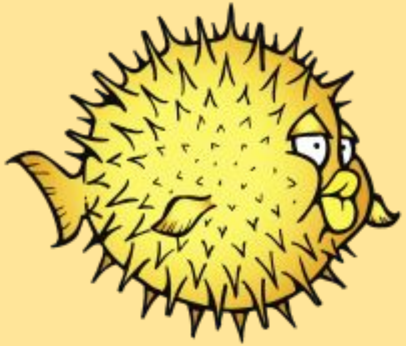


Disk Scheduling

COMP3301 - Week 8 Applied Class

OpenBSD

COMP3301 - 2025

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

A2 Help - DMA Allocation

- General flow of logic

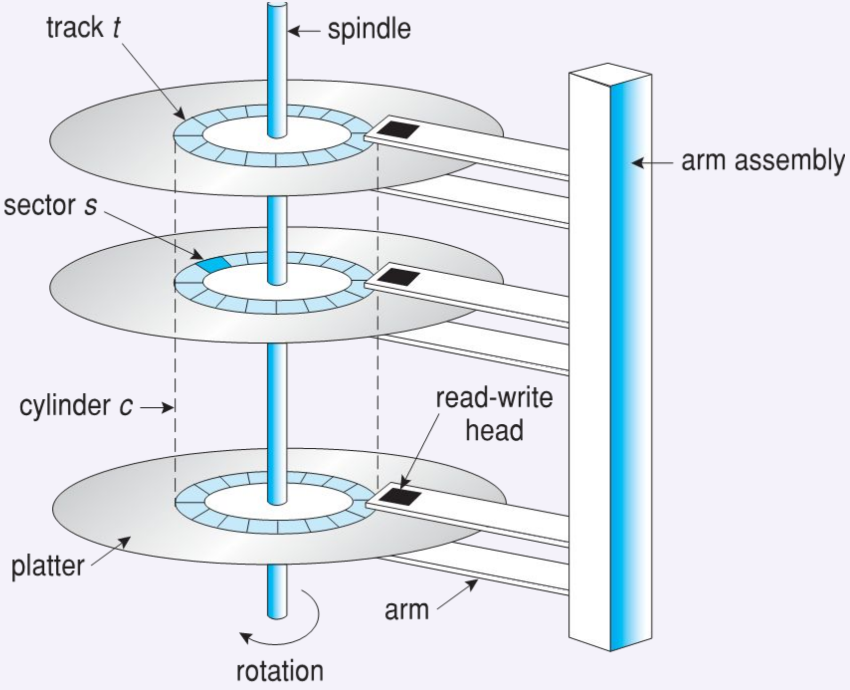
```
bus_dmamap_create();            // Create DMA map.
  bus_dmamem_alloc();           // Allocate memory to use for DMA.
    bus_dmamem_map();           // Map allocated memory into kernel address space.
      // Fill out DMA memory (setup the rings or use uiomove etc).
      bus_dmamap_load();        // Associate the memory with our DMA map.
        bus_dmamap_sync();      // Flush any DMA caches.
        // Start DMA (send START command or write to doorbell etc).
        bus_dmamap_sync();      // Flush any DMA caches.
      bus_dmamap_unload();      // Unassociate the memory with our DMA map.
    bus_dmamem_unmap();         // Unmap the allocated memory from kernel AS.
  bus_dmamem_free();            // Free the allocated memory.
bus_dmamap_destroy();           // Destroy the DMA map.
```

A2 Help - DMA Allocation

- You can load an UIO to a DMA map and do DMA with the memory directly
  - Better performance, less steps
  - bus_dmamap_create, bus_dmamap_load_uio, bus_dmamap_unload, bus_dmamap_destroy
  - No need to allocate new memory and copy contents over
  - But won’t work if the UIO is too fragmented to fit in 4 DMA pointers (the UIO must be 4 or less contiguous blocks of physical memory)
  - Also won’t work for non-blocking mode, because it does the DMA directly on userland memory
    - But in non-blocking mode, it returns immediately and user could be doing something else with that memory
  - So up to you if you want to use it
  - Resorting to bounce buffer for everything is fine, it makes logic simpler

