

Partitionierung unter Linux

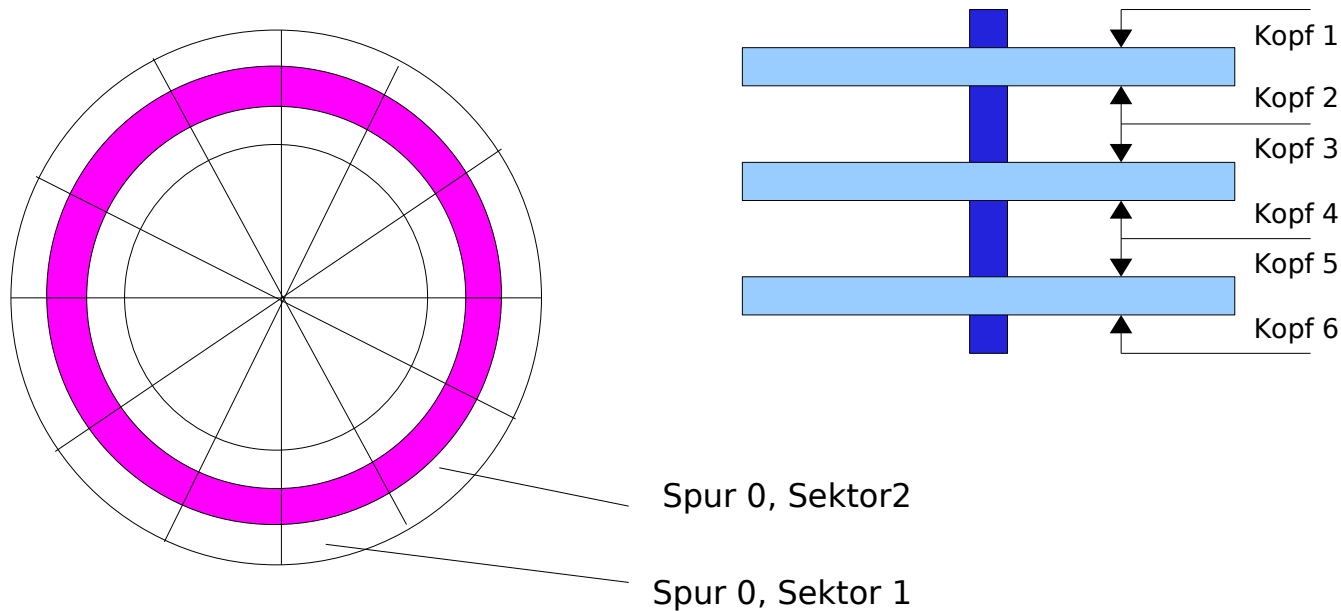
- Die Struktur einer Festplatte
- Aufbau der Partitionstabelle und Regeln
- Programme zum Partitionieren
- Partitionslayouts
- Dateisysteme
- Volume Label

Warum partitionieren?

- Mehrere Betriebssysteme können auf einer Platte nebeneinander existieren
- Abgrenzung der Datenbereiche eines Systems gegeneinander
- Begrenzung der Auswirkung von Dateisystemfehlern
- Optimierung der Kapazitätsausnutzung
- Sicherheitsaspekte

Die Struktur einer Festplatte

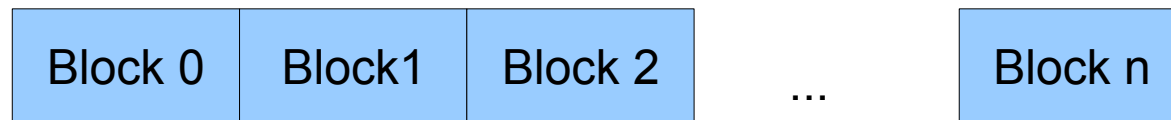
- Zylinder, Köpfe, Sektoren (CHS)



Sektorgröße 512 Bytes

Die Struktur einer Festplatte

- Eine lineare Folge von Blöcken (LBA)



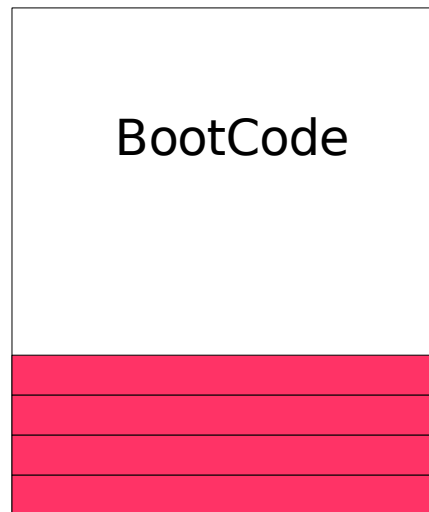
Blockgröße 512 Bytes

Partitionsschemata

- Historisch haben sich mehrere Partitionsschemata entwickelt, die einander ähneln. Wir besprechen das häufigste, das aus M\$ DOS resultiert.
- Es gibt einige Grundregeln, die eingehalten werden müssen, darüber hinaus Empfehlungen und Merkwürdigkeiten.

Aufbau der Partitionstabelle bis DOS 3

Partitionssektor
Zyl 0, Kopf 1, Sek 1 / Block 0

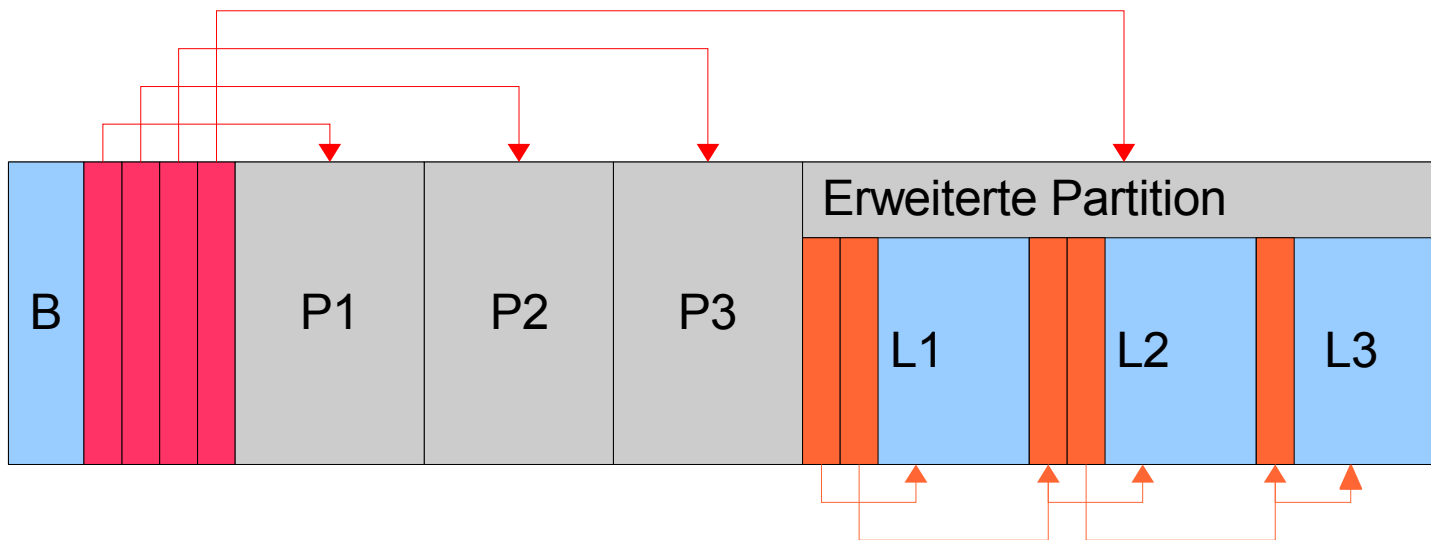


Partitionstabelle mit max.
4 primären Partitionen

Ein Eintrag enthält :

- Partitionsstatus
- Start der Part. In CHS
- Partionstyp
- Ende der Part. in CHS
- Start der Part. in LBA
- Länge der Part. in LBA

Primäre und erweiterte Partitionen



Vier primäre Partitionen, davon eine erweiterte Partition mit drei logischen Partitionen

Partitionsregeln (muss)

- Max. 4 primäre Partitionen
- Maximal eine primäre Partition darf eine erweiterte Partition sein.
- Eine erweiterte Partition enthält selbst keinen Datenbereich sondern logische Partitionen.
- Die Anzahl der logischen Partitionen ist unter Linux auf 15 / 63 (SCSI/IDE) pro Platte begrenzt.

Partitionsregeln (soll)

- Keine überschneidenden Datenbereiche
- Die Datenbereiche sollen aufsteigend angeordnet sein.
- Beginn und Ende einer Partition sollen auf Zylindergrenzen liegen.

Flags und „Partitionstypen“

- Das „boot“ Flag kennzeichnet die Partition, von der gebootet werden soll(!)
- Das Feld „Partitionstyp“ enthält eine krause Mischung aus verschiedenen Informationen, die von den unterschiedlichen Betriebssystemen möglicherweise ausgewertet wird (siehe Praxisteil).

