PURDUE UNIVERSITY

CS 25200: Systems Programming

Lecture 9: File Systems

Prof. Turkstra

Lecture 09

- Program loading
- Storage
- Partitioning
- RAID
- Encryption
- inodes
- File systems

Loader

- Essential step in starting a program
- Historically allocated space for all sections of the executable (text, data, bss, etc)
- Now simply establishes mappings
  - Page faults actually populate the memory
  - For executable as well as (shared) libraries

- Also resolves any values in the executable to point to the functions/variables in the shared libraries
- Jumps to _start
  - init()’s all libraries
  - _then calls main()
  - …and exit()
- Sometimes loaders are called “runtime linkers”

Interpreter

```
readelf --headers /bin/ls
```

Lazy binding

- Binding a function call to a library can be expensive
  - Have to go through code and replace the symbol with its address
- Delay until the call actually takes place
  - Calls stub PLT function
  - Invokes dynamic linker to load the function into memory and obtain real address
    - Rewrites address that the sub code references
    - Only happens once
- Procedure Lookup Table (PLT)

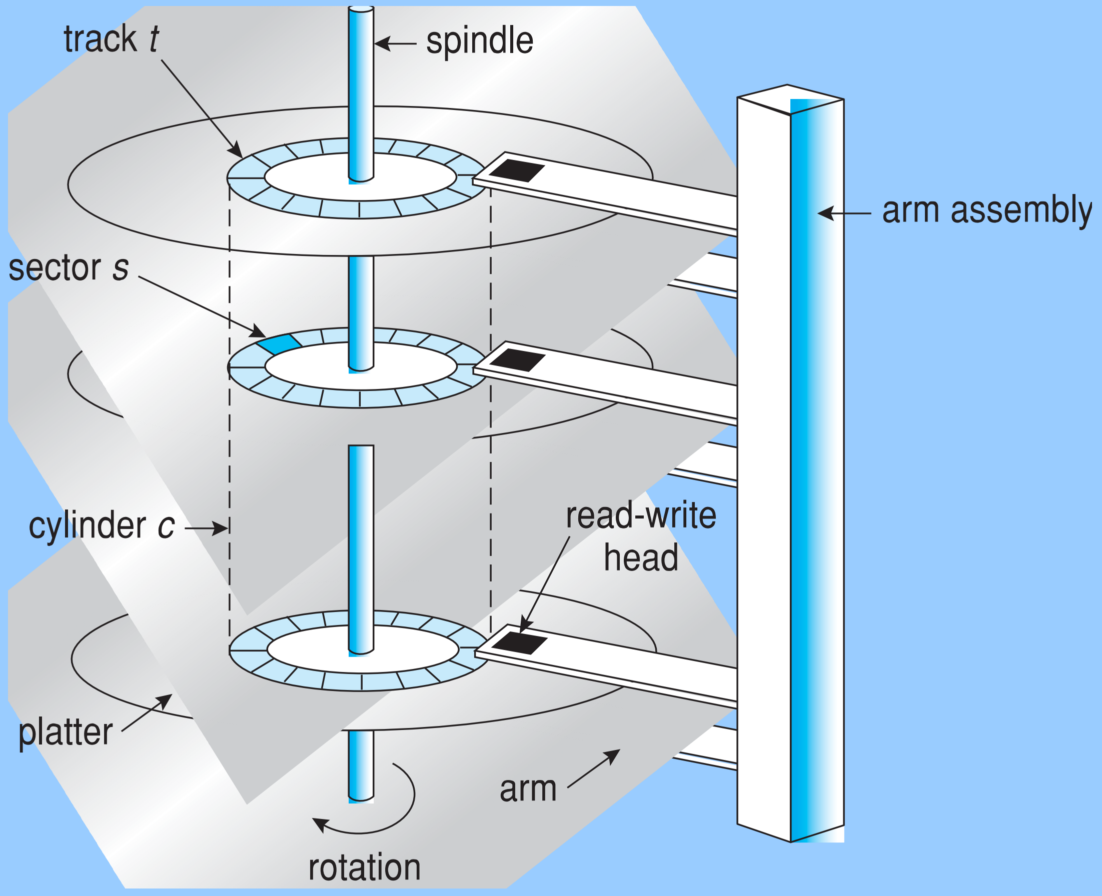
Storage

Starting at the bottom

- Block device
  - Hard disk
  - SSD
  - Tapes
  - More
- At least an order of magnitude (or more) slower than main memory
  - Fastest SSDs ~550MB/sec
  - DDR4 ~16,155MB/s
  - Latency worse

