Fun with disks

ComS 252 — Iowa State University

Andrew Miner

Non-volatile storage

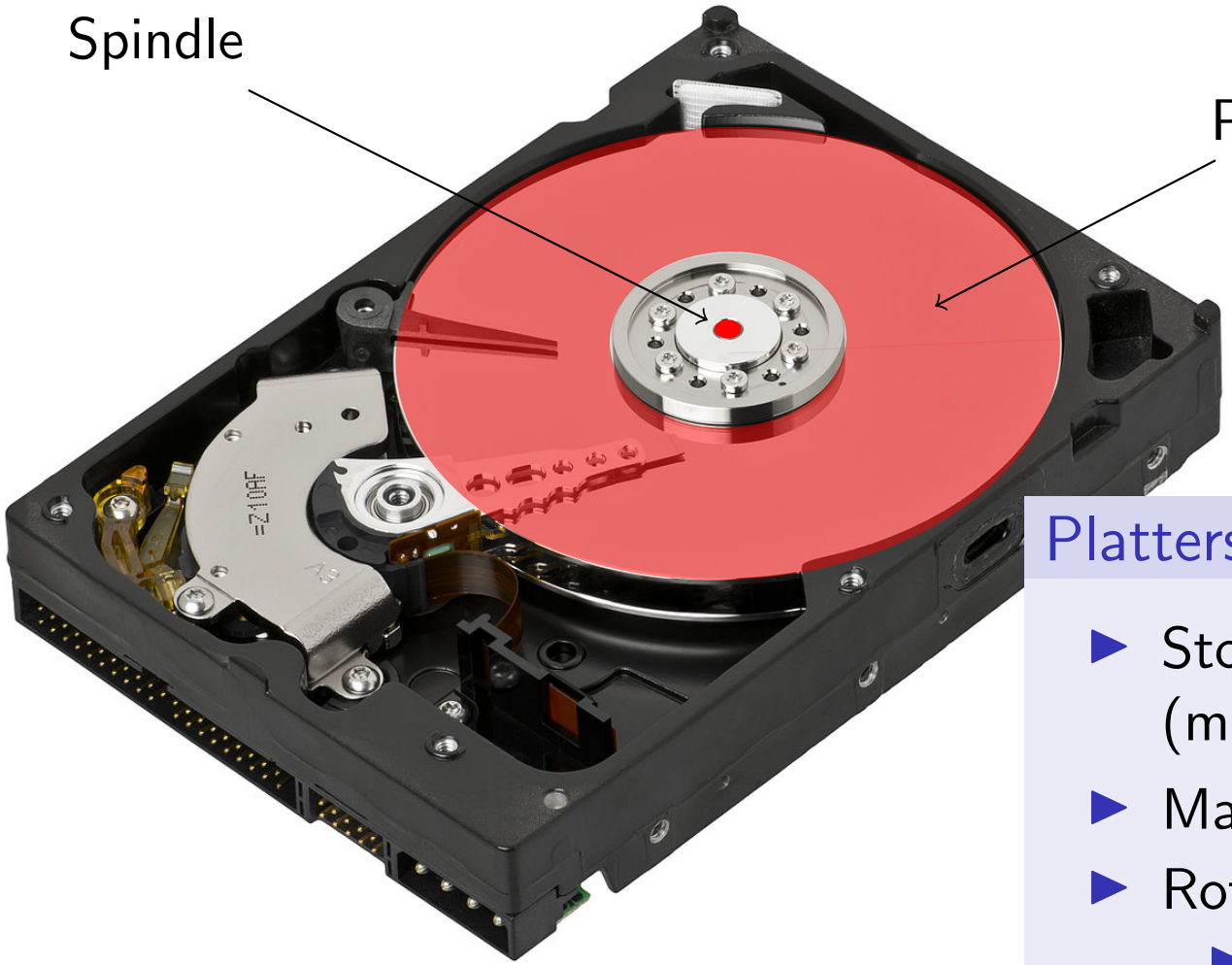
- Does not lose data when power is cut
- Main technologies:
  - Magnetic tape
    - Very slow
    - Still useful for backups and archival storage
  - Optical drives
    - Low capacity (around 4 Gibi) by today's standards
  - USB flash drives
  - Hard disk drives (HDDs)
    - Spinning magnetic disks
    - Still viable
  - Solid state drives (SSDs)
    - Electronic
    - Have become fast/cheap enough for mainstream use

Basic anatomy of a hard disk drive

Drives usually sealed

Assume the drive will never work again if you open it

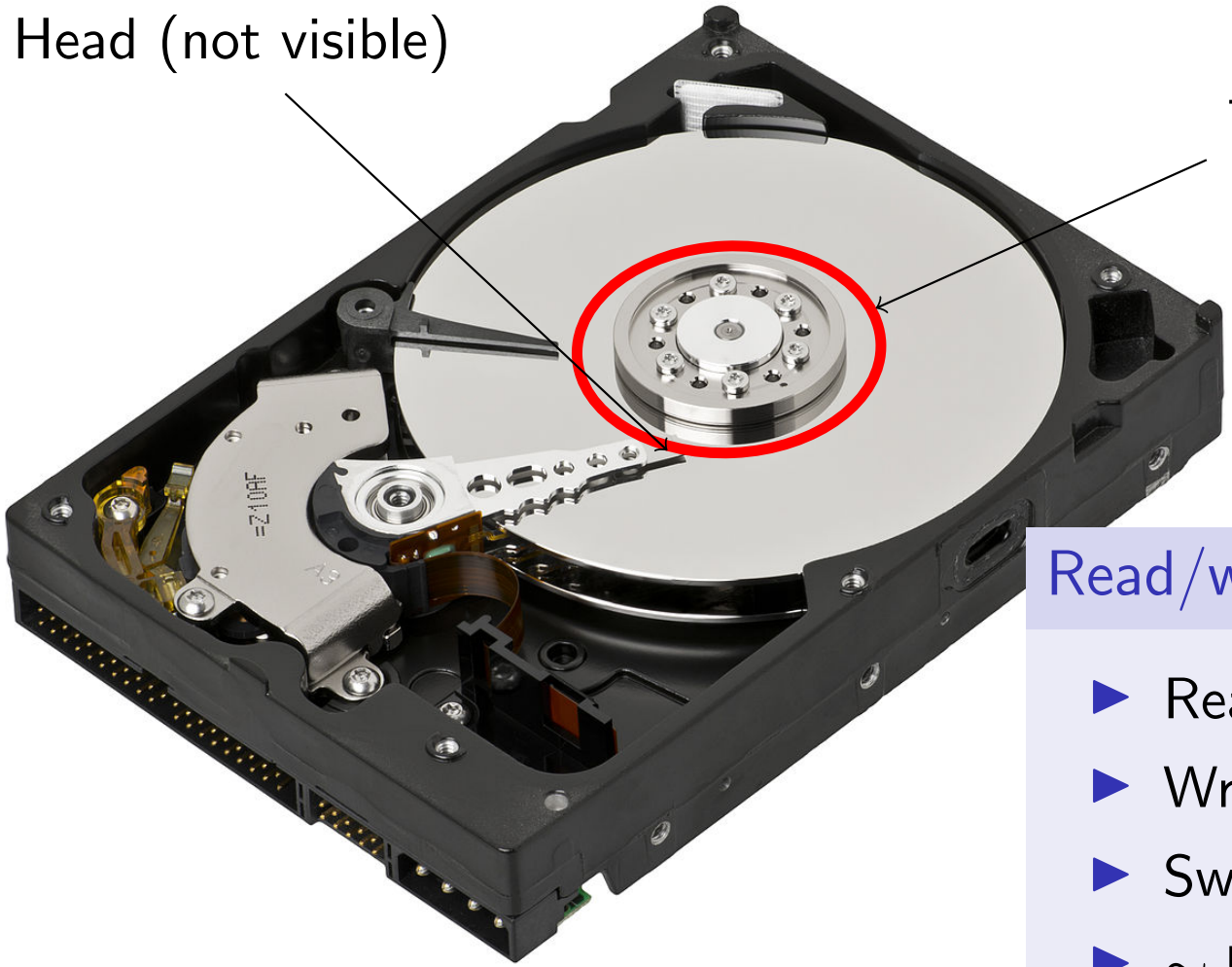
Basic anatomy of a hard disk drive



Platters

- Store data (magnetic coating)
- May be stacked
- Rotate around spindle
  - 4 — 15 krpm

