ISTANBUL KEMERBURGAZ UNIVERSITESI

Solaris I/O Management and Disk Scheduling

NAME: AMNA ABURAS

ID :153103037

Department : Electrical and Computer Engineering

Prof Dr. Hasan Balik

Solaris I/O Management and Disk Scheduling

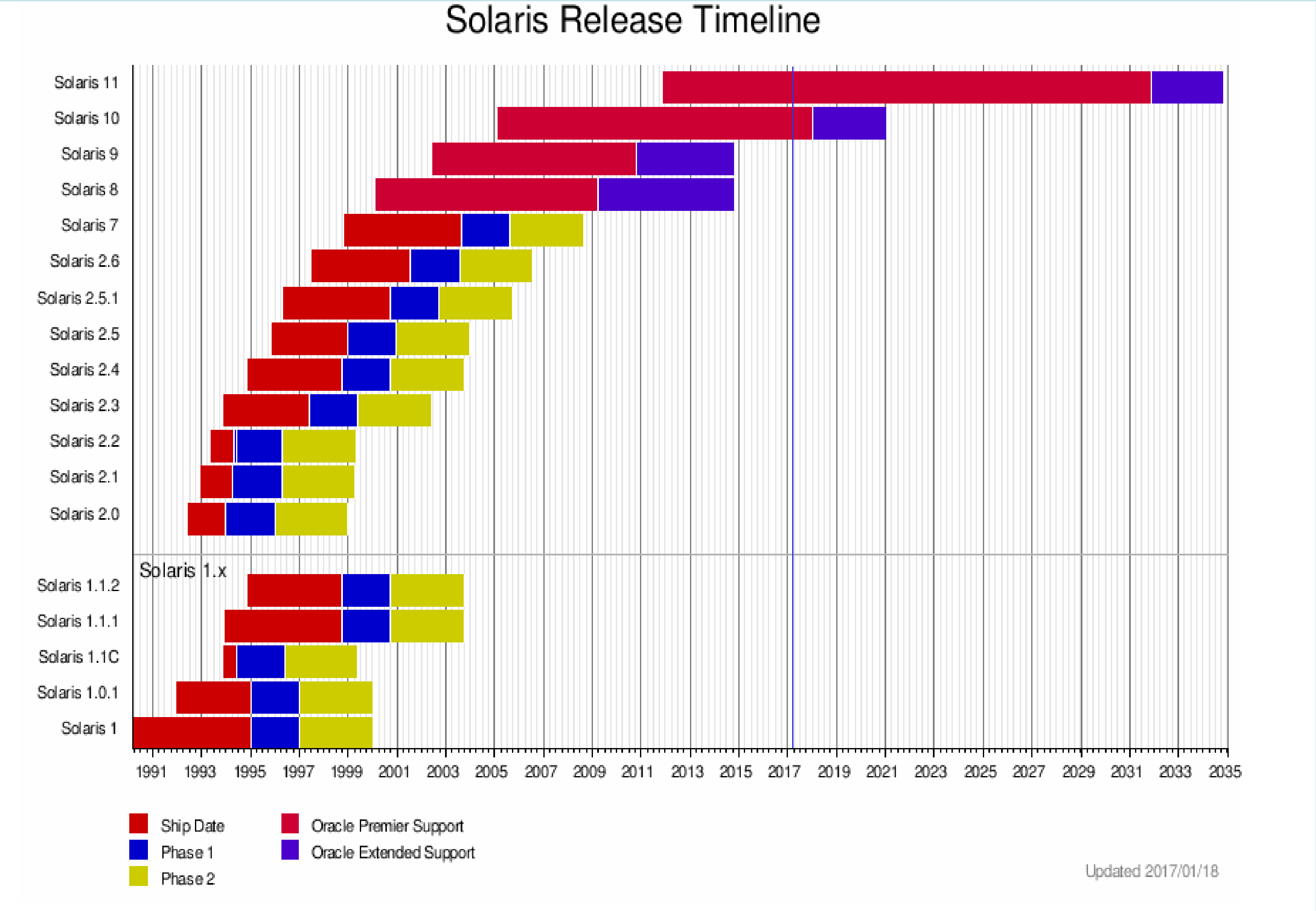
General Categories :

The I/O system consists of:
1-A buffer-caching system
2-A general device-driver interface
3- Drivers for specific hardware devices

Solaris

- Solaris is a Unix operating system originally developed by Sun Microsystems. It superseded their earlier SunOS in 1993. Oracle Solaris, so named as of 2010, has been owned by Oracle Corporation since the Sun acquisition by Oracle in January 2010.

