5. Platten und Filesysteme

Festplatten: (Harddisks, Raid-Systeme, ZIP, Jazz, MOs, CD-ROM's, Floppy's)

Struktur:

Zylinder, Spuren, Sektoren

Partitionen:

a,b,c,d,e,f,g,h
in der Regel mit festen Funktionen:
a - root
b - swap
c - gesammte Platte
d,e,f,g,h - Datenpartitionen, können auch andere überlappen

Partitionsgrössen und -lage verschieden festgelegt:
- dynamisch durch die Partitionstabelle
- durch ein File z.B. /etc/disktab (DEC)
- statisch durch Übersetzung des Drivers (HP-UX)
zur Information: /etc/disktab

Namenskonventionen für Festplatten:

	Struktur	Beispiel
SunOS:	/dev/[r]sdNP	/dev/sd0a
Solaris:	/dev/[r]dsk/cCtSdUsP	/dev/dsk/c0t3d0s6
DEC-UNIX:	/dev/[r]rzNP	/dev/rz12c
HP-UX:	/dev/[r]cNdUsP	/dev/dsk/c1000d0s13
AIX:	/dev/hdiskN	/dev/hdisk8
Linux:	/dev/sdNP	/dev/sda3

- C – Controller
- N – Plattennummer
- S – SCSI-ID
- U – Unit
- P – Partition
- r – Raw-Device

Über Platten-IDs

z.B. Linux

```
/dev/disk/by-id/ata-HITACHI_HTS543232L9SA00_081025FB0432LEH5X9DA-part5
/dev/disk/by-id/ata-HITACHI_HTS543232L9SA00_081025FB0432LEH5X9DA-part6
/dev/disk/by-id/ata-HITACHI_HTS543232L9SA00_081025FB0432LEH5X9DA-part7
/dev/disk/by-id/ata-HITACHI_HTS543232L9SA00_081025FB0432LEH5X9DA-part1
```

Filesysteme:

	SunOS	Solaris	Tru64	HP-UX	Linux	AIX
local	4.2	ufs zfs	ufs advfs	vxfs	ext4 reiserfs	jfs2 gpfs
NFS	nfs2	nfs4	nfs3	nfs3	nfs3(4)	nfs3
CD-ROM	hsfs	hsfs	cdfs	cdfs	iso9660	cdrfs
swap	swap	swap	zfs	-	swap	-
DOS	pcfs	pcfs	pcfs	-	umsdos	-
/proc	-	procfs	procfs	-	procfs	procfs
RAM	-	tmpfs	mfs	-	ramfs tmpfs	-
sonstige	-	cachefs nfs3	-	hfs	ext2 vfat smb jfs	-

locale Filesysteme: Aufbau des Datenbereichs in der Regel mit unterschiedlicher Struktur und Leistungs-Fähigkeit.

NFS: Version 3 (Standard), Version 4 (neu, sicherer)

CD-ROM: standardisiert ISO9660

swap: individuell

DOS: standardisiert

/proc: Prozeßinformationen in Filesystemstruktur

Bootbare Filesysteme: Bestimmte Bereiche des Filesystems werden mit zusätzlichen Informationen gefüllt.

Eigenschaften von lokalen Filesystemen

	SunOS	Solaris		Tru64	HU-UX		Linux		AIX
Typ	4.2	ufs	zfs	ufs	advfs	vxfs	ext4	reiserfs	jfs
Journal	nein	ja	ja	nein	ja	ja	ja	ja	ja
>2GB	nein	ja	ja	ja	ja	ja	ja	ja	ja
Dynamische Anpassung	nein	ja (ab 2.9)	ja	nein	ja	ja	ja	ja	ja
Lückenfiles	nein	ja	ja	ja	ja	ja	ja	ja	ja
NFSv3	nein	ja	ja	ja	ja	ja	ja	ja	ja
NFSv4	nein	ja	ja	nein	nein	nein	ja	ja	nein
Dump	ja	ja	ja	ja	ja	ja	ja	nein	ja

Zuordnung von Partitionen und Filesystemen

Klassisch: 1 : 1, einer Partition wird einem Filesystem zugeordnet.

Moderne Systeme:
Striping, Raid-Technologie, Logical Volume Managers, Logical Storage Manager (alles später)

Filesysteme müssen auf Partitionen angelegt werden – mkfs (später)

Filesysteme müssen gemountet werden, damit sie benutzt werden können.

Root-Verzeichnis ist beim Booten automatisch gemountet.

Mount-Kommando

```
   mount [optionen] block-special-device mount-point
             mount-point:   Dircetory

SunOs:
  /usr/etc/mount [-p]
  /usr/etc/mount -a [-fnv] [-t type]
  /usr/etc/mount [-fnrv][-t type][-o options] filesystem directory
  /usr/etc/mount [-vfn] [-o options] filesystem | directory
  /usr/etc/mount -d [-fnvr][-o options] RFS-resource | directory

Solaris:
  mount [ -p | -v ]
  mount [ -F FSType ] [ generic_options ]
        [ -o specific_options ] [ -O ] special | mount_point
  mount [ -F FSType ] [ generic_options ]
        [ -o specific_options ] [ -O ] special mount_point
  mount -a [ -F FSType ] [ -V ] [ current_options ]
           [ -o specific_options ] [ mount_point. . . ]

DEC-UNIX:
  /usr/sbin/mount [-el] [-t [no]type]
  /usr/sbin/mount -a [-fv] [-t [no]type]
  /usr/sbin/mount [-d] [-r|-u|-w] [-o option, ...] [-t [no]type]
                 file-system directory
  /usr/sbin/mount [-d] [-r|-u|-w] [-o option, ...] [-t [no]type]
                 file-system | directory
```

```
HP-UX:
  /etc/mount [fsname directory [-frv] [-s|-u] [-o options] [-t type]]
  /etc/mount -a [-fv] [-s|-u]
  /etc/mount [-p] [-l|-L] [-s|-u]

AIX:
  mount [-f][-n Node][-o Options][-p][-r][-v VfsName]
        [-t Type | [Device | Node:Directory] Directory |
        all |  -a ]

Linux:
       mount [-hV]
       mount -a [-fnrvw] [-t vfstype]
       mount [-fnrvw] [-o options [,...]] device | dir
       mount [-fnrvw] [-t vfstype] [-o options] device dir
```

Mountoptionen: Beschreiben Filesystemeigenschaften, sind von System zu System verschieden

rw	– lesen und schreiben
ro	– nur lesen
nosuid	– keine S–Bit–Unterstützung
noauto	– kein automatisches Mounten beim booten
noexec	– keine X–Bit–Unterstützung
nodev	– kein Gerätezugriff (AIX,LINUX,True64)
user	– mounten durch Nutzer erlaubt (fstab)
remount	– gemountetes Filesystem erneut mounten
nogrpid	– keine Vererbung von Gruppen–ID
resuid=UID	– UID für reservierte Blöcke
resgid=GID	– GID für reservierte Blöcke
largefiles	– Unterstütung von Files über 2 GB
logging	– Journaling ein (Solaris, linux ext4)
delaylog	– Verzögertes Schreiben von Log–Einträgen (schneller)
writeback	– Schreiben von Blöcken in optimaler Reihenfolge
nolog	– kein Transaktionslog (HP–UX)
nologging	– Journaling aus (Solaris)
forcedirectio	– keine Pufferung (direct io)
resize=nn	– Blockgröße beim Mounten anpassen
rsize=nn	– Blockgröße beim Lesen (NFS)
wsize=nn	– Blockgröße beim Schreiben (NFS)
rq	– Quotas aktiv (True64)
userquota	– Quota aktiv (Linux)
quota	– Quota aktiv (Solaris)
pri=nn	– Priorisierung von SWAP–Bereichen

Konfigurationsfile für automatisches Mounten

SunOs – /etc/fstab:

```
# dev to mount  mountpoint  FS  option freq  pass
#                                      dump  fsck
/dev/sd0a       /           4.2    rw     1     1
/dev/sd0h       /home       4.2    rw     1     3
/dev/sd0g       /usr        4.2    rw     1     2
/dev/fd0   /     pcfs       pcfs   rw,noauto 0  0
```

Solaris – /etc/vfstab für ufs

```
#   dev to mount    dev to fsck      mountpoint FS  FSCK mount opt
#                                                   pass boot
/dev/dsk/c0t3d0s0 /dev/rdsk/c0t3d0s0 /          ufs  1    no    –
/dev/dsk/c0t3d0s6 /dev/rdsk/c0t3d0s6 /usr       ufs  1    no    –
/dev/dsk/c0t3d0s5 /dev/rdsk/c0t3d0s5 /opt       ufs  2    yes   –
/dev/dsk/c0t3d0s1      –               –        swap –    no    –
/dev/dsk/c0t1d0s0 /dev/rdsk/c0t1d0s0 /usr/local ufs  6    yes   –
/dev/dsk/c0t2d0s6 /dev/rdsk/c0t2d0s6 /usr1      ufs  7    yes   –
```

DEC-UNIX - /etc/fstab:

```
# dev      mountpoint   FS  OP freq pass
#                            dump fsck
/dev/rz3a       /       ufs rw  1    1
/dev/rz3g       /usr    ufs rw  1    2
/dev/rz0c       /usr/local ufs rw 1  2
/dev/rz3b       swap1   ufs sw  0    2
/dev/rz8c       /usr1   ufs rw  1    2
/dev/rz9c       /usr2   ufs rw  1    2
/dev/rz10c      /usr3   ufs rw  1    2
/dev/rz11c      /usr4   ufs rw  1    2
/dev/rz16c      /usr5   ufs rw  1    2
/dev/rz17c      /usr6   ufs rw  1    2
/dev/rz18c      /usr7   ufs rw  1    2
/dev/rz19c      /usr8   ufs rw  1    2
/dev/rz20c      /usr9   ufs rw  1    2
```

HP-UX - /etc/checklist:

```
# device            mountpoint FS     options   backup  pass
#                                                       fsck
/dev/dsk/c1000d0s13 /          vxfs   defaults      0       1
/dev/dsk/c1001d0s2  /usr1      hfs    rw            0       1
/dev/dsk/c1000d0s15 swap       ignore sw            0       0
/dev/dsk/c1001d0s2 /usr1 swapfs min=0,lim=2000,res=0,pri=0 0 0
```

```
Linux     -    /etc/fstab:
  # device   MNT-Point Type     options       dump fsck-seq
  #
  /dev/sda3  swap      swap     defaults        0   0
  /dev/sda2  /         ext2     defaults        1   1
  /dev/sda1  /dos      msdos    defaults        0   0
  /dev/hda   /cdrom    iso9660  ro,noauto,user  0   0
  /proc      /proc     proc     defaults        0   0

AIX        -     /etc/filesystems:

   default:
        vol             ="AIX"
        mount           =false
        check           =false
   /:
        dev             = /dev/hd4
        vfs             = jfs
        log             = /dev/hd8
        mount           = automatic
        check           = false
        type            = bootfs
        vol             = root
        free            = true
```

```
   /home:
        dev         = /dev/hd1
        vol         = "/home"
        mount       = true
        check       = true
        free        = false
        vfs         = jfs
        log         = /dev/hd8
   /usr:
        dev             = /dev/hd2
        vfs             = jfs
        log             = /dev/hd8
        mount           = automatic
        check           = false
        type            = bootfs
        vol             = /usr
        free            = false

AIX       -     /etc/swapspaces:

   hd6:
        dev = /dev/hd6
```

umount-Kommando

```
    umount mount-point | block-special-device

SunOS:
     /usr/etc/umount [-t type] [-h host]
     /usr/etc/umount -a [-v]
     /usr/etc/umount [-v] filesystem|directory
     /usr/etc/umount [-d] RFS-resource | directory

Solaris:
     umount [-V] [-o specific_options] special | mount_point
     umount -a [-f] [-V] [-o specific_options] [mount_point]

DEC-UNIX:
  /usr/sbin/umount -a|-A -b [-fv] [-t type] [-h host]

  /usr/sbin/umount [-fv] file-system ... | directory ...

HP-UX:
      /etc/umount [-v] [-s] fsname
      /etc/umount [-v] [-s] directory
      /etc/umount -a [-v] [-s] [-h host]] [-t type]]

AIX:
  {unmount | umount} [-f] [-a] | [all | allr | Device |
          Directory | File | FileSystem | -n Node |  -t Type ]
```

Linux:

```
       umount [-hV]
       umount -a [-dflnrv] [-t vfstype] [-O options]
       umount [-dflnrv] dir | device [...]
```

Umount nur wenn Filesystem nicht benutzt!!!!

Kommando fuser:

```
   fuser [-kufc] filename|ressource
```

Ab Solaris 2.8, Linux 2.6:

umount -f – gewaltsames entmounten von Fifesystemen (für NFS)

Prüfen der Konsistenz von Filesystemen

Sehr wichtig. Sollte in regelmässigen Abständen erfolgen, bei einem Absturz wird es in der Regel automatisch beim Booten gemacht (/etc/fstab,/etc/checklist,..).
Achtung!!!! Es gibt einige Systeme, die dies bei ordentlichem Shutdown nie machen.