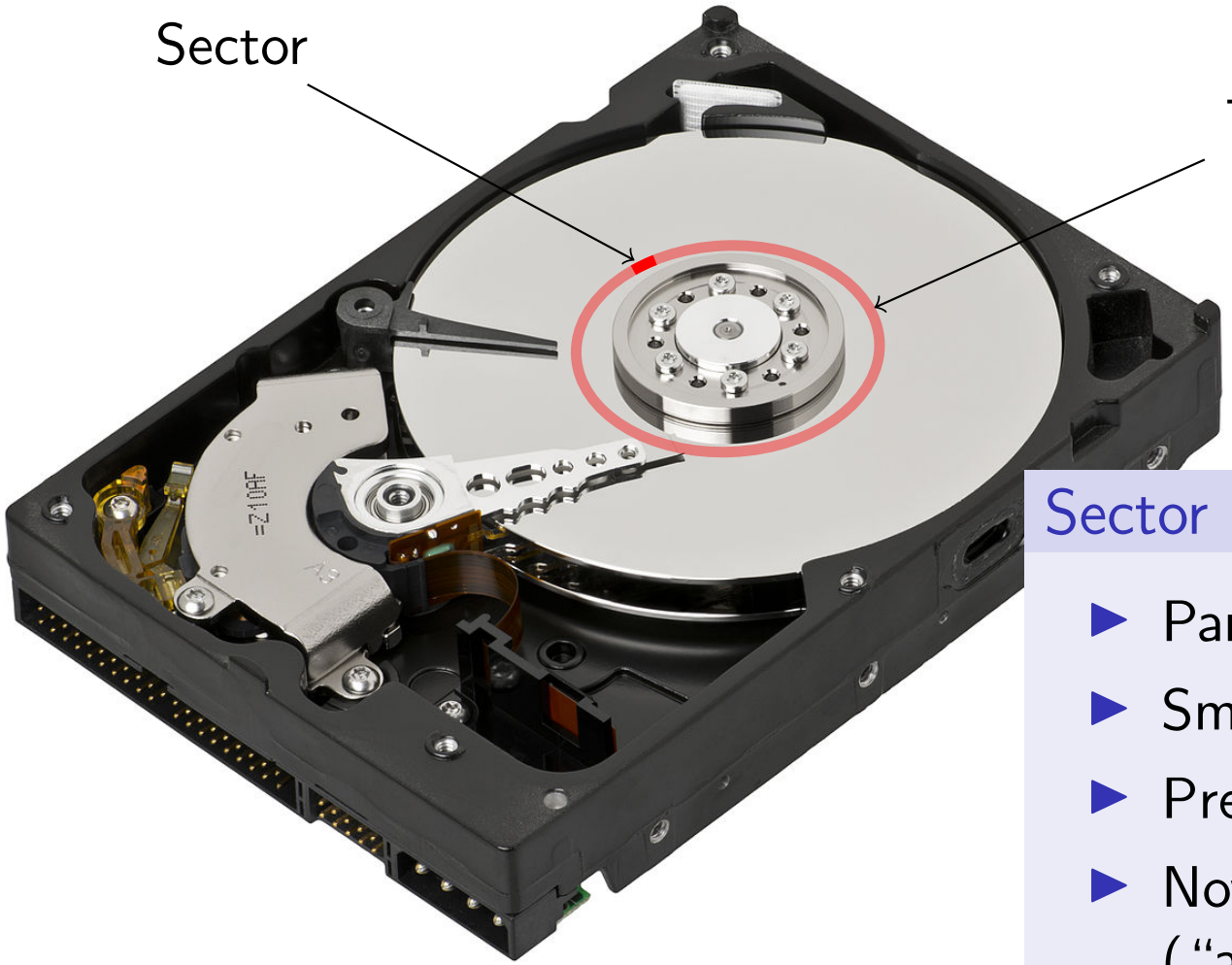
Basic anatomy of a hard disk drive



Read/write heads

- Read data
- Write data
- Sweep out *tracks*
- $\sim$ 5nm from platters

Basic anatomy of a hard disk drive



Sector

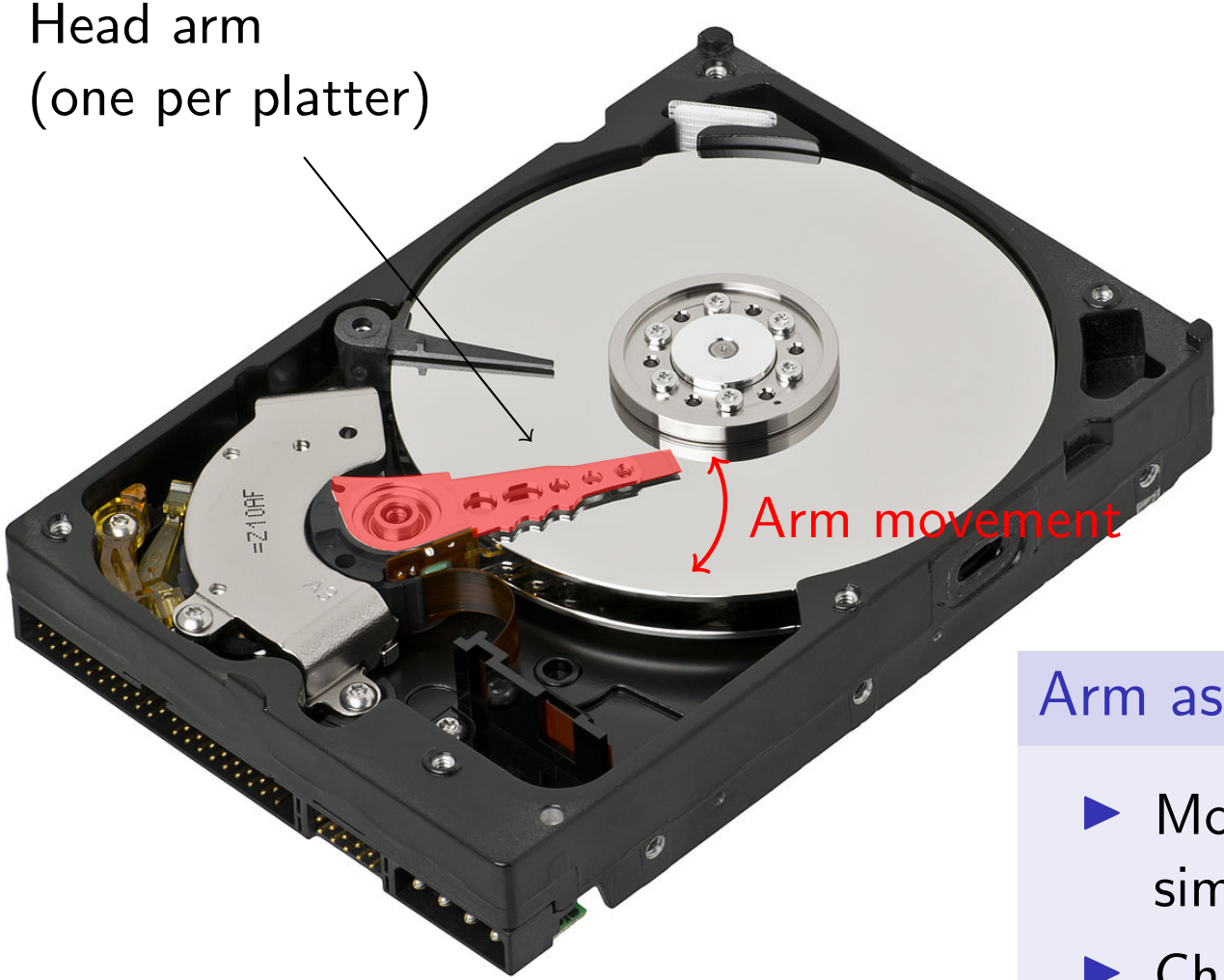
- Part of a track
- Smallest unit of data
- Previously: 512 bytes
- Now: 4 kibi ("advanced format")

Basic anatomy of a hard disk drive

Cylinder

- Collection of tracks over all platters

Basic anatomy of a hard disk drive



Arm assembly

- Moves all head arms simultaneously
- Changes cylinder(s)

Reading or writing a sector

Reading or writing a sector

1. Move heads to correct cylinder (seek)
   - Averages between 3 and 12 ms

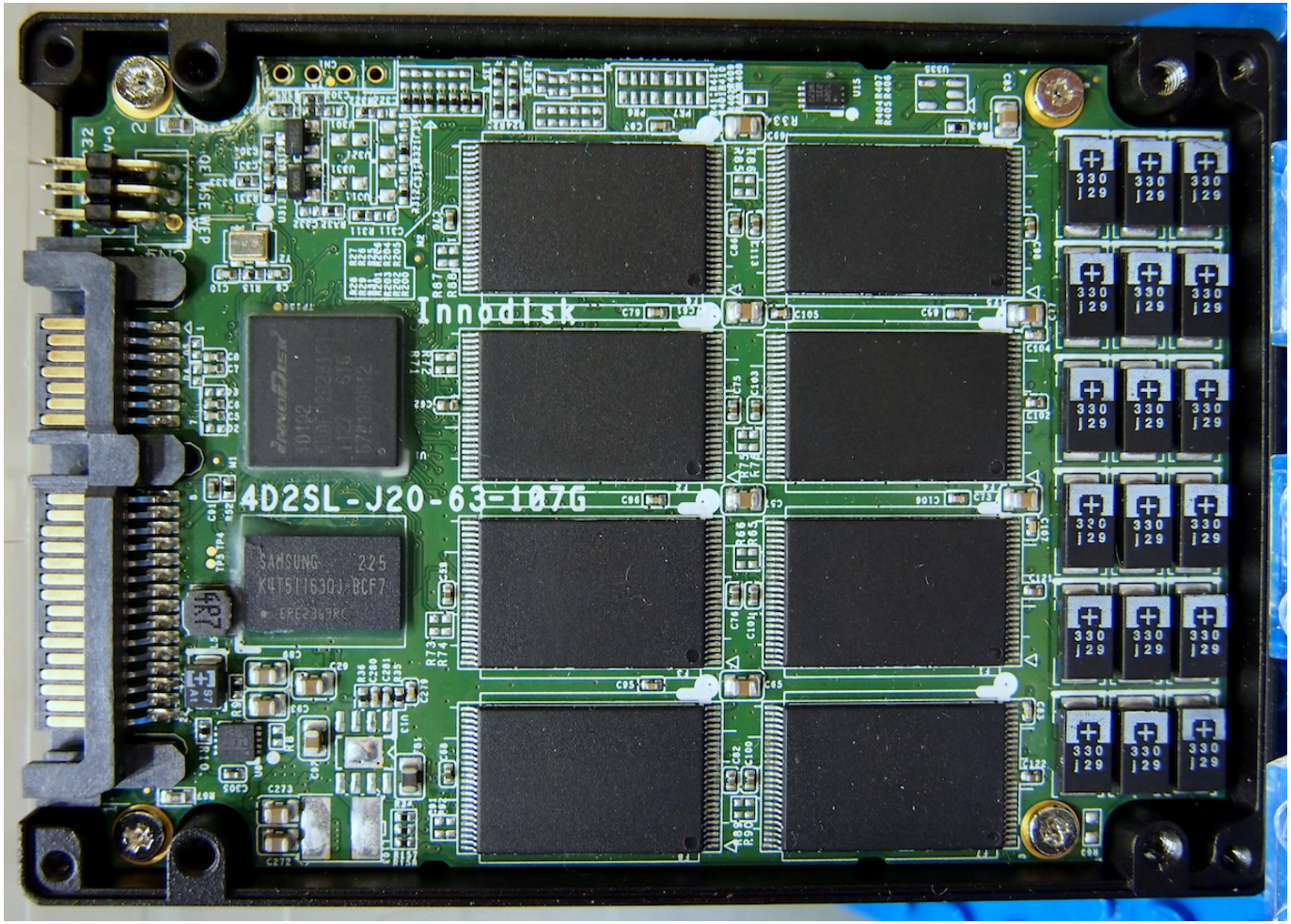
Reading or writing a sector

1. Move heads to correct cylinder (seek)
   - Averages between 3 and 12 ms
2. Wait until sector is under head (rotational latency)
   - Average is 30/rpm seconds

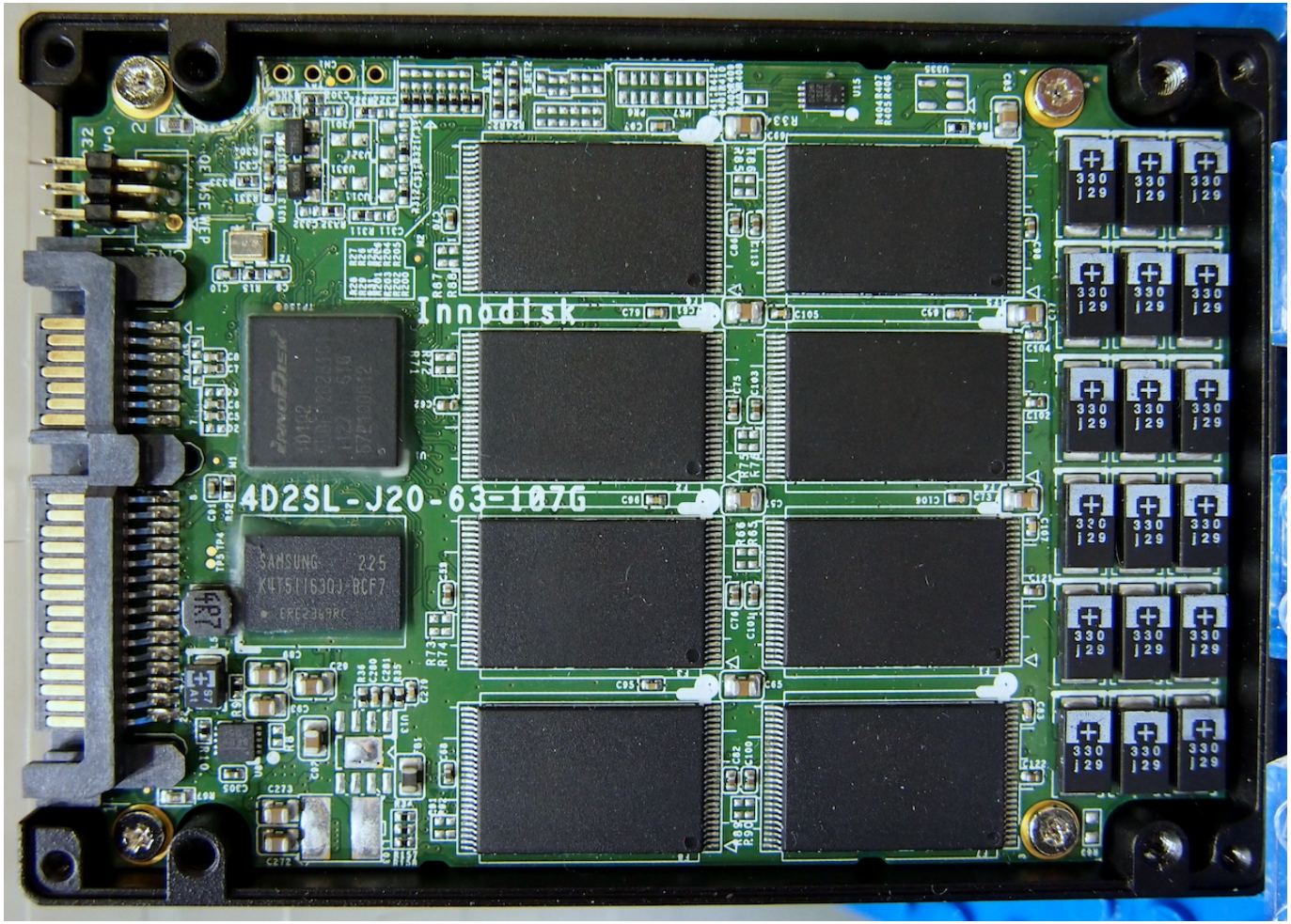
Reading or writing a sector

1. Move heads to correct cylinder (seek)
   - Averages between 3 and 12 ms
2. Wait until sector is under head (rotational latency)
   - Average is 30/rpm seconds
3. Read/write sector