- Solaris is known for its scalability, especially on SPARC systems, and for originating many innovative features such as DTrace, ZFS and Time Slider. Solaris supports SPARC-based and x86-based workstations and servers from Oracle and other vendors, with efforts underway to port to additional platforms. Solaris is registered as compliant with the Single Unix Specification .



Solaris Release Timeline
Solaris 11
Solaris 10
Solaris 9
Solaris 8
Solaris 7
Solaris 2.6
Solaris 2.5.1
Solaris 2.5
Solaris 2.4
Solaris 2.3
Solaris 2.2
Solaris 2.1
Solaris 2.0
Solaris 1.x
Solaris 1.1.2
Solaris 1.1.1
Solaris 1.1C
Solaris 1.0.1
Solaris 1
1991 1993 1995 1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2017 2019 2021 2023 2025 2027 2029 2031 2033 2035
Ship Date
Phase 1
Phase 2
Oracle Premier Support
Oracle Extended Support
Updated 2017/01/18

Why Study Operating Systems?

- Understand the relationship between hardware & software
- Understand future Operating Systems
- Understand powerful tools to help you in your computer use

Categories of I/O Devices

- Machine readable
  - Used to communicate with electronic equipment
  - Disk drives
  - USB keys
  - Sensors
  - Controllers
  - Actuators

Relationship Among Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Evolution of the I/O Function

1. Processor directly controls a peripheral device .
2. A controller or I/O module is added .
3. Same configuration as step 2, but now interrupts are employed .
4. The I/O module is given direct control of memory via DMA .
5. The I/O module is enhanced to become a separate processor, with a specialized instruction set tailored for I/O .
6. The I/O module has a local memory of its own and is, in fact, a computer in its own right .

Direct Memory Address

- Processor delegates I/O operation to the DMA module
- DMA module transfers data directly to or from memory
- When transfer is complete, DMA module sends an interrupt signal to the processor

DMA

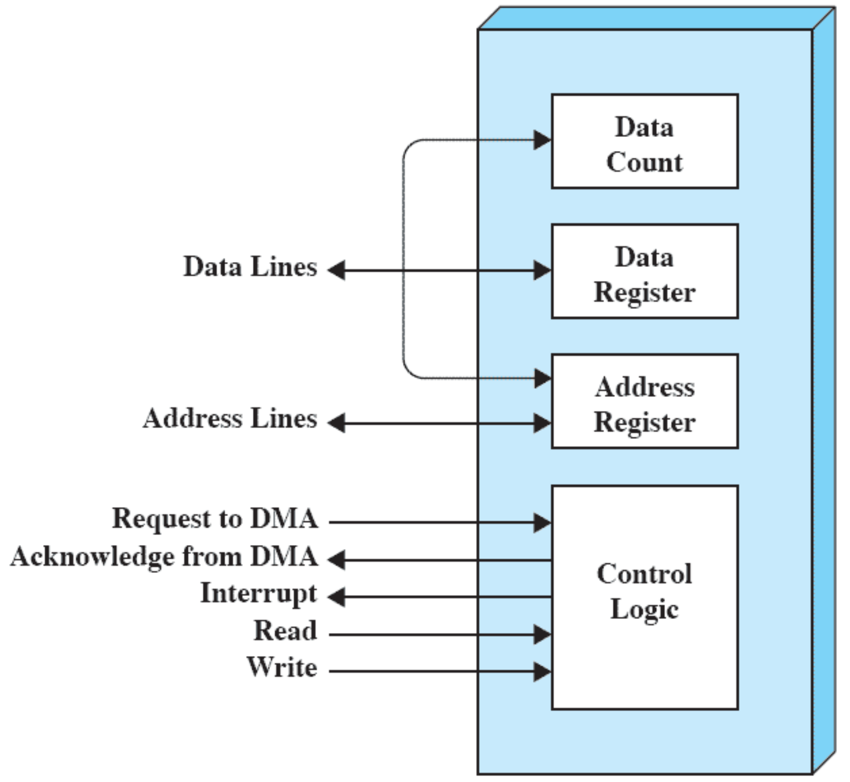
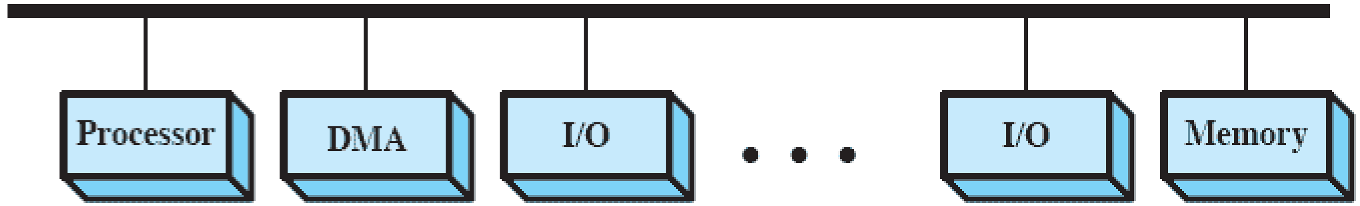