Kommando fsck:

```
fsck [-pPfcnyo] [-b superblock] [-l number] [-m mode] [-F type]
                        filesystem
```

- -p – nicht interaktiv, automatische Korrektur
- -c – konvertieren
- -y – yes-Antwort
- -n – no-Antwort
- -o – unbedingt prüfen, optionen weiterreichen(Solaris)
- -b number – Nummer eines alternativen Superblocks
- -l number – Anzahl der parallelen fsck-Läufe
- -m mode – Mode für lost+found
- -w – nur schreibbare Filesysteme prüfen
- -P,-f – prüfen nur wenn nicht korrekt entmountet (clean)

Fehler, die fsck erkennen und beheben kann:

- **Blöcke finden, die mehreren Inodes zugeordnet sind**
- **Blöcke finden, die als frei gekennzeichnet sind, aber noch in einem Inode benutzt werden**
- **Blöcke finden, die als benutzt gekennzeichnet sind, aber frei sind**
- **Inkorrekte Linkcounts**
- **Inkonsitenz der Filegrösse**
- **Illegale Blöcke in Files (Systemtabellen)**
- **Inkonsitenz der Filesystemtabellen**
- **Inodes ohne Filenamen**
- **Falsche oder nichtzugewiesene Inodenummern in Directories**

Protokoll von fsck unter LINUX:

```
> fsck /dev/sda2
e2fsck 1.06, 7-Oct-96 for EXT2 FS 0.5b, 95/08/09
/dev/sda2 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
UNREF FILE I= 523  OWNER = 501 MODE = 100660
SIZE = 0  MTIME = May 5 16:33 1997
CLEAR? y
Pass 5: Checking group summary information
Fix summary information? yes

Block bitmap differences: -31841 -31842 -31843 -31844 -31845
                -31846 -31847 -126738 -126739 -126740 -126741
                -126742 -126743 -126744.  IGNORED
/dev/sda2: 23814/69360 files (2.1% non-contiguous),
                                        243069/276480 blocks
Parallelizing fsck version 1.06 (7-Oct-96)
>
```

Bemerkung:
fsck wird in der Regel bei nicht gemounteten Filesystemen durch-
geführt. Aussnahme: root-Filesystem. Wenn fsck beim einem
gemounteten Filesystem etwas gemacht hat, darf hinterher keine
sync-Operation veranlasst werden (reboot -n).

Hinzufügen einer neuen Festplatte zu einem bestehenden System

allgemeine Aktionen:

1. Neue Platte einbauen
2. Betriessystem auf das neue Gerät vorbereiten.
   (Kern muss neu konfiguriert werden)
3. Gerätedateien für das neue Gerät anlegen
4. Festplatte eventuell formatieren
5. Partitionen auf der Festplatte anlegen
6. Filesystem(e) oder swap–Bereiche anlegen
7. neue Filesysteme checken (fsck)
8. Anlegen eines neuen Mountpoints
9. Mounten von Hand
10. Einfügen des neuen Filesystems in die System
    konfigurationsfiles (/etc/fstab,...)
11. Bootinformationen eintragen
12. Verwaltungsarbeiten (quota einrichten)

Etwas SCSI-Technik: SCSI ist nicht gleich SCSI

Name	Geschwindigkeit	Busbreite	Kabellänge
SCSI-1/SCSI-2	5 MB/s	8 Bits	6m
Fast SCSI	10 MB/s	8 Bits	3m
Fast Wide SCSI	20 MB/s	16 Bits	3m
Ultra SCSI	20 MB/s	8 Bits	1,5m
Wide Ultra SCSI	40 MB/s	16 Bits	1,5m
Ultra2 SCSI	40 MB/s	8 Bits	3m
Wide Ultra2 SCSI	80 MB/s	16 Bits	3m
Ultra3 SCSI	160 MB/s	16 Bits	3m
Ultra160 SCSI	160 MB/s	16 Bits	3m
Ultra320 SCSI	320 MB/s	16 Bits	3m

SCSI ist ein Bus und muß terminiert werden. Terminierung kann
am Gerät oder am Kabel erfolgen.
Es gibt aktive, passive, hybride und Forced Perfect Termination.
Es gibt diverse Kabeltypen und Stecker.

- Stecker: DB-25 (SCSI-1)
  - 50-Pin Centronics (SCSI-1, SCSI-2)
  - 50-Pin Mikrostecker (SCSI-2)
  - 68-Pin Mikrostecker (SCSI-3)
  - ... diverse Ministecker für Ultra-SCSI

SAN – Storage Area Network

Anschluss von Platten über LWL-Netzwerk
2,4,8,16 GBit z.Z. üblich

Spezielle Treiber simmulieren die Festplatten

In der Regel mehrer Netzwerkverbindungen zwischen SAN-Server und SAN-Client.
Spezielle zusätzliche Software notwendig – dynapath, mpath – dadurch höhere Sicherheit beim Ausfall einer Netzwerkverbindung.

Virtualisierung ist üblich.

Bei der Formatierung der Platten Label beachten
- SMI-Label – normal nicht portabel
- EFI-Label – Portabel (die Partitionen und Filesysteme sind später zwischen verschiedenen Rechnerarchitekturen portierbar)
  z.B. Solaris Sparc, Solaris X86

Arme Leute Variante: iscsi – über normales Netzwerk

Der mknod-Befehl

Gerätedateien werden mit dem Befehl mknod angelegt. Dadurch wird eine Verbindung im Filesystem zwischen Name und Treiber hergestellt

```
mknod  <Dateiname>  b <majornummer> <minornummer>
mknod  <Dateiname>  c <majornummer> <minornummer>

   b – block-orientiertes Gerät
   c – character-orientiertes Gerät
   u – ungepufferte Gerät
   p – FIFO (named pipe)
   majornummer – Geräteklasse
   minornummer – Gerätenummer in der Geräteklasse
```

Erstellen eines Filesystems in einer Partition mit:

```
mkfs [-V][-t fstype] [fs-optionen] <gerät> [<blocks>]
  fs-optionen  – spezielle Optionen für das Filesystem
       -c
       -b block-size
       -f fragment-size
       -i bytes-per-inode
       -N number-of-inodes
       -m reserved-blocks-percentage
```

```
LINUX:
------
eventuell Gerätedateien anlegen, heute automatisch:
  cd /dev; MAKEDEV sdc

Partitionieren:
  fdisk [-l][-v][-u][-b sectorsize] [-s partition] <device>
        Partitionen 1-4 -->  /dev/sdc1.../dev/sdc4
     oder
  cfdisk [-agvz][-c cylinders][-h heads]
             [-s sectors-per-track][-P opt] [device]
Filesystem anlegen:
  mkfs [-c] [-b bytes] [-i bytes/inode] -t ext4 -j /dev/sdc1 # normal
  mkreiserfs /dev/sdc1  # Reiser
Filesystem prüfen:
  fsck -f -y /dev/sdc1  # normal
  reiserfsck -x /dev/sdc1 # Reiser
Plattenzugriff tunen:
  tune2fs -l /dev/sdc1          # show
  tune2fs -i 0 -c 25 /dev/sdc1 # max mount counts
Mounten:
  mkdir /newdisk
  mount /dev/sdc1 /newdisk
  vi /etc/fstab

Bootinformationen:
     lilo, grub
```

```
SunOs:
------
  Kernel eventuell neu bilden (File: /usr/sys/...)
     default:   Controller   SCSI ID    Device
                    0            3         sd0
                    0            1         sd1
                    0            2         sd2
                    0            0         sd3
                    1            3         sd4
                    1            1         sd5
                    1            2         sd6
                    1            0         sd7
  format-Programm: formatieren, partitionieren, definieren,
                   Ersatzspurzuweisung
    Kochrezept:
      format [/dev/sd2]
           Subkommandos:
                disk    -  Festplatte auswählen
                type    -  Type festlegen, wenn nicht automatisch erkennbar
                defect  -  Defektspurmanagment
                   origin    - Herstellertabelle laden
                   commit    - übertragen
                   quit
                format  -  formatieren und prüfen
                partition(print,a,b,c,d,e,f,g,h,label,quit)
                  print      - anzeigen
                  a,b,c,g    - Partition a definieren
                  label      - Partitionstabelle schreiben
                  quit
                label   -  Alles schreiben
```

```
 Anzeigen der Platteninformationen bei laufendem Betrieb:
   > dkinfo sd0
   sd0: SCSI CCS controller at addr f0800000, unit # 24
   2036 cylinders 14 heads 72 sectors/track
   a: 66528 sectors (66 cyls)
      starting cylinder 0
   b: 139104 sectors (138 cyls)
      starting cylinder 66
   c: 2052288 sectors (2036 cyls)
      starting cylinder 0
   d: No such device or address
   e: No such device or address
   f: No such device or address
   g: 684432 sectors (679 cyls)
      starting cylinder 204
   h: 1162224 sectors (1153 cyls)
      starting cylinder 883

Erzeugen eines neuen Filesystems:
  newfs [-b blocksize] [-f fragmentsize] [-i bytes-per-inode]
              [-m free-procent] raw-device
    > newfs /dev/rsd1g
    /dev/rsd1g:  12345 sectors in 234 cylinders of 3 tracks,64 sectors
                 333 MB in 32 cyl groups (.....)
    super-block backups (for fsck -b #) at:
    32, 4192, 8352, 12512, 16416, .....
Prüfen des neuen Filesystems:
    > fsck  -y /dev/rsd1g
Mount-Point anlegen:
    > mkdir /dir
```

```
Manuelles Mounten:
    > mount /dev/sd1g /dir
Vorbereitung des automatischen Mounten:
    > vi /etc/fstab
    /dev/sd1g /dir 4.2  rw  1 4
Testen des automatischen Mounten:
    > umount /dir
    > mount -a -t 4.2
Bootinformationen eintragen:  installboot
```

Solaris:

Gerätedateien:
/dev/dsk/c0t3d0s0

Formatieren und partitionieren mittels Kommando format:

```
# > format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
          /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
       1. c0t2d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
          /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@2,0
       2. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
          /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0

Specify disk (enter its number): 2
                                 ---
selecting c0t3d0
[disk formatted]
Warning: Current Disk has mounted partitions.
```

```
FORMAT MENU:

        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        quit

format>
```

```
format> partition

PARTITION MENU:
        0      – change ‘0’ partition
        1      – change ‘1’ partition
        2      – change ‘2’ partition
        3      – change ‘3’ partition
        4      – change ‘4’ partition
        5      – change ‘5’ partition
        6      – change ‘6’ partition
        7      – change ‘7’ partition
        select – select a predefined table
        modify – modify a predefined partition table
        name   – name the current table
        print  – display the current table
        label  – write partition map and label to the disk
        quit

partition>
```

```
partition> print

Current partition table (original):
Total disk cylinders available: 2036 + 2 (reserved cylinders)

Part      Tag    Flag   Cylinders        Size            Blocks
  0       root    wm     0 -   91       45.28MB    (92/0/0)     92736
  1       swap    wu    92 -  376      140.27MB    (285/0/0)   287280
  2     backup    wm     0 - 2035     1002.09MB    (2036/0/0) 2052288
  3 unassigned    wm     0               0         (0/0/0)          0
  4 unassigned    wm     0               0         (0/0/0)          0
  5      stand    wm   377 - 1189      400.15MB    (813/0/0)   819504
  6        usr    wm  1190 - 2035      416.39MB    (846/0/0)   852768
  7 unassigned    wm     0               0         (0/0/0)          0

partition>

       format-Kommando-Kochrezept:  analog SunOs

partition> label
partition> quit
format> quit
```

Anzeigen der Platteninformationen bei laufendem Betrieb:

```
               prtvtoc  /dev/rdsk/c0t3d0s0

# prtvtoc  /dev/rdsk/c0t3d0s0
* /dev/rdsk/c0t3d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
*      72 sectors/track
*      14 tracks/cylinder
*    1008 sectors/cylinder
*    2038 cylinders
*    2036 accessible cylinders
* Flags:
*   1: unmountable
*  10: read-only
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00          0     92736     92735   /
       1      3    01      92736    287280    380015
       2      5    00          0   2052288   2052287
       5      6    00     380016    819504   1199519   /opt
       6      4    00    1199520    852768   2052287   /usr
#
```

Anlegen eines neuen Filesystems:

```
# newfs -Nv /dev/rdsk/c0t3d0s6

mkfs -F ufs -o N /dev/rdsk/c0t3d0s6 852768 72 14 8192 1024 16 10 90 \
              2048 t 0 -1 8 -1
/dev/rdsk/c0t3d0s6:     852768 sectors in 846 cylinders of 14 tracks,
                        72 sectors
        416.4MB in 53 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696, 145904, 162112,
 178320, 194528, 210736, 226944, 243152, 258080, 274288, 290496, 306704,
 322912, 339120, 355328, 371536, 387744, 403952, 420160, 436368, 452576,
 468784, 484992, 501200, 516128, 532336, 548544, 564752, 580960, 597168,
 613376, 629584, 645792, 662000, 678208, 694416, 710624, 726832, 743040,
 759248, 774176, 790384, 806592, 822800, 839008,
#
```

Prüfen:

```
  fsck -y /dev/rdsk/c0t3d0s6
```

/etc/vfstab aktualisieren:

```
  /dev/dsk/c0t3d0s6   /dev/rdsk/c0t3d0s6  /usr1  ufs    1   yes   -
```

Bootinformationen: installboot <bootblk> <raw-disk-device>

```
   installboot /usr/platform/sun4m/lib/fs/ufs/bootblk /dev/rdsk/c0t3d0s0
   installboot /usr/platform/‘uname -i‘/lib/fs/ufs/bootblk \
               /dev/rdsk/c0t3d0s0
```