Zwischenspiel: Geträtdateien

- Unter Unix ist alles eine Datei! Also ist auch die Festplatte, die wir partitionieren wollen, eine Datei. Aber welche?
- Geräte wie Festplatten sind als „Gerätdateien“ in /dev zu finden:

```
ls /dev/hda  
brw-rw---- 1 root disk 3, 0 Mr 23 2002 /dev/hda
```

- IDE Festplatten: /dev/hda bis /dev/hdd
- SCSI Festplatten: /dev/sda bis /dev/sdf

Zwischenspiel: Geträtdateien

- Für jede Partition einer Platte gibt es ebenfalls eine Gerätedatei:
`/dev/hda1` für die erste Partition der ersten IDE Platte usw.

Programme zum Partitionieren

- Textbasiert
 - cfdisk: einfach, mit Menü, kann aber nicht alle Partionstabellen bearbeiten
 - fdisk: etwas komplexer, kann alles bearbeiten, lässt „Schweinkram“ zu.
 - parted: macht viel mehr als Partitionierung, kann Partitionen mit Dateisystemen verkleinern!
- Bunt:
 - gparted: GUI für parted

Die Programme in der Praxis

...

Partitionlayout

- Partitionen nehmen auf:
 - Dateisysteme
 - Swapspace
 - Rohdaten (z.B. für Datenbanksysteme)
- Klein und flexibel:
 - Eine große Partition für das / Dateisystem
 - Eine Partition für Swap
- Wieviel Swap braucht ein Linux System?
genug ;-) (stark lastabhängig)

Partitionlayout

- Besser strukturiert:
 - /
 - /boot (notwendig bei altem BIOS)
 - /usr (könnte ro gemountet werden)
 - /opt (könnte ro gemountet werden)
 - /var
 - /home
 - (/srv)
 - swap

Partitionlayout

- Wir machen jetzt
 - /
 - /boot
 - /home
 - Swap
- ...

Dateisysteme

- Verschiedene Implementierungen stehen zur Auswahl: ext2/3, jfs, xfs, reiserfs, ...
(für Performance Aussagen bitte aktuelle Benchmarks lesen)
- einige Parameter können verändert werden:
Blockgröße, Inode Dichte (die default Werte passen aber schon ganz gut für Desktop Rechner)

Journaling

- Dateisysteme enthalten Nutzdaten und sogenannte Metadaten (interne Strukturen).
- Stürzt das System beim Schreiben der Metadaten ab, so müssen diese repariert werden (mit fsck). Das kann bei großen Dateisystemen sehr lange dauern.
- Dateisysteme mit Journaling führen über Veränderungen der Metadaten ein Protokoll. Damit ist eine Rekonstruktion der Metadaten im Falle eines Absturzes sehr schnell möglich.
- Hinweis: ext3 ist ext2 mit Journaling!

Anlegen eines Dateisystems

- `mkfs -t <typ> ...` oder `mkfs.<typ> ...` macht keinen Unterschied
- Beispiele:
 - `mkfs.ext2 /dev/hda1 ext2` mit default Werten
 - `mkfs.ext2 -j /dev/hda1 ext3` mit default Werten
 - `mkfs.ext2 -j -b 1024 /dev/hda1 ext3` mit **blocksize 1024**

/etc/fstab

- Die Datei gibt an, welche Dateisysteme beim Systemstart wo gemountet werden sollen.

#<fs>	<mountpoint>	<type>	<opts>	<dump/pass>
/dev/hda1	/boot	ext2	defaults	0 0
/dev/hda2	none	swap	sw	0 0
/dev/hda3	/	ext3	defaults,noatime	0 0
/dev/fd0	/mnt/floppy	minix	noauto,user	0 0

Volume Label

- Ermöglichen es, Dateisysteme nicht nach Partitionsname zu mounten sondern nach einem Label, das im Superblock steht.
- Label anlegen mit
`e2label HOME /dev/hdb2` oder
`mkfs.ext2 -L HOME /dev/hdb2`
- In der `/etc/fstab` steht dann
`LABEL=HOME /home ext3 ...`
- Schützt vor Überraschungen beim Vertauschen von Platten oder Ausfall einer Platte.

Links

- Filesystem Hierarchy Standard
<http://www.pathname.com/fhs/>
- Linux Partition HOWTO
<http://tldp.org/HOWTO/Partition/>
- Filesystems HOWTO (incomplete)
<http://tldp.org/HOWTO/Filesystems-HOWTO.html>