Hard drives

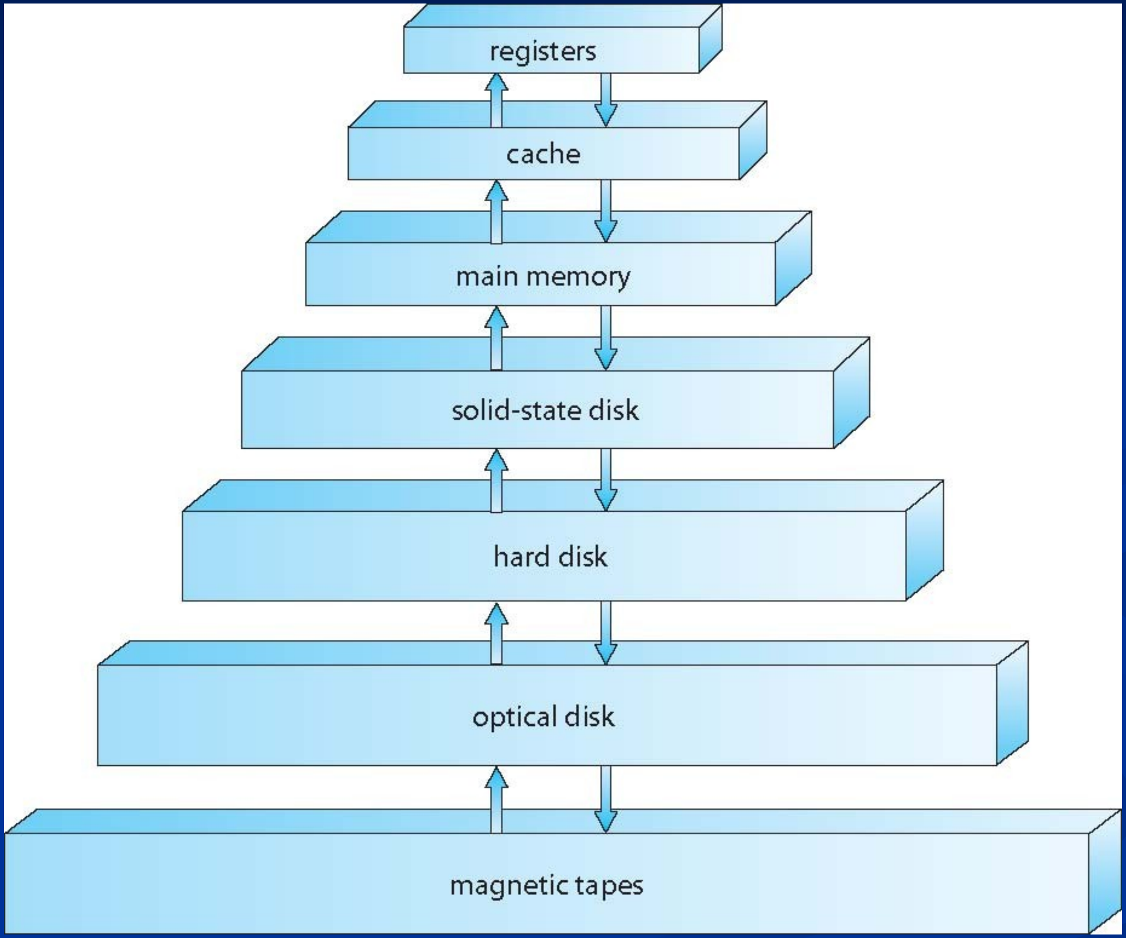
- Mechanical
  - Spinning platters
  - Moving heads
- Modern
  - Lie about sector size
  - On-board cache
  - ECC (Reed-Solomon)
  - Controller handles physical sector remaps


track *t*
spindle
arm assembly
sector *s*
cylinder *c*
read-write head
platter
arm
rotation

SSDs

- Solid state
  - No moving parts
  - Wear leveling
- 4-5 times faster than HDs
- Hybrid drives