Nach dem Einbauen eines neuen Gerätes:

```
      reboot --r
```

oder

```
                touch /reconfigure
                reboot
```

Das File /reconfigure bewirkt die Abarbeitung von /usr/sbin/drvconfig beim Booten

oder

```
    cfgadm -al
    cfgadm -c configure c1::dsk/c1t3d0
    cfgadm -c configure c1::21000011c6b82b4e
```

```
Tru64 ohne LVM
--------------
als root:
    cd /
    mv vmunix vmunix.org
    cp genvmunix vmunix
    shutdown -h now
  Ausschalten, Festplatte einbauen, einschalten
als root:
    cd /dev
    MAKEDEV rz19
    doconfig      (neuen Kern bilden und zu /vmunix machen, reboot)
/etc/disktab aktualisieren:
  Eintrag in disktab:
      rz24|RZ24|DEC RZ24 Winchester:\
              :ty=winchester:dt=SCSI:ns#38:nt#8:nc#1348:\
              :oa#0:pa#131072:ba#8192:fa#1024:\
              :ob#131072:pb#262144:bb#8192:fb#1024:\
              :oc#0:pc#409792:bc#8192:fc#1024:\
              :od#0:pd#0:bd#8192:fd#1024:\
              :oe#0:pe#0:be#8192:fe#1024:\
              :of#131072:pf#278720:bf#8192:ff#1024:\
              :og#393216:pg#16576:bg#8192:fg#1024:\
              :oh#0:ph#0:bh#8192:fh#1024:
Partitionieren:
    disklabel -r -w /dev/rrz19a RZ24
    disklabel -r /dev/rrz19a
```

Neues Filesystem anlegen:

```
    newfs /dev/rrz19c
    fsck -y -o /dev/rrz19a
```

Mounten:

```
   mount -t ufs /dev/rz19a /usr8
```

/etc/fstab aktualisieren:

```
  /dev/rz19c      /usr8   ufs rw 1 2
```

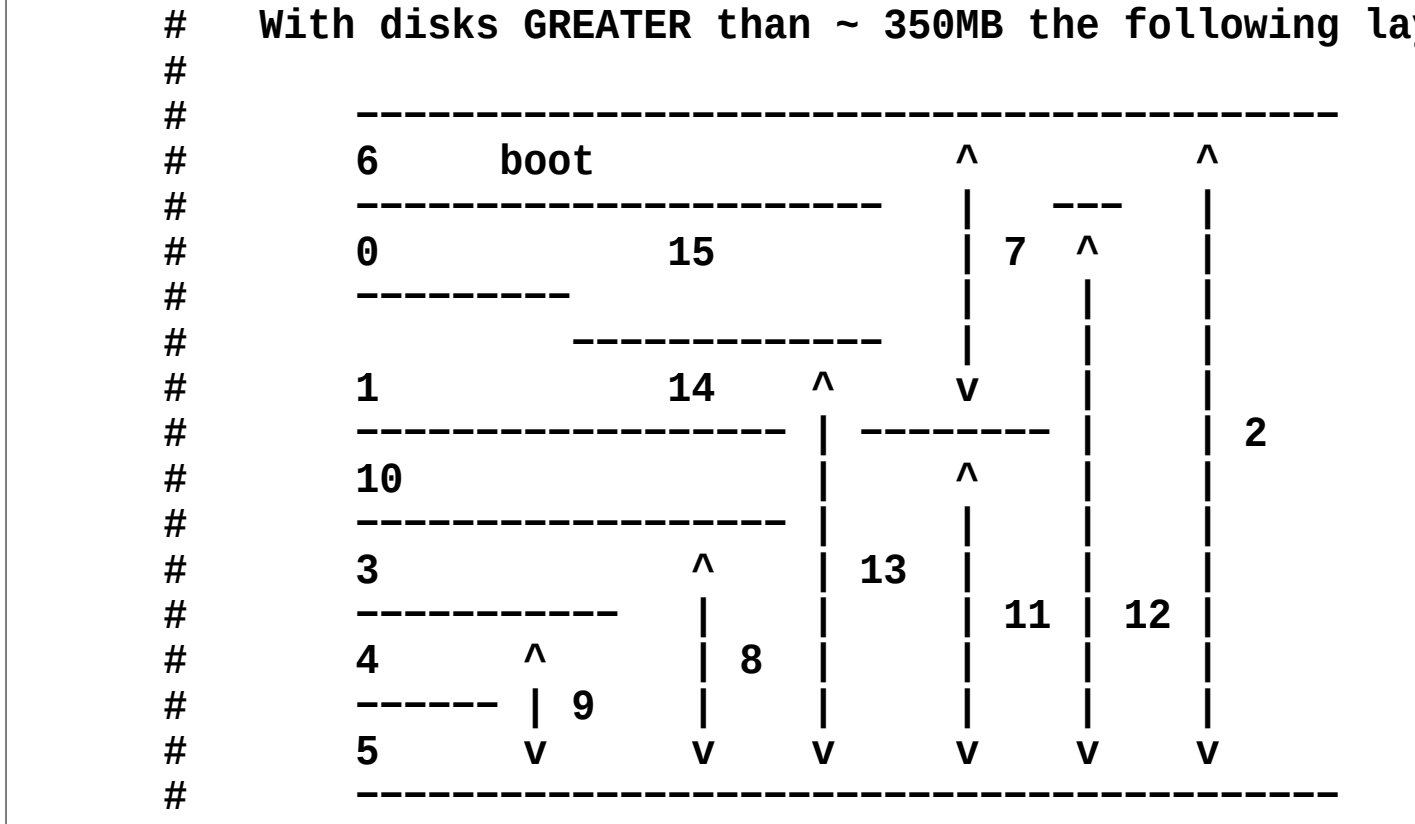
Bootinforamtionen: disklabel

```
      /sbin/disklabel -w [-r] [-t ufs | advfs] <disk-disktype>\
                      [packid] [primary-boot secondary-boot]
```

Hardwareabhängige primary und seondary Bootblöcke in: `/mdec/`

HP-UX bis 9.0 ohne LVM

/etc/disktab




```
hp7936|hp79360:\
        :ty=winchester:ns#30:nt#7:nc#1396:rm#3600:\
        :s0#24280:b0#8192:f0#1024:\
        :s1#16384:b1#8192:f1#1024:\
        :s2#300489:b2#8192:f2#1024:\
        :s3#19532:b3#8192:f3#1024:\
        :s4#36864:b4#8192:f4#1024:\
        :s5#200828:b5#8192:f5#1024:\
```

```
              :s6#1998:b6#8192:f6#1024:\
              :s7#43050:b7#8192:f7#1024:\
              :s8#257438:b8#8192:f8#1024:\
              :s9#237850:b9#8192:f9#1024:\
              :s12#298336:b12#8192:f12#1024:\
              :s13#281762:b13#8192:f13#1024:\
              :s14#24280:b14#8192:f14#1024:\
              :s15#16384:b15#8192:f15#1024:

Formatieren:
      mediainit  /dev/rdsk/c201d0s2

Einrichten und prüfen des Filesystems:
      newfs /dev/rdsk/c2001d0s12  hp7936
      fsck −y /dev/rdsk/c2001d0s12

Vorbereitung automatisiertes Mounten:
  /etc/checklist
      /dev/dsk/c2001d0s12 /usr1 hfs rw 0 1

Bootinformationen ablegen:  mkboot
```

Spezielle Filesysteme (CD-Laufwerke, Diskettenlaufwerke)

CD-ROM-Laufwerke:

```
  Linux:    /dev/cdrom,  /dev/sonycd, /dev/aztcd, /dev/scd0, /dev/sr0
  SunOs:    /dev/sr0
               Vordefiniert(Gerät - SCSI-Bus/SCSI-ID):
                  sr0 - 0/6, sr1 - 0/5, sr2 - 0/1,
                  sr3 - 0/0, sr4 - 1/6, sr5 - 3/6
  Solaris:  /dev/dsk/c0t#d0s0   standard: /dev/dsk/c0t6d0s0
  DEC-Unix: /dev/rz#c  standard: /dev/rz4c, /dev/disk/cdrom0c
  AIX:      /dev/cd0
  HP-UX:    /dev/dsk/c201d#s0
Bemerkung:  # - SCSI-ID am entsprechenden Kanal
```

Mount-Kommandos für CD-RROM-Laufwerke:

```
  Linux:    mount -r -t iso9660 /dev/cdrom /mnt
  SunOs:    mount -r -t hsfs /dev/sr0 /mnt
  Solaris:  mount -r -t hsfs /dev/dsk/c0t6d0s0 /cdrom
  DEC-Unix: mount -r -t cdfs /dev/rz4c /cdrom
  AIX:      mount  /cdrom
             Vorbedingungen:
                 mkdev -c cdrom -r cdrom1 -s scsi \
                       -p scsi0 -w 5,0
                 mkdir /cdrom
                 crfs -v cdrfs -p ro -d cd0 -m /cdrom -A no
  HP-UX:   mount -r -t cdfs /dev/dsk/c201d6s0 /mnt
```

```
Disketten-Laufwerke:
--------------------

Gerätedateien:

  Linux:    /dev/fd0, /dev/rfd0
  SunOs:    /dev/fd0, /dev/rfd0
  Solaris:  /dev/diskette
  DEC-Unix: /dev/fd0, /dev/rfd0
  AIX:      /dev/fd0, /dev/rfd0
  HP-UX:    /dev/dsk/c0t1d0, /dev/rdsk/c0t1d0

Geräte: normalerweise 1,44 MB Diskette 3,5"
         Aussnahme: AIX 1,44MB und 2,88 MB Diskette 3,5"

Benutzung:
    Blockorientiertes Medium für tar, cpio, ...
    Blockorientiertes Medium mit UNIX-Filessystem
    Blockorientiertes Medium mit DOS-Filesystem
        (gemountet und nicht gemountet)

DOS-Unterstützung
-----------------
Linux:
   Formatieren: fdformat  /dev/fd0
                mkfs -t msdos /dev/fd0
   Mounten:     mount -t msdos /dev/fd0 /mnt
   M-Tools:     siehe SunOs
```

SunOs:

Formatieren: fdformat

Zugriff nur über M-Tools:

```
mattrib – change MSDOS file attribute flags
mcd – change MSDOS directory
mcopy – copy MSDOS files to/from Unix
mdel – delete an MSDOS file
mdir – display an MSDOS directory
mformat – add an MSDOS filesystem to a low-level formatted diskette
mlabel – make an MSDOS volume label
mmd – make an MSDOS subdirectory
mrd – remove an MSDOS subdirectory
mread – low level read (copy) an MSDOS file to Unix
mren – rename an existing MSDOS file
mtype – display contents of an MSDOS file
mwrite – low level write (copy) a Unix file to MSDOS
```

Solaris:
über Volume Management Dämon /usr/sbin/vold

Konfiguration /etc/vold.conf

```
# @(#)vold.conf 1.20     95/01/09 SMI
#
# Volume Dämon Configuration file
#
# Database to use (must be first)
db db_mem.so
# Labels supported
label dos label_dos.so floppy pcmem
label sun label_sun.so floppy pcmem
# Devices to use
use floppy drive /dev/rdiskette[0-9] dev_floppy.so floppy%d
# Actions
insert dev/diskette[0-9]/* user=root /usr/sbin/rmmount
eject dev/diskette[0-9]/* user=root /usr/sbin/rmmount
# List of file system types unsafe to eject
unsafe ufs hsfs pcfs
```

Kommandos:

- Diskette bekannt machen: volcheck [-v] <pathname>
- Formatieren: fdformat -d -b fdbell1 /dev/diskette
- Auswerfen: eject
- Files kopieren mit cp

oder M-Tools

AIX:

Formatieren: dosformat
Lesen und Schreiben: dosread, doswrite
Streichen von Files: dosdel
Directory anzeigen: dosdir

Tru64:

keine Unterstützung für DOS

HP-UX:

formatieren mit: mediainit
mediainit -v -i2 -f16 /dev/rdsk/c0t1d0
newfs -n /dev/rdsk/c0t1d0 ibm1440
doscp [-fvu] file1 [file2 ...] directory
doscp [-fvu] file1 file2
doschmod [-u] mode device :file ...
dosdf device[:]
dosls [-aAudl] device:[file]
dosll [-aAudl] device:[file]
dosmkdir [-u] device:directory ...
dosrm [-friu] device:file ...
dosrmdir [-u] device:file ...

Moderne Plattenverwaltung

1.Disk-Striping

Zusammenfassung mehrerer physischer Platten oder Partitionen von Platten zu einer logischen Partition.

Zielstellung:

1. Erhöhung des Durchsatzes durch Parallelisierung der E/A-Operationen
2. Vergrösserung der Kapazität eines Filesystems

Kritische Punkte:

1. gute Parallelisierung nur mit mehrere E/A-Controler möglich
2. ähnliche Platten notwendig
3. kleinste Platte bestimmt die Partitionsgrösse
4. Platten sind nicht für andere Zwecke nutzbar
5. Nicht für root-Filesysteme verwendbar
6. Alles weg, wenn eine Platte ausfällt.

Unterstützung durch HP-UX, Solaris, DEC-OSF, AIX, Linux

2.RAID-Geräte

Redundant Array of Independent Disks

Zielstellung:

Fehlerredundante Speicherung von Daten

Realisierung:

Hardware (Controler), Software (z.B.: Solaris, Tru64, Linux)

Level	Funktion	Vorteile
0	Disk-Striping	Hoher E/A-Durchsatz
1	100%ige Plattenspiegelung	100%ige Datenredundanz
2	mit eigener ECC Einrichtung	nur Fehlerkorrektur möglich
3	Disk-Striping mit seperater Parity-Platte (Byte-Verteilung	Datenrekonstruktion beim Ausfall einer Platte möglich
4	wie 3, aber mit Block-Verteilung	schlechte Performance
5	Disk-Striping mit verteilten Parity-Informationen	Datenrekonstruktion beim Ausfall einer Platte möglich höher E/A-Leistung als Raid 4
6	wie 5, aber mit zwei Parity-Informationen	Datenrekonstruktion beim Ausfall zweier Platten möglich
0+1	Spiegelung gestripter Platten	höherer E/A-Leistung mit Raid-Sicherheit

Unterstützung durch HP-UX, Solaris, DEC-OSF, AIX, Linux

Raid unter LINUX

1.Ansatz (Kernel 2.4)

Softwareunterstützung für Raid-Level 0,1,4,5
Raid für root-Device möglich, aber nicht ganz einfach
Raid wird für SCSI-Platten und IDE-Platten unterstützt.

mittels /etc/raidtab wird die Konfiguration des Raid-Systems vorgegeben.