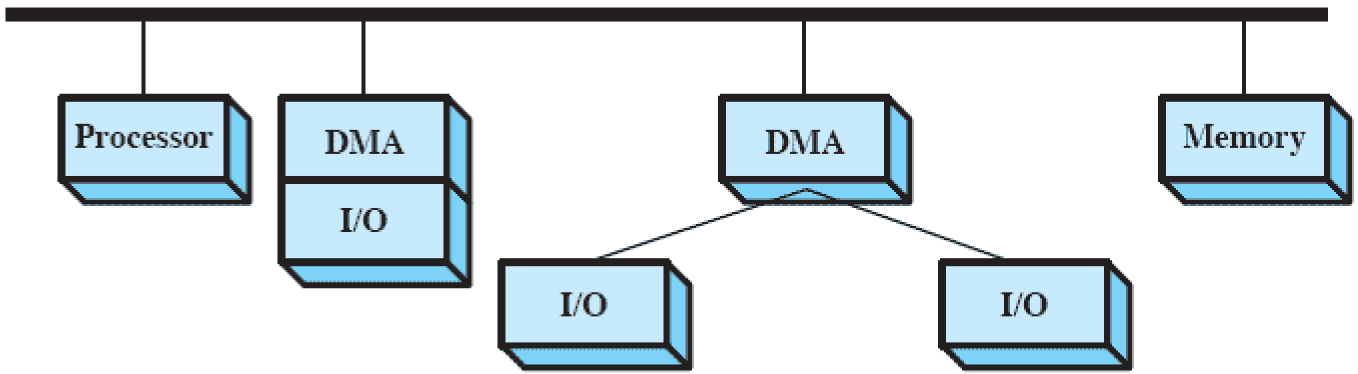
Figure 11.2 Typical DMA Block Diagram

DMA Configurations



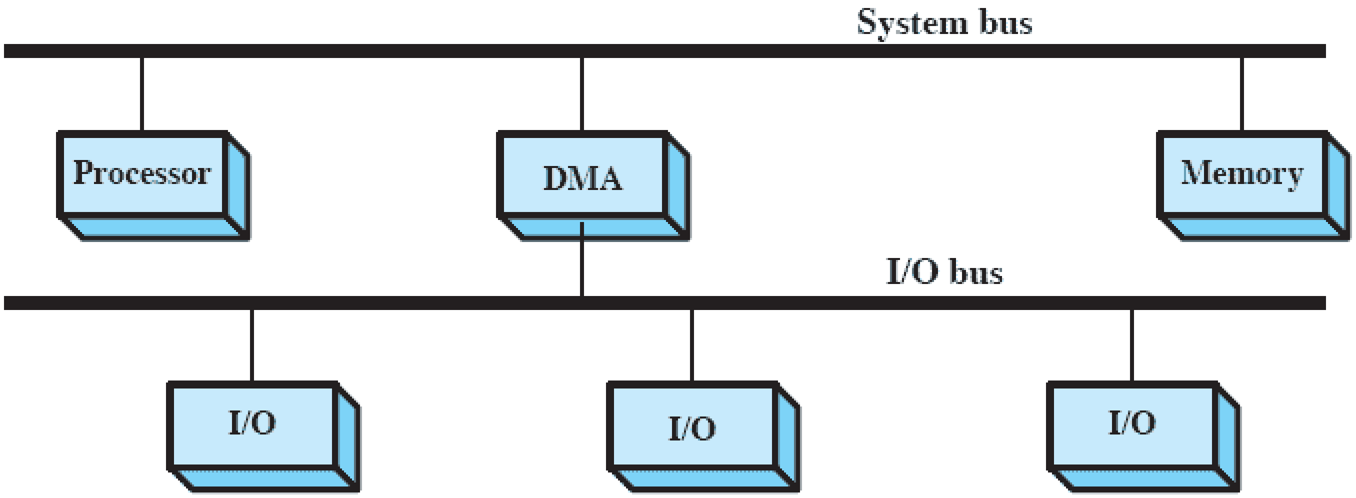
(a) Single-bus, detached DMA

DMA Configurations



(b) Single-bus, Integrated DMA-I/O

DMA Configurations

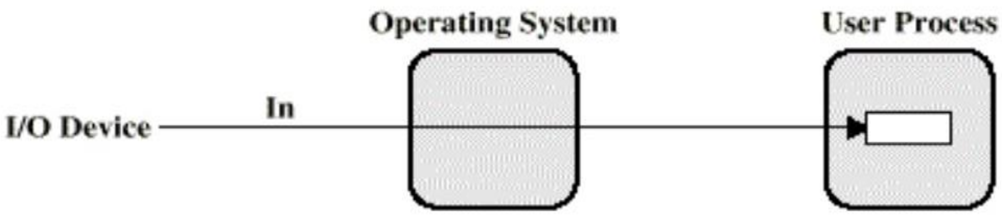


(c) I/O bus

I/O Buffering

- Block-oriented
  - Information is stored in fixed sized blocks
  - Transfers are made a block at a time
  - Used for disks and USB keys
- Stream-oriented
  - Transfer information as a stream of bytes
  - Used for terminals, printers, communication ports, mouse and other pointing devices, and most other devices that are not secondary storage