Storage hierarchy



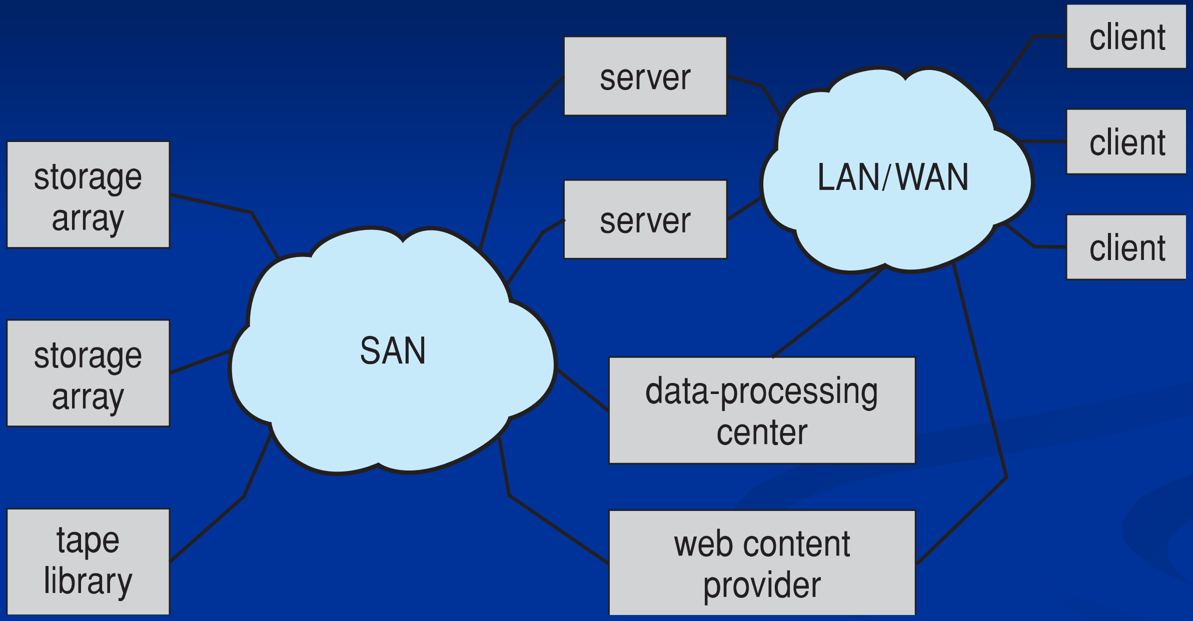
Disk structure

- Large one-dimensional arrays of logical blocks
  - Smallest unit of transfer
- Blocks mapped onto sectors
  - Sector 0 first sector, first track, outermost cylinder
  - Non-constant number of sectors per track
    - Constant angular velocity
- Bad sectors

Disk attachment

- Host-attached: SCSI, SATA, etc
- Fibre Channel
  - Often basis of a Storage Area Network (SAN)
- Network attached storage (NAS)

Storage area network



- Common in large storage environments
- Multiple hosts attached to multiple storage arrays

SAN

- One or more storage arrays
  - Connected to one or more Fibre Channel switches
- Hosts attach to switches as well
- Storage made available via LUN Masking

Network attached storage

- NAS, storage made available over network
- Remotely attaching file systems
- NFS, CIFS, Samba
- Remote procedure calls (RPCs) between hosts
- iSCSI
  - Uses IP network to carry SCSI protocol

Formatting

- Low-level or physical formatting
  - Divides disks into sectors
  - Each sector holds header information, data, and error correction code (ECC)
  - Usually 4096 bytes now
    - Used to be 512 bytes
    - Many disks can mimic 512 byte sectors
      - There's a cost if misaligned
- Logical formatting