Schlüsselwörter in /etc/raidtab:

raiddev <device>	- Raid-Gerät /dev/md?
nr-raid-lever <nr>	- Raid-Level (0,1,4,5)
nr-Raid-disks <nr>	- Anzahl der Platten
nr-spare-disks<nr>	- Anzahl der Lehrplatten für Erweiterungen
persitent-superblock <nr>	- 0/1 Plattenbereich auf jeder Platte zum Erkennen der Platten
parity-algorithm	- left-symmetric, right-symmetric, left-asymmetric, right-asymmetric
chunk-size <nr>	- 2 - 4M, Stripe-Größe (4k-128k empfohlen)
device <device>	- Partiton (/dev/hda1,/dev/sda1)
raid-disk <index>	- Raid-Plattenindex im Raid-System (0..7)
spare-disk <index>	- Spare-Plattenindex im Raid-System
parity-disk <index>	- Parity-Plattenindex
failed-disk <index>	

/proc/mdstat enthält Statusinformationen über den Zustand der Raidsysteme eines Systems.

Kommandos
----

mkraid – neues Raid anlegen
mkraid [-c configfile] [-f] [-h] [-o] [-V] /dev/md?

raidstart – Raid-System starten
raidstart [-c configfile] [-a] [-h] [-V] /dev/md?

raidstop – Raid-System stoppen
raidstop [--configfile configfile] [--all] [--help]
[--version] /dev/md?

raidhotadd – neue Platte ins Raidsystem einfuegen
raidhotadd [--configfile configfile] [--all] [--help]
[--version] /dev/md?

raidhotremove – defekte Platte aus Raidsystem ausgliedern
raidhotremove [--configfile configfile] [--all] [--help]
[--version] /dev/md?

Kochrezept:
1. **/etc/raidtab anlegen**
2. **mkraid /dev/md0**
3. **mke2fs -b 4096 -R stride=4 /dev/md0**
4. **mkdir /usr3**
5. **mount /dev/md0 /usr3**

Beispiel:

```
raiddev /dev/md0
        raid-level      5
        nr-raid-disks   3
        nr-spare-disks  0
        persistent-superblock   1
        parity-algorithm        left-symmetric
        chunk-size      16
        device  /dev/hdb1
        raid-disk       0
        device  /dev/hdc1
        raid-disk       1
        device  /dev/hdd1
        raid-disk       2

raiddev /dev/md1
        raid-level      0
        nr-raid-disks   3
        persistent-superblock   1
        chunk-size      4
        device  /dev/hdb2
        raid-disk       0
        device  /dev/hdc2
        raid-disk       1
        device  /dev/hdd2
        raid-disk       2
```

Raid unter LINUX

2.Ansatz (Kernel 2.6)

Unterstützt Raid0, Raid1, Raid4, Raid5, Raid6, Raid10, Multipath
Alle Operationen zur Konfiguration des Raid-Systems werden durch

mdadm

realisiert. Funktionen von mdadm

- **create** – erzeugen eines neuen Array mit Superblock
- **build** – erzeugen eines neuen Arrays ohne Superblock
- **assemble** – einbinden eines zuvor erzeugten Arrays
- **monitor** – überwachen eines md-Devices
- **manage** – add oder remove Arrays für ein md-Device

/etc/mdadm.conf – Beschreibt existierende Raid-Systeme
für das jeweilige System

```
gruenau4:/etc # cat mdadm.conf
ARRAY /dev/md127 metadata=1.0 name=141.20.21.167:2
                                  UUID=21ff187f:f84be895:a38a4db0:ede247b9
ARRAY /dev/md126 metadata=1.0 name=141.20.21.167:1
                                  UUID=dd8a6d3c:59d1251d:b2a55a93:685e7b3d
ARRAY /dev/md125 metadata=1.0 name=141.20.21.167:0
                                  UUID=8bfb774c:b01a98b9:1d48b1be:aeb77d71

mdadm --query --detail /dev/md0     # MD-Gerät
mdadm --query --examine /dev/sda1   # SD-Gerät
```

Beispiel:

Erstellen eines Software-Raid-1
vorhanden: /dev/hdb1 (z.Z. genutzt als /data)
/dev/hdc1 (neu eingebaut) – an einem zweiten Controler

md0 erzeugen (Raid 1 (-l 1), 2 Platten, eine missing):

```
mdadm -C /dev/md0 -l 1 -n 2 missing /dev/hdc1
```

Filesystem auf md0 erzeugen:

```
mkfs.ext4 /dev/md0
   oder
 mkreiserfs /dev/md0
```

Mounten von md0:

```
mkdir /mnt/neu
mount /dev/md0 /mnt/neu
```

Daten von /data hach md0 kopieren:

```
rsync -avH --progress /data /mnt/neu

umount /mnt/neu
```

Austauschen von hdb1 und md0 für /data:

```
umount /data      # entmounten von /data
# editieren von /etc/fstab
#   /dev/md0  /data ext4  auto,rw  1 3
mount /data      # mounten von /data
```

Für Neugierige:

```
cat /proc/mdstat
```

Zweite Platte zu md0 hinzufügen:

```
mdadm /dev/md0 −a /dev/hdb1
#  Nocheinmal für Neugierige:
cat /proc/mdstat
```

LVM – Logischer Volume Manager, LSM – Logical Storage Manager
VM – Volume Manager

Standardmäßig bei AIX, HP-UX und Tru64 benutzt.
Optional bei Solaris und Linux möglich.

Vorteile von LVM:
- **Dateisysteme und Dateien können größer als einzelne Platten sein.**
- **Dateisysteme können vergrößert werden, ohne daß Regeneration notwendig ist.**
- **Software-Mirroring und Raid werden unterstützt.**
- **Disk-Striping wird unterstützt.**

Achtung!!!! Begriffsbildung nicht einheitlich!!!!

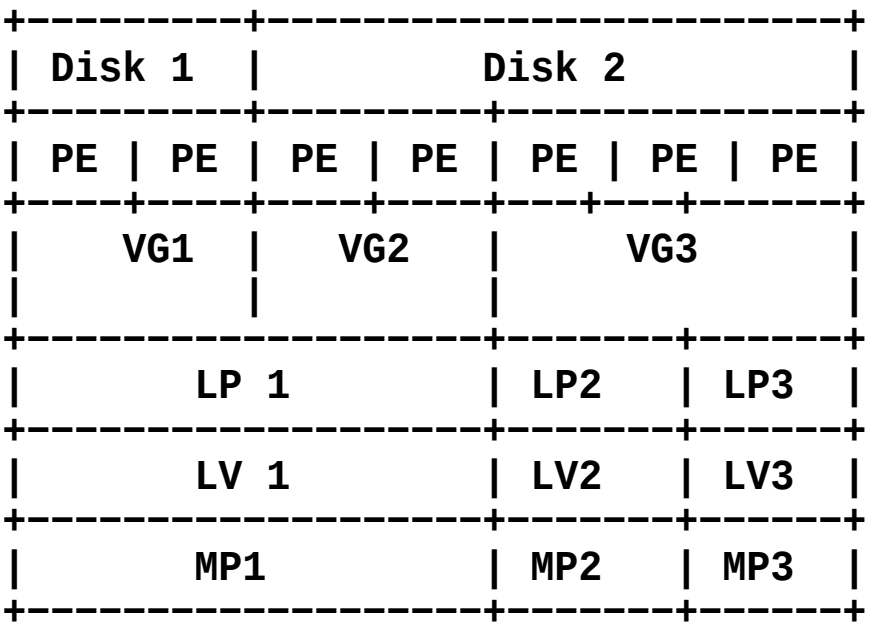
Physical Volume (PV):
die gute alte Festplatte/Partition im formatierten Zustand.

Physical Partition (pysical Extent) (PE):
Kleinste Menge von zusammenhängenden Blöcken in einem physical Volume, die verwaltet werden kann.
1..256 MB gross. Standard 4 MB.

Volume Group (VG):
Menge von Physical Volumes von einer oder mehreren Festplatten (Physical Volumes).

Logical Partition (Logical Extents):
Menge von physical Extens zur Aufnahme von Daten.
Einer logical Partition können pysical Extents aus
mehrere Volume Groups (bis zu drei) zugeordnet sein
(2,3 Datenspiegelung).

Logical Volume (LV):
Menge von Logical Partitions (logical Extents)
die ein Filesystem bilden. Auf einen logischen Volume
kann ein Filesystem aufgesetzt werden(alte Partition).



Bemerkungen:

Volume Groups können vergrössert und verkleinert werden.
Logical Volumes können vergrössert und verkleinert werden.
Durch die Verteilung eines Logical Volumes auf mehrere Physical Partitions mehrerer Festplatten ist Striping möglich.
Durch die Zuordnung von mehreren Physical Partitions zu einer Logical Partition ist Datenspiegelung möglich

LVM-Begriffe im Vergleich

Begriff/OS	AIX	HP-UX	Linux	Solaris	Tru64	Linux
Manager	LVM	LVM	LVM	VM	LSM	LVM
Virtuelle Platte	Volume Group	Volume Group	Volume Group	Volume	Disk-Group	Volume-Group
Logical Volume	Logical Volume	Logical Volume	Logical Volume	Soft Partition	Volume	Logical Volume
Verteilungs-einheit	Extent	Extent	Extent	Extent	Extent	Extent

Kommandos für AIX (Der Erfinder)

```
 Verwaltung von Festplatten:
    Erzeugen eines neuen Gerätes:    mkdev
    Ändern eines Gerätes:            chdev
 Verwaltung von Volumegruppen:
    Erzeugen einer Volumegruppe:
          mkvg
    Erweitern einer Volumegruppe:
          extendvg, chvg
    Verringern einer Volumegruppe:
          reducevg
    Volumegruppe ans System anhängen:
          importvg
    Volumegruppe vom System abhängen:
          exportvg
    Volumegruppe aktivieren:
          varyonvg
    Volumegruppe deaktivieren:
          varyoffvg
 Verwalten von Logische Volumes:
    Erzeugen eines Logischen Volumes:
          mklv
    Ändern eines Logischen Volumes:
          chlv, extendlv
    Streichen eines Logischen Volumes:
          rmlv
 Verwalten von Filesystemen:
    Erzeugen eines Filesystems auf einem logischen
    Volume:   crfs
```

```
 Diverse List-Kommandos:
       lsdev -C -c disk   - Platten anzeigen
       lspv  - Festplatten
       lsvg  - Volumegruppen
       lsfs  - Filesysteme
       lsvgfs- Filesysteme einer Volumegruppe
       lslv  - Logical Volumes

Beispiele:

Erzeugen eines neuen Filesystems:

 mkdev  hdisk3
 mkdev  hdisk4                          - Festplatten verfügbar machen
 mkvg -y "volgrp" hdisk3 hdisk4         - neue Volumegruppe def.
 varyonvg volgrp                        - Volumegruppe aktivieren
 mkvg   hdisk5
 extendvg volgrp hdisk5                 - neue Disk in Volumegruppe
 mklv -y "logvol" volgrp 100  hdisk3 - neuer logical Volume mit 100
                                         logischen Partitionen def.
                                          = 400 MB
 mklv -y "strip" -S 32K 100 hdisk3 hdisk4
                                        - ... mit Striping
 crfs -v jfs -d logvol  -m /mountpoint -A yes
                                        - Filesystem erzeugen
 mount /mountpoint                      - erstes Mounten von Hand
```

Abhängen einer defekten Platte:

!!!!!!defekte Platte muss noch im System sein!!!!

```
umount /mountpoint          – Platte entmounten
rmfs   /mountpoint          – Filesystem streichen
rmlvcopy logvol 2 hdisk3    – Spiegelung entfernen(falls vorhanden)
chps –a n paging–bereich    – Paging–Bereich deaktivieren
shutdown –r now             – reboot
chpv –v r hdisk3            – Festplatte deaktivieren
reducevg volgrp hdisk3      – Festplatte aus Volumegroup
rmdev –l hdisk3 –d          – Gerätedatei streichen
```

Kommandos für Tru64

Manipulation von Physical Volumes:

- pvcreate
- pvdisplay
- pvchange
- pvmove

Manipulation von Volume Group:

- vgcreate
- vgdisplay
- vgchange
- vgextend
- vgreduce
- vgremove
- vgsync

Manipulation von Logical Volumes

- lvcreate
- lvdisplay
- lvchange
- lvextend
- lvreduce
- lvremove
- lvsync

Beispiel:

Mirror:

```
pvcreate −t rz55 /dev/rrz1c
pvcreate −t rz55 /dev/rrz2c
mkdir /dev/vg16
mknod /dev/vg16/group c 16 0
vgcreate /dev/vg16 /dev/rz1c /dev/rz2c
pvdisplay /dev/rz1c  (anzeigen der physical Extents)
lvcreate −s y −l 316 −m −1 /dev/vg16
newfs /dev/vg16/lvol1 rz55
```

Kommandos für Solaris > 2.7

Zentraler Begriff: Metadevice

Programme:

- metadb – Erzeugen und Streichen von Replicas der Datenbasis für Metadevices (Es werden midestens 3 Kopien der Datenbasis an mindestens zwei Controlern gewünscht)
- solstice – Grafische Oberfläche zum Verwalten von Metadevices (nicht mehr bei Solaris 10)

