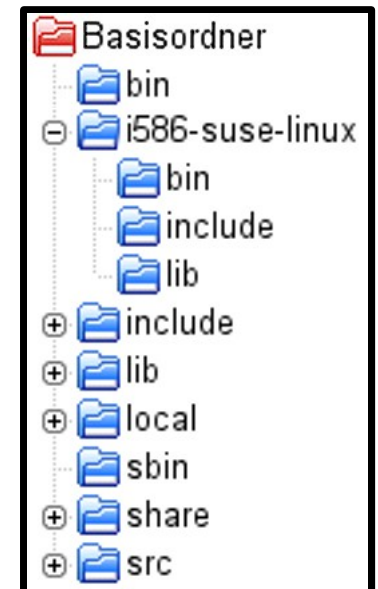
1.) `mount /dev/sda1 /`

`/dev/sda1`



2.) `mount /dev/sda2 /usr`

`/dev/sda2`



Syntax:

mount [-a]

mount [options] *device* (Eintrag in /etc/fstab vorhanden)

mount [options] *directory* (Eintrag in /etc/fstab vorhanden)

mount [options] *device directory*

Hängt das Gerät *device* im Verzeichnisbaum an die Stelle *directory*.

Optionen

-a all, alle Einträge in /etc/fstab einhängen (außer: noauto)

-h help

-r read only, nur lesende Zugriffe möglich

-v verbose

-t typ Dateisystemtyp (ext2, vfat, iso9660, nfs, swap, proc, ...)

-o *mount_opt* Mount Optionen (siehe nächste Folie), mehrere durch Komma getrennt (ohne Leerzeichen)

mount ohne weitere Angaben zeigt die derzeitig gemounteten Geräte und Einhänge-Punkte (Mount-Points).



sync	synchrone I/O (sofort auf Datenträger schreiben)
async	asynchrone I/O (verzögertes schreiben, d.h. Pufferung)
noauto	kein automatisches mounten bei mount -a
auto	Gegenteil von noauto
dev	Interpretiert Gerätedateien
nodev	Gegenteil von dev
exec	Gegenteil von noexec
noexec	Ausführen von Programmen verbieten
suid	SUID und SGID erlauben
nosuid	Gegenteil von suid
user	Ein User darf mounten. Nur der User der gemountet hat (und root) darf unmounten.
users	Ein User darf mounten, ein anderer unmounten
nouser	Gegenteil von user und users
ro	read only
rw	read write
remount	erneutes Einhängen (z.B. ro -> rw)
loop=datei	Datei wird über das Loopback-Gerät gemountet
defaults	entspricht "rw, suid, dev, exec, auto, nouser, asyn"

Syntax:

```
umount [options] device  
umount [options] directory
```

Hängt das Gerät *device* bzw. im Verzeichnisbaum an der Stelle *directory* aus.

Optionen

-a	all, alle Einträge in <code>/etc/mtab</code> aushängen
-t <i>typ</i>	Alle Partitionen vom Dateisystemtyp <i>typ</i> (ext2, vfat, iso9660, nfs, swap, proc, ...) aushängen

In der Datei `/etc/mtab` vermerkt das Betriebssystem welche Dateisysteme gerade gemountet sind.



Um auf beim Booten alle Partitionen zu mounten wird die Datei
`/etc/fstab` verwendet:

Mount Optionen

Dump (man dump)
0: kein Dump (kein ext2)
1: Dump (ext2)

Beispiel:

#Kommentarzeile					
/dev/sda1	/	ext2	defaults	1	1
/dev/sda2	/home	ext2	defaults	1	2
/dev/sda3	/usr	ext2	defaults	1	2
/dev/sda5	/var	ext2	defaults	1	2
/dev/sda6	swap	swap	defaults	0	0
/dev/hdc	/media/cdrom	iso9660	noauto,ro,users	0	0
serv1:/share	/serv1	nfs	defaults	0	0

Gerätedatei

Mount-Point

Dateisystem

`/media` – lt. FHS vorgesehener Mount-Point
für Geräte die ein User einhängen darf! Auch
Standard bei dynam. Mounts v. Gnome/KDE

fsck -A Reihenfolge
0: kein fsck
1: zuerst
2: nachfolgend

- Erfolgt automatisch während des Bootvorgangs
- Informationen für „mount“ (incl. Zusatzoptionen zum Mounten) ➔ **/etc/fstab**