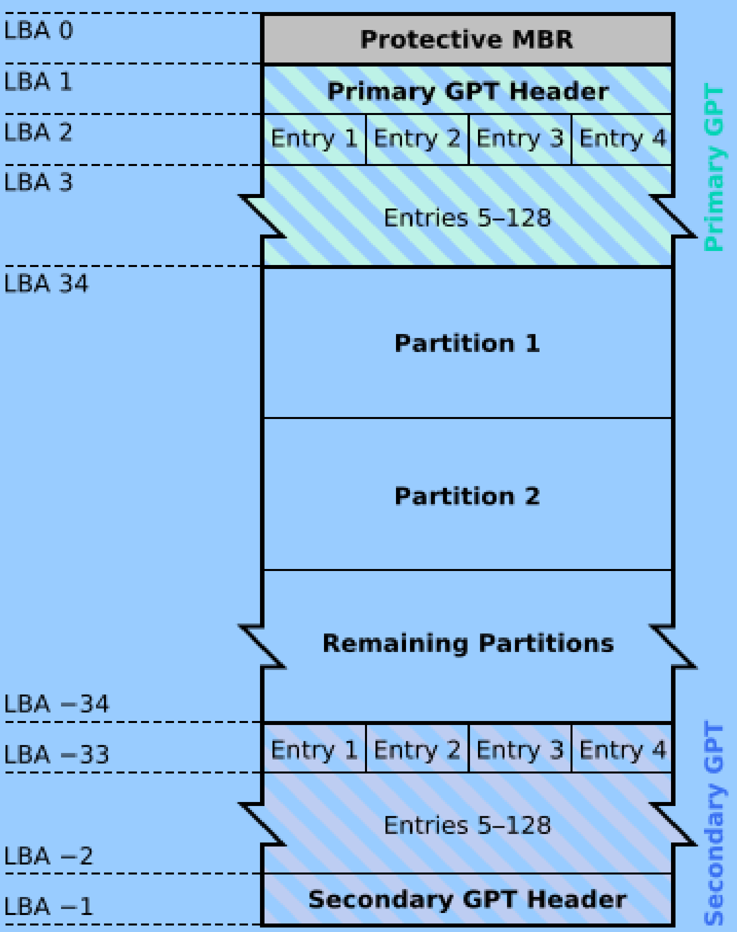
Partitions

- MBR – Master Boot Record
  - IBM PC DOS 2.0, 1983
  - Limit of 2TiB disk and partition size
  - Four primary partitions
  - Extended partitions
- GPT – GUID Partition Table
  - Part of UEFI
  - Relaxes above limitations
  - 128 partitions for Windoze
  - CRC ECC
  - Protective MBR

Boot sector

- Region of storage containing machine code that is loaded into RAM
  - Has just enough information to locate the OS kernel and begin loading it
- Why? BIOS knows nothing about the OS or file system

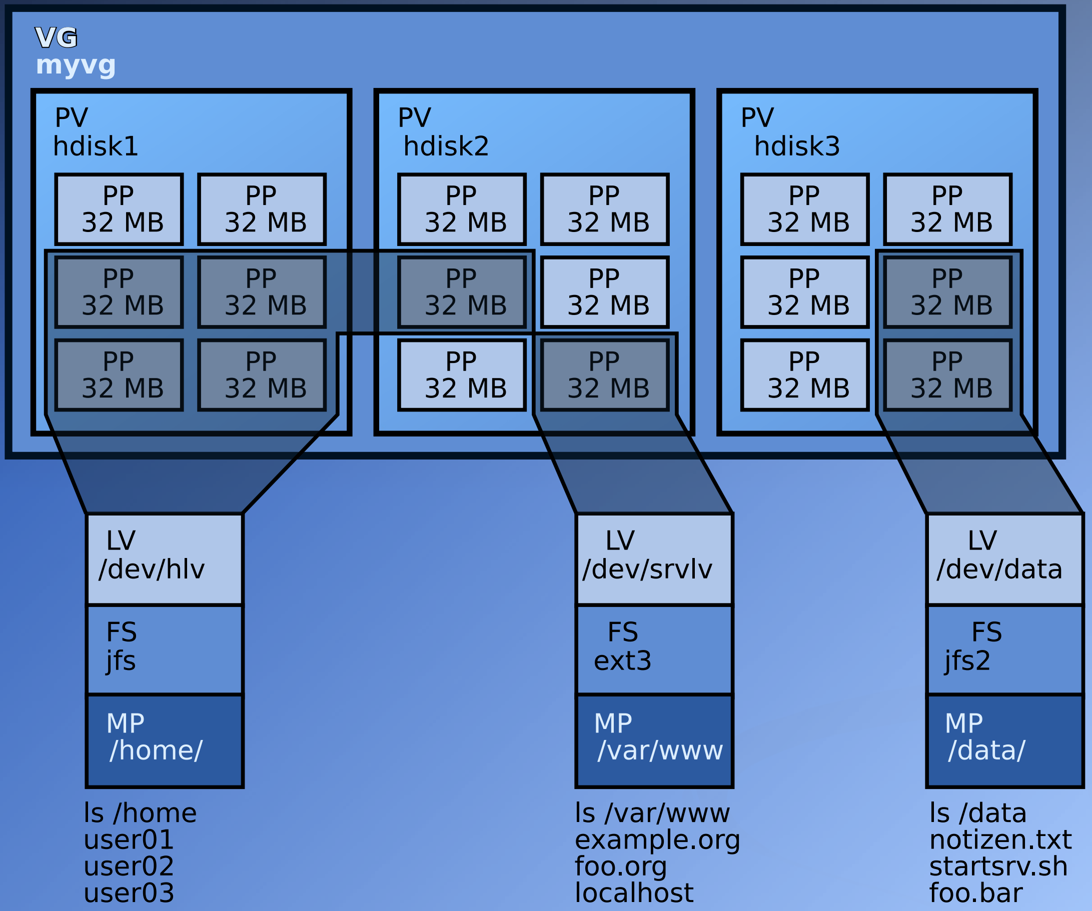
GUID Partition Table Scheme



Partition formats

- Regular FS (0x83)
- Swap (0x82)
- LVM Physical Disk (0x8e)
- Linux raid autodetect (0xfd)
- Often ignored

- fdisk/gdisk demo



VG
myvg

PV
hdisk1

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PV
hdisk2

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PV
hdisk3

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

PP
32 MB

LV
/dev/hlv

FS
jfs

MP
/home/

ls /home
user01
user02
user03

LV
/dev/srvlv

FS
ext3

MP
/var/www

ls /var/www
example.org
foo.org
localhost

LV
/dev/data

FS
jfs2

MP
/data/

ls /data
notizen.txt
startsrv.sh
foo.bar

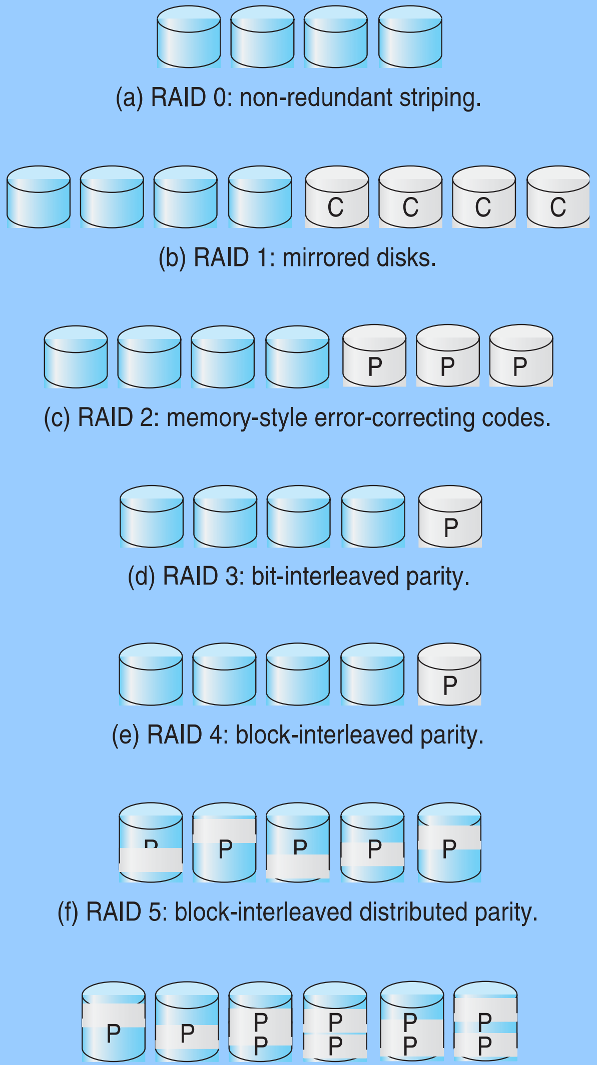
PP: Physical Partition
PV: Physical Volume
VG: Volume Group
LV: Logical Volume
FS: Filesystem
MP: Mounting Point

LVM
Logical Volume Manager

md (multiple device)

- Virtual devices created from one or more independent underlying devices
  - RAID-0: Block level striping
  - RAID-1: Mirrored
  - RAID-4: RAID-0 + parity
  - RAID-5: Distributed parity
  - RAID-6: RAID-5, except two parity segments
  - RAID 10: RAID-0 striped over RAID-1

RAID levels



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

RAID

- Redundant array of independent disks
- RAID is not a backup
- Fault tolerant
  - Hot spares

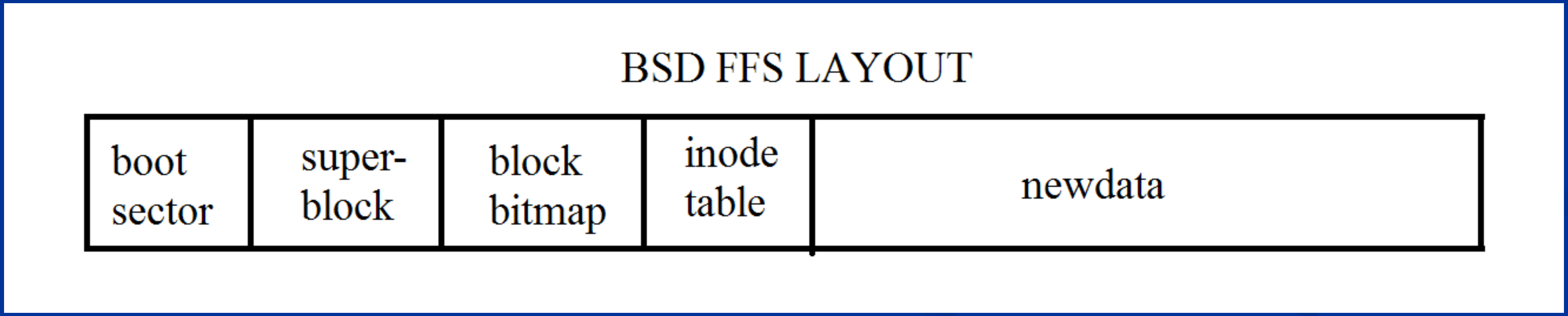
dm-crypt and LUKS

- dm-crypt
  - Encrypted block devices
- LUKS
  - Linux Unified Key Setup
  - Standardizes partition headers and data formats
- cryptsetup
  - Convenient interface to create encrypted block devices using the LUKS extension

Purdue trivia

- Purdue University is known as the "cradle of astronauts" with twenty one alumni having been chosen for space travel. Purdue astronauts include the first and last men on the moon, Neil Armstrong and Gene Cernan as well as one of America's original Project Mercury astronauts, Gus Grissom. Jerry Ross has logged 58 hours and 18 minutes in nine spacewalks - more than any other NASA astronaut.
- Purdue is also home to the oldest university-based airport in the nation.

BSD File System



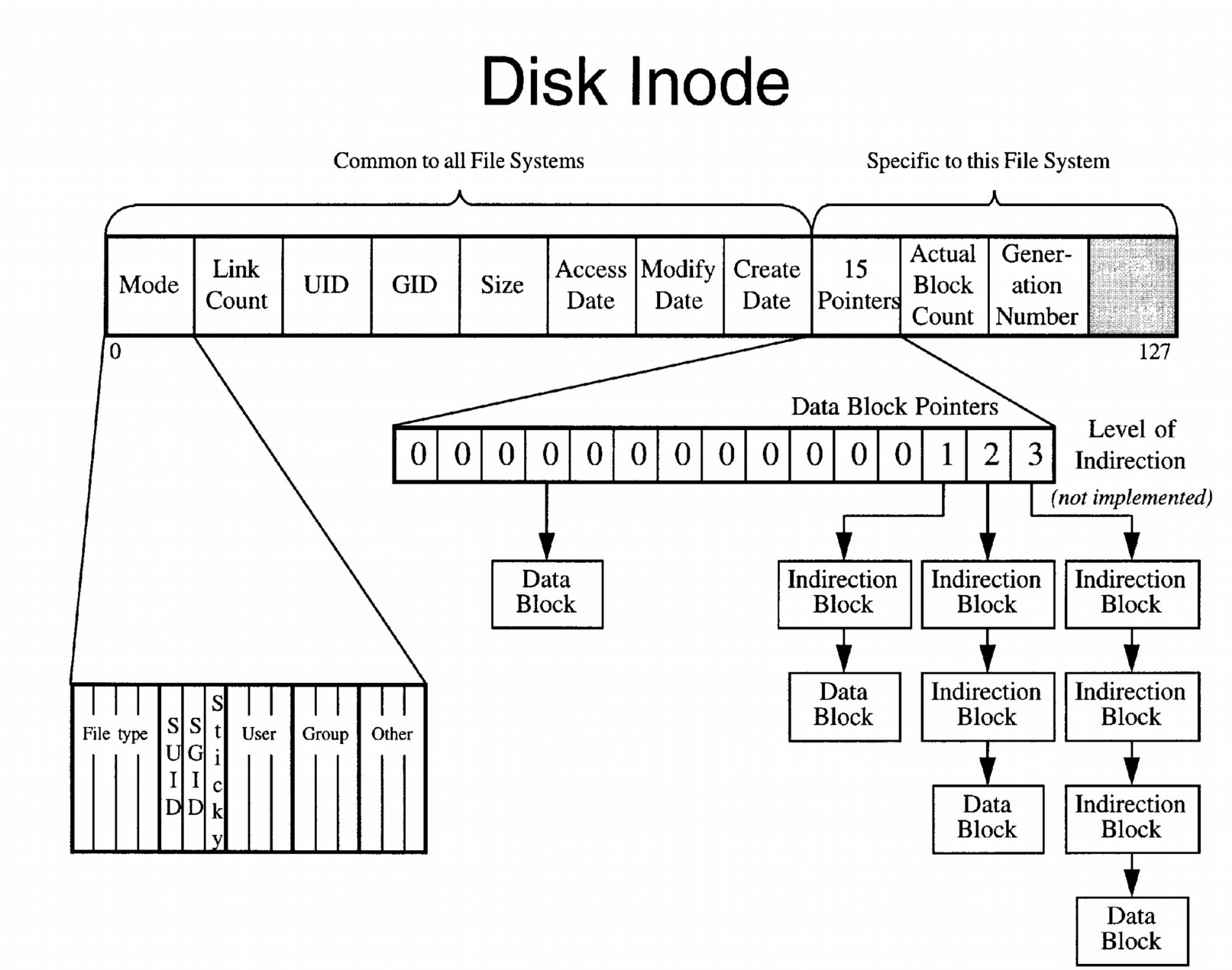
BSD FFS LAYOUT

boot sector	super-block	block bitmap	inode table	newdata

Superblock

- Record of the characteristics of a filesystem
  - Size
  - Block size
  - Empty/filled blocks
  - Size and location of inode table(s)
  - Block map and sizes
  - Location of root inode
- Copy maintained in memory while running