Disk Scheduling

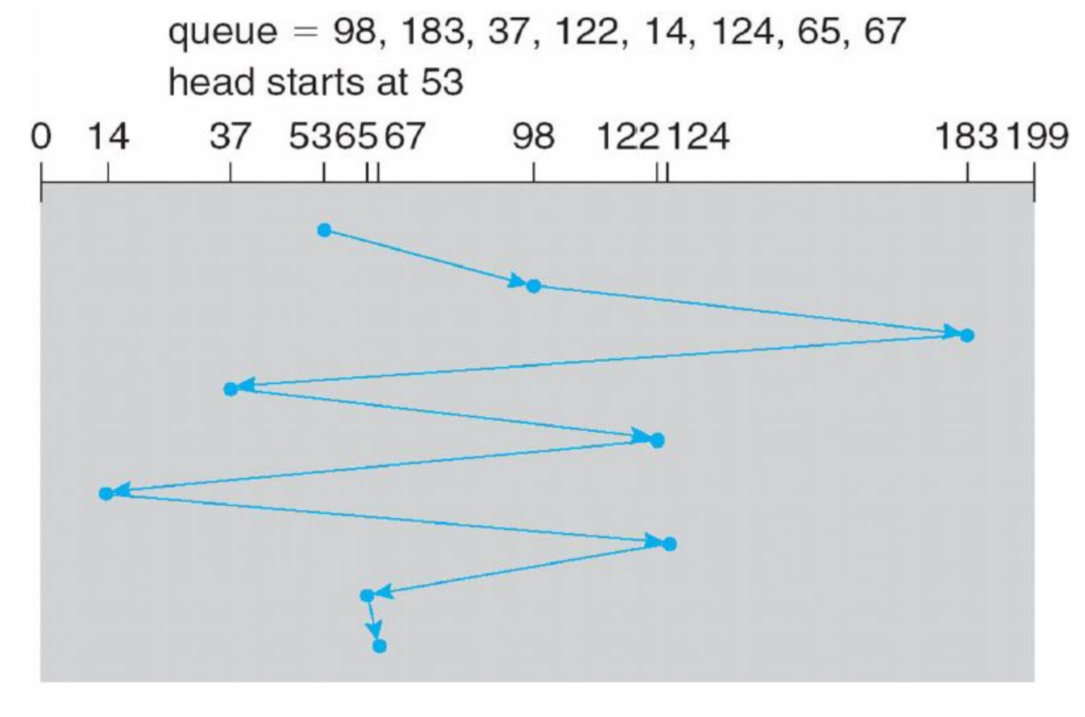
-



FCFS (First Come First Serve)

- Head moves

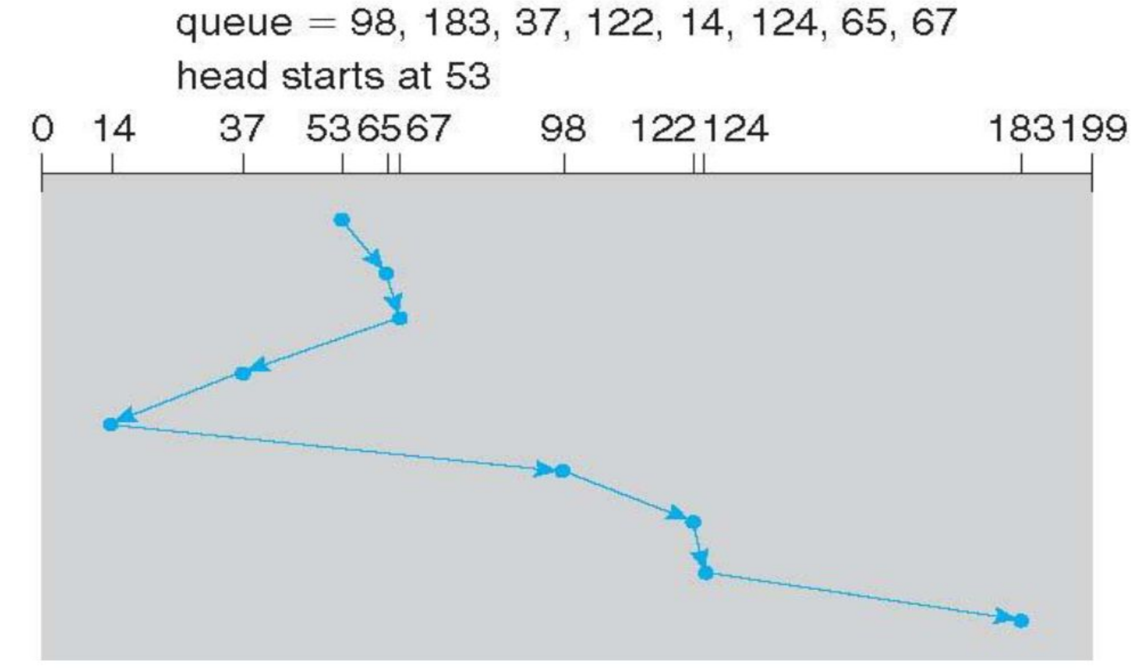
  640 cylinders



SSTF (Shortest Seek Time First)