NO BUFFERING



(a) No buffering

- **Process must read/write a device a byte/word a time.**
- **Each individual system call adds significant overhead.**
- **Process must what until each I/O is complete.**

  i) Blocking/waking adds to overhead

  ii) Many short run of a process is inefficient.
- **What happens if buffer is paged out to disk**

  i) Could lose data while buffer is paged in

  ii) Could lock buffer in memory (needed for DMA), however many processes doing I/O reduce RAM available for paging.

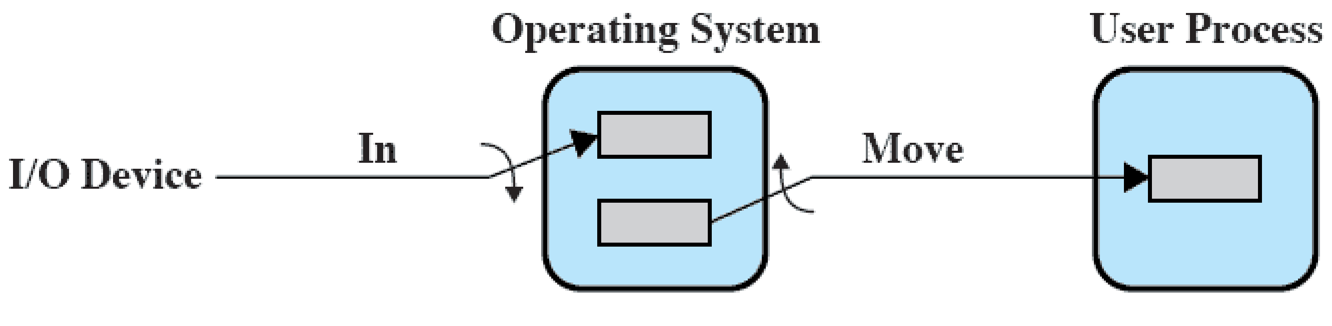
  iii) Can cause deadlock as RAM is limited