inode



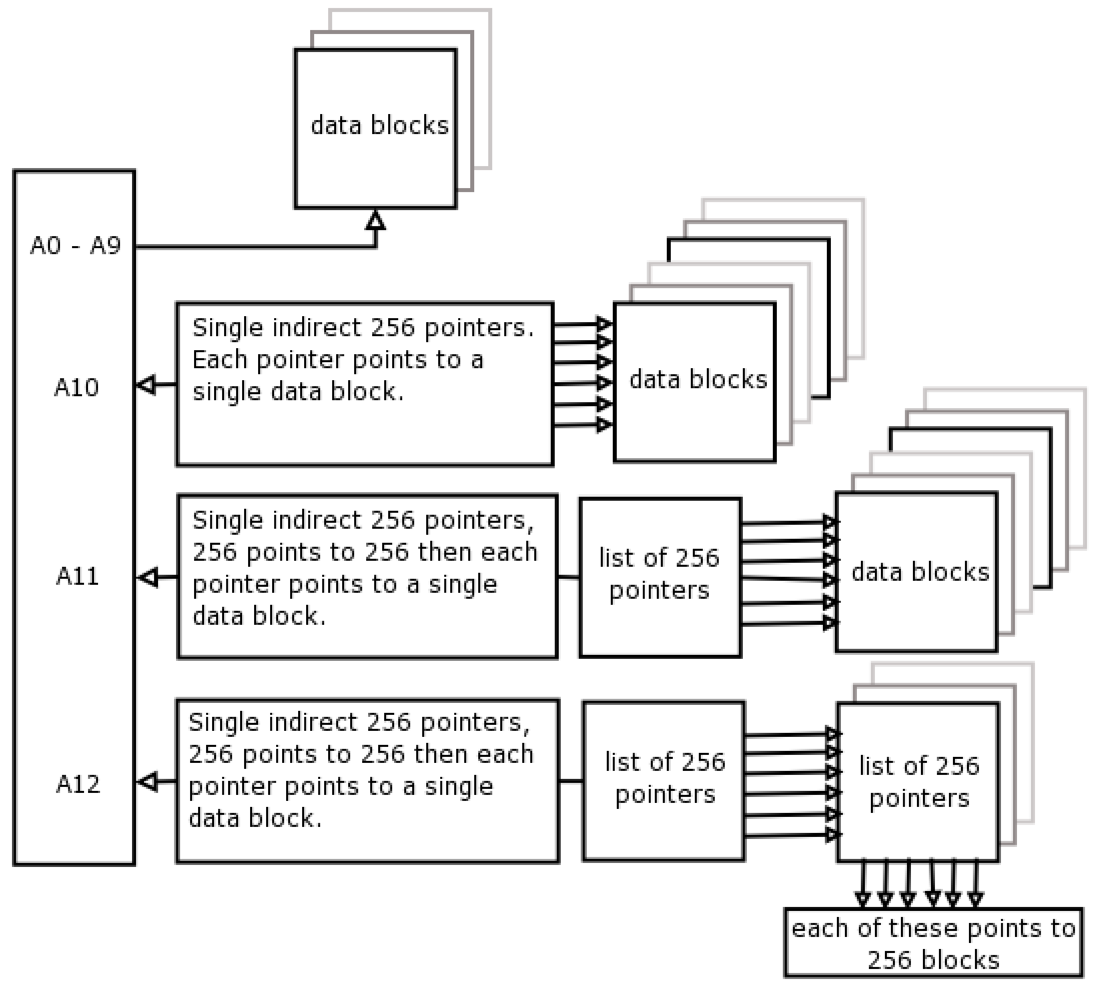
* http://web.cs.ucla.edu/classes/spring14/cs111/scribe/12b/

inode

- File size
- User ID (uid)
- Group ID (gid)
- Mode (rwx, special flags)
- Timestamps (ctime, atime, mtime)
- Link count
- Pointers to data blocks
- Many dictated by POSIX

Block addressing

- Data blocks are 1KiB in Linux
  - man 8 vmstat
- Direct block pointers – twelve pointers straight to a data block
- Single indirect – block of pointers to blocks (1024 / 4 = 256 pointers)
- Double indirect – block with 256 pointers to blocks with 256 pointers
- Triple indirect – yet another level of indirection



Single indirect 256 pointers. Each pointer points to a single data block.

Single indirect 256 pointers, 256 points to 256 then each pointer points to a single data block.

Single indirect 256 pointers, 256 points to 256 then each pointer points to a single data block.

each of these points to 256 blocks

* http://www.iakovlev.org/index.html?p=1251

Size matters not

- 12 * 1KiB = 12 KiB
- 256 * 1KiB = 256KiB
- 256 * 256 * 1KiB = 64MiB
- 256 * 256 * 256 * 1KiB = 16GiB

inode information

- Most files are small
- Having direct pointers gives us two disk reads to get a data block
- Lots of alternatives
- Linked list?
  - Terrible for random access
  - Great for sequential, though

Modern file systems

- Are considerably more complex
- ZFS: variable block sizes, dynamic striping, adaptive endianness, deduplication, etc
  - Dynamic inode allocation

Security

- Sometimes information security involves forensics
  - Knowing that there may be unwiped flash cells due to wear-leveling
  - Exploring the free blocks on a disk
  - FAT – put a NULL for the first character to delete the file
    - Exceptionally easy to “undelete”
    - Still relevant!

Linux file systems

- Actual file system varies
  - ext2/3/4
  - XFS
  - btrfs
  - ZFS
  - …and others

Questions?