- Head moves

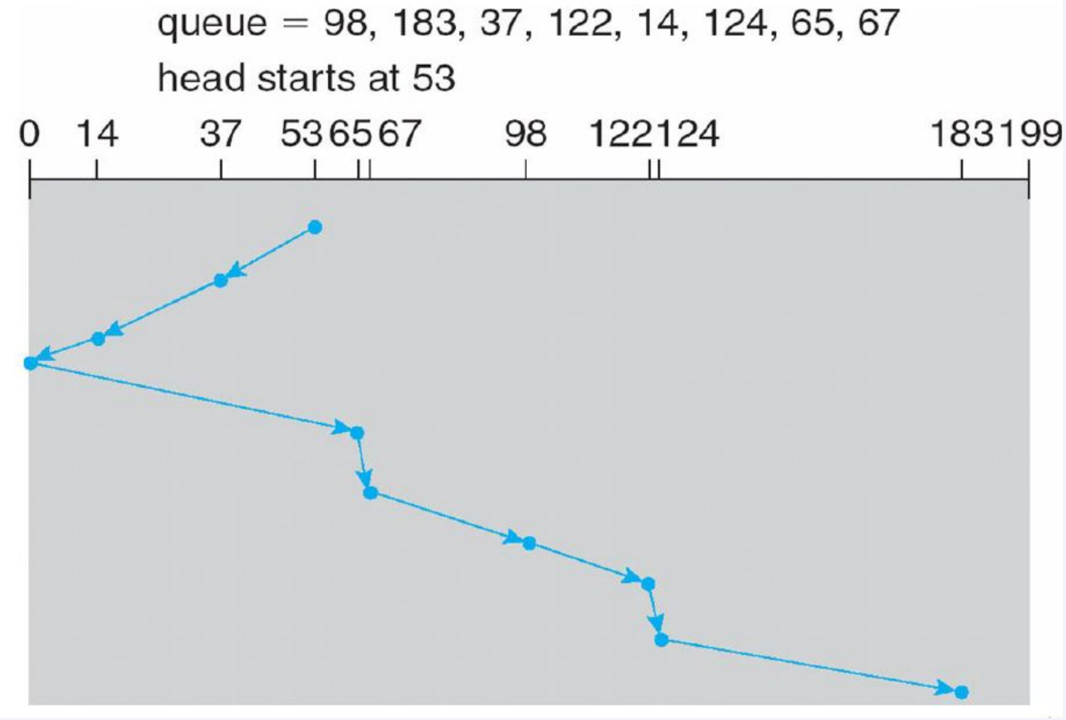
  236 cylinders



SCAN

- Head moves