USER LEVEL BUFFERING

- User process can process one block of data while next block is read in.
- Swapping can occur since input is taking place in system memory, not user memory.
- Operating system keeps track of assignment of system buffers to user processes.
- Assume:
  - T is transfer time for a block from device
  - C is computation time to process incoming block
  - M is time to copy kernel buffer to user buffer
- Computation and transfer can be done in parallel

Double Buffer

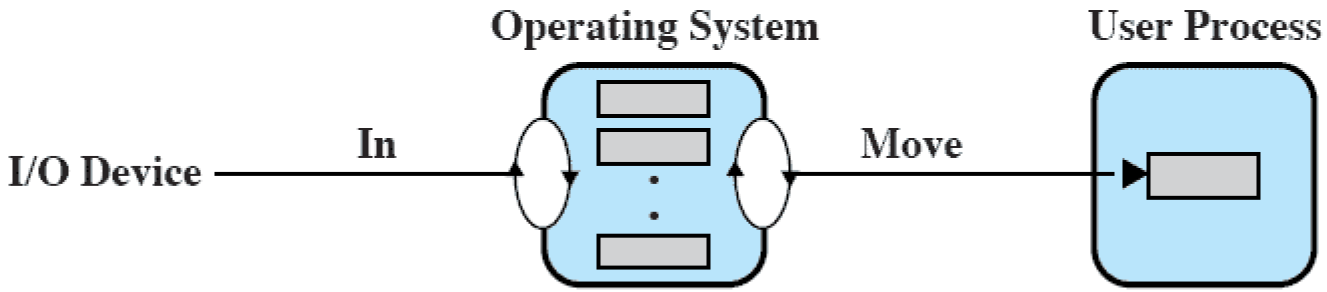
- Use two system buffers instead of one
- A process can transfer data to or from one buffer while the operating system empties or fills the other buffer



(c) Double buffering

Circular Buffer

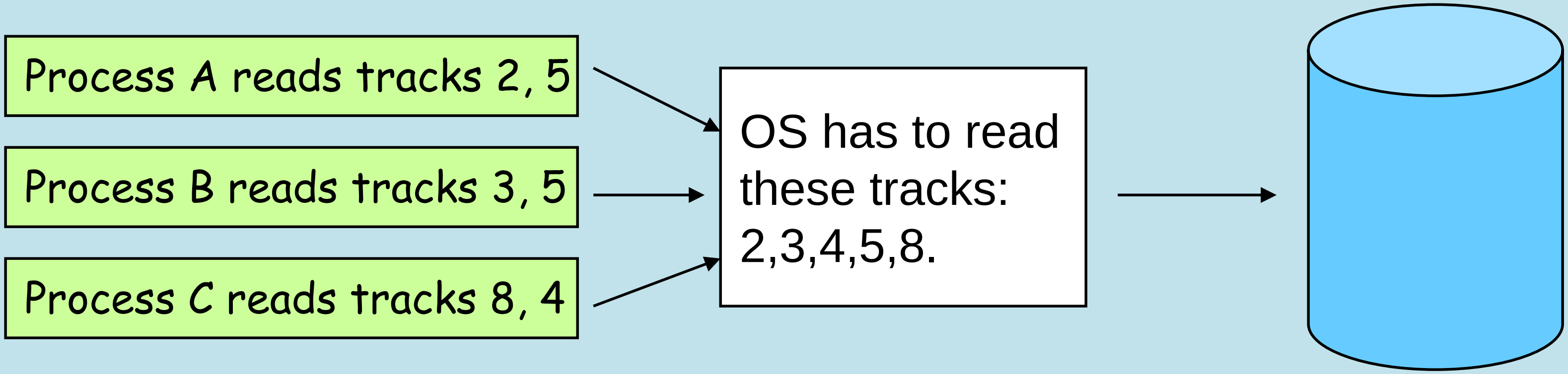
- More than two buffers are used
- Each individual buffer is one unit in a circular buffer
- Used when I/O operation must keep up with process