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>    <type>    <options>          <dump>  <pass>
proc        /proc                proc      defaults            0        0
/dev/sda1   /                    ext3      relatime,errors=remount-ro 0    1
/dev/sda5   swap                sw        0                    0
/dev/scd0   /media/cdrom0       udf,iso9660 user,noauto,exec,utf8 0    0
/dev/fd0    /media/floppy0      auto      rw,user,noauto,exec,utf8 0    0
```

- **ACHTUNG:** Vergabe der Device-Namen kann sich durch Einfügen von Datenträgern (z.B. auch USB-Stick) ändern!!! Ältere Kerneltreiber führen zu „hd“-Devices und anders herum!

- „Besseres“ Mounten als über Devicenamen
möglichst durch eindeutige Kennungen für einen
Datenträger
 - ID: ata-WDC_WD3200JS-00PDB0_WD-WCAPD2174202-part5
 - UUID: 179330be-ba6f-4481-8dbc-2124bc95a14a
 - Pfad (Bus): pci-0000:00:1f.2-scsi-0:0:0:0-part5
 - oder Label der Partition: myroot

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>    <type>  <options>          <dump>  <pass>
proc          /proc                proc     defaults            0        0
# /dev/sda1
UUID=1774896c-bf76-4394-acd4-31febb3c6c48 / ext3  relatime,errors=remount-ro 0 1
# /dev/sda5
UUID=aa7e952f-54bb-41fa-b1ea-1a8a0d79be29 none  swap  sw                0        0
/dev/scd0      /media/cdrom0        udf,iso9660 user,noauto,exec,utf8 0        0
/dev/fd0       /media/floppy0       auto     rw,user,noauto,exec,utf8 0        0
```



Syntax:

```
df [optionen] [directories]
```

Anzeige des freien Festplattenplatzes einer Partition.

Optionen

- h** human, menschlich lesbarere Ausgabe mit der Angabe der Belegung in MB bzw. GB
- i** inode, Ausgabe der Anzahl der freien I-Nodes

Beispiel:

```
root@abc:~# df -h
```

Dateisystem	Größe	Benut	Verf	Ben%	Eingehängt auf
/dev/hdc1	14G	3,5G	9,7G	27%	/
/dev/hda1	3,0G	382M	2,5G	14%	/boot
/dev/hdc6	33G	9,7G	21G	32%	/home
/dev/hdc5	33G	18G	13G	58%	/usr/local

```
root@abc:~#
```



Syntax:

`du [optionen] [directories]`

Anzeige des belegten Festplattenplatzes in einem Verzeichnis (inkl. Unterverzeichnissen). Ohne *directories* wird nur das aktuelle Verzeichnis betrachtet!

Optionen

- `-h` human, Ausgabe mit der Angabe der Belegung in MB bzw. GB
- `-a` all, auch Dateien anzeigen
- `-c` Anzeige der Gesamtsumme
- `-s` Anzeige einer Summe für jedes angegebenes Verzeichnis
- `-s` Anzeige für jedes Verzeichnis, dabei ausschließen der Unterverzeichnisse in der Summe



Beispiele:

```
root@abc:~# du /etc/init.d/  
24      /etc/init.d/ipsec.d/policies  
28      /etc/init.d/ipsec.d  
708     /etc/init.d
```

```
root@abc:~# du -s /etc/init.d/  
708     /etc/init.d
```

```
root@abc:~# du -sS /etc/init.d/  
680     /etc/init.d
```

```
root@abc~#cd /etc/init.d  
root@abc:/etc/init.d# du -csh  
708K    .  
708K    insgesamt
```




Beispiele:

```
root@berkely:~# du -cs /home/* |sort -nr
10034948    insgesamt
7194756     /home/walter
1100280     /home/connie
883052      /home/jonas
766752      /home/hanna
90092       /home/siggi
16          /home/lost+found
root@berkely:~#
```

Es gibt mehrere Befehle, um die Strukturdaten von ext2 anzuzeigen und zu manipulieren.