Basic anatomy of a solid state drive



Basic anatomy of a solid state drive



- Just electronics (NAND flash)
- No moving parts

Solid state vs hard disk drives

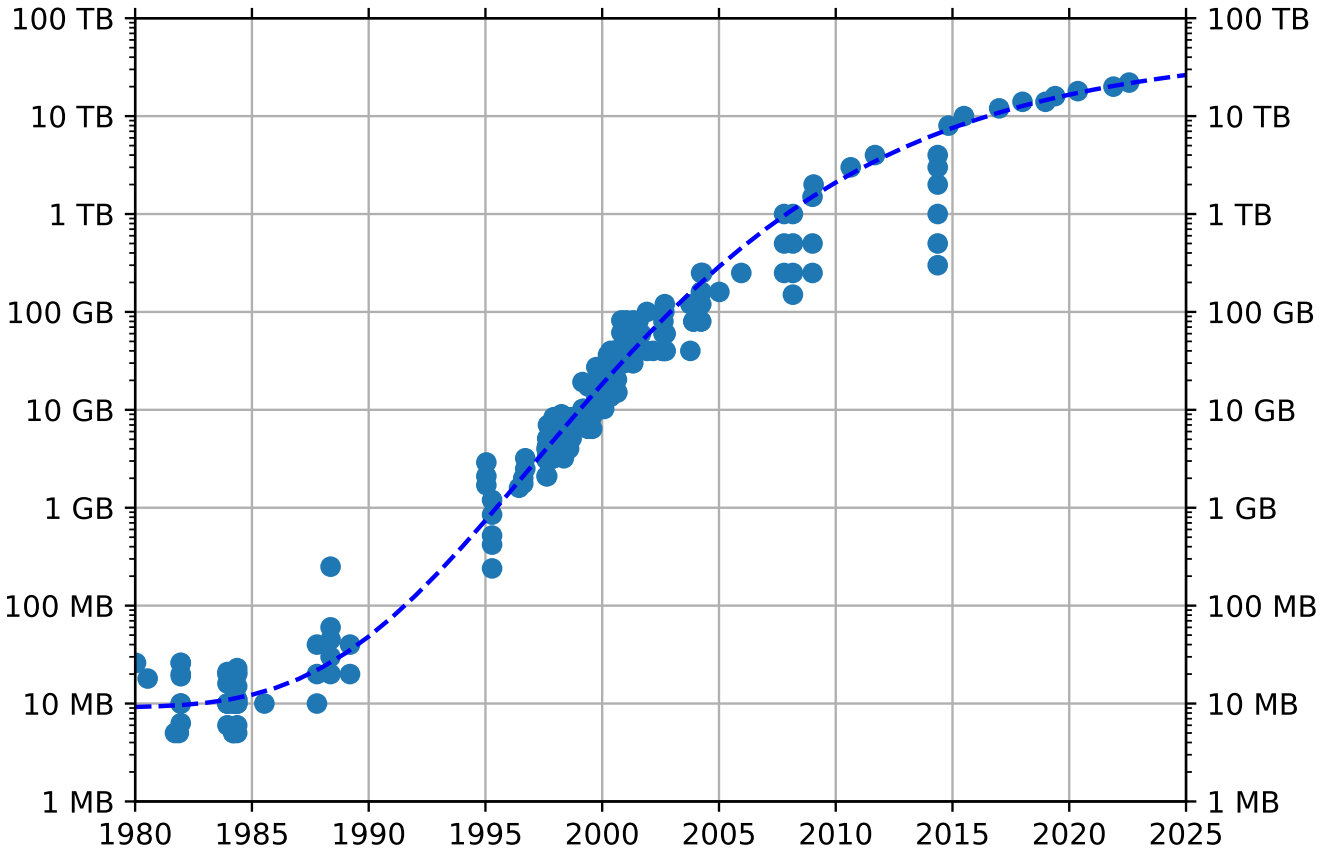
Wins for SSD

- Much lower latency
- Lower power consumption
  - No motors
- No risk of head crashes