(d) Circular buffering

Disk Scheduling

- At runtime, I/O requests for disk tracks come from the processes
- OS has to choose an order to serve the requests

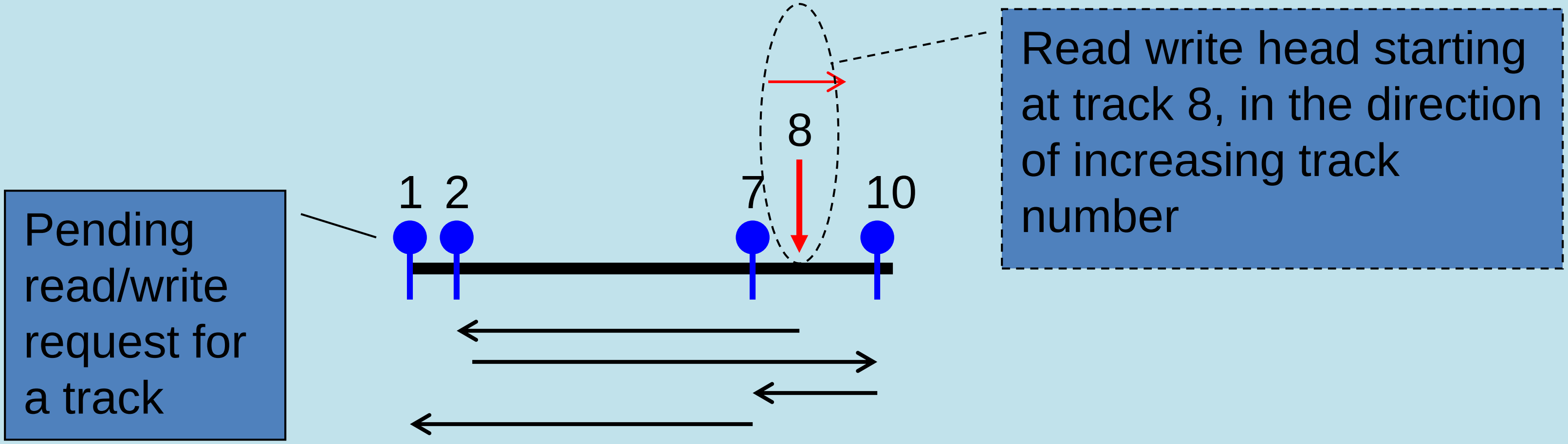


Disk Scheduling Policy

- FIFO: fair, but near random scheduling
- SSTF: possible starvation
- SCAN: favor requests for tracks near the ends
- C-SCAN
- FSCAN: avoid “arm stickiness” in SSTF, SCAN and C-SCAN

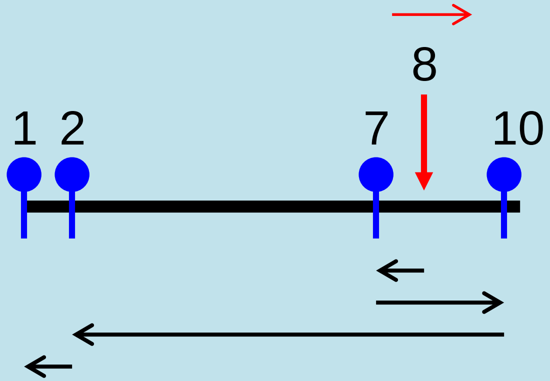
First-in-first-out, FIFO

- process requests in the order that the requests are made
- fair to all processes
- approaches random scheduling in performance if there are many processes



