6. File System

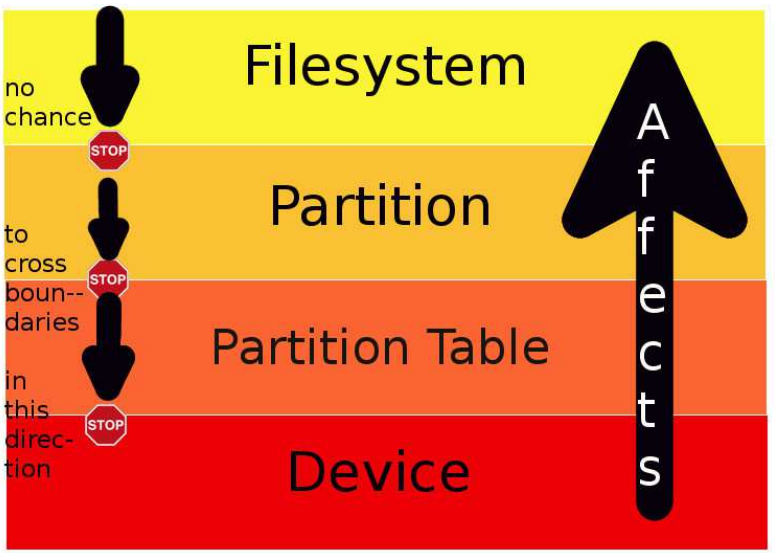
Drives and Capacity

as of 2014

Drive	Bandwidth (read)	Capacity	EUR/GB
hard disk drive	1.6 GB/s	60 GB. . . 4 TB	0.06. . . 0.20
solid state drive	2.7 GB/s	120 GB. . . 2 TB	0.70. . . 0.85
secure digital memory card	150 MB/s	4 GB. . . 128 GB	0.68. . . 0.85
USB memory stick	60 to 90 MB/s	4 GB. . . 256 GB	0.69. . . 2.00
digital versatile disk	61.7 MB/s (16x)	4.7 GB (1s, 1l)	0.69. . . 2.00

http://en.wikipedia.org/wiki/Hard_disk_drive
http://www.intel.com/content/www/us/en/
solid-state-drives/solid-state-drives-ssd.html
http://www.tomshardware.com/charts/-usb-3.0-card-reader-charts-2014/
-01-Compact-Flash-Sequential-Read-MB-s,3542.html
http://www.tomshardware.com/reviews/DVD-Burner,2447-8.html

Logical Layering



Partition Mess on Intel Systems

- first ,,OS" for Intel-based system was MS-DOS
- fundamental design error: four partitions on a hard disk named C:, D:, E:, F: (restriction 32 MB in MS-DOS 3.3 in 1987)
- disks grew bigger $\rightsquigarrow$more ,,logical" partitions G:, H:...
- disks grew still bigger $\rightsquigarrow$larger partitions
- MBR: still four *primary* partitions
- MBR: ,,extended" partition contains *logical* partitions
- MBR: disk limit 2 TB,
- MBR: no backup
- MBR: no error correcting code

UEFI Unified Extensible Firmware Interface

EFI: Itanium platform 1998 (Intel)

UEFI: April 2011 (Intel, AMD, Microsoft, Apple,...)

- GPT = GUID Partition Table
- pre-OS environment, including network capability
- 8 ZiB = 8000 EiB

SI-Prefixes: kilo-mega-giga-tera-peta-exa-zetta-yotta-...

IEEE1541: kibi-mebi-gibi-tebi-pebi-ebi-zebi-yobi-...

Linux / Windows 64-bit / HP-UX / HP-OpenVMS / Apple(Intel) / FreeBSD(GPT)

GUID = Globally Unique Identifier

What is a File System?

A file system is a *logical* unit of (background) memory.

Inodes are local to a file system.

A file system can live on

- a hard disk
- a floppy disk
- a CDROM
- a DVD
- a memory stick
- a part of RAM (RAMDISK)
- ...

FreeBSD Device Naming

The name determines what type of driver handles the storage device:

device name	drive type
ad	IDE (ATA, SATA) hard drives
da	USB mass storage, SCSI hard drives
acd	IDE CDROM drives
cd	SCSI CDROM drives
scd,mcd	non–standard CDROM drives
sa	SCSI tape drives
ast	IDE tape drives
fla	flash drives
aacd,mlxd,mlyd,idad,twed	RAID drives

Linux Device Naming

- /dev/hda first drive, first IDE controller
- /dev/sda first drive, first SATA/SCSI controller
  - first partition /dev/sda1.
  - second partition /dev/sda2.
- /dev/sdb 2nd drive
  - first partition /dev/sdb1.
  - second partition /dev/sdb2.

Type of device is irrelevant (HDD/CDROM).

Which devices are found?

Look at the boot messages.

Example:

```
# dmesg

ada0 at ata0 bus 0 scbus2 target 0 lun 0
ada0: <ST3250310AS 3.AAB> ATA-7 SATA 2.x device
ada0: 238475MB (488397168 512 byte sectors: 16H 63S/T 16383C)

ada1 at ata1 bus 0 scbus3 target 1 lun 0
ada1: <ST3500418AS CC38> ATA8-ACS SATA 2.x device
ada1: 476940MB (976773168 512 byte sectors: 16H 63S/T 16383C)

acd0: DVDROM <TSSTcorpDVD-ROM SH-D162C/TS04> at ata1-master UDMA33
acd1: CDRW <CW088D ATAPI CD-R/RW/V110F> at ata1-slave UDMA33
```


FreeBSD GPT Device and Partition Naming

/dev/ada0 is the first drive

Its first partition is /dev/ada0p1 (boot).

Its second partition is /dev/ada0p2 (usually /).

```
# gpart show ada0
=>       34  488397101  ada0  GPT  (233G)
         34       1024     1  freebsd-boot  (512K)
       1058   10485760     2  freebsd-ufs  (5.0G)
   10486818  209715200     3  freebsd-ufs  (100G)
  220202018   25165824     4  freebsd-ufs  (12G)
  245367842    8388608     5  freebsd-ufs  (4.0G)
  253756450  125829120     6  freebsd-ufs  (60G)
  379585570    8388608     7  freebsd-swap  (4.0G)
  387974178  100422957     8  freebsd-ufs  (48G)
```


Example: booting different partition

```
gpart unset -a bootme -i 2 ada0

gpart set -a bootme -i 6 ada0
```

File System (FS)

- lives within a partition
- maps directory-tree structure and files to disk
- inodes (meta–data) and directories/files (data)
- faetures: max FS size, max file size, crash recovery...

⇝several file system types

`http://linux-xfs.sgi.com`
`/projects/xfs/papers/xfs_white/xfs_white_paper.html`

File System Types

- FreeBSD
  - ufs (UNIX filesystem), FFS (Berkeley Fast Filesystem)
  - ext2fs
  - cd9660 – CD–ROM file system
  - new: ZFS (Sun Microsystems)
  - ...
- Linux
  - ext2 – standard linux FS
  - ext3 – journaling extension of ext2
  - ext4 – extension of ext3 (performance/features)
  - reiserfs – file system based on balanced trees
  - jfs – IBM's journaled FS
  - xfs – journaled FS
  - iso9660 – CD–ROM file system
  - ...

`http://www.tech-analyser.com/2011/10/`
`understanding-file-systemsntfs-fat.html`

`http://www.enterprisestorageforum.com/technology/features/`
`article.php/3849556/10-Reasons-Why-ZFS-Rocks.htm`

Show supported FS types

```
$ ls -l /sbin/mount_*
-r-xr-xr-x  /sbin/mount_cd9660
-r-xr-xr-x  /sbin/mount_fusefs
-r-xr-xr-x  /sbin/mount_mfs
-r-xr-xr-x  /sbin/mount_msdosfs
-r-xr-xr-x  /sbin/mount_nfs
-r-xr-xr-x  /sbin/mount_nullfs
-r-xr-xr-x  /sbin/mount_oldnfs
-r-xr-xr-x  /sbin/mount_udf
-r-xr-xr-x  /sbin/mount_unionfs
```

Partitioning/FS/Mounting

action	GPT
partition disk	`gpart`
init filesystem	`newfs/mkfs`
dev $\rightsquigarrow$ dir tree	`mount`

command	parameters
`gpart`	disk
`newfs`	partition, FS type
`mount`	partition, directory

Partitioning (1)

concept: additional layer between disk and FS

advantage:

- separated file storage
- controlled subsystems

disadvantage:

- fixed size (though `growfs` may resize)
- each partition to be configured

Partitioning (2)

Should be done carefully (fixed sizes).

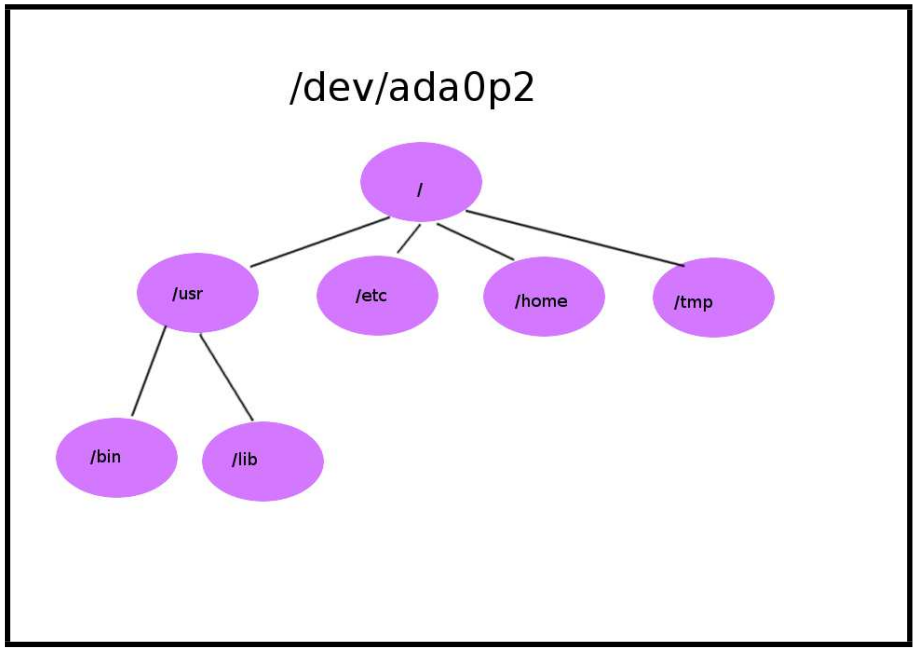
The system core should not be affected by file I/O of users.

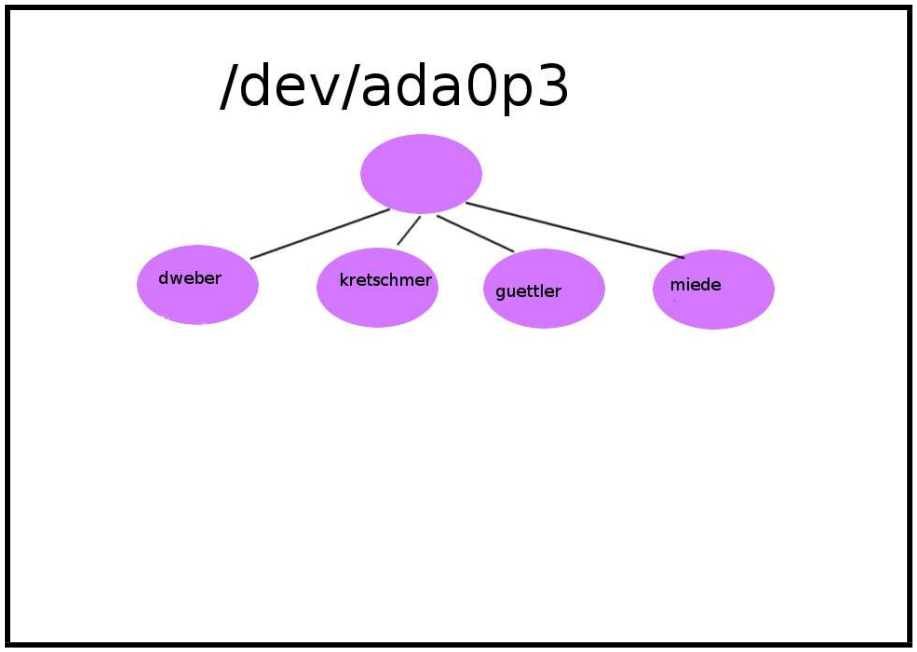
$\rightsquigarrow$`/`, `/home`, `/var`, `/tmp` should be on different file systems

swap at least as big as RAM

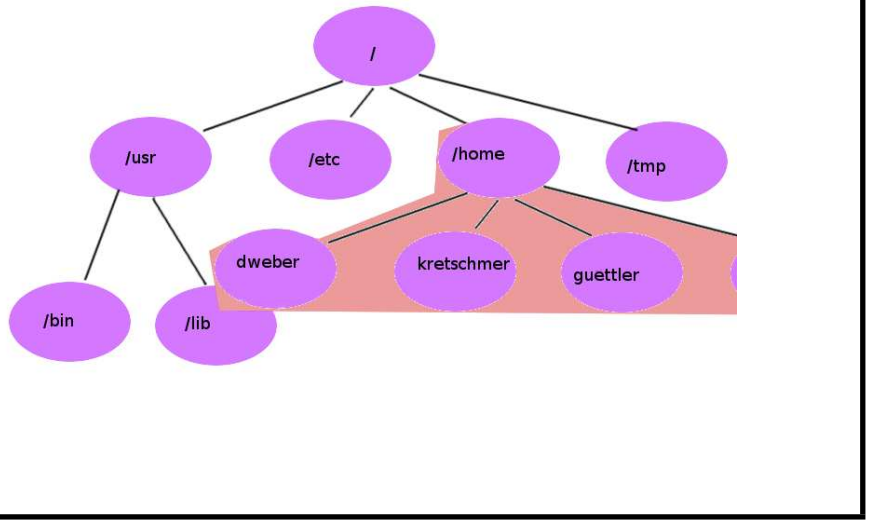
/var at least as big as RAM

Mounting an FS (1)





Mounting an FS (2)



Mounting an FS (3)

Example:

```
# mount /dev/ada0p6 /tmp
```

Mounts partition `/dev/ad0p6` as directory `/tmp`.

⇝`/tmp` is called a mount point

⇝mount point = empty directory

Mounting is usually done at boot time.

File `/etc/fstab` contains device-mount-mapping.

/etc/fstab

```
# Device        M-point   FStype  Options Dump Pass#
/dev/ada0p2     /         ufs     rw      1       1
/dev/ada0p3     /usr      ufs     rw      2       2
/dev/ada0p4     /var      ufs     rw      2       2
/dev/ada0p5     /tmp      ufs     rw      2       2
/dev/ada0p10    /TMP      ufs     rw      2       2
134.96.216.92:/home /home nfs     rw      0       0
/dev/acd0       /cdrom  cd9660  ro,noauto 0       0
```

order of entriers important for `mount`, `fsck`

dump (# days), pass = order of FS check

Unmounting an FS (1)

Simple:

```
# umount /tmp
```

Or not so easy:

```
# umount /tmp
umount: unmount of /tmp failed: Device busy
```

We should *not* unmount an FS which is currently in use.

But we could:

```
# umount -f /tmp
```

This does *not* work for the *root filesystem*.

Unmounting an FS (2)

Which process uses a disk/file?

```
$ lsof | grep /home
COMMAND  PID   USER  FD TYPE        DEVICE SIZE/OFF    NODE NAME
bash    3627 dweber cwd VDIR 255,117440514     1536 3379712 /home/dweber
lsof    3696 dweber cwd VDIR 255,117440514     1536 3379712 /home/dweber
grep    3697 dweber cwd VDIR 255,117440514     1536 3379712 /home/dweber
```

- alert corresponding users
- kill offending processes
- unmount the FS

Filesystems on a RAMDISK

- create device node for this filesystem
  FreeBSD: `mdconfig`, OpenBSD/NetBSD: `vnconfig`,
  Solaris `ramdiskadm`
  - need info whether to use
    - simply allocated memory (malloc())
    - a file
    - swap space
  - need size
  - should provide a device number
- create filesystem on the device
- mount it

Filesystems on a RAMDISK, Examples

using swap space

```
mdconfig -a -t swap -s 128M -u 10
newfs -U /dev/md10
mount /dev/md10 /tmp
chmod 1777 /tmp
```

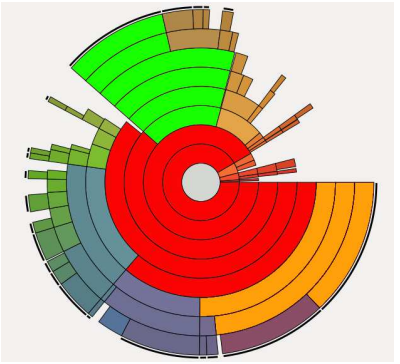
using a file (with bsdlabel)

```
dd if=/dev/zero of=somebackingfile bs=1k count=5k
mdconfig -a -t vnode -f somebackingfile -u 0
gpart create -s gpt md0
gpart add -t freebsd-ufs md0
newfs md0p1
mount /dev/md0p1 /mnt
```


Space Usage on file system: df = disk free

shows mounted file systems with

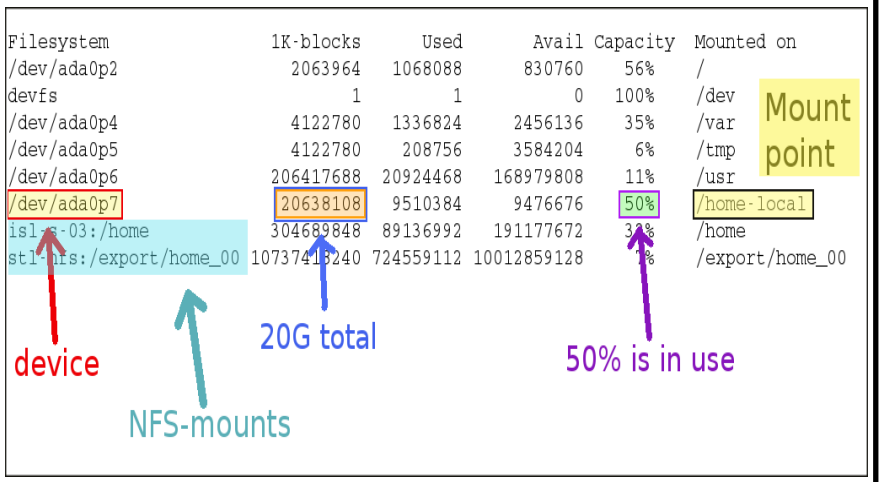
- name
- size in blocks (1K)
- number of used blocks
- number of available blocks
- percentage of use
- mount point



Note:

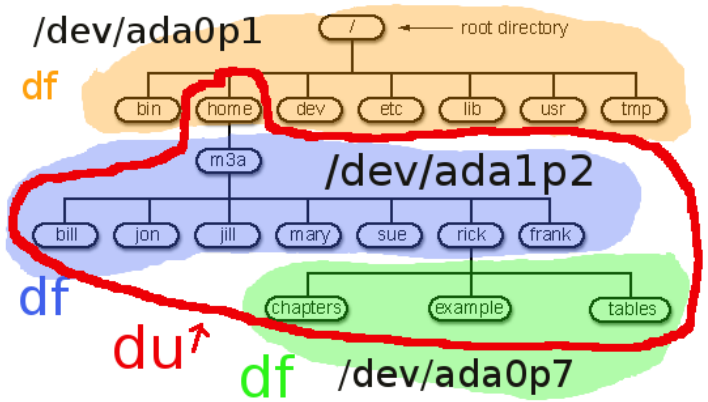
- must be checked periodically to avoid system failure
- likely overflows in `/home`, `/var`, `/tmp`

Example: df

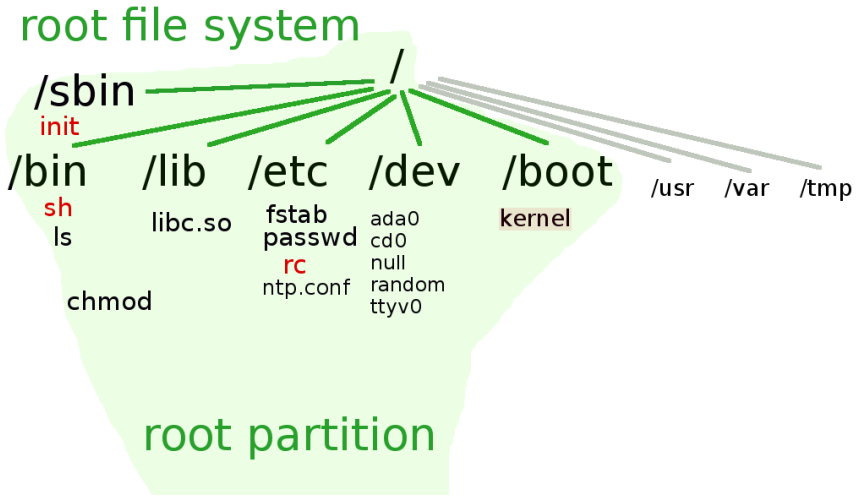


Space Usage on Directory: du = disk usage

```
$ du -s /home
13574540 /home
```



/ – The Root File System (1)



/ – The Root File System (2)

- **system core files**
  - system configuration
  - important shared libs
  - device entries
  - admin commands
- **boot scripts** and corresponding commands
- **mount points** for all other filesystems

/ – The Root File System (3)

Directory	Description	Example
/bin	user commands	/bin/ls
/dev	device entries	/dev/ada0
/etc	configuration	/etc/passwd
/lib	shared libraries	/lib/libc.so
/sbin	system administration commands	/sbin/shutdown
/boot	kernel binary, kernel modules	/boot/kernel/kernel
(/proc)	process information	/proc/curproc/status

The Boot Problem

operating system does

- process management
- memory management
- file system
- I/O

but needs I/O and file system to read the operating system

- must determine system disk
- must read boot code from disk
- must read OS kernel from directory tree

⇝**chicken–and–egg problem**

Baron Münchhausen

Mein Pferd und ich wären hoffnungslos versunken,
wenn ich es nicht geschafft hätte,
mich an meinem eigenen Haarschopf aus dem Sumpf zu ziehen.

Baron Münchhausen

Es kann eben doch von Vorteil sein,
wenn man einen gut trainierten Körper hat.

Booting