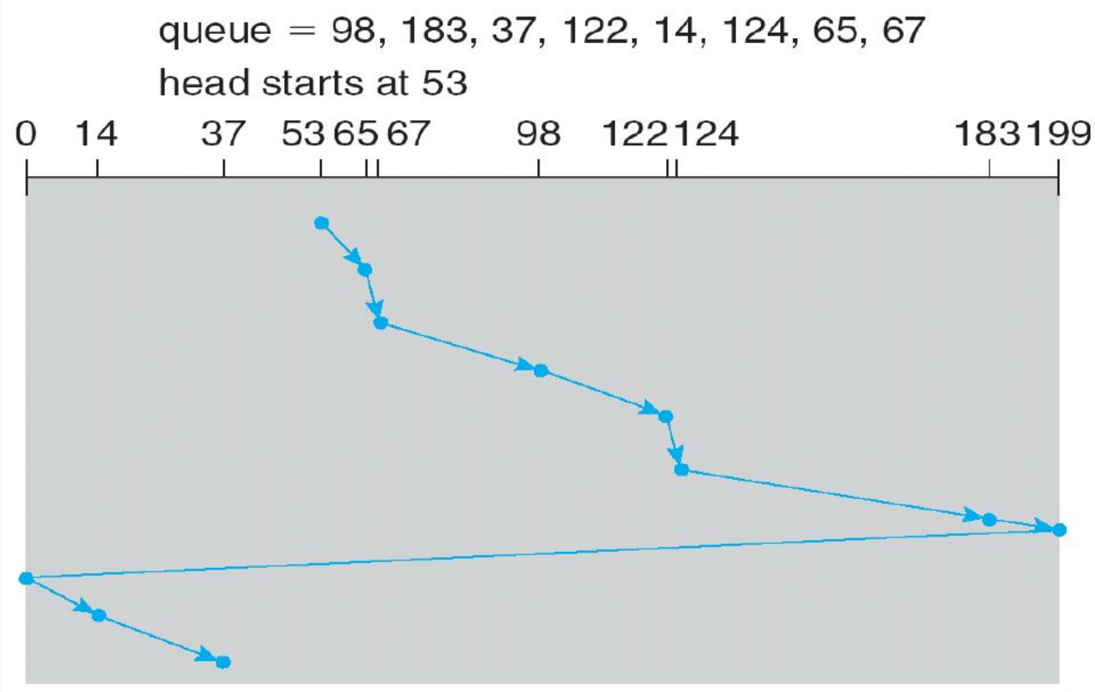
  208 cylinders



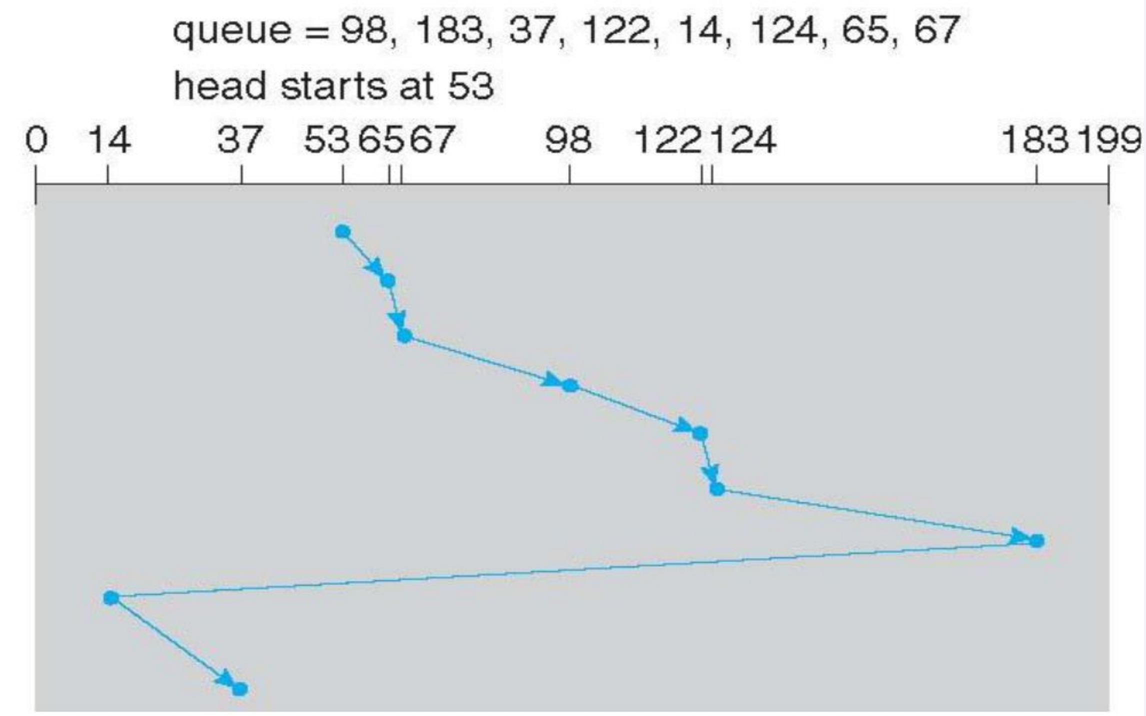
C-SCAN

Why?