Beispiel: Spiegelung einer Systemplatte

- Datenbasis (Replicas) erzeugen:
  - metadb –a –f –c 2 c0t0d0s7
    c0t0d0s7 freie kleine Partition
  - metadb –a –f –c 2 c0t1d0s7
    c0t0d0s7 freie kleine Partition
- reboot
- solstice (disksuite) aufrufen
  1. Filesystem der ersten Festplatte (z.B. /) in das Concat/Stripe–Device ziehen (submirror1)
  2. zweites Concat/Stripe–Device erzeugen (submirror 2)
  3. submirror1 in ein mirror–Device ziehen und commit betätigen (one–way–mirror)
- reboot
  solstice (disksuite) aufrufen
  1. submirror2 in das mirror–Device ziehen und commit betätigen (Synchronisation des Filesystems beginnt)

  wiederholen für alle Filesysteme

```
Fehlerfall – (disk0 – bootplatte) defekt:
  System rebooten:
     boot disk1
           bootet nur in single–user–mode
           root–passwort eingeben, Replica auf defekter Platte
           streichen:
                 metadb –d c0t0d0s7
     halt
     setenv boot–device disk1
     boot
           Rechner arbeitet wieder , jedoch nur mit einer Festplatte

  Spiegelung wiederherstellen
     – neue Festplatte für disk0 einsetzen und
       möglichst deckungsgleich zu disk1 partitionieren
     – booten
     – Replica auf disk0 einrichten mit
          metadb –a –c 2 c0t0d0s7
                    c0t0d0s7 freie kleine Partition disk0
     – solstice (disksuite) aufrufen:
        slice doppelt anklicken
        enable einer Partition
        commit (Synchronisation des Filesystems beginnt)
     – wiederholen für alle Filesysteme
     – halt
     – setenv boot–device disk
     – boot
    und alles ist wie vorher
```

Anstelle der grafischen Oberfläche muß der Guru ab Solaris 10 Kommandos benutzen:

```
  metastat    – Status anschauen
  metadb      –  Erzeugen und Verwalten der Datenbank
  metatinit   –  Erzeugen und Konfigurieren von Metadevices
  metaparam   –  Parameter der Metadevices einstellen
  metattach   –  Platte/Partition hinzufügen
  metadetach  –  Platte/Partition abhängen
  metareplace–  Platte/Partition ersetzen
  metaclear   –  Status säubern
  metaoffline – offline
  metaonline  – online
  metaroot    – Gerät als Root–Filesystem festlegen
  metaset
  metasync    – sync metadb
```

Konfigurations–Files:

```
   /etc/lvm/md.cf,
   /etc/lvm/md.tab  – für metainit und mdetadb zur Konfiguration
                      von Metadevices
   /etc/lvm/mddb.cf – von metadb erzeugt – nie verändern!!!!
```

Beispiel Solaris 2.10
(ohne graphische Oberfläche)

Plattenstruktur des Mailserver "mail":

Platte 0 – c1t0d0 Spiegel: Platte 2 – c1t2d0

Part	Tag	Flag	Cylinders	Size	Blocks	
0	root	wm	0 – 7088	9.77GB	(7089/0/0) 20480121	/
1	swap	wu	7089 – 8506	1.95GB	(1418/0/0) 4096602	swap
2	backup	wm	0 – 24619	33.92GB	(24620/0/0) 71127180	
3	unassigned	wm	0	0	(0/0/0) 0	
4	unassigned	wm	0	0	(0/0/0) 0	
5	unassigned	wm	0	0	(0/0/0) 0	
6	unassigned	wm	8507 – 24548	22.10GB	(16042/0/0) 46345338	space1
7	unassigned	wm	24549 – 24619	100.16MB	(71/0/0) 205119	metadb

Platte 1 – c1t1d0 Spiegel: Platte 3 – c1t3d0

Part	Tag	Flag	Cylinders	Size	Blocks	
0	unassigned	wm	0	0	(0/0/0) 0	
1	unassigned	wm	0	0	(0/0/0) 0	
2	backup	wm	0 – 24619	33.92GB	(24620/0/0) 71127180	
3	var	wm	0 – 5671	7.81GB	(5672/0/0) 16386408	/var
4	unassigned	wm	5672 – 7089	1.95GB	(1418/0/0) 4096602	/tmp
5	unassigned	wm	0	0	(0/0/0) 0	
6	unassigned	wm	7090 – 24548	24.05GB	(17459/0/0) 50439051	space2
7	unassigned	wm	24549 – 24619	100.16MB	(71/0/0) 205119	metadb

Aktivieren von Metadb:

```
metadb –a –f –c 2 c1t0d0s7 c1t1d0s7 c1t2d0s7 c1t3d0s7
   Initialisieren Datenbasis mit je 2 Replica
      auf der 7.Partition jeder Platte
```

Aufteilung der logischen Plattenname:

```
  /   –   d10
          d11   –   c1t0d0s0
          d12   –   c1t2d0s0
 swap –   d20
          d21   –   c1t0d0s1
          d22   –   c1t2d0s1
 /var –   d30
          d31   –   c1t1d0s3
          d32   –   c1t3d0s3
/tmp  –   d40
          d41   –   c1t1d0s4
          d42   –   c1t3d0s4
space1–   d50
          d51   –   c1t0d0s6
          d52   –   c1t2d0s6
space2–   d60
          d61   –   c1t1d0s6
          d62   –   c1t3d0s6
```

Einrichten der Spiegel

root Spiegeln

```
 1.Teil
    metainit -f d11 1 1 c1t0d0s0
    metainit    d12 1 1 c1t2d0s0
    metainit    d10 -m d11
    metaroot d10
    lockfs -fa
    reboot
 2.Teil
    metattach d10 d12
```

swap Spiegeln

```
    metainit -f d21 1 1 c1t0d0s1
    metainit    d22 1 1 c1t2d0s1
    metainit    d20 -m d21
```

/var Spiegeln

```
    metainit -f d31 1 1 c1t1d0s3
    metainit    d32 1 1 c1t3d0s3
    metainit    d30 -m d31
```

```
/tmp Spiegeln
-------------
    metainit -f d41 1 1 c1t1d0s4
    metainit    d42 1 1 c1t3d0s4
    metainit    d40 -m d41

space1  - Plattencontainer 1 Spiegeln
-------------------------------------
    metainit    d51 1 1 c1t0d0s6
    metainit    d52 1 1 c1t2d0s6
    metainit    d50 -m d51

space2  - Plattencontainer 2 Spiegeln
-------------------------------------
    metainit    d61 1 1 c1t1d0s6
    metainit    d62 1 1 c1t3d0s6
    metainit    d60 -m d61
    reboot
    metattach d20 d22
    metattach d30 d32
    metattach d40 d42
    metattach d50 d52
    metattach d60 d62
```

Verwalten von logischen Platten

Anlegen einer logische Platte (Soft-Partition d101) mit 5 GByte in d50

```
metainit d101 -p d50 5g
newfs /dev/md/rdsk/d101
       neues Filesystem erzeugen
```

Anlegen einer Logische Platte (Soft-Partition d102) mit 10 GByte in d60

```
metainit d102 -p d60 10g
newfs /dev/md/rdsk/d102
       neues Filesystem erzeugen
```

/etc/vfstab-Eintrag:

```
  mkdir -p /zones/mail           # d101
  mkdir -p /zones/mail-data      # d102
  /dev/md/dsk/d101   /dev/md/rdsk/d101      /zones/mail ufs 2 yes -
  /dev/md/dsk/d102   /dev/md/rdsk/d102      /zones/mail-data ufs 2 yes -
```

Kommandos für meta-Geräte-Verwaltung:

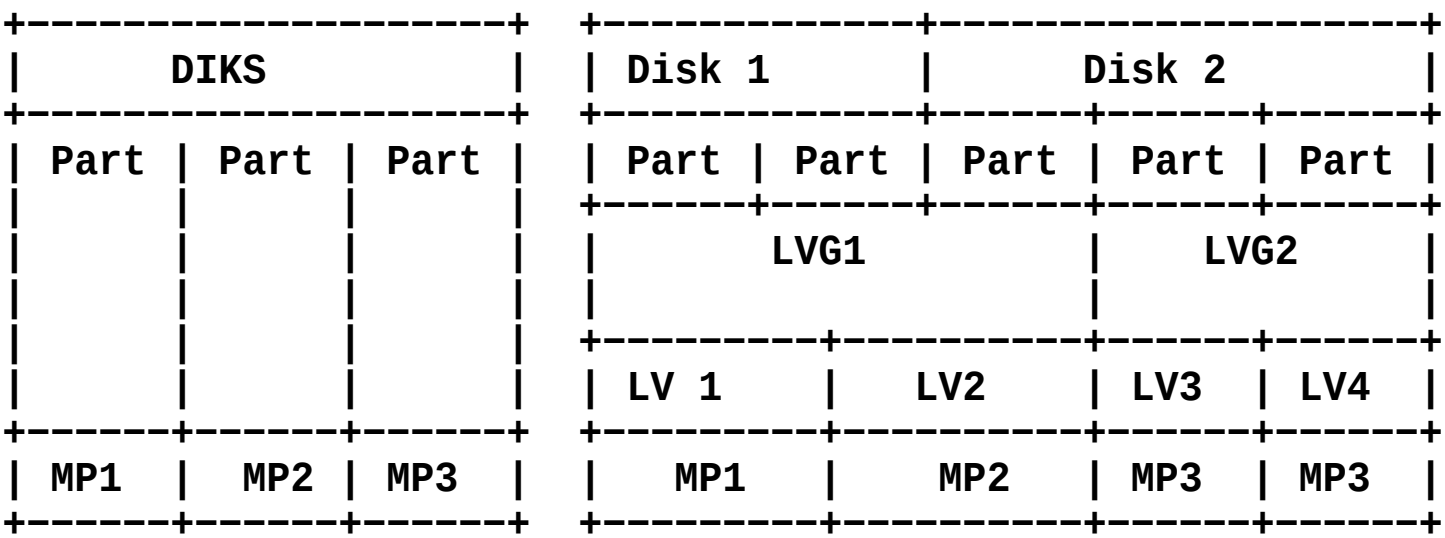
Vergrössern der Soft-Partition d101 um 2 GByte

```
   metattach d101 2g
   growfs -M /zones/mail d101 /dev/md/rdsk/d101
```

Erzeugen von /etc/lvm/md.tab

```
     metastat -p >/etc/lvm/md.tab
```

LVM unter LINUX



Part: klassische physikalische Partition
LVG: logische Volume Group (auch VG – Volume Group)
LV: Logischer Volume
MP: Mount Punkt (logische Partition)

Kommandos:

Masterkommando: lvm

```
lvm help              Liste der Subkommandos
lvm help <command>    Beschreibung der Subkommmandos
```

```
Subkommandos

  help            Display help for commands

  dumpconfig      Dump active configuration
                  dumpconfig <filename>

  formats         List available metadata formats

  lvchange        Change the attributes of logical volume(s)
                   [-A|--autobackup y|n]
                   [-a|--available [e|l]y|n]
                   [--addtag Tag]
                   [--alloc AllocationPolicy]
                   [-C|--contiguous y|n]
                   [-d|--debug]
                   [--deltag Tag]
                   [-f|--force]
                   [-h|--help]
                   [--ignorelockingfailure]
                   [-M|--persistent y|n] [--major major] [--minor minor]
                   [-P|--partial]
                   [-p|--permission r|rw]
                   [-r|--readahead ReadAheadSectors]
                   [--refresh]
                   [-t|--test]
                   [-v|--verbose]
                   [--version]
                   LogicalVolume[Path] [LogicalVolume[Path]...]
```

```
lvcreate        Create a logical volume (1)

      lvcreate
                  [-A|--autobackup {y|n}]
                  [--addtag Tag]
                  [--alloc AllocationPolicy]
                  [-C|--contiguous {y|n}]
                  [-d|--debug]
                  [-h|-?|--help]
                  [-i|--stripes Stripes [-I|--stripesize StripeSize]]
                  {-l|--extents LogicalExtentsNumber |
                   -L|--size LogicalVolumeSize[kKmMgGtT]}
                  [-M|--persistent {y|n}] [--major major] [--minor minor]
                  [-n|--name LogicalVolumeName]
                  [-p|--permission {r|rw}]
                  [-r|--readahead ReadAheadSectors]
                  [-t|--test]
                  [--type VolumeType]
                  [-v|--verbose]
                  [-Z|--zero {y|n}]
                  [--version]
                  VolumeGroupName [PhysicalVolumePath...]
```

```
lvcreate        Create a logical volume (2)

     lvcreate -s|--snapshot

                  [-c|--chunksize]
                  [-A|--autobackup {y|n}]
                  [--addtag Tag]
                  [--alloc AllocationPolicy]
                  [-C|--contiguous {y|n}]
                  [-d|--debug]
                  [-h|-?|--help]
                  [-i|--stripes Stripes [-I|--stripesize StripeSize]]
                  {-l|--extents LogicalExtentsNumber |
                   -L|--size LogicalVolumeSize[kKmMgGtT]}
                  [-M|--persistent {y|n}] [--major major] [--minor minor]
                  [-n|--name LogicalVolumeName]
                  [-p|--permission {r|rw}]
                  [-r|--readahead ReadAheadSectors]
                  [-t|--test]
                  [-v|--verbose]
                  [--version]
                  OriginalLogicalVolume[Path] [PhysicalVolumePath...]
```

```
lvdisplay       Display information about a logical volume
lvextend        Add space to a logical volume
lvmchange       With the device mapper, this is obsolete
                and does nothing.
lvmdiskscan     List devices that may be used as physical
                volumes
lvmsadc         Collect activity data
lvmsar          Create activity report
lvreduce        Reduce the size of a logical volume
lvremove        Remove logical volume(s) from the system
lvrename        Rename a logical volume
lvresize        Resize a logical volume
lvs             Display information about logical volumes
lvscan          List all logical volumes in all volume groups

pvchange        Change attributes of physical volume(s)
pvcreate        Initialize physical volume(s) for use by LVM
pvdata          Display the on-disk metadata for physical volume(s)
pvdisplay       Display various attributes of physical volume(s)
pvmove          Move extents from one physical volume to another
pvremove        Remove LVM label(s) from physical volume(s)
pvresize        Resize a physical volume in use by a volume group
pvs             Display information about physical volumes
pvscan          List all physical volumes

segtypes        List available segment types
```

```
vgcfgbackup     Backup volume group configuration(s)
vgcfgrestore    Restore volume group configuration
vgchange        Change volume group attributes
vgck            Check the consistency of volume group(s)
vgconvert       Change volume group metadata format
vgcreate        Create a volume group
vgdisplay       Display volume group information
vgexport        Unregister volume group(s) from the system
vgextend        Add physical volumes to a volume group
vgimport        Register exported volume group with system
vgmerge         Merge volume groups
vgmknodes       Create the special files for volume group
                devices in /dev
vgreduce        Remove physical volume(s) from a volume group
vgremove        Remove volume group(s)
vgrename        Rename a volume group
vgs             Display information about volume groups
vgscan          Search for all volume groups
vgsplit         Move physical volumes into a new volume group
version         Display software and driver version information
```