- | | |
|-------------------------------------|---|
| <code>dumpe2fs device</code> | Anzeige der Strukturdaten (Inhalt Superblock, Anzahl freier I-Nodes in jeder Blockgruppe, ...) |
| <code>debugfs device</code> | Wie <i>dumpe2fs</i> , aber interaktiv und Manipulation an den Strukturdaten sind teilweise möglich (VORSICHT!) |
| <code>tune2fs device</code> | Parameter von ext2 ändern (Journaling ein/aus, max. Mount-Count, Reserved Blocks, ...) |

Syntax:

`fsck [optionen] [-t typ] device`

Überprüft die Integrität der Dateistruktur von *device* (verwendet aber selbst **`fsck.ext2`**, **`fsck.ext3`**, ... diese wieder untereinander und zu **`e2fsck`** binärkompatibel ... **`fsck.xfs`** bzw. **`xfs_repair`**, ...).

Optionen:

<i>typ</i>	ext2, ext3, minix, xfs, vfat, ... (Abhängig von instal. Paketen)
<i>device</i>	Gerätedatei (z.B. <code>dev/hda3</code>)
-A	all, Alle Dateisysteme in <code>/etc/fstab</code> prüfen (z.B. beim Booten)
-n	no execute, nichts verändern, nur testen

Für jedes Dateisystem gibt es weitere spez. Optionen, z.B. **ext2**:

-b <i>superbl</i>	Verwende einen anderen Superblock (z.B. <code>-b 8193</code>)
-c	check, nach defekten Blöcken suchen
-f	force, Überprüfung erzwingen, auch wenn die Partition „clean“ ist
-p	no prompt, repariere ohne Nachfrage
-y	yes, Antwortet auf alle Fragen mit yes

Links setzen

In [optionen] *datei link* *link* zeigt auf *datei*

In [optionen] *verzeichnis link* *link* zeigt auf *verzeichnis*

In [optionen] *dateien verzeichnis* In *verzeichnis* werden Links auf *dateien* angelegt

Wichtige Optionen:

- s** soft, erstellt einen Softlink (default: Hardlink)
- f** force, überschreibt existierende Dateien
- i** interactive, fragt vor dem Überschreiben

Werden Softlinks gelöscht (**rm**), wird nur der Link entfernt. Bei Hardlinks wird die Datei gelöscht, wenn kein Link mehr vorhanden (Linkcount = 0).



Links anzeigen:

Mit `ls -l` werden Softlinks inkl. Linkziel angezeigt.
Hardlinks müssen mit Hilfe der I-Node-Suche von `find` gefunden werden (s.u.). Der I-Node einer Datei/eines Verzeichnisses wird mit `ls -li` angezeigt.

Softlinks suchen:

```
find /verz -lname xyzlink
```

I-Nodes suchen:

```
find /verz -inum 12345
```

Filesystem Hierarchy Standard (FHS) - <http://www.pathname.com/fhs/>
trennt Dateisystem zuerst primär in Bereiche die systemspezifisch – und damit nicht mit anderen teilbar sind – und Bereiche die z.B. für mehrere Hosts gemeinsam genutzt werden können. Beispiel im FHS:

	"Shareable" da nicht systemspezifisch	"Unshareable" da systemspezifisch
Statisch (kaum Änderungen)	<code>/usr</code> <code>/usr/local</code>	<code>/etc</code> <code>/boot</code>
Dynamisch (ständige Änderungen)	<code>/var/mail</code> <code>/home</code>	<code>/var/log</code> <code>/proc</code>

- Entsteht ein Fehler im Dateisystem, so ist die Partition nicht mehr brauchbar
- Ist die root-Partition voll, kann das System nicht mehr arbeiten
- Quotas beziehen sich immer auf eine Partition
- Kleinere Partitionen sind u.U. schneller, da der Kopf nicht so weit bewegt werden muss
- => Ein System sollte in mehrere Partitionen aufgeteilt werden (zumindest auf dem Server)

/boot eventuell unbedingt als eigene Partition

Ein älteres BIOS kann maximal Festplatten mit 1024 Zylindern verwalten. Zum Booten ist Linux auf das BIOS angewiesen, deswegen muss für diese BIOS-Arten die Partition für das Verzeichnis **/boot** unterhalb dieser Grenze liegen. Sobald Linux gestartet ist, verwendet es das BIOS nicht mehr, so dass dann diese Grenze unerheblich ist.

Bei Verwendung des LBA-Modes und aktueller Boot-Manager (bei LILO mit Param. lba32) die LBA beherrschen spielt dies meist keine Rolle mehr!

Mögliche Aufteilung
des Dateibaums in
Partitionen und ein
Richtwert für deren
Größe. Reale Werte
aus der Praxis!!!



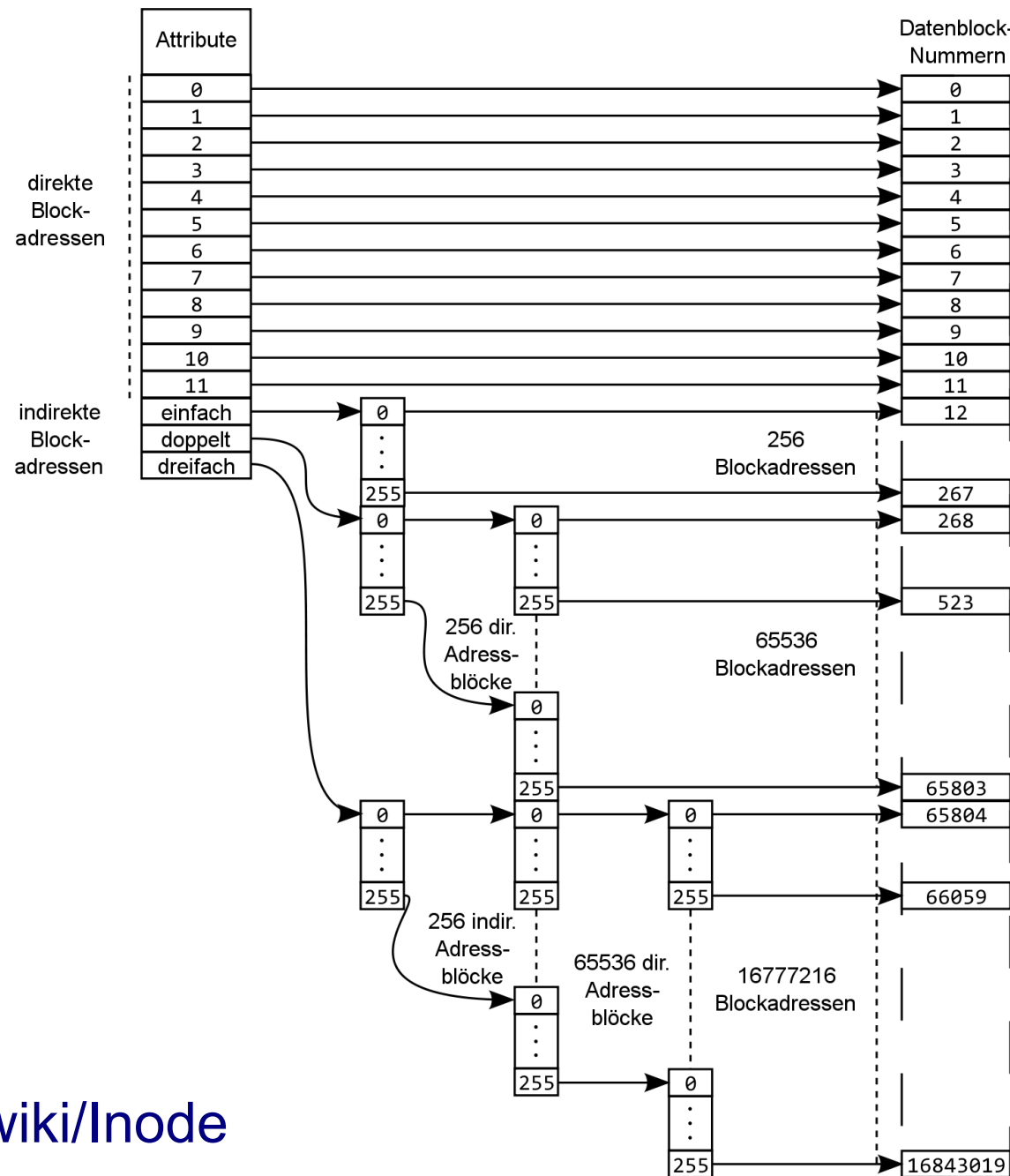
▪ /	500MB
▪ /boot	100MB
▪ /var	2GB
▪ /tmp	500MB
▪ /usr	6GB
▪ /home	> 50GB
▪ /opt	1GB
▪ swap	~ 1 bis 2-fach RAM