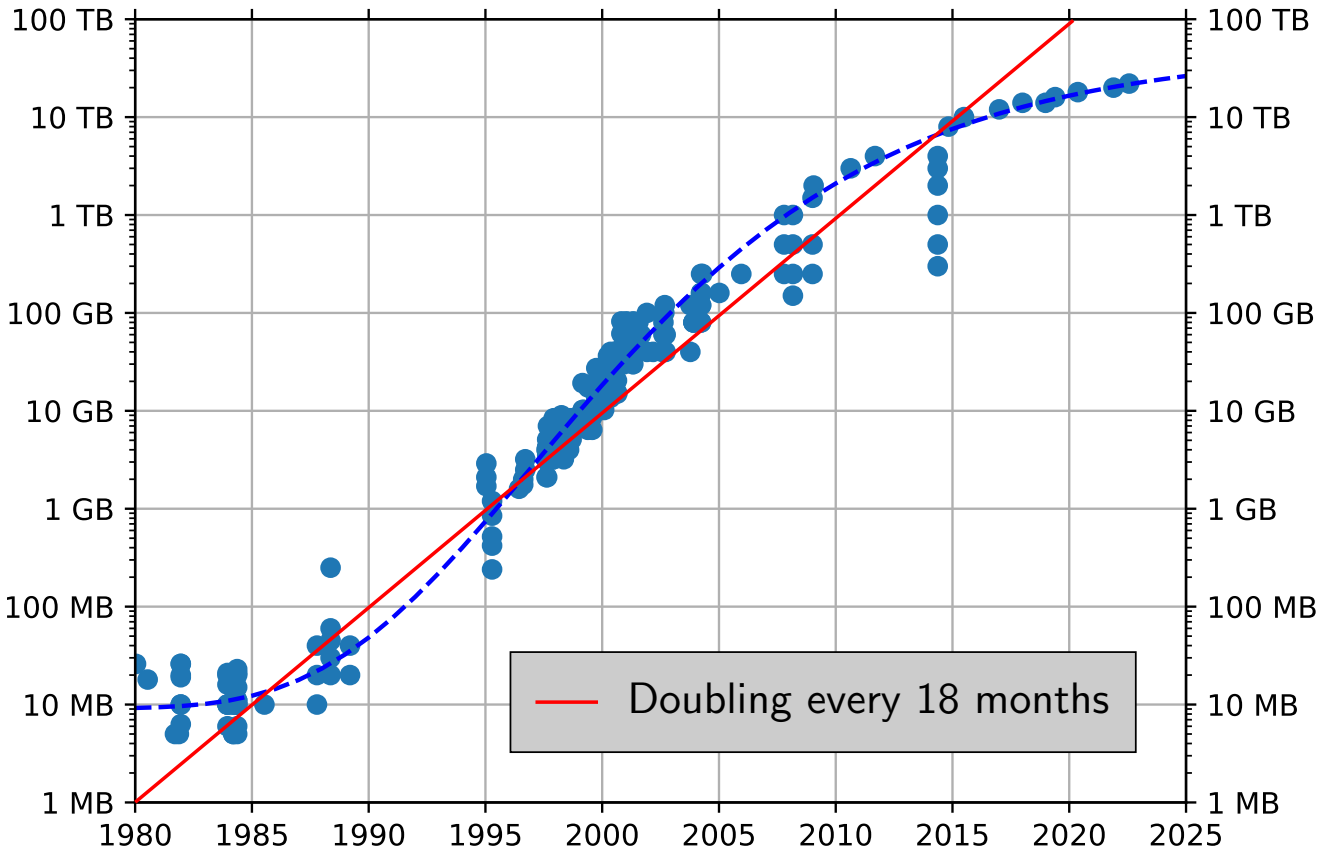
Wins for HDD

- Cost per byte is lower

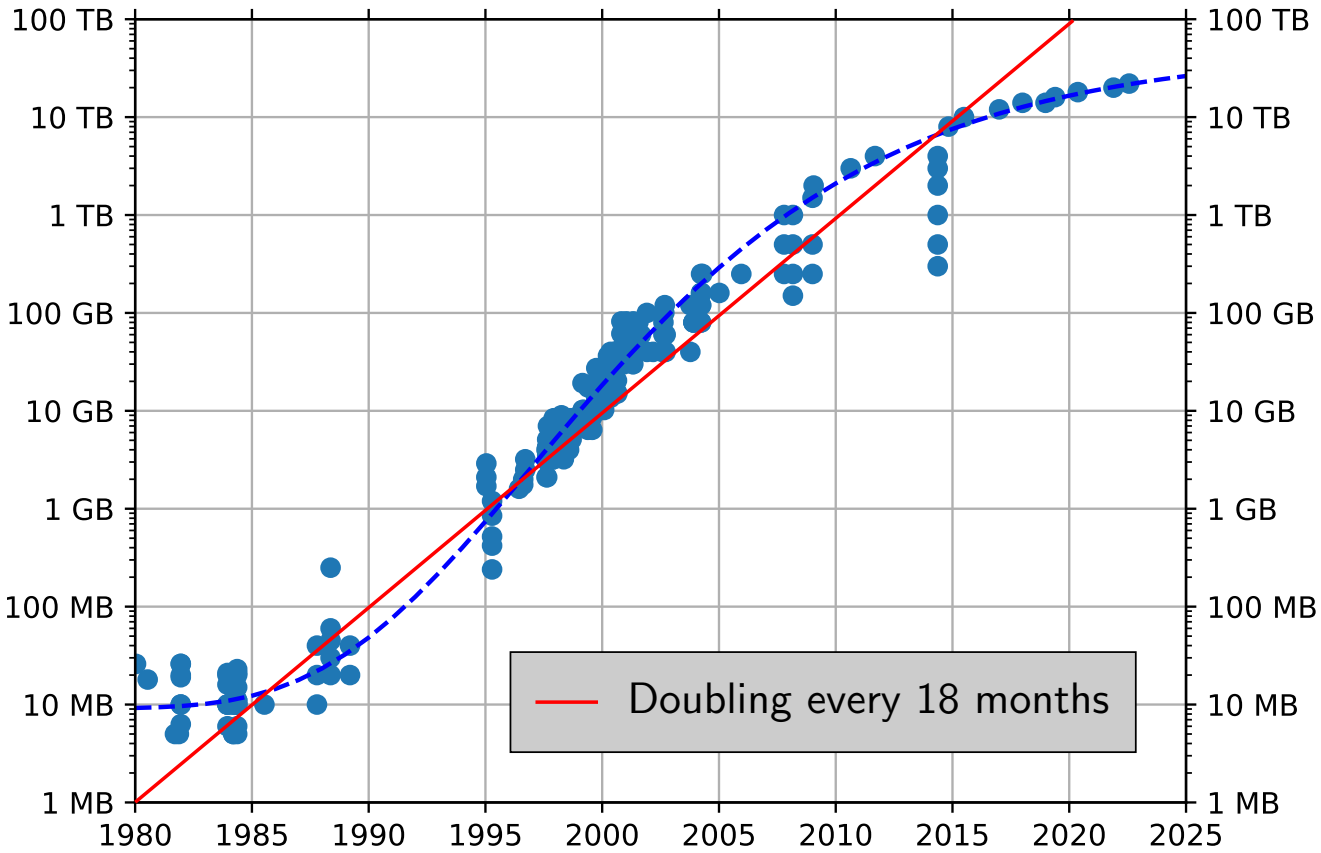
Drive Capacities Over Time



Drive Capacities Over Time

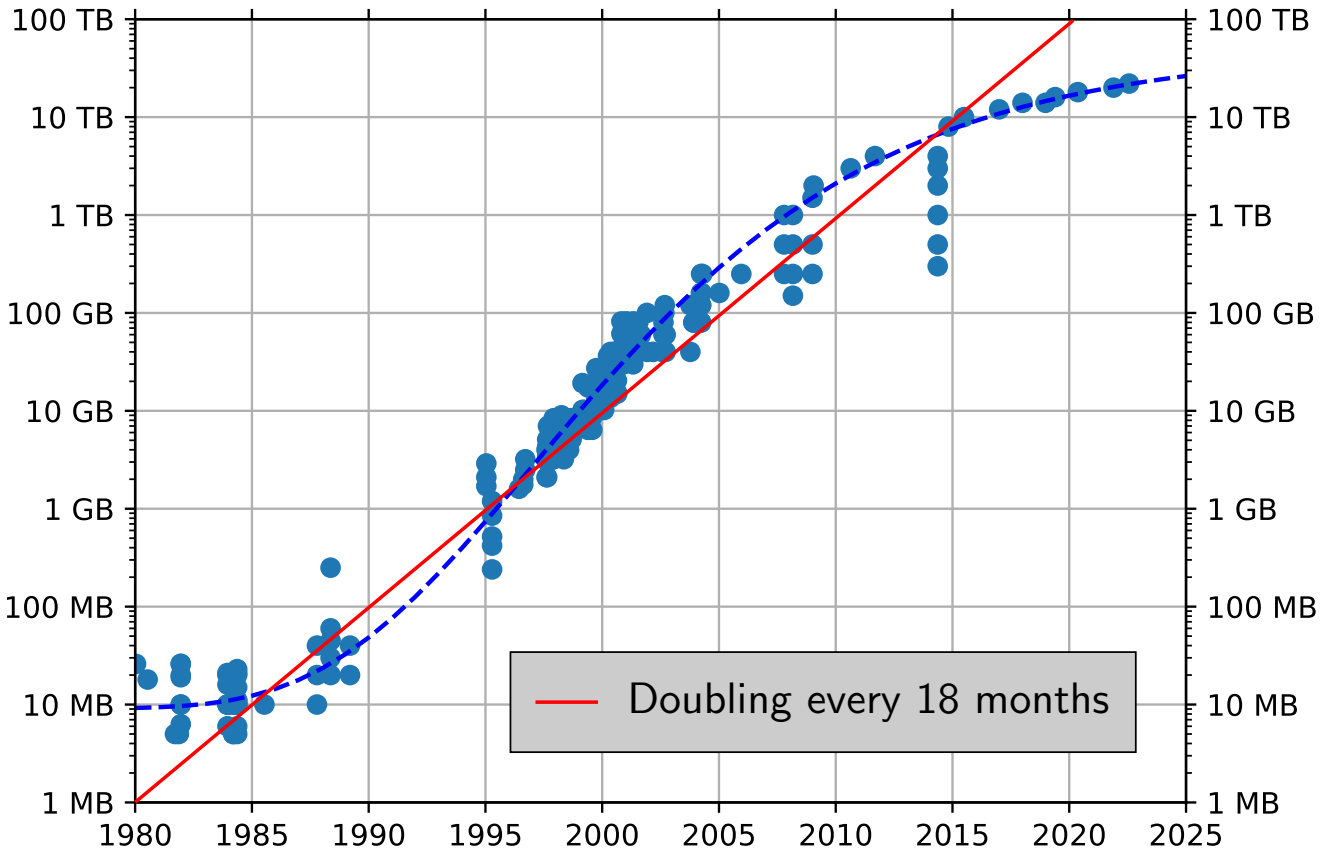


Drive Capacities Over Time



- $2\times$ every 18 months $\approx$ $10\times$ every 5 years $(2^{5/1.5} \approx 10.08)$

Drive Capacities Over Time



- $2\times$ every 18 months $\approx$ $10\times$ every 5 years $(2^{5/1.5} \approx 10.08)$
- Growth rate slowed around 2010

What does doubling every 18 months predict?

2010	$\approx$	1	Tb	(terabytes)
2015	$\approx$	10	Tb	(terabytes)
2020	$\approx$	100	Tb	(terabytes)
2025	$\approx$	1	Pb	(petabytes)
2030	$\approx$	10	Pb	(petabytes)
2035	$\approx$	100	Pb	(petabytes)
2040	$\approx$	1	Eb	(exabytes)

What does doubling every 18 months predict?

2010	$\approx$	1	Tb	(terabytes)
2015	$\approx$	10	Tb	(terabytes)
2020	$\approx$	100	Tb	(terabytes)
2025	$\approx$	1	Pb	(petabytes)
2030	$\approx$	10	Pb	(petabytes)
2035	$\approx$	100	Pb	(petabytes)
2040	$\approx$	1	Eb	(exabytes)

- How much data is that?

What does doubling every 18 months predict?

2010	≈	1	Tb	(terabytes)
2015	≈	10	Tb	(terabytes)
2020	≈	100	Tb	(terabytes)
2025	≈	1	Pb	(petabytes)
2030	≈	10	Pb	(petabytes)
2035	≈	100	Pb	(petabytes)
2040	≈	1	Eb	(exabytes)

- How much data is that?
  - 100 Tb ≈ 2,000 dual–layered Blu-ray discs
  - 4 Pb ≈ data generated by Facebook daily (Dec. 2022)
  - 300 Pb ≈ YouTube total storage capacity (Dec. 2022)
  - 5 Eb ≈ "All words ever spoken by humans" (as ASCII text?)
  - 500 Eb ≈ World's total digital content (May 2009)

Files and Filesystems

File: An ordered collection of data, with attributes; e.g.

- Name
- Date and time of last modification

Files and Filesystems

File: An ordered collection of data, with attributes; e.g.

- Name
- Date and time of last modification

Directory: A container for files and other directories

- Essential for organization

Files and Filesystems

File: An ordered collection of data, with attributes; e.g.

- Name
- Date and time of last modification

Directory: A container for files and other directories

- Essential for organization

File system: Rules for mapping files and directories to sectors

- Specifies supported file attributes
- Has limits, e.g.,
  - Largest overall filesystem size
  - Largest file size
  - Number of items per directory
  - Allowed file names

Filesystem Families

FAT (File Allocation Table), Microsoft

- Common for DOS and Windows 9x machines

NTFS (New Technology File System), Microsoft

- Used for Windows NT, XP, and later
- Version 3.1 is current

HFS (Hierarchical File System), Apple

- HFS: 1985
- HFS+: 1998

APFS (Apple File System), Apple

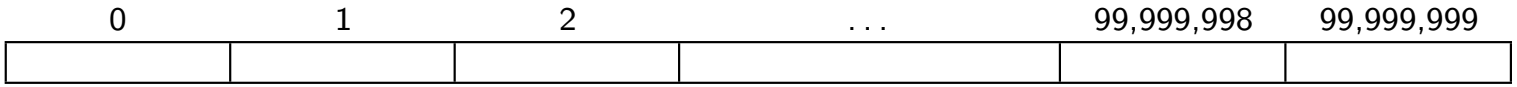
- Released 2017
- Optimized for solid-state drive storage

ext (extended file system), Linux

- Started with ext (1992)
- ext4 (2008) is current

Drive partitions

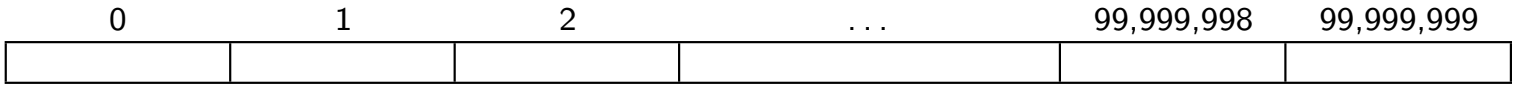
- Logically, a disk is a huge pool of numbered sectors



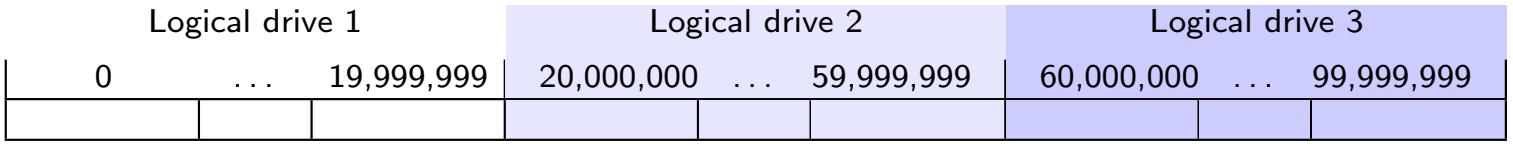
  - The disk shown above has 100,000,000 sectors
  - If sectors are 512 bytes: total capacity is around 50 Gb

Drive partitions

- Logically, a disk is a huge pool of numbered sectors



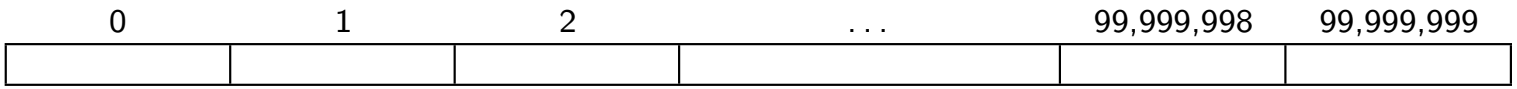
  - The disk shown above has 100,000,000 sectors
  - If sectors are 512 bytes: total capacity is around 50 Gb
- A disk can be partitioned into multiple "logical" drives



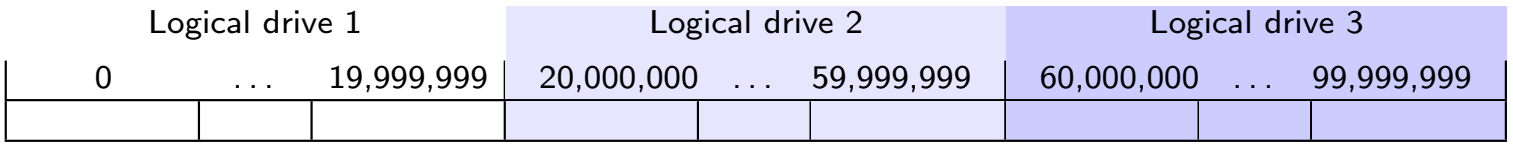
  - Logical drive 1: about 10 Gb
  - Logical drive 2: about 20 Gb
  - Logical drive 3: about 20 Gb

Drive partitions

- Logically, a disk is a huge pool of numbered sectors



  - The disk shown above has 100,000,000 sectors
  - If sectors are 512 bytes: total capacity is around 50 Gb
- A disk can be partitioned into multiple "logical" drives

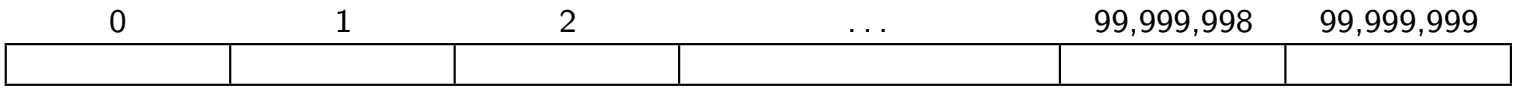


  - Logical drive 1: about 10 Gb
  - Logical drive 2: about 20 Gb
  - Logical drive 3: about 20 Gb
- OS will treat each logical drive as a separate disk; e.g.:

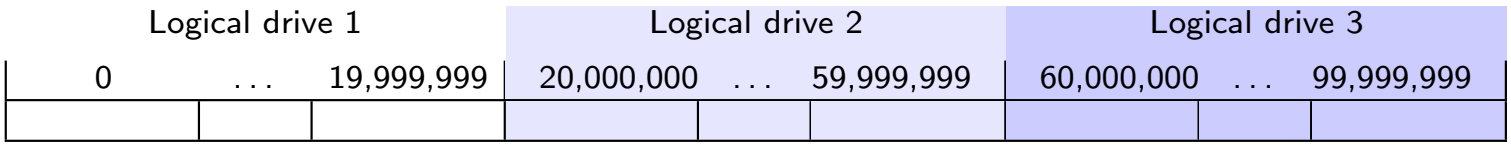
  C: Logical drive 1
  D: Logical drive 2
  E: Logical drive 3

Drive partitions

- Logically, a disk is a huge pool of numbered sectors



  - The disk shown above has 100,000,000 sectors
  - If sectors are 512 bytes: total capacity is around 50 Gb
- A disk can be partitioned into multiple "logical" drives



  - Logical drive 1: about 10 Gb
  - Logical drive 2: about 20 Gb
  - Logical drive 3: about 20 Gb
- OS will treat each logical drive as a separate disk; e.g.:

  C: Logical drive 1
  D: Logical drive 2
  E: Logical drive 3
- Partition information is stored on the disk (in a partition table)

Drive partions: why? how?