Konfiguration:

```
    Directory:  /etc/lvm
                        lvm.conf  – zentrales Konfigurationsfile
                        archive   – Archiv–Directory
                        backup    – Backup–Directory
                        metadata  – Directory Metadaten
```

Beispiele für /etc/lvm/lvm.conf:

```
# This is an example configuration file for the LVM2 system.
# It contains the default settings that would be used if there was no
# /etc/lvm/lvm.conf file.
#
# Refer to 'man lvm.conf' for further information including the file layout.
#
# To put this file in a different directory and override /etc/lvm set
# the environment variable LVM_SYSTEM_DIR before running the tools.
# This section allows you to configure which block devices should
# be used by the LVM system.
devices {
    # Where do you want your volume groups to appear ?

    dir = "/dev"

    # An array of directories that contain the device nodes you wish
    # to use with LVM2.

    scan = [ "/dev" ]
```

```
    # A filter that tells LVM2 to only use a restricted set of devices.
    # The filter consists of an array of regular expressions.  These
    # expressions can be delimited by a character of your choice, and
    # prefixed with either an 'a' (for accept) or 'r' (for reject).
    # The first expression found to match a device name determines if
    # the device will be accepted or rejected (ignored).  Devices that
    # don't match any patterns are accepted.
    # Be careful if there there are symbolic links or multiple filesystem
    # entries for the same device as each name is checked separately against
    # the list of patterns.  The effect is that if any name matches any 'a'
    # pattern, the device is accepted; otherwise if any name matches any 'r'
    # pattern it is rejected; otherwise it is accepted.
    # Remember to run vgscan after you change this parameter to ensure
    # that the cache file gets regenerated (see below).
    # By default we accept every block device:

    filter = [ "a/.*/" ]

    # Exclude the cdrom drive
    # filter = [ "r|/dev/cdrom|" ]
    # When testing I like to work with just loopback devices:
    # filter = [ "a/loop/", "r/.*/" ]
    # Or maybe all loops and ide drives except hdc:
    # filter =[ "a|loop|", "r|/dev/hdc|", "a|/dev/ide|", "r|.*|" ]
    # Use anchors if you want to be really specific
    # filter = [ "a|^/dev/hda8$|", "r/.*/" ]
```

```
    # The results of the filtering are cached on disk to avoid
    # rescanning dud devices (which can take a very long time).  By
    # default this cache file is hidden in the /etc/lvm directory.
    # It is safe to delete this file: the tools regenerate it.

    cache = "/etc/lvm/.cache"

    # You can turn off writing this cache file by setting this to 0.

    write_cache_state = 1

    # Advanced settings.
    # List of pairs of additional acceptable block device types found
    # in /proc/devices with maximum (non-zero) number of partitions.
    # types = [ "fd", 16 ]
    # If sysfs is mounted (2.6 kernels) restrict device scanning to
    # the block devices it believes are valid.
    # 1 enables; 0 disables.

    sysfs_scan = 1

    # By default, LVM2 will ignore devices used as components of
    # software RAID (md) devices by looking for md superblocks.
    # 1 enables; 0 disables.

    md_component_detection = 1

}
```

```
# This section that allows you to configure the nature of the
# information that LVM2 reports.

log {
    # Controls the messages sent to stdout or stderr.
    # There are three levels of verbosity, 3 being the most verbose.

    verbose = 0

    # Should we send log messages through syslog?
    # 1 is yes; 0 is no.

    syslog = 1

    # Should we log error and debug messages to a file?
    # By default there is no log file.
    #file = "/var/log/lvm2.log"
    # Should we overwrite the log file each time the program is run?
    # By default we append.

    overwrite = 0

    # What level of log messages should we send to the log file and/or syslog?
    # There are 6 syslog-like log levels currently in use - 2 to 7 inclusive.
    # 7 is the most verbose (LOG_DEBUG).

    level = 0
```

```
    # Format of output messages
    # Whether or not (1 or 0) to indent messages according to their severity

    indent = 1

    # Whether or not (1 or 0) to display the command name on each line output

    command_names = 0

    # A prefix to use before the message text (but after the command name,
    # if selected).  Default is two spaces, so you can see/grep the severity
    # of each message.

    prefix = "  "

    # To make the messages look similar to the original LVM tools use:
    #   indent = 0
    #   command_names = 1
    #   prefix = " -- "
    # Set this if you want log messages during activation.
    # Don't use this in low memory situations (can deadlock).
    # activation = 0
}
```

```
# Configuration of metadata backups and archiving.  In LVM2 when we
# talk about a 'backup' we mean making a copy of the metadata for the
# *current* system.  The 'archive' contains old metadata configurations.
# Backups are stored in a human readeable text format.

backup {
    # Should we maintain a backup of the current metadata configuration ?
    # Use 1 for Yes; 0 for No.
    # Think very hard before turning this off!

    backup = 1

    # Where shall we keep it ?
    # Remember to back up this directory regularly!

    backup_dir = "/etc/lvm/backup"

    # Should we maintain an archive of old metadata configurations.
    # Use 1 for Yes; 0 for No.
    # On by default.  Think very hard before turning this off.

    archive = 1

    # Where should archived files go ?
    # Remember to back up this directory regularly!

    archive_dir = "/etc/lvm/archive"
```

```
    # What is the minimum number of archive files you wish to keep ?

    retain_min = 10

    # What is the minimum time you wish to keep an archive file for ?

    retain_days = 30
}

# Settings for the running LVM2 in shell (readline) mode.



shell {
    # Number of lines of history to store in ~/.lvm_history

    history_size = 100

}
```

```
# Miscellaneous global LVM2 settings
global {
    # The file creation mask for any files and directories created.
    # Interpreted as octal if the first digit is zero.

    umask = 077

    # Allow other users to read the files
    #umask = 022
    # Enabling test mode means that no changes to the on disk metadata
    # will be made.  Equivalent to having the -t option on every
    # command.  Defaults to off.

    test = 0

    # Whether or not to communicate with the kernel device-mapper.
    # Set to 0 if you want to use the tools to manipulate LVM metadata
    # without activating any logical volumes.
    # If the device-mapper kernel driver is not present in your kernel
    # setting this to 0 should suppress the error messages.

    activation = 1

    # If we can't communicate with device-mapper, should we try running
    # the LVM1 tools?
    # This option only applies to 2.4 kernels and is provided to help you
    # switch between device-mapper kernels and LVM1 kernels.
    # The LVM1 tools need to be installed with .lvm1 suffices
    # e.g. vgscan.lvm1 and they will stop working after you start using
```

```
    # the new lvm2 on-disk metadata format.
    # The default value is set when the tools are built.
    # fallback_to_lvm1 = 0
    # The default metadata format that commands should use - "lvm1" or "lvm2".
    # The command line override is -M1 or -M2.
    # Defaults to "lvm1" if compiled in, else "lvm2".
    # format = "lvm1"
    # Location of proc filesystem

    proc = "/proc"

    # Type of locking to use. Defaults to file-based locking (1).
    # Turn locking off by setting to 0 (dangerous: risks metadata corruption
    # if LVM2 commands get run concurrently).

    locking_type = 1

    # Local non-LV directory that holds file-based locks while commands are
    # in progress.  A directory like /tmp that may get wiped on reboot is OK.

    locking_dir = "/var/lock/lvm"

    # Other entries can go here to allow you to load shared libraries
    # e.g. if support for LVM1 metadata was compiled as a shared library use
    #   format_libraries = "liblvm2format1.so"
    # Full pathnames can be given.
    # Search this directory first for shared libraries.
    #   library_dir = "/lib"
}
```

```
activation {
    # Device used in place of missing stripes if activating incomplete volume.
    # For now, you need to set this up yourself first (e.g. with 'dmsetup')
    # For example, you could make it return I/O errors using the 'error'
    # target or make it return zeros.
    missing_stripe_filler = "/dev/ioerror"

    # Size (in KB) of each copy operation when mirroring
    mirror_region_size = 512

    # How much stack (in KB) to reserve for use while devices suspended
    reserved_stack = 256

    # How much memory (in KB) to reserve for use while devices suspended
    reserved_memory = 8192

    # Nice value used while devices suspended
    process_priority = -18

    # If volume_list is defined, each LV is only activated if there is a
    # match against the list.
    #   "vgname" and "vgname/lvname" are matched exactly.
    #   "@tag" matches any tag set in the LV or VG.
    #   "@*" matches if any tag defined on the host is also set in the LV or VG
    #
    # volume_list = [ "vg1", "vg2/lvol1", "@tag1", "@*" ]
}
```

```
####################
# Advanced section #
####################
# Metadata settings
#
# metadata {
    # Default number of copies of metadata to hold on each PV.  0, 1 or 2.
    # You might want to override it from the command line with 0
    # when running pvcreate on new PVs which are to be added to large VGs.
    # pvmetadatacopies = 1
    # Approximate default size of on-disk metadata areas in sectors.
    # You should increase this if you have large volume groups or
    # you want to retain a large on-disk history of your metadata changes.
    # pvmetadatasize = 255
    # List of directories holding live copies of text format metadata.
    # These directories must not be on logical volumes!
    # It's possible to use LVM2 with a couple of directories here,
    # preferably on different (non-LV) filesystems, and with no other
    # on-disk metadata (pvmetadatacopies = 0). Or this can be in
    # addition to on-disk metadata areas.
    # The feature was originally added to simplify testing and is not
    # supported under low memory situations - the machine could lock up.
    #
    # Never edit any files in these directories by hand unless you
    # you are absolutely sure you know what you are doing! Use
    # the supplied toolset to make changes (e.g. vgcfgrestore).
    # dirs = [ "/etc/lvm/metadata", "/mnt/disk2/lvm/metadata2" ]
#}
```

Einrichten eines LVM-Systems unter LINUX

1. Platten Partitionieren und Partitionen mit
   "8E" kennzeichnen – Physical Volume

2. vgscan -v # finden der Partitionen

3. Erstellen der Physical Volumes
   pvcreate /dev/hdb1
   pvcreate /dev/hdc1

4. Einrichten der Volumegruppe
   vgcreate volg1 /dev/hdb1 /dev/hdc1

5. Einrichten des Logischen Volumes

   lvcreate -n logv1 -L 500M volg1

6. Formatieren des Logischen Volumes
   mkfs -t ext4 /dev/volg1/logv1

7. Einhängen in den Verzeichnisbaum

   mkdir /mntpoint
   mount -t ext4 /dev/volg1/logv1 /mntpoint

Verändern von Logischen Volumes

1. Erweitern des Logischen Volumes um 200 MB (auf 700 MB)

```
lvextend -L +200M /dev/volg1/logv1
```

   Vergrößern des Filesystems:

```
umount /mntpoint
e2fsck -f /dev/volg1/logv1
resize2fs /dev/volg1/logv1
mount -t ext4 /dev/volg1/logv1 /mntpoint
```

2. Verkleinern des Logischen Volumes um 200 MB (auf 500 MB)

```
umount /mntpoint
e2fsck -f /dev/volg1/logv1
resize2fs /dev/volg1/logv1 512000  # 1024 Byte_Blöcke
lvreduce -l -200M /dev/volg1/logv1
mount -t ext4 /dev/volg1/logv1 /mntpoint
```

Verändern von Volume Gruppen

1. Volume Gruppe erweitern

```
pvcreate /dev/hdd1  # neuer
vgextend volg1 /dev/hdd1
vgdisplay /dev/volg1   # anzeigen der Volume Gruppe
```

2. Volume Gruppen verkleinern (theoreisch)
   Nur nicht benutze physical Volumes können entfernt werden!!

```
vgreduce −a volg1  # alle freien Physical Volumes entfernen

pvdisplay −v /dev/hdb7 # Anzeigen ob benutzt
pvmove −v /dev/hdd1    # eventuell freischaufeln
vgreduce volg1 /dev/hdd1
```

Solaris 10

ZFS – Ablösung des LVM unter Solaris

ZFS kann nicht:

- Volume Management
- Filesystemchecks
- Logging/Journaling
- RAID5 oder RAID6

braucht es auch nicht, kann dafür aber

- Storage virtualisieren
- immer konsistent bleiben (dank copy-on-write)
- die Datenvalidität garantieren
- kommt auch ohne teures NVRAM aus (ist mit aber schneller)

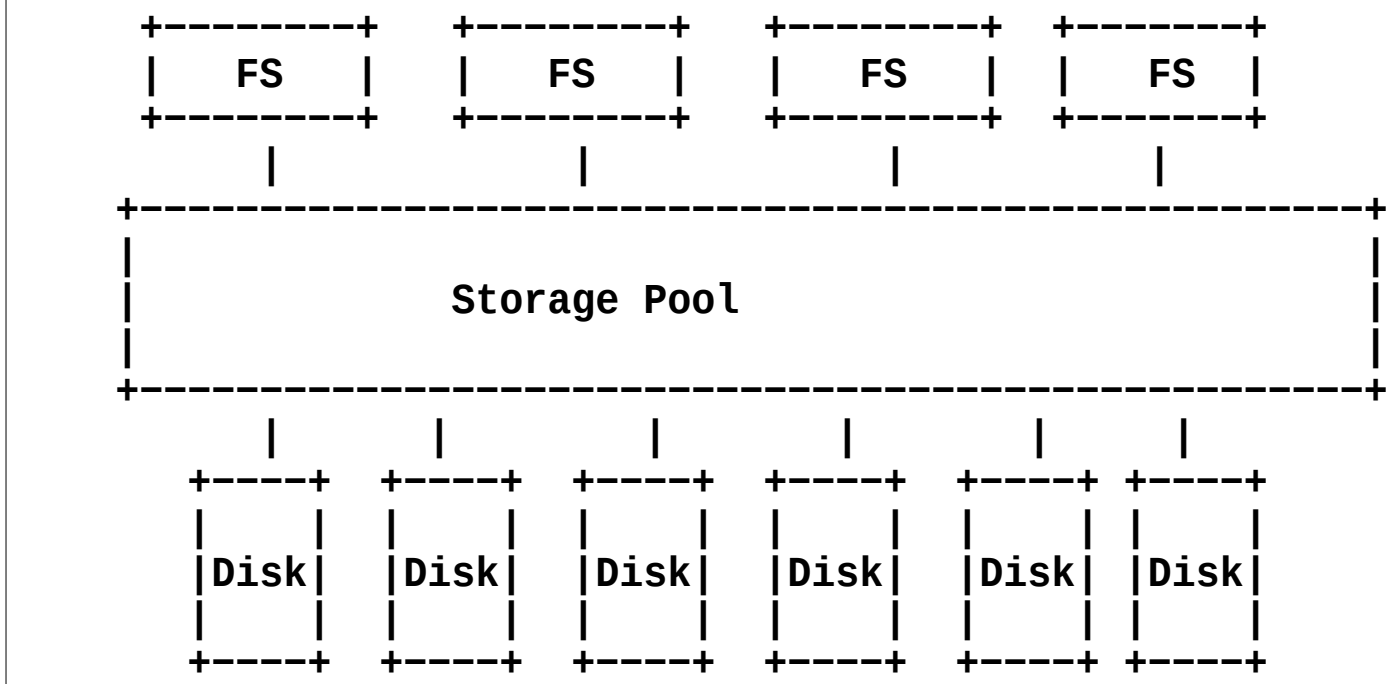
ZFS – Zettabyte File System

128 Bit Adressraum, Name beschreibt nur einen Teil der Kapazität

- Terabyte: ca. 2 hoch 40 Bytes (1 Tebibyte)
- Petabyte: ca. 2 hoch 50 Bytes (1 Bebibyte)
- Zettabyte: ca. 2 hoch 70 Bytes (1 Zebibyte)
- ZFS: ca 2 hoch 128 Bytes oder mehr als 8 Petabytes mal 1 Zettabyte

Einen Speicher dieser Größe ist auf der Erde nicht baubar – zu wenig Materie

Struktur von ZFS



- Keine Volumes
- automatisches Vergrößern und Verkleinern
- Jedem Filesystem steht die Bandbreite des IOPS zur Verfügung
- Ein gemeinsamer Storagepool

Wie stellt ZFS die Datenintegrität sicher?

Alles wird mit copy on write geschrieben
- **Aktive Daten werden nie überschrieben**
- **der Zustand auf der Platte ist niemals inkonsistent**
- **kein Filesystemcheck**

Alles wird transaktionsbasiert geschrieben
- **Alles oder Nichts**
  Achtung!!! Es muß im Storage Pool die Kapatzität für die vollständig Operation vorhanden sein. Auch für rm werden freie Blöcke benötigt!!!!
  Alle zu verändernden Daten werden erst auf die Platte geschrieben, so daß durch Schreiben des letzten Blockes, der konsistente Zustand hergestellt wird. Erst wenn diese Operation erfolgreich war, werden die alten Blöcke gelöscht.
  Dadurch ist ein Snapshot leicht möglich – alte Blöcke nicht löschen.
- **daher kein Journaling notwendig**

Alle Datenblöck werden mit einer Checksumme geschrieben
- **Checksummen werden auch im Parentblock gespeichert**
- **erkennbare Fehler:**
  BitRot, Phantom Writes, Misdirected Reads/Writes, DMA Parity Errors, Driver Errors, versehentliches Überschreiben
- **keine schleichenden Datenkorruption**
- **keine Panics wegen schleichender Datenkorruption bei den Metadaten**

Zuerst ein Beispiel:

/usr1 und /usr2 auf einem Server anlegen (T5200 mit 8 Platten).
2 Platten für das System (Pool SYSTEM), 6 Platten als Raid 5 mit
einer Spare-Platte für Pool DATEN. Pool SYSTEM ist bei der
Installation bereits erzeugt worden.

Erzeugen des Plattenpools "DATEN" aus c1t2d0-c1t6d0 für das Raidsystem
mit Reserverplatte c1t7d0:

```
zpool create DATEN raidz1 c1t2d0 c1t3d0 c1t4d0 c1t5d0 c1t6d0 spare c1t7d0
```

Der Pool ist jetzt unter /DATEN vollständig gemountet.
Löschen des Mountpunktes:

```
zfs set mountpoint=legacy DATEN
```

Anlegen von neuen Filesystems im Pool DATEN:

```
    zfs create DATEN/usr1
    zfs create DATEN/usr2
```

Diese Filesysteme sind jetzt nicht gemountet!!!

Mountpoint anlegen:

```
    zfs set mountpoint=/usr1 DATEN/usr1
    zfs set mountpoint=/usr2 DATEN/usr2
```

Beide Filesysteme haben jetzt die volle Kapazität von DATEN.
Die Blöcke sind also "mehrfach" vergeben. Buchführung stimmt aber,
ein verbrauchter Block wird in beiden Filesystemen abgezogen

Quotas fuer die Filesysteme anlegen

```
zfs set quota=300GB DATEN/usr1
zfs set quota=100GB DATEN/usr2
```

ZFS-Kommandos

Es gibt nur zwei Kommdos

1. zpool – zur Verwaltung der Platten
2. zfs – zur Verwaltung der Filesysteme

Übersicht an Hand von täglichen Aufgaben:

Zur Information für Neugierige:

Anschauen des Pools (Plattenbelegung)

```
zpool status [-x] [-v]
          -x  - nur Status
          -v  - verbose
zpool list [-H]
          -H  -  ohne header
zpool history
zpool iostat [-v] [<sekunden>]
        -v  -   verbose
```

Anschauen der Filesysteme

```
zfs list     -  anzeigen aller Filesysteme
zfs get all  -  anzeigen aller Properties aller Filesysteme
zfs get mountpoint DATEN/usr1   - anzeigen einzelner Properties
                                  einzelner Filesysteme
zfs get quota DATEN/usr1
```

Beispiel:

```
bellus# zpool history
History for 'rpool':
2009-04-20.15:01:12 zpool create -f -o failmode=continue -R /a -m \
            legacy -o cachefile=/tmp/root/etc/zfs/zpool.cache rpool c1t0d0s0
2009-04-20.15:01:13 zfs set canmount=noauto rpool
2009-04-20.15:01:13 zfs set mountpoint=/rpool rpool
2009-04-20.15:01:14 zfs create -o mountpoint=legacy rpool/ROOT
2009-04-20.15:01:15 zfs create -b 8192 -V 8192m rpool/swap
2009-04-20.15:01:15 zfs create -b 131072 -V 4096m rpool/dump
2009-04-20.15:02:42 zfs create -o canmount=noauto rpool/ROOT/system
2009-04-20.15:02:43 zfs create -o canmount=noauto rpool/ROOT/system/var
2009-04-20.15:02:44 zpool set bootfs=rpool/ROOT/system rpool
2009-04-20.15:02:44 zfs set mountpoint=/ rpool/ROOT/system
2009-04-20.15:02:45 zfs set canmount=on rpool
2009-04-20.15:02:46 zfs create -o mountpoint=/export rpool/export
2009-04-20.15:02:46 zfs create rpool/export/home
2009-04-23.10:51:17 zfs create rpool/ROOT/system/tmp
2009-04-23.10:53:09 zfs create rpool/DATEN
2009-04-23.10:53:29 zfs create rpool/DATEN/home
2009-04-23.10:53:53 zfs set mountpoint=legacy rpool/DATEN
2009-04-23.10:56:04 zfs set mountpoint=/home rpool/DATEN/home
2009-04-23.11:31:43 zfs destroy rpool/export/home
2009-04-23.11:31:47 zfs destroy rpool/export
2009-04-23.11:49:40 zpool attach -f rpool c1t0d0s0 c1t1d0s0
2009-04-23.12:01:47 zpool scrub rpool
2009-04-23.12:23:14 zfs set quota=50G rpool/DATEN/home
2009-04-23.12:25:00 zfs set quota=10G rpool/ROOT/system/var
2009-04-23.12:25:06 zfs set quota=10G rpool/ROOT/system/tmp
```

Prüfen des Pools (vorsicht)

```
zpool scrub DATEN   # Achtung!! Alle Blöcke werden gelesen, mehr als fsck
```

Platte ausbauen (wenn genug vorhanden sind, im Beispiel kann man eine ausbauen)

```
zpool offline DATEN c1t5d0
```

Platte eingliedern

```
zpool online DATEN c1t5d0
```

Mirror nach Installation von Solaris mit ZFS-Root-Filesystem einschalten

```
zpool status
```

Platten (Partion)anschauen (in der Regel Partition 0)
Entsprechende Partion auf der neuen Platte mit format anlegen.
Platte als mirror der Orginal-Platte zuordnen

```
zpool attach SYSTEM c1t0d0s0 c1t1d0s0
```

Bootblock auf der 2.Platte installieren:
Notwendig!!! Sehr Wichtig!!!! Sonst geht boot disk1 nicht!!!!!

```
installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk \
                   /dev/rdsk/c1t1d0s0
```

Abhängen von nicht benötigten Filesystemen

```
zfs destroy SYSTEM/export/home
zfs destroy SYSTEM/export
```

Erzeugen von neuen Filesystemen

```
zfs create SYSTEM/DATEN
zfs create SYSTEM/DATEN/usr2
zfs set mountpoint=legacy SYSTEM/DATEN
zfs set mountpoint=/usr2 SYSTEM/DATEN/usr2
```

```
Platte defekt
=============
Platte in einem Spiegel defekt (Systemplatte)
---------------------------------------------
    Platte c1t0d0s0  ist defekt

    Defekte Platte aus Spiegel entfernen

        zpool detach SYSTEM c1t0d0s0
        zpool status SYSTEM

    defekte Platte ausbauen
    neue Platte einbauen und einbinden

         zpool attach SYSTEM c1t1d0s0 c1t0d0s0
         zpool status SYSTEM

    Pruefen und Bootblock installieren

       installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk \
                                                  /dev/rdsk/c1t0d0s0

    warten bis neue Platten ok (zpool status SYSTEM)

        zpool scrub SYSTEM


Hilfe aus dem Internet:
  http://www.opensolaris.org/os/community/zfs/boot/zfsbootFAQ/
```

Platte im Raid defekt (mit Spare-Platte)

Platte c1t2d0 ist defekt

Das ZFS aktiviert automatisch die SPARE-Platte (c1t7d0)

```
zpool status DATEN
```

zeigt den Mirror bestehend aus c1t2d0 und c1t7d0
defekte Platte offline setzen

```
zpool offline DATEN c1t2d0
```

defekte Platte ausbauen
neue Platte einbauen

```
zpool online DATEN c1t2d0
zpool scrub  DATEN
```

warten bis alles wieder ok ist, dann

```
zpool detach DATEN c1t7d0
```

Damit ist die SPARE-Platte wieder verfuegbar

Mit

```
zpool history
```

sieht man, was man alles wann gemacht hat (mit Datum und Uhrzeit)
Ich bin es nicht gewesen!!!!!

