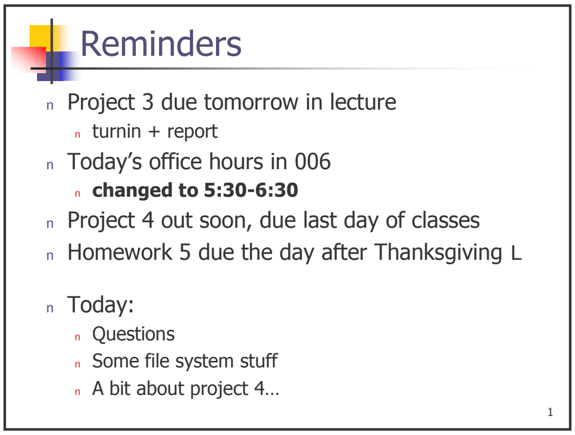
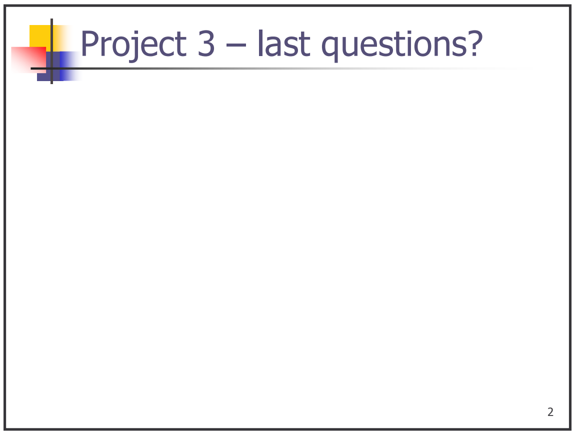
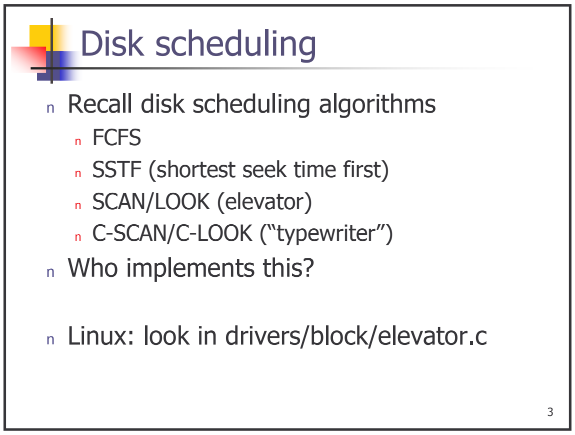


Reminders

- Project 3 due tomorrow in lecture
  - turnin + report
- Today's office hours in 006
  - **changed to 5:30-6:30**
- Project 4 out soon, due last day of classes
- Homework 5 due the day after Thanksgiving L
- Today:
  - Questions
  - Some file system stuff
  - A bit about project 4…

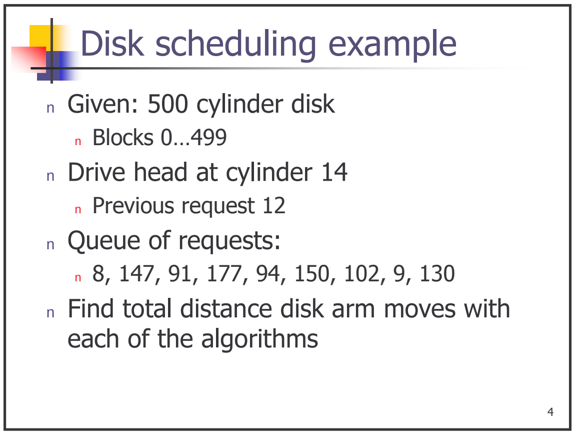


Project 3 – last questions?



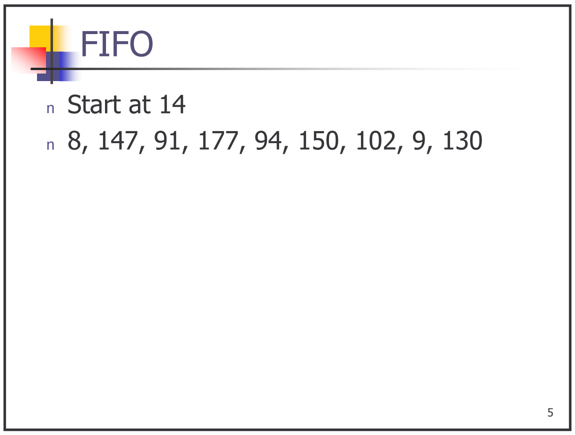
Disk scheduling

- Recall disk scheduling algorithms
  - FCFS
  - SSTF (shortest seek time first)
  - SCAN/LOOK (elevator)
  - C-SCAN/C-LOOK ("typewriter")
- Who implements this?
- Linux: look in drivers/block/elevator.c



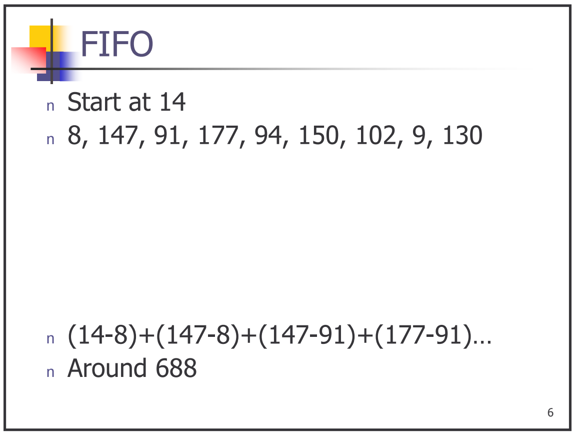
Disk scheduling example

- Given: 500 cylinder disk
  - Blocks 0…499
- Drive head at cylinder 14
  - Previous request 12
- Queue of requests:
  - 8, 147, 91, 177, 94, 150, 102, 9, 130
- Find total distance disk arm moves with each of the algorithms



FIFO

- Start at 14
- 8, 147, 91, 177, 94, 150, 102, 9, 130



FIFO

- Start at 14
- 8, 147, 91, 177, 94, 150, 102, 9, 130
- (14-8)+(147-8)+(147-91)+(177-91)…
- Around 688

Shortest Seek Time First

- Start at 14
- 8, 147, 91, 177, 94, 150, 102, 9, 130

Shortest Seek Time First

- Start at 14
- 8, 147, 91, 177, 94, 150, 102, 9, 130
- Head path:
  - 9, 8, 91, 94, 102, 130, 147, 150, 177
- 5+1+(91-8)+3+8+28+17+3+15+2
- = 165 (<< 688)

C-LOOK

- Start at 14 (was at 12 before 14)
- 8, 147, 91, 177, 94, 150, 102, 9, 130

C-LOOK

- Start at 14 (was at 12 before 14)
- 8, 147, 91, 177, 94, 150, 102, 9, 130
- Head path (it's moving right!):
  - 91, 94, 102, 130, 147, 150, 177, 8, 9
- (91-14)+3+8+28+17+3+27+(177-8)+1 = 333

More disk stuff

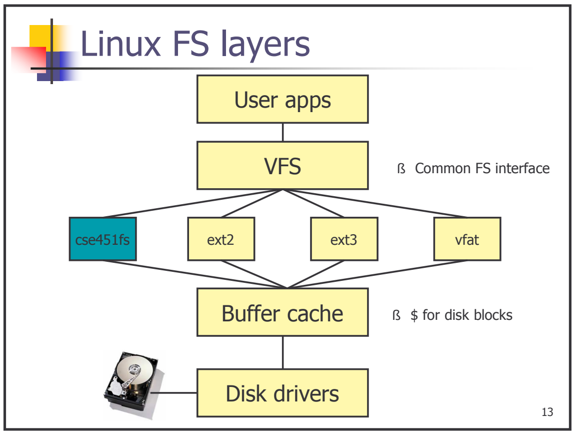
- Why not consider the rotational latency in any of disk scheduling algorithms?
- Why does SSTF favor middle cylinders?
- Cost of typical disk access?
  - Time to transfer a 4K block

300GB Seagate Barracuda 7200.8:

Internal Transfer Rate (Mbits/sec)	760
Max. External Transfer Rate (Mbytes/sec)	100
Avg. Sustained Transfer Rate (Mbytes/sec)	>
Average Seek (msec)	8
Average Latency (msec)	4.16
Multisegmented Cache	8192
Spindle Speed (RPM)	7200

Project 4

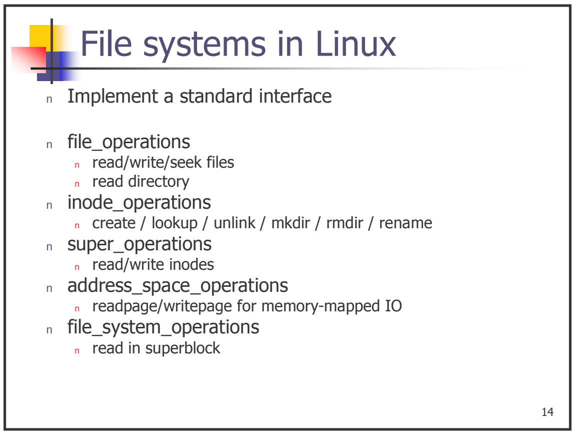
- Work with a real file system
- Given:
  - cse451fs: simplified file system for Linux
- Goals:
  - Understand how it works
  - Modify implementation to:
    - Increase maximum size of files (currently 13KB)
    - Allow for longer file names (currently 30 chars)



Linux FS layers

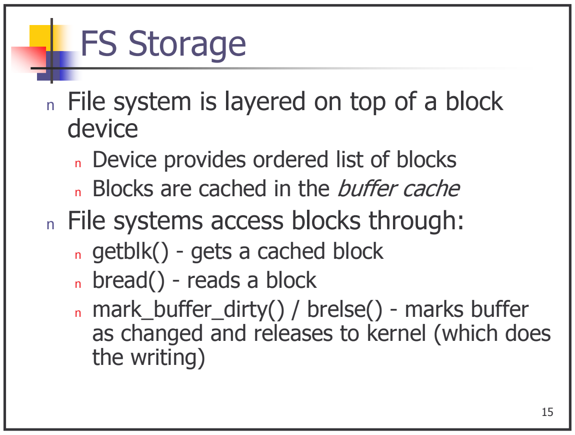
ß Common FS interface

ß $ for disk blocks



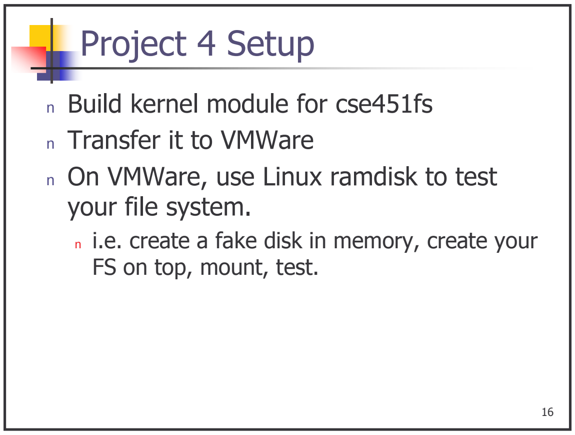
File systems in Linux

- Implement a standard interface
- file_operations
  - read/write/seek files
  - read directory
- inode_operations
  - create / lookup / unlink / mkdir / rmdir / rename
- super_operations
  - read/write inodes
- address_space_operations
  - readpage/writepage for memory-mapped IO
- file_system_operations
  - read in superblock



FS Storage

- File system is layered on top of a block device
  - Device provides ordered list of blocks
  - Blocks are cached in the *buffer cache*
- File systems access blocks through:
  - getblk() - gets a cached block
  - bread() - reads a block
  - mark_buffer_dirty() / brelse() - marks buffer as changed and releases to kernel (which does the writing)



Project 4 Setup

- Build kernel module for cse451fs
- Transfer it to VMWare
- On VMWare, use Linux ramdisk to test your file system.
  - i.e. create a fake disk in memory, create your FS on top, mount, test.