Why partition?

Because we want more than one filesystem on the disk

- Reserve space for things; for example
  - 50 Gb reserved for the OS
  - 150 Gb for user files
- Installing more than one OS
  - Windows requires NTFS
  - Linux requires ext4 or xfs
- Different needs
  - Sensitive data: use an encrypted filesystem
  - Other data: use an unencrypted filesystem

How to partition?

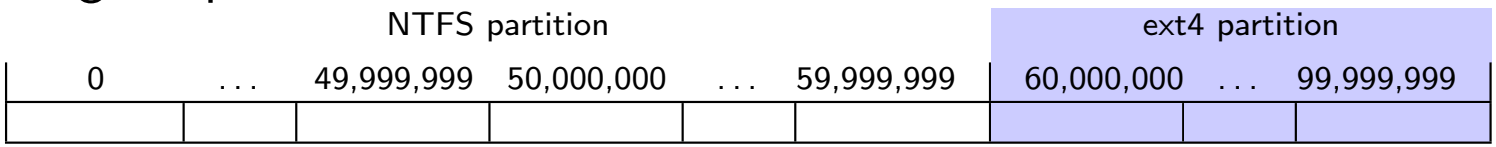
Use disk partitioning software (carefully)...

Changing partitions

- Some partitioning tools modify disk partitions and filesystems
- Some partitioning tools modify only the disk partitions
- What happens if we change only the disk partitions?

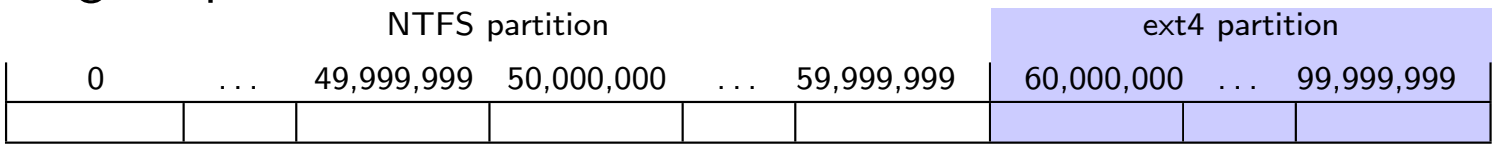
Changing partitions

- Some partitioning tools modify disk partitions and filesystems
- Some partitioning tools modify only the disk partitions
- What happens if we change only the disk partitions?
  - Original partition:

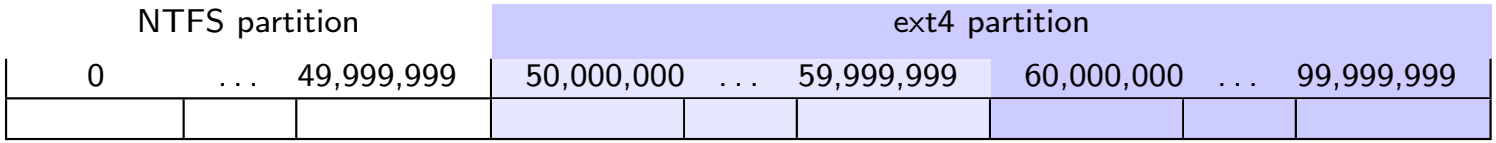


Changing partitions

- Some partitioning tools modify disk partitions and filesystems
- Some partitioning tools modify only the disk partitions
- What happens if we change only the disk partitions?
  - Original partition:

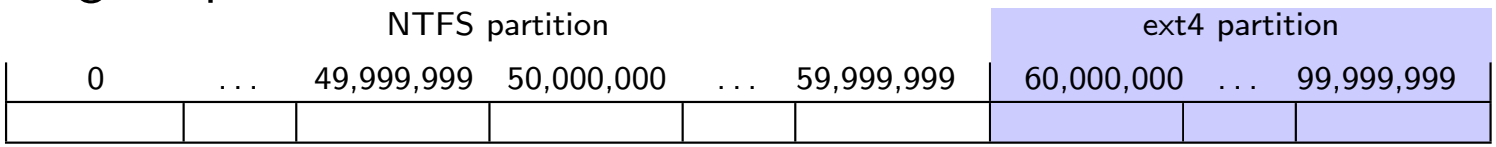


  - New partition (without changing filesystems):

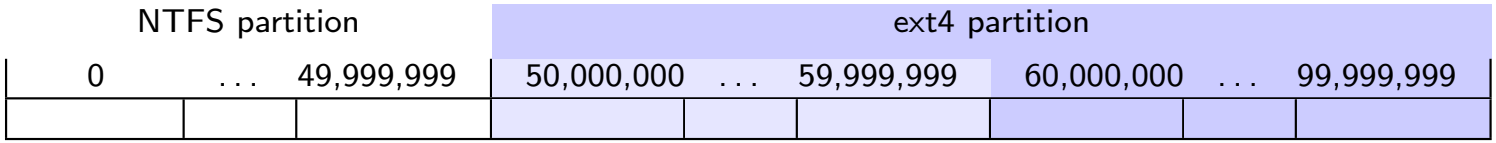


Changing partitions

- Some partitioning tools modify disk partitions and filesystems
- Some partitioning tools modify only the disk partitions
- What happens if we change only the disk partitions?
  - Original partition:



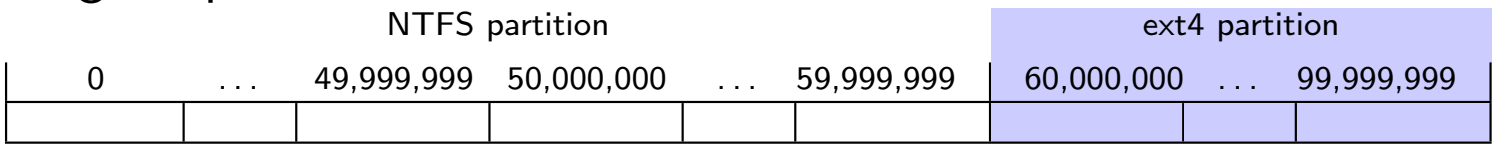
  - New partition (without changing filesystems):



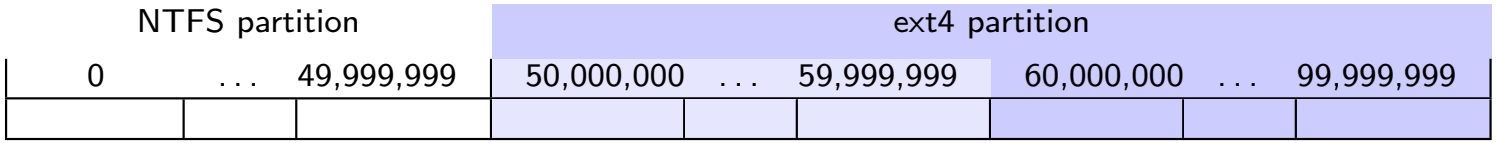
  - Sectors 50,000,000 — 59,999,999 still have NTFS data

Changing partitions

- Some partitioning tools modify disk partitions and filesystems
- Some partitioning tools modify only the disk partitions
- What happens if we change only the disk partitions?
  - Original partition:



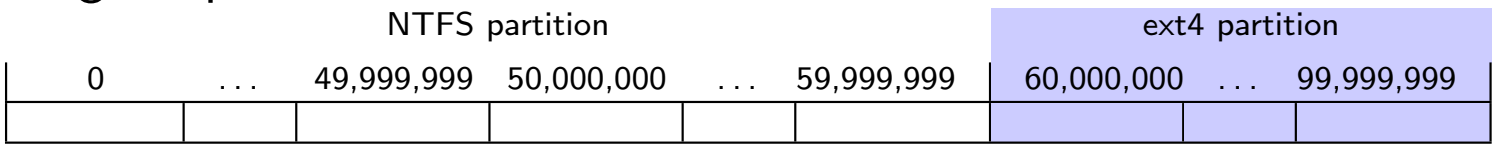
  - New partition (without changing filesystems):



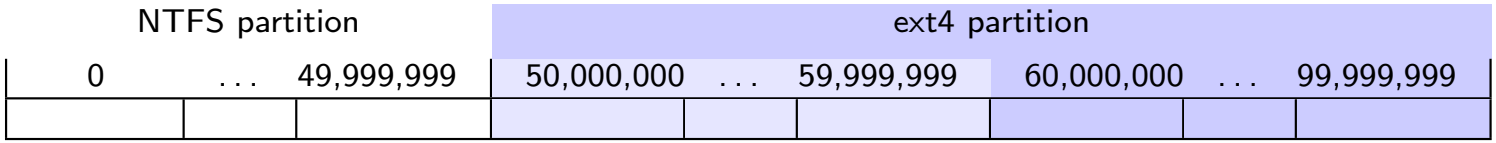
  - Sectors 50,000,000 — 59,999,999 still have NTFS data
  - Best–case scenario:
    - Corrupt ext4 partition is fixable
    - Corrupt NTFS partition is fixable except for some missing files

Changing partitions

- Some partitioning tools modify disk partitions and filesystems
- Some partitioning tools modify only the disk partitions
- What happens if we change only the disk partitions?
  - Original partition:



  - New partition (without changing filesystems):



  - Sectors 50,000,000 — 59,999,999 still have NTFS data
  - Best–case scenario:
    - Corrupt ext4 partition is fixable
    - Corrupt NTFS partition is fixable except for some missing files
  - Worst–case scenario:
    - Corrupt ext4 partition unreadable by OS
    - Corrupt NTFS partition unreadable by OS

How to change partitions and keep data

- Method 1:
  1. BACK UP ALL DATA just in case
  2. Run a utility to shrink, move, or expand affected filesystems
  3. Change partitions (might be part of above utility)

How to change partitions and keep data

- Method 1:
  1. BACK UP ALL DATA just in case
  2. Run a utility to shrink, move, or expand affected filesystems
  3. Change partitions (might be part of above utility)
- Method 2 (more work, but safer):
  1. BACK UP ALL DATA
  2. Change partitions
  3. Re-format changed partitions
  4. Copy data back
     (May require OS reinstall)

Virtual Memory

- Allows OS to use more than available physical RAM
- Disk is used for extra storage
- Each memory page can be on disk or in RAM
- OS needs disk space for virtual memory, called swap space
  - Windows: uses a single paging file
  - Mac OS X: uses multiple swap files
  - Linux: can use swap files or swap partitions

Miscellaneous Filesystems

Linux swap Used for Linux swap partitions

ISO 9660 Standard for optical disks

- `iso` files are images of an ISO 9660 file system

LVM (Logical Volume Manager)

- Mechanism to map logical volumes to physical drives or partitions
- Can "virtually" partition using LVM
  - LVM can shrink, expand, and move logical volumes . . . and their data
  - Makes it easy to change virtual partitions

Typical PC boot sequence

1. Power on
2. BIOS is loaded first
   - BIOS: Basic Input Output System
   - Used to configure hardware
3. BIOS then loads the MBR
   - MBR: Master Boot Record
   - Usually, sector 0 on the first disk
   - Holds disk partition table
4. MBR contains a bootloader
   - bootloader: a small program that can boot an OS
5. Bootloader loads OS kernel

What about multiple–boot machines?