Das Kommando zpool

/usr/sbin/zpool

```
zpool [-?]
            Möglichkeiten von zpool anzeigen - help

zpool create [-fn] [-o property=value] ... [-m mountpoint] [-R root]
    pool vdev ...
            Erzeugen eines neuen Pools  pool  bestehend aus
            virtuellen Devices  vdev
            virtuelles Device:
              Platten:   c0t1d0 c0t2d0
              File:      regular File - /date/filesystem
              Mirror:    mirror c0t1d0 c0t2d0
              Raid:      raidz1 c0t1d0 c0t2d0 c0t3d0
                         raidz2 c0t1d0 c0t2d0 c0t3d0 c0t4d0
              Spare:     spare c0t5d0
              Log-Gerät: log c0t0d0
    zpool create pool mirror c0t1d0 c1t1d0 spare c2t1d0 c3t1d0
    zpool create pool c0t1d0 c1td0 log c2t1d0

zpool destroy [-f] pool
            Löschen eines Pools

zpool add [-fn] pool vdev ...
            Hinzufügen eins virtuellen Devices zu einem Pool
```

```
zpool remove pool device ...
          Löschen eines Gerätes aus einem Pool

zpool list [-H] [-o property[,...]] [pool] ...
          Anzeigen  der Eigenschaften eines Pools (Detail-Informationen)

zpool iostat [-v] [pool] ... [interval[count]]
          Anzeigen der Zahl der E/A-Operationen

zpool status [-xv] [pool] ...
          Anzeigen des Status

zpool online pool device ...
          Gerät online setzen

zpool offline [-t] pool device ...
          Gerät offline setzen

zpool clear pool [device]
          Löschen von Gerätefehlern

zpool attach [-f] pool device new_device
          Hinzufügen eines Gerätes zu einem vorhanden Gerät
          (Erzeugen eines Mirrors)

zpool detach pool device
          Abhängen eines Gerätes von einem Mirror
```

```
zpool replace [-f] pool device [new_device]
            Ersetzen eines Grätes (attach, detach)

zpool scrub [-s] pool ...
            Überprüfung aller Daten!!!
            Alle Blöcke werden gelesen und die Prüfsummen überprüft.
            ZFS nimmt sich hierbei alle E/A-Ressourcen, die es bekommen
            kann!!!!!!   SAN!!!!!!

zpool import [-d dir] [-D]
            Auflisten aller Pools, die hinzugefügt werden können
            (inaktive Pools, die zuvor exportiert wurden)

zpool import [-o mntopts] [-p property=value] ... [-d dir | -c cachefile]
                [-D] [-f] [-R root] -a
            Importieren aller  nicht benutzten Pools

zpool import [-o mntopts] [-o property=value] ... [-d dir | -c cachefile]
    [-D] [-f] [-R root] pool |id [newpool]
            Importieren des spezifizierten Pools  pool

zpool export [-f] pool ...
            Abhängen eines Pools (exportieren). Er kann dann durch ein
            anderes System importiert werden (SAN).
```

```
zpool upgrade
            Anzeigen von Pools mit abweichender ZFS-Version

zpool upgrade -v
            Anzeigen aller unterstützten ZFS-Versionen

zpool upgrade [-V version] -a | pool ...
            Upgraden eines oder aller Pools auf eine neue ZFS-Version

zpool history [-il] [pool] ...
            Anzeigen der History

zpool get "all" | property[,...] pool ...
            Anzeigen aller Eigenschaften.
            zpool get all rpool

zpool set property=value pool
            Setzen von Eigenschaften
```

Das Kommando zfs

/usr/sbin/zfs

Ein Filesystem (ZFS dataset) wird durch den Pool-Namen und den Filesystemnamen bestimmt.

```
rpool/SYSTEM      - Pool rpool, Filesystem SYSTEM
rpool/SYSTEM/root - Pool rpool, Filesystem SYSTEM/root
                    Filesystem im Filesystem ist möglich
```

Mehrere Filesysteme können in einem Pool liegen.

```
zfs [-?]
        Hilfe

zfs create [-p] [-o property=value] ... filesystem
        Erzeugen eines Filesystems. Durch den Namen ist die
        Lage im Pool bestimmt (vollständiger Name notwendig)

zfs destroy [-rRf] filesystem|volume|snapshot
        Löschen eines Filesystems bzw Snapshots

zfs snapshot [-r] filesystem@snapname|volume@snapname
        Anlegen eines Snapshots
        zfs snapshot rpool/SYSTEM/root@root1

zfs rollback [-rRf] snapshot
        Zurückspeichern eines Snapshots auf den alten Stand

zfs clone [-p] snapshot filesystem|volume
        Einspielen des Snapshots auf ein neues Filesystem
```

```
zfs promote clone-filesystem
            Abkoppeln des Clone-Filesystems vom Vater

zfs rename [-p] filesystem|volume filesystem|volume
            Umbenennen eines Filesystems

zfs rename -r snapshot snapshot
            Umbenennen eines Snapshots

zfs list [-rH] [-o property[,...]] [-t type[,...]]
    [-s property] ... [-S property ... [filesystem|volume|snapshot] ...
          Anzeigen der Filesystem einschließlich deren Eigenschaften
          zfs list -o all

zfs set property=value filesystem|volume ...
          Setzen von Eigenschaften

zfs get [-rHp] [-o field[,...]] [-s source[,...]] "all" | property[,...]
     filesystem|volume|snapshot ...
          Anzeigen einzelner Eigenschaften.

zfs inherit [-r] property filesystem|volume ...
          Löschen von einzelnen Eigenschaften

zfs upgrade [-v]
          Anzeigen der ZFS-Version bzw. aller unterstützten ZFS Versionen

zfs upgrade [-r] [-V version] -a | filesystem
          upgraden eines Filesystems auf die aktuelle Version
```

```
zfs mount
          Anzeigen der gemounteten ZFS-Filesysteme

zfs mount [-vO] [-o options] -a | filesystem
          Mounten eines ZFS-Filesystems (bei boot)

zfs unmount [-f] -a | filesystem|mountpoint
          Entmounten eines ZFS-Filesystems

zfs share -a | filesystem
          Exportieren eines ZFS-Filesystems (nfs, smb)

zfs unshare  -a filesystem|mountpoint
          Export eines ZFS-Filesystems zurücknehmen

zfs send [-vR] [-[-iI] snapshot] snapshot
          Ausgabe eines Snapshots in einen Stream

zfs receive [-vnF] filesystem|volume|snapshot
          Erzeugen eines Snapshots aus einem Stream  in
          dem angegebenen Filesystem/Volume.
```

```
zfs allow [-ldug] "everyone"|user|group[,...] perm|@isetname[,...]
    filesystem|volume
          Delegieren der ZFS-Administrationsrechte an einen none-root
          Nutzer

zfs unallow [-rldug] "everyone"|user|group[,...] [perm|@setname[,... ]]
    filesystem|volume
          Rücknahme der ZFS-Administrationsrechte für ein ZFS-Volume
          bzw. ZFS-Filesystem
```

Einige Messungen mit ZFS, UFS und EXT3

unzip 10_Recommended.zip (998 MB --> 2.491 MB)

zfs: mirror->raid 5 T2000
```
real    3m49.969s
user    2m8.565s
sys     1m38.960s
```

ufs: mirror->raid 5 T2000
```
real    8m25.185s
user    2m7.709s
sys     1m4.703s
```

ext4: disk1->disk2 X4600
```
real    1m0.167s
user    0m21.945s
sys     0m17.641s
```

ufs: disk1-->disk2 X4600
```
real    3m9.361s
user    0m24.527s
sys     0m33.603s
```

```
find . -print | wc
------------------

  zfs:  raid5            T2000
    real    0m3.314s
    user    0m1.108s
    sys     0m2.432s

  ufs:  raid5            T2000
    real    0m2.045s
    user    0m0.990s
    sys     0m1.278s

  ext4: disk             X4600
    real    0m0.273s
    user    0m0.096s
    sys     0m0.224s

  ufs: disk              X4600
    real    0m0.823s
    user    0m0.098s
    sys     0m0.585s
```

Linux

btrfs – B-tree FS, Butter FS, Better FS

ab SuSE 12.1

Eigenschaften:

- Speicherbereich 2 hoch 64 Byte
- dynamische Inodes
- dynamische Filesystemgröße
- Schnappschüsse
- integriertes Raid
- Prüfsummen für Metadaten und Daten
- Fehlerkorrektur
- Datenkompression und Verschlüsselung
- SSD-Unterstützung
- Dateisystemüberprüfung und Defragmentierung während des Betriebes
- internes inkrementelles Backup
- Copy-On-Write
- EXT-Filesystem-Konvertierung
- boot-Device ?? z.Z. nicht empfohlen

Einige Kommandos:

```
# erzeugen eines BTRFS auf zwei Platten
mkfs.btrfs /dev/sdb1 /dev/sdc1

# anzeigen, dass beide Platten im neuen FS
btrfs filesystem show /dev/sdb1

# mounten
mkdir /media/btrfs1
mount /dev/sdb1 /media/btrfs1

# anzeigen
btrfs filesystem df /media/btrfs1

# vergrößern auf Maximum
btrfs filesystem resize max /media/btrfs1

# verkleinern um 2g Byte
btrfs filesystem resize −2g /media/btrfs1

# Kompression aktivieren
mount −o compress /dev/sdb1 /media/btrfs1
```

```
# konvertierung ext4 nach btrfs
#
#   Prüfen
fsck.ext4  -f /dev/sdd1
#   Konvertieren
btrfs-convert /dev/sdd1
#   Einbinden
mkdir /media/btrfs2
mount /dev/sdd1 /media/btrfs2
#
# snapshot  ext2_save mit ursprünglichem Filesystem
# Wenn nicht ok:
     # Rückkonvertieren
     umount /media/btrfs2
     btrfs -convert -r /dev/sdd1
# Wenn ok
     # löschen  des Snapshots
     btrfs subvolume delete /media/btrfs2/ext2_saved

# Subvolumes
btrfs subvolume create /media/btrfs1/sub1
mkdir /media/sub1
mount -o subvol=sub1 /dev/sdb1 /media/sub2
```

Übersicht btrfs-Kommandos

```
mkfs.btrfs          - Erzeugen eines BTRFS in einer Partition
btrfs               - allgemeines Administrationstool (create, delete, ...)
btrfsctl            - Snapshot-Verwaltung, Defragmentierung (veraltet)
btrfs-dump-super    - Dump Superblock nach /tmp
btrfs-map-logical   - bei Fehlern
btrfs-show          - anzeigen (veraltet)
btrfs-zero-log      - bei Fehlern
btrfsck             - fsck
btrfs-debug-tree    - bei Fehlern
btrfs-find-root     - bei Fehlern
btrfs-restore       - restore Files von einem defektem Gerät
btrfstune           - tune
btrfs-convert       - Convert ext4 -> btrfs und zurueck
btrfs-dev-clear-sb -  ???
btrfs-image         - Backup und Restore (Image)
btrfs-select-super - bei Fehlern
btrfs-vol           - Volume-Managment (veraltet)
```

```
  btrfs --help

Usage:
        btrfs subvolume snapshot [-r] <source> [<dest>/]<name>
                Create a writable/readonly snapshot of the subvolume <source> with
                the name <name> in the <dest> directory.

        btrfs subvolume create [<dest>/]<name>
                Create a subvolume in <dest> (or the current directory if
                not passed).

        btrfs subvolume delete <subvolume>
                Delete the subvolume <subvolume>.

        btrfs subvolume list [-p] <path>
                List the snapshot/subvolume of a filesystem.

        btrfs subvolume set-default <id> <path>
                Set the subvolume of the filesystem <path> which will be mounted
                as default.

        btrfs subvolume get-default <path>
                Query which subvolume of the filesystem <path> will be mounted
                as default.

        btrfs subvolume find-new <subvolume> <last_gen>
                List the recently modified files in a filesystem.
```

```
btrfs filesystem defragment -c[zlib|lzo] [-l len] [-s start] [-t size] -[vf]
                                                  <file>|<dir> [<file>|<dir>...]
        Defragment a file or a directory.

btrfs filesystem sync <path>
        Force a sync on the filesystem <path>.

btrfs filesystem resize [<devid>:][+/-]<size>[gkm]|max <path>
        Resize the file system.

btrfs filesystem show [--all-devices|<uuid>|<label>]
        Show the info of a btrfs filesystem. If no argument
        is passed, info of all the btrfs filesystem are shown.

btrfs filesystem df <path>
        Print allocated and used data for all block group types.

btrfs filesystem balance start [-d [filters]] [-m [filters]]
                                                [-s [filters]] [-vf] <path>
        Balance chunks accross the devices on filesystem under path. Control
        operation by subcommands.

btrfs filesystem balance pause <path>
        Pause balance operation at the first possible occasion.

btrfs filesystem balance cancel <path>
        Cancel balance operation at the first possible occasion.
```

```
btrfs filesystem balance resume <path>
        Resume balance operation.

btrfs filesystem balance status [-v] <path>
        Show status of running or paused balance operation.

btrfs filesystem label <dev> [newlabel]
        With one argument, get the label of filesystem on <device>.
        If <newlabel> is passed, set the filesystem label to <newlabel>.
        The filesystem must be unmounted.

btrfs filesystem csize [-s start] [-e end] <file>
        Read regular and compressed size of extents in the range [start,end).

btrfs scrub start [-Bdqru] {<path>|<device>}
        Start a new scrub.

btrfs scrub cancel {<path>|<device>}
        Cancel a running scrub.

btrfs scrub resume [-Bdqru] {<path>|<device>}
        Resume previously canceled or interrupted scrub.

btrfs scrub status [-d] {<path>|<device>}
        Show status of running or finished scrub.
```

```
        btrfs device scan [--all-devices|<device> [<device>...]
                Scan all device for or the passed device for a btrfs
                filesystem.

        btrfs device add <dev> [<dev>..] <path>
                Add a device to a filesystem.

        btrfs device delete <dev> [<dev>..] <path>
                Remove a device from a filesystem.

        btrfs inspect-internal inode-resolve [-v] <inode> <path>
                Resolve given inode number to path name.

        btrfs inspect-internal logical-resolve [-v] [-P] <logical> <path>
                Resolve given logical block number to path name or inode number.


        btrfs help|--help|-h
                Show the help.

        btrfs <cmd> --help
                Show detailed help for a command or
                subset of commands.

Btrfs v0.19+20120406
```