PURDUE UNIVERSITY®

CS 25200: Systems Programming

Lecture 10: File Systems

Prof. Turkstra

© 2019 Dr. Jeffrey A. Turkstra

Lecture videos

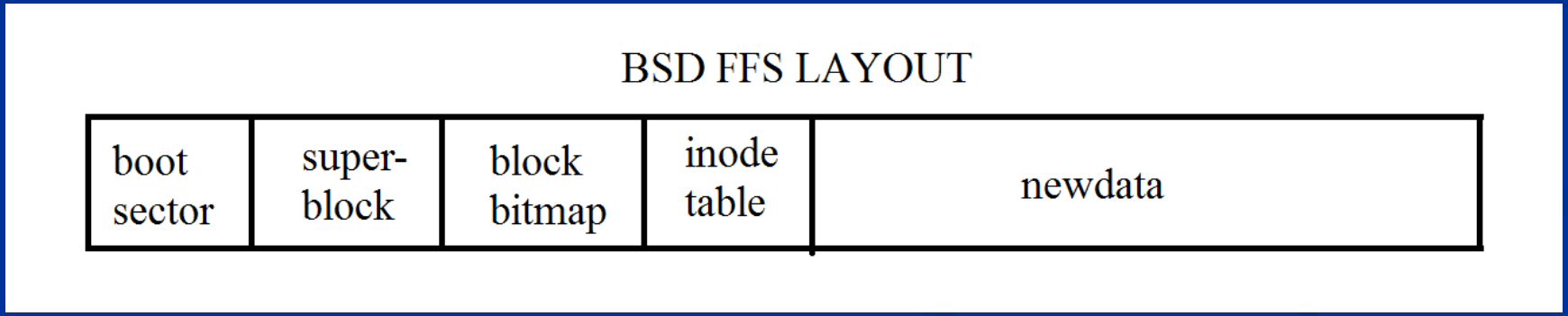
- https://mediaspace.itap.purdue.edu/

© 2019 Dr. Jeffrey A. Turkstra

Lecture 10

- inodes
- File systems
- Types
- Permissions
- Special bits
- Extended attributes
- DAC vs. MAC