- Partition disk(s) for separate filesystems
- Install different OS's on different partitions
- Use a bootloader that can boot different OS's
  - For example, `grub`
  - Get a menu at boot time; can select OS

What about multiple–boot machines?

- Partition disk(s) for separate filesystems
- Install different OS's on different partitions
- Use a bootloader that can boot different OS's
  - For example, `grub`
  - Get a menu at boot time; can select OS
- How many OS's can I have?

What about multiple–boot machines?

- Partition disk(s) for separate filesystems
- Install different OS's on different partitions
- Use a bootloader that can boot different OS's
  - For example, `grub`
  - Get a menu at boot time; can select OS
- How many OS's can I have?
  - Practical limits:
    - Number of disks
    - Number of allowed partitions per disk
    - Total disk capacity

MBR partition table limitations

- Only space for 4 entries
  - Called primary partitions
  - Some OS's (Windows) must boot from a primary partition
  - Can subdivide primary partitions
    1. Set the partition as extended
    2. Define logical partitions within the extended partition
- Sector size is fixed at 512 bytes
- Partition lengths stored as 32–bit integers
  - Maximum partition size is 2^{32} sectors

MBR partition table limitations

- Only space for 4 entries
  - Called primary partitions
  - Some OS's (Windows) must boot from a primary partition
  - Can subdivide primary partitions
    1. Set the partition as extended
    2. Define logical partitions within the extended partition
- Sector size is fixed at 512 bytes
- Partition lengths stored as 32–bit integers
  - Maximum partition size is 2^{32} sectors
    $2^{32} \cdot 512 = 2$ Tibi

MBR partition table limitations

- Only space for 4 entries
  - Called primary partitions
  - Some OS's (Windows) must boot from a primary partition
  - Can subdivide primary partitions
    1. Set the partition as extended
    2. Define logical partitions within the extended partition
- Sector size is fixed at 512 bytes
- Partition lengths stored as 32–bit integers
  - Maximum partition size is 2^{32} sectors
    $2^{32} \cdot 512 = 2$ Tibi
- Partition starts stored as 32–bit integers

MBR partition table limitations

- Only space for 4 entries
  - Called primary partitions
  - Some OS's (Windows) must boot from a primary partition
  - Can subdivide primary partitions
    1. Set the partition as extended
    2. Define logical partitions within the extended partition
- Sector size is fixed at 512 bytes
- Partition lengths stored as 32–bit integers
  - Maximum partition size is 2^{32} sectors
    $2^{32} \cdot 512 = 2$ Tibi
- Partition starts stored as 32–bit integers
- Largest supported drive is TiBi

MBR partition table limitations

- Only space for 4 entries
  - Called primary partitions
  - Some OS's (Windows) must boot from a primary partition
  - Can subdivide primary partitions
    1. Set the partition as extended
    2. Define logical partitions within the extended partition
- Sector size is fixed at 512 bytes
- Partition lengths stored as 32–bit integers
  - Maximum partition size is 2^{32} sectors
    $2^{32} \cdot 512 = 2$ Tibi
- Partition starts stored as 32–bit integers
- Largest supported drive is 4 TiBi
  - The last 2 TiBi must be a single, separate partition

Unified Extensible Firmware Interface

- UEFI standard replaced EFI in 2005
- Designed as a replacement for BIOS
  - Can run in 32–bit or 64–bit mode
    - If you have a 64–bit CPU
    - BIOS runs in 16–bit mode
  - Need EFI bootloaders
- Can use MBR partitioning
- Can also use GUID partition tables (GPT)
  - GUID: Globally unique identifier (128 bits)
  - Not tied to 512–byte sectors
  - Use 64–bit integers for partition locations
    - For 512–byte sectors: 8 Zb limit
    - For 4,096–byte sectors: 64 Zb limit
    - Capacity–wise, this should be fine through 2060
  - 128 partitions are allowed (or more. . .)

OS support for GPT

- Linux, FreeBSD, Solaris, HP-UX: YES
- Mac OS X: YES
  - All Intel Macintoshes use UEFI/GPT
- Windows, 32–bit: YES
  - Since Windows 8
  - Requires UEFI
- Windows, 64–bit: YES
  - Since Windows Server 2003
  - Requires UEFI

OS support for GPT

- Linux, FreeBSD, Solaris, HP-UX: YES
- Mac OS X: YES
  - All Intel Macintoshes use UEFI/GPT
- Windows, 32–bit: YES
  - Since Windows 8
  - Requires UEFI
- Windows, 64–bit: YES
  - Since Windows Server 2003
  - Requires UEFI

But before Windows 8 came out, how did Macs run both Mac OS and 32–bit Windows?

OS support for GPT

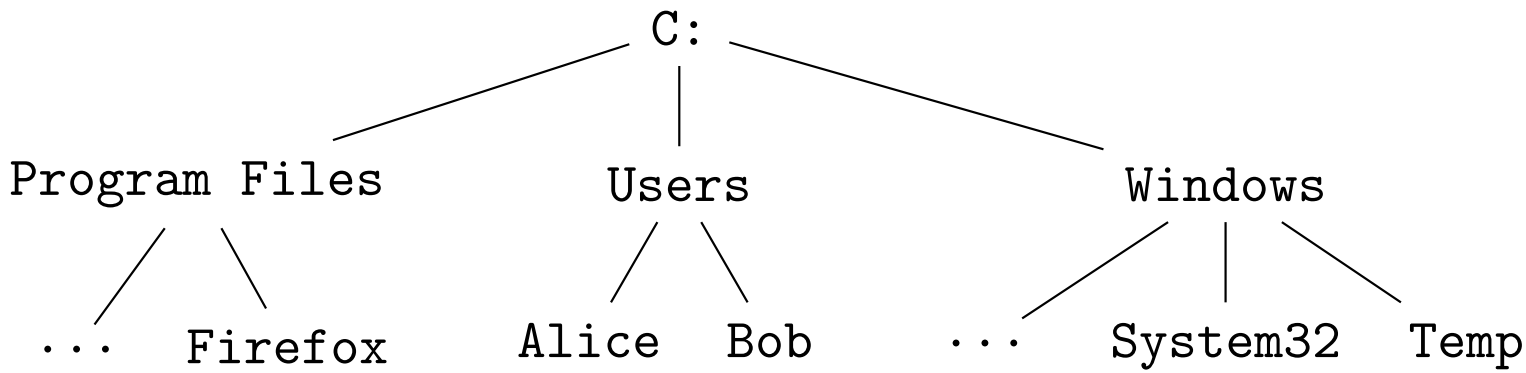
- Linux, FreeBSD, Solaris, HP-UX: YES
- Mac OS X: YES
  - All Intel Macintoshes use UEFI/GPT
- Windows, 32–bit: YES
  - Since Windows 8
  - Requires UEFI
- Windows, 64–bit: YES
  - Since Windows Server 2003
  - Requires UEFI

But before Windows 8 came out, how did Macs run both Mac OS and 32–bit Windows?

- Hack: Use a "hybrid" MBR
  - MBR and GPT on the same system
  - Need to be careful with this . . .

Filesystem management in Windows

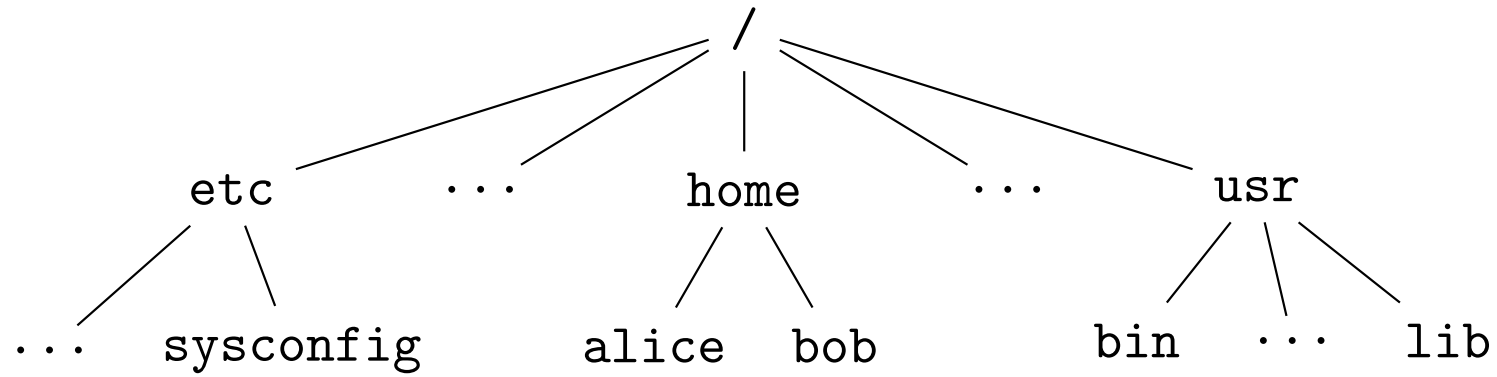
- Each logical drive is distinguished by a "drive letter"; e.g.,
  - C: NTFS filesystem on first partition of first drive
  - D: FAT filesystem on second partition of first drive
  - F: ISO-9660 filesystem in CD–ROM drive
- Subdirectory structure gives a tree for each drive letter



- Subdirectory separator is a backslash, e.g.,
  `C:\Windows\System32`

Filesystem management in *NIX

- There is only one system–wide tree of files



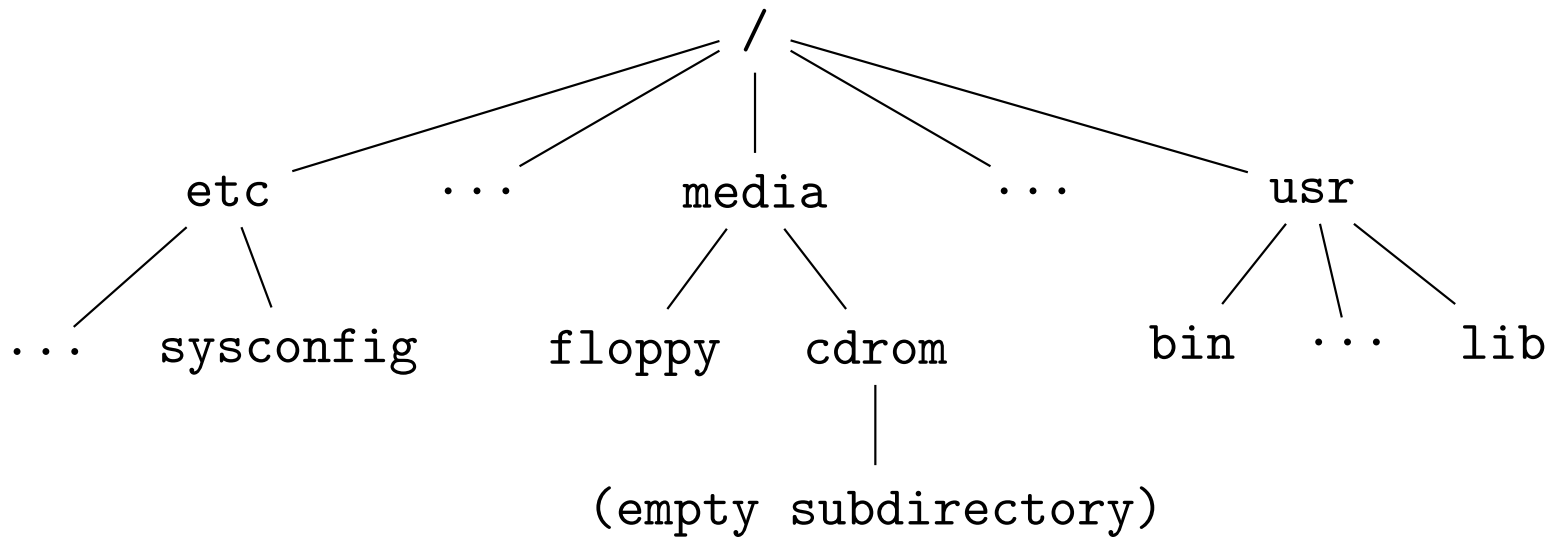
- Subdirectory separator is (forward) slash, e.g., `/home/alice/.bashrc`
- Logical drives are integrated into the system–wide tree