- Provides more uniform wait time



C-LOOK



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

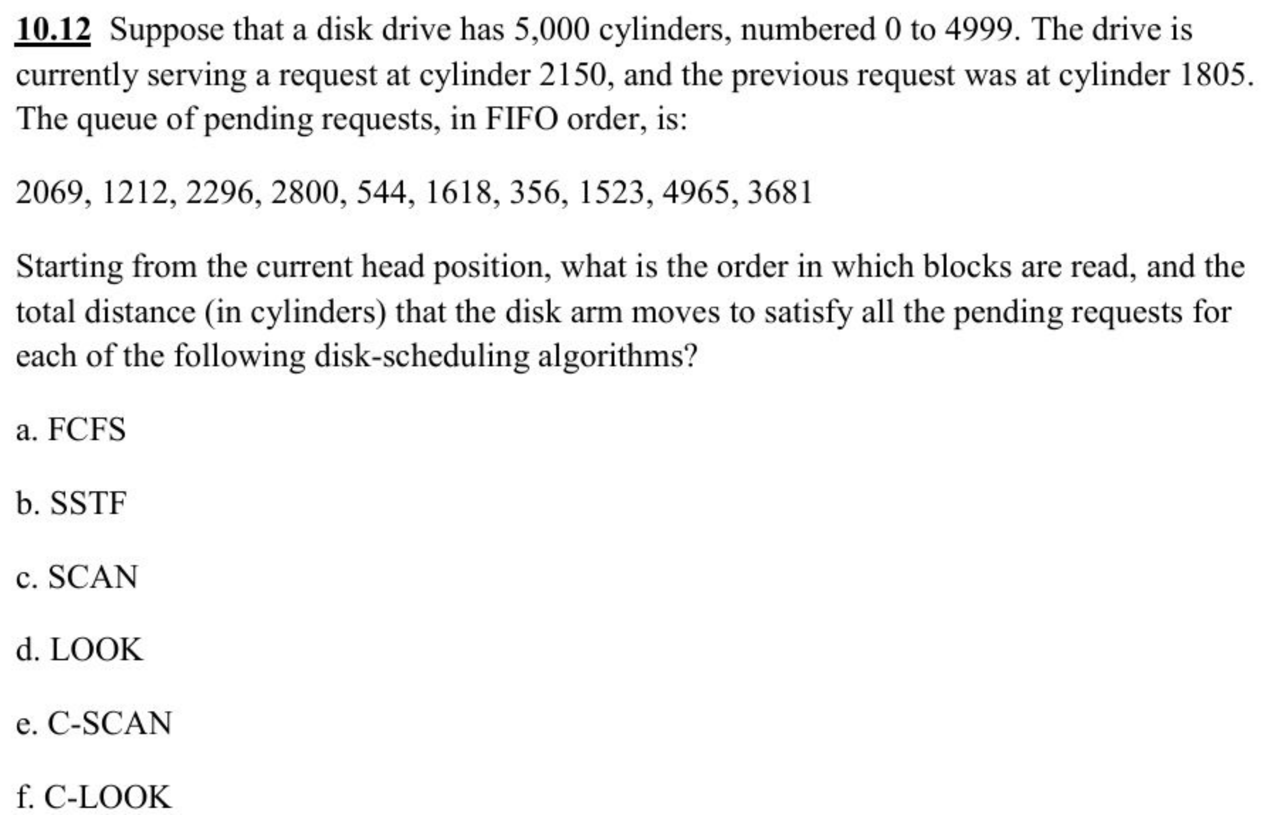
head starts at 53

Disk Scheduling

Are these algorithms still useful today?

- Yes, but slightly less relevant compared to when they were invented
- SSDs
- OS no longer have control over where the head is
    - Zoned bit recording
    - Sector remapping
    - Controlled by the firmware, OS don’t know exactly where head is
- But they still help!

Exercise



10.12 Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is:

2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

Starting from the current head position, what is the order in which blocks are read, and the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

a. FCFS

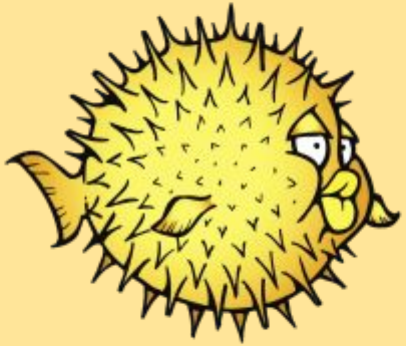
b. SSTF

c. SCAN

d. LOOK

e. C-SCAN

f. C-LOOK



Happy devving!

Thanks for coming.

OpenBSD

COMP3301 - 2025

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA