CS 471 Operating Systems

Yue Cheng

George Mason University
Fall 2017
Review: Disks
Device I/O Protocol Variants

- Status checks
  - Polling
  - Interrupts

- Data
  - PIO
  - DMA
Disks

- Doing an disk I/O requires:
  - Seek
  - Rotate
  - Transfer

- What is expensive?
Give a stream of I/O requests, in what order should they be served?
Disk Scheduling
Disk Scheduling

- OS is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth

- **Strategy:** reorder requests to meet some goal
  - Performance (e.g., by making I/O sequential)
  - Fairness
  - Consistent latency

- Usually implemented in both OS and hardware
Disk Scheduling

- Performance objective: minimize seek+rotation time
  - Minimize the distance the head needs to go

- Disk bandwidth:
  - The total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer
Disk Scheduling

- There are many sources of disk I/O requests:
  - OS
  - System processes
  - User processes

- I/O request:
  - Read/write, disk address, memory address, number of sectors to transfer

- OS maintains queue of requests, per disk or device

- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimization algorithms make sense only when a queue exists
Disk Scheduling

- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)

- Disk scheduling algorithms:
  - Algorithms that schedule the orders of disk I/O requests
Disk Scheduling

- Note that **drive controllers have small buffers** and can manage a queue of I/O requests (of varying “depth”)

- Disk scheduling algorithms:
  - Algorithms that schedule the orders of disk I/O requests

- The analysis is true for one or many platters

- We illustrate scheduling algorithms with an example request queue (0-199)

  98, 183, 37, 122, 14, 124, 65, 67

Initially, head pointer pointing to 53
FIFO

- Idea: Serve the I/O request in the order they arrive
**FIFO**

- Idea: Serve the I/O request in the order they arrive

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
FIFO

- Idea: Serve the I/O request in the order they arrive

Illustration shows total head movement of 640 cylinders
Shortest Seek Time First (SSTF)

- Idea: Selects the request with the minimum seek time from the current head position
Shortest Seek Time First (SSTF)

- Idea: Selects the request with the minimum seek time from the current head position

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

Illustration shows total head movement of **236** cylinders. SSTF scheduling is a form of SJF scheduling: may cause starvation of some requests!
SCAN

- Idea: Sweep back and forth, from one end of disk to the other, serving requests as you go
  - The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues
  - AKA Elevator Algorithm
Illustration shows total head movement of 208 cylinders.

**Issue:** Cylinders in the middle get better service; Requests at the other end wait the longest!
C-SCAN (Circular-SCAN)

- Idea: Only sweep in **ONE** direction
  - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Provides a more uniform wait time than SCAN
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
C-SCAN

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

Total number of cylinders?
C-LOOK

- Idea: Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk
  - LOOK: A version of SCAN
  - C-LOOK: A version of C-SCAN
C-LOOK

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

Total number of cylinders?
Work Conservation

- **Work conserving** schedulers always try to do I/O if there’s I/O to be done.

- Sometimes, it’s better to wait instead if you anticipate another request will appear nearby.

- Such non-work-conserving schedulers are called anticipatory schedulers.
CFP (Linux Default)

- Completely Fair Queueing
- Queue for each process
- Do weighted round-robin among queues, with slice time proportional to priority
- Optimize order within queue
- Yield slice only if idle for a given time (anticipation)
Selecting A Disk Scheduling Algorithm

- SSTF is common and has a natural appeal
  - Starvation
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - Less starvation
- Performance depends on the workload (i.e., number and types of requests)
- Requests for disk service can be influenced by the file-allocation method
  - And metadata layout
- The disk-scheduling algorithm should be written as a separate OS module, allowing it to be replaced with a different algorithm if necessary