  Mount : Add a filesystem (on a logical drive) to the tree

  Mount point : Directory where a filesystem is mounted

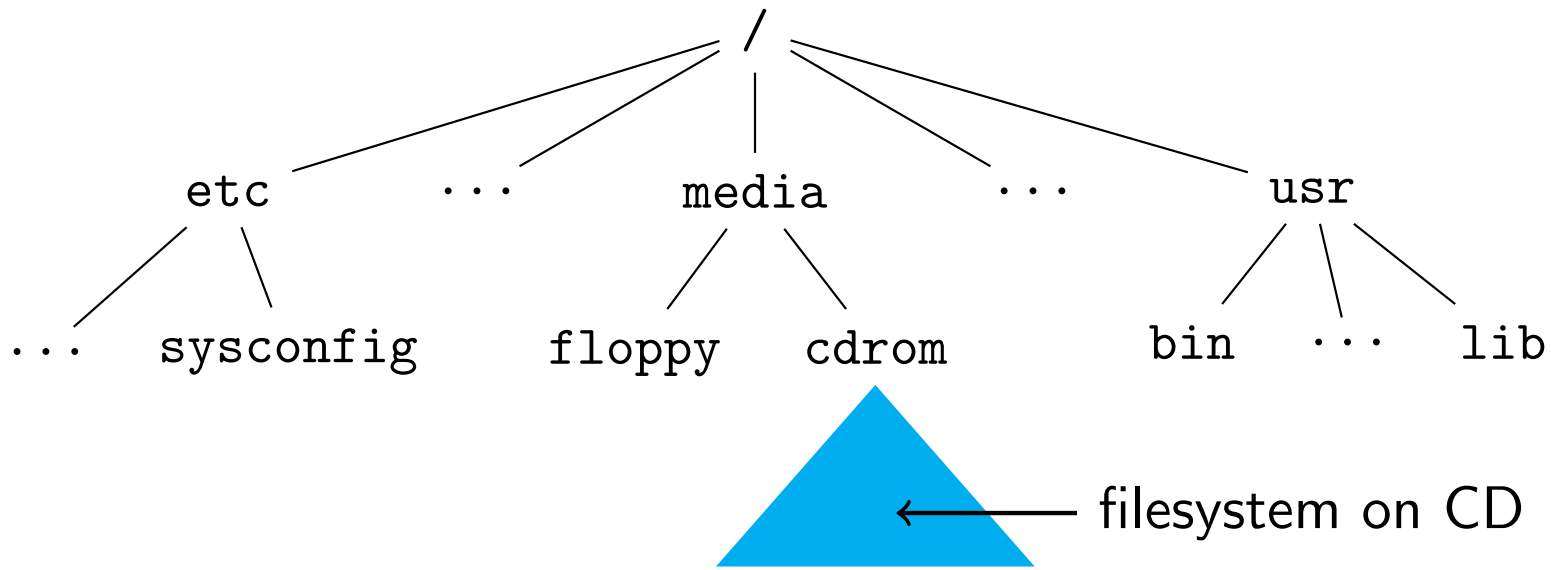
  Unmount : Remove a logical drive from the tree

Example: mounting and unmounting



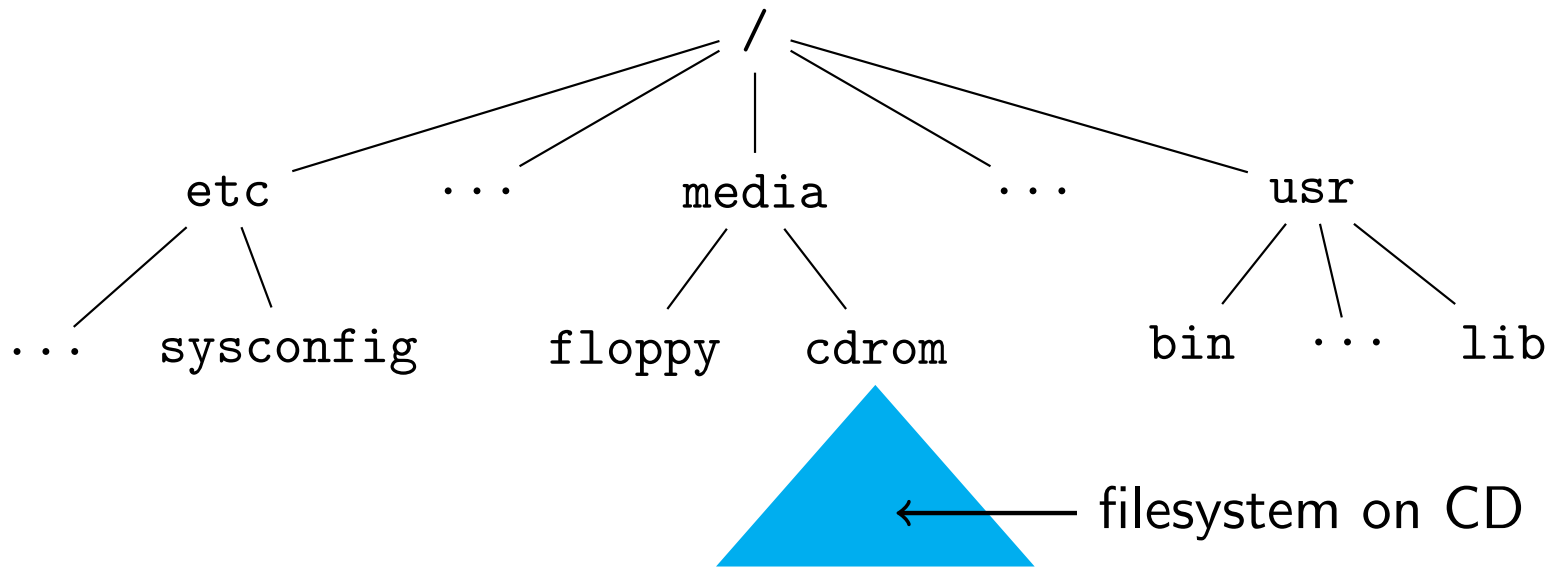
1. Insert CD
2. Mount CD filesystem at `/media/cdrom`
3. Work with files ...
4. Close all files
5. Unmount CD filesystem
6. Eject CD

Example: mounting and unmounting



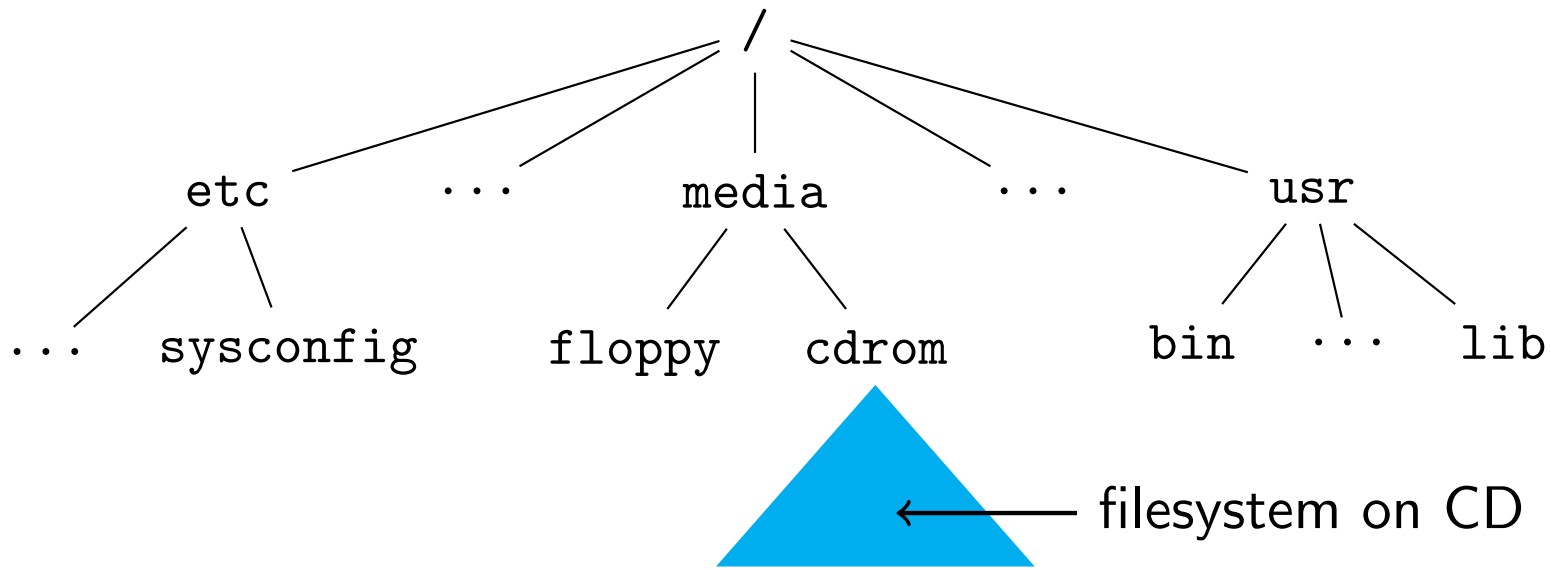
1. Insert CD
2. Mount CD filesystem at `/media/cdrom`
3. Work with files ...
4. Close all files
5. Unmount CD filesystem
6. Eject CD

Example: mounting and unmounting



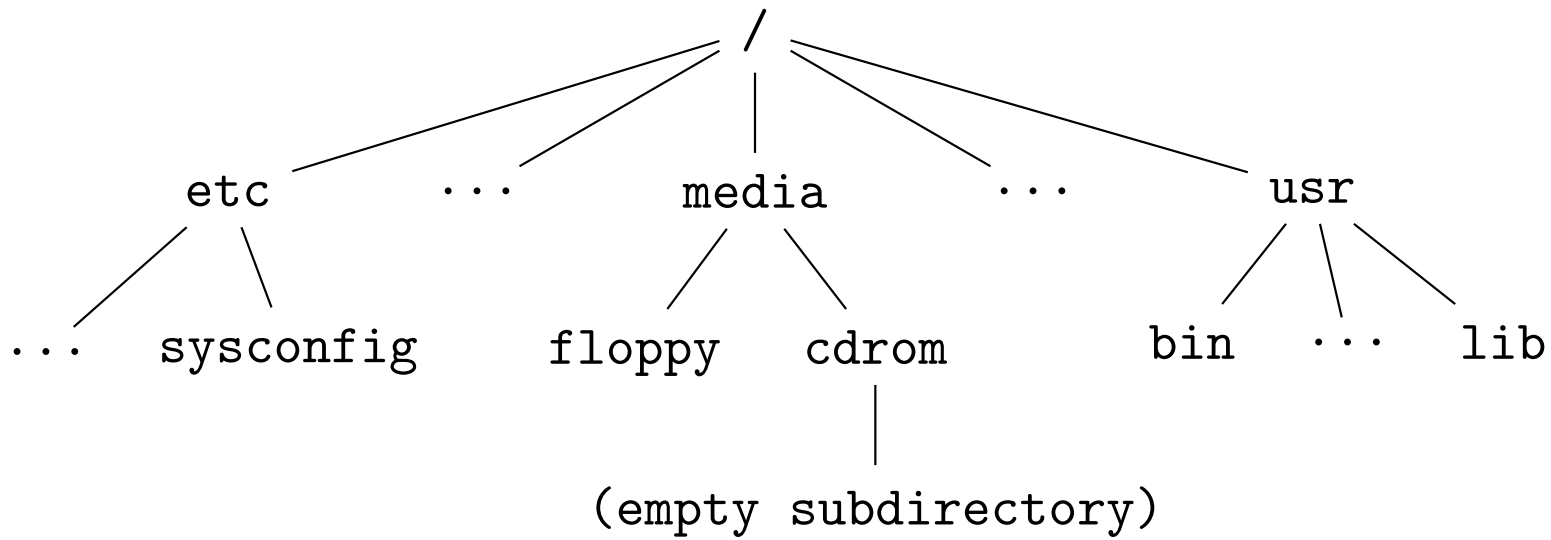
1. Insert CD
2. Mount CD filesystem at `/media/cdrom`
3. Work with files . . .
4. Close all files
5. Unmount CD filesystem
6. Eject CD