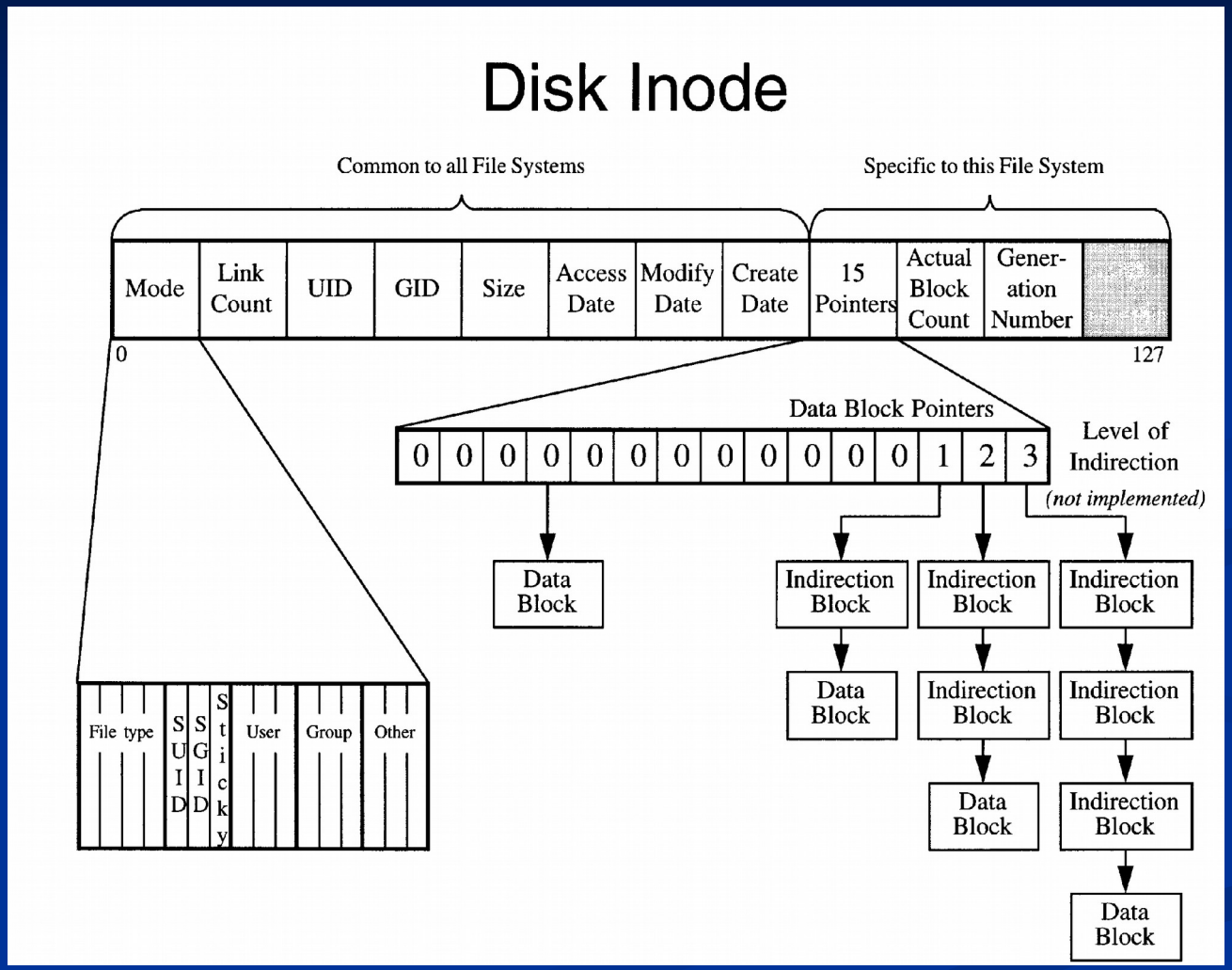
BSD File System



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



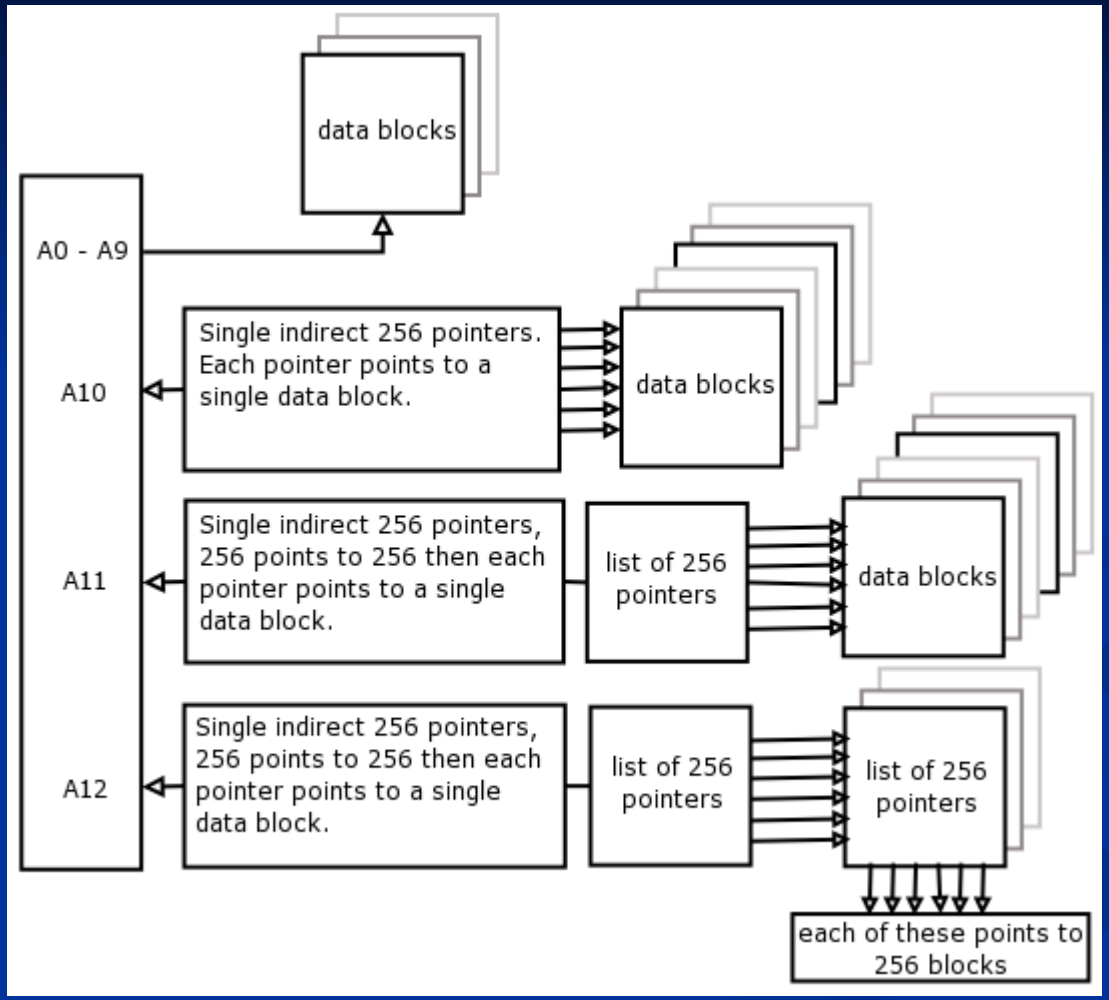
* http://web.cs.ucla.edu/classes/spring14/cs111/scribe/12b/

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



* 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

File systems

- Userland view is generally the same
  - Ownership
  - Permissions
  - Date and time information
  - Number of links
  - File size
  - Extended attributes
  - Directory hierarchy
- Kernel VFS layer

Exploring

- fdisk/gdisk
- mkfs
- tune2fs -l
- cryptsetup luksDump
- ls -ia
- stat

The UNIX mantra

- “On a UNIX system, everything is a file; if something is not a file, it is a process.”
- No difference between a file and a directory
  - Directory is just a file containing names of other files

Types of files

- Directories
  - Lists of other files
- Special files
  - Mechanisms for input/output
  - Often in /dev
- Links
  - Symbolic links
  - Hard links

Types cont.

- (Domain) Sockets
  - Inter-process networking protected by file system's access control
- Named pipes
  - Similar to sockets, without the networking semantics
- Regular files

ls -l

Symbol	Meaning
-	Regular file
d	Directory
l	Symbolic link
c	Special file
s	Socket
p	Named pipe
b	Block device

- Or, maybe ls -F

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.

File permissions

- Read: access the contents of a file
  - For directories, list the file names in a directory
- Write: modify a file
  - For directories, create/delete/rename
- Execute or search: execute a file
  - Not necessarily read its contents, though
  - Must be readable for interpreted files (eg, shell scripts, python, etc)
  - Directories: access a file given its explicit path. Cannot list files without the read bit

Classes

- User: the file owner
- Group: members of the group that owns the file
- Other: anyone that does not fall into the first two classes

Setting the mode

- chmod
  - Symbolic: ugoa[+-]rwxXst
  - Numeric:
    read = 4 (0b100)
    write = 2 (0b010)
    execute = 1 (0b001)
    - Eg, chmod 0711 myfile
  - 4000 for setuid
  - 2000 for setgid
  - 1000 for sticky

setuid/setgid bits

- setuid: when executed, file runs as the user/owner
  - Specifically, the process' effective uid is the owner's
- setgid: same idea, but with gid
  - Except for directories: files created within a setgid directory inherit its group

sticky bit

- Applies to directories only
  - Well, almost
- Users cannot rename/move/delete files owned by other users
  - Even if they have write permission to the directory
  - Doesn’t apply to directory owner
- Why?

Examples

- ls -l, ls -al
- chmod 4700 /usr/bin/vim
  - Or chmod u+s
- chmod 2700 /usr/bin/vim
  - Or chmod g+s
- chmod 1755 /tmp
  - Or chmod +t
- stat

Directories

- Are files that simply contain a (file name, inode number) mapping
- inodes may appear in multiple directories
- Reference count tracks the number of directories in which it appears
- When refcnt == 0, the file is removed
  - Almost. Also cannot have open file descriptors

Links

- Hard
  - Files only (no cycles)
  - Directory entry + duplicate inode
  - $ ln target linkname
- Soft (Symbolic, symlink)
  - File itself (has an inode)
  - Holds a path
  - Cycles permitted
  - Can cross file systems
  - $ ln -s target linkname

Extended attributes

- Extension to the normal attributes associated with every inode in the system
- name:value pairs associated with files
- Eg, setfacl, getfacl
- -rwxr-xr-x+
- setfacl -x to remove
- getfattr

Examples

- setfacl -m u:apache:r /some/path
  getfacl /some/path
  ls -l

Discretionary Access Control

- User dictated
- Eg, classic file permissions
- POSIX Access Control Lists (ACLs)

Mandatory Access Control

- ...or MAC.
- Policy-based access control

SELinux

- Security-Enhanced Linux
- Implements MAC
- Set of kernel modifications and userland tools
  - Originally from the NSA
- Added to mainline kernel as of 2.6
- Originally included in RedHat
  - CentOS and Scientific Linux
  - Fedora *by default*
- Now Debian, Ubuntu, openSUSE, etc optionally

How?

- ls -Z
- chcon
- restorecon
- Etc

  chcon -R -t httpd_user_content_t
  setsebool -P httpd_can_network_connect on
  setsebool -P httpd_can_sendmail on

Sample policy

Questions?