- Erstellen Sie mit **fdisk** mind. eine Partition in Ihrem System
- Formatieren Sie die Partition(en) mit einem spezifischen Dateisystem (vorher meist Reboot notwendig um neue Partition zu erkennen)
- Sorgen Sie dafür, dass Ihre neue Partition zukünftig unter **/home** gemountet wird.
Verschieben Sie die bisherigen Daten unter **/home** dazu auf die neue Partition.



- Mehrere Blockgruppen => schneller
- Superblock (anz. Blöcke, Inodes je insg./pro Gruppe)
- Inode Bitmap: Belegung der Inodes (Welcher Inode ist frei?)
- D-Zone Bitmap: Belegung der Datenblöcke (Welcher Datenblock ist frei?)
- z.B. Blockgröße 4kB => max. Dateigröße 2048GB, max.Partitionsgröße 16384GB



In Unix ist alles eine Datei. Deswegen gibt es verschiedene Arten von Dateien:

- Gerätedateien
 - Gepuffert, für ein blockorientiertes Gerät -> Festplatten (*b*)
 - Ungepuffert, für ein zeichenorientiertes Gerät -> Konsole (*c*)
- Verzeichnis (*d*)
- benannte Pipeline (FiFo), darüber können 2 oder mehrere Prozesse kommunizieren (-> man fifo) (*p*)
- normale Datei (-, *f*)
- symbolischer Link (*l*)
- Socket (*s*), darüber können 2 oder mehrere Prozesse kommunizieren (-> man socket),

Suchen kann man diese Dateien mit *find* (Schalter: -type *sym*)

- Ein Verzeichnis ist eine Datei, in welcher der I-Node und der Dateiname eingetragen ist.

15432	Länge	Dateiname
15501	Länge	abc.txt
15456	Länge	name.sh
15444	Länge	long_file_name

- Hardlink:
 - Ein I-Node kann in mehreren Verzeichnissen eingetragen sein (Vorteil: sehr schnell)
- Softlink
 - Im Datenblock steht als Text der Pfad zur richtigen Datei (Vorteil: Link geht auch über Partitions Grenzen)

- Setzen Sie einen Softlink namens **rcups** von der Datei **/etc/init.d/cups** in das Verzeichnis **/sbin**.
- Setzen Sie einen Hardlink von der Datei **/etc/init.d/cups** nach **/root/cups** und **/root/cups2**. Welchen Linkcount erhalten Sie (Hinweis: **ls**).
- Löschen Sie wieder die oben erstellten Links.
- Suchen Sie für jeden Dateityp (Blockgerät, Zeichengerät, ...) einen Vertreter in Ihrem System.
- Suchen Sie Dateien mit 3 Hardlinks (Hinweis: **find**).
- Suchen Sie zu diesen Dateien alle 3 Datei- bzw. Verzeichnisnamen (Hinweis: **find**).
- Warum kann es vorkommen, dass es nicht möglich ist, eine noch so kleine Datei anzulegen (Fehlermeldung: Disk full), obwohl noch 50 MB Speicher frei sind?

- Welche Bedeutung haben die Schalter `-i`, `-N` und `-T` für den Befehl `mke2fs`?
- Formatieren Sie die vorher erzeugte Partition mit dem Dateisystem `ext2`.
- Testen Sie die Dateisystemintegrität des neuen Dateisystems.
- Mounten Sie das neue Dateisystem in das Verzeichnis `/mnt/dummy`.
- Wie groß ist diese neue Partition?
- Wechseln Sie in das Verzeichnis `/mnt/dummy` und hängen Sie dann das System wieder aus. Was passiert?
- Wie viele Daten liegen in `/etc` (inkl./exkl. der Unterverzeichnisse)?
- Wie viele I-Nodes hat die root-Partition (`/`)?
- Wie groß ist in der root-Partition (`/`) ein Block?