Example: mounting and unmounting



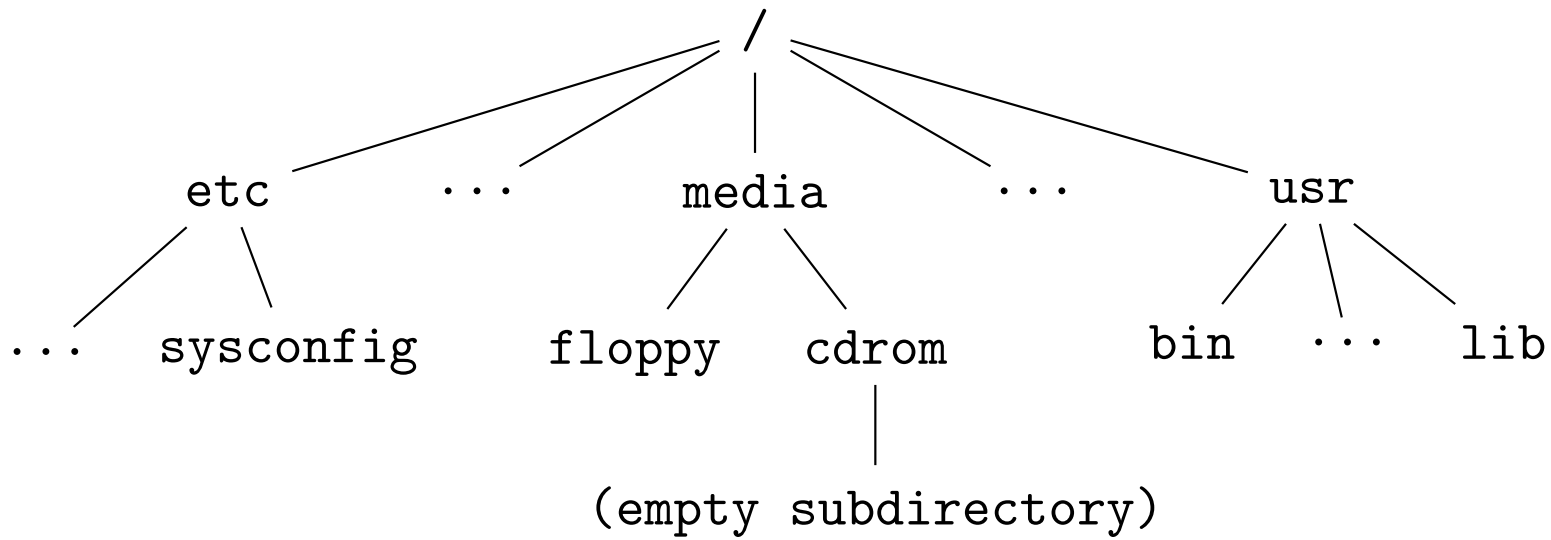
1. Insert CD
2. Mount CD filesystem at `/media/cdrom`
3. Work with files ...
4. Close all files
5. Unmount CD filesystem
6. Eject CD

Example: mounting and unmounting



1. Insert CD
2. Mount CD filesystem at `/media/cdrom`
3. Work with files ...
4. Close all files
5. Unmount CD filesystem
6. Eject CD

Example: mounting and unmounting



1. Insert CD
2. Mount CD filesystem at `/media/cdrom`
3. Work with files ...
4. Close all files
5. Unmount CD filesystem
6. Eject CD

Typical Linux filesystem layout

/bin : Essential system binaries

/boot : Kernel image and configuration

/dev : Devices. More on these later.

/etc : System configuration files

/home : Directories for user files

/lib : Software libraries (similar to `dlls`)

/media : Mount points for removable media

/tmp : Temporary files, automatically deleted

/usr : Most other system binaries

/var : Variable, runtime stuff

Example dual–boot setup (old laptop, 11.2 Gb drive)

#	Size	Type	Mount point	Description
1	140 Mb	ext3	/boot	Linux boot
2	4.3 Gb	ext3	/	Linux root
3	2.0 Gb	NTFS	(none)	Windows C:
4	4.8 Gb		Extended partition	
5	2.5 Gb	ext3	/home	Linux user files
6	1.4 Gb	FAT32	/D	Windows D:
7	384 Mb	swap	(none)	Linux swap space

Why this was chosen:

1. Swap space should be larger than RAM
   - Original RAM was 128 Mb
2. Separate /home partition keeps files safe during upgrades
3. FAT32 D: drive to move files between Linux and Windows
   - This was before NTFS support in Linux

Example Macintosh triple–boot setup[1]

GUID Partition Table:

#	Size	Type	Linux mount	Description
1	210 Mb	FAT32	(none)	EFI tables
2	220 Gb	HFS+	`/mnt/Mac`	Macintosh FS
3	50 Gb	NTFS	`/mnt/Win`	Windows FS
4	524 Mb	ext3	`/boot`	Linux boot
5	21 Gb	ext4	`/home`	User files
6	28 Gb	LVM	(Linux root and swap space)	

Notes:

1. MBR partition table contains only the first 4 entries
   - Works because only Windows reads the MBR
2. Newer versions of Linux can mount NTFS and HFS+
3. LVM for / and swap: can resize without changing GPT, MBR

[1] 32-bit and 64-bit machines use different, non-trivial setups. YMMV.

Example Macintosh triple–boot setup[1]

GUID Partition Table:

#	Size	Type	Linux mount	Description
1	210 Mb	FAT32	(none)	EFI tables
2	220 Gb	HFS+	`/mnt/Mac`	Macintosh FS
3	50 Gb	NTFS	`/mnt/Win`	Windows FS
4	524 Mb	ext3	`/boot`	Linux boot
5	21 Gb	ext4	`/home`	User files
6	28 Gb	LVM	(Linux root and swap space)	

Notes:

1. MBR partition table contains only the first 4 entries
   - Works because only Windows reads the MBR
2. Newer versions of Linux can mount NTFS and HFS+
3. LVM for / and swap: can resize without changing GPT, MBR

[1] 32-bit and 64-bit machines use different, non-trivial setups. YMMV.

Example Macintosh triple–boot setup[1]

GUID Partition Table:

#	Size	Type	Linux mount	Description
1	210 Mb	FAT32	(none)	EFI tables
2	220 Gb	HFS+	`/mnt/Mac`	Macintosh FS
3	50 Gb	NTFS	`/mnt/Win`	Windows FS
4	524 Mb	ext3	`/boot`	Linux boot
5	21 Gb	ext4	`/home`	User files
6	28 Gb	LVM	(Linux root and swap space)	

Notes:

1. MBR partition table contains only the first 4 entries
   - Works because only Windows reads the MBR
2. Newer versions of Linux can mount NTFS and HFS+
3. LVM for / and swap: can resize without changing GPT, MBR

[1] 32-bit and 64-bit machines use different, non-trivial setups. YMMV.

Example Macintosh triple–boot setup[1]

GUID Partition Table:

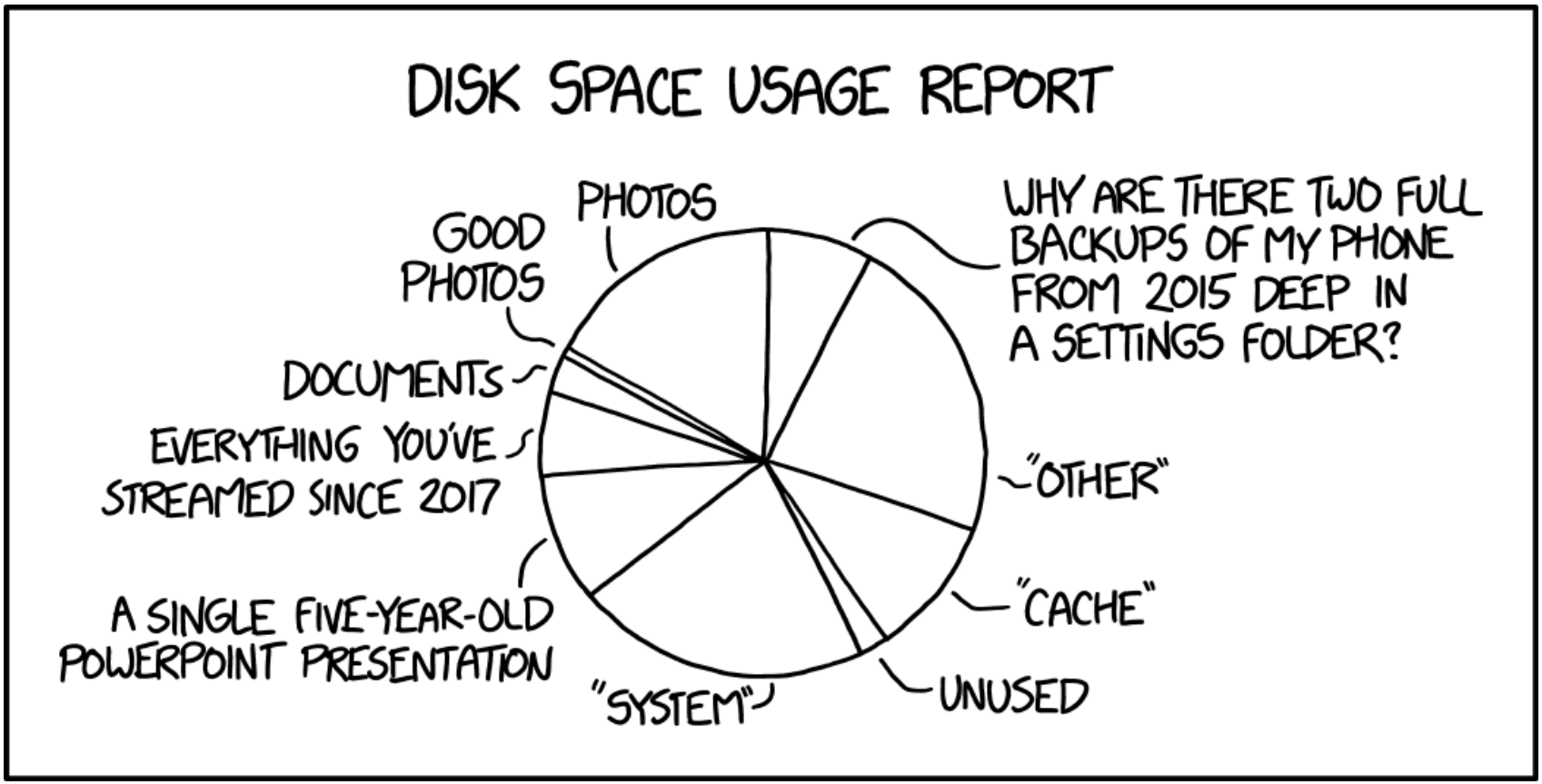
#	Size	Type	Linux mount	Description
1	210 Mb	FAT32	(none)	EFI tables
2	220 Gb	HFS+	`/mnt/Mac`	Macintosh FS
3	50 Gb	NTFS	`/mnt/Win`	Windows FS
4	524 Mb	ext3	`/boot`	Linux boot
5	21 Gb	ext4	`/home`	User files
6	28 Gb	LVM	(Linux root and swap space)	

Notes:

1. MBR partition table contains only the first 4 entries
   - Works because only Windows reads the MBR
2. Newer versions of Linux can mount NTFS and HFS+
3. LVM for / and swap: can resize without changing GPT, MBR

[1] 32-bit and 64-bit machines use different, non-trivial setups. YMMV.

An appropriate xkcd comic: http://www.xkcd.com/2143



End of lecture