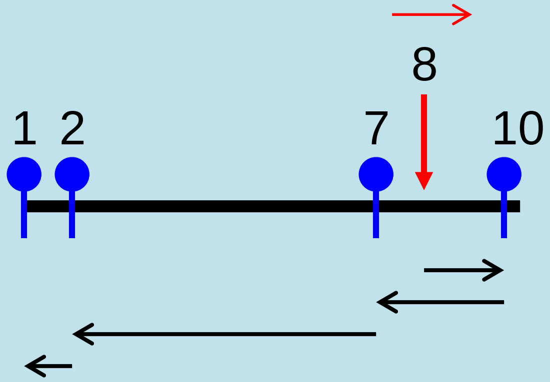
Shortest Service Time First, SSTF

- select the disk I/O request that requires the least movement of the disk arm from its current position
- always choose the minimum seek time
- new requests may be chosen before an existing request



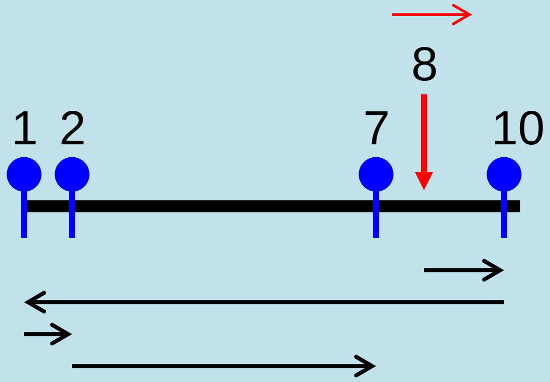
SCAN

- arm moves in one direction only, satisfying all outstanding requests until there is no more requests in that direction. The service direction is then reversed.
- favor requests for tracks near the ends



C-SCAN

- restrict scanning to one direction only
- when the last track has been visited in one direction, the arm is returned to the opposite end of the disk and the scan begins again.



FSCAN

- “Arm stickiness” in SSTF, SCAN, C-SCAN in case of repetitive requests to one track
- FSCAN uses two queues. When a SCAN begins, all of the requests are in one of the queues, with the other empty. During the scan, all new requests are put into the other queue.
- Service of new requests is deferred until all of the old requests have been processed.

Disk Scheduling Algorithms

(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	**Average seek length**	27.5	**Average seek length**	27.8	**Average seek length**	35.8

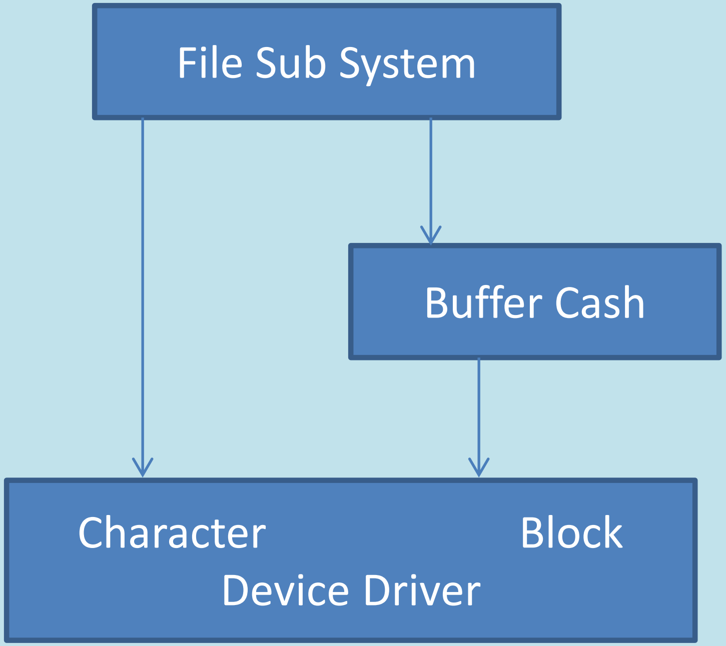
UNIX SVR4 I/O

Devices are Files

- Each I/O device is associated with a special file
  - Managed by the file system
  - Provides a clean uniform interface to users and processes.
- To access a device, read and write requests are made for the special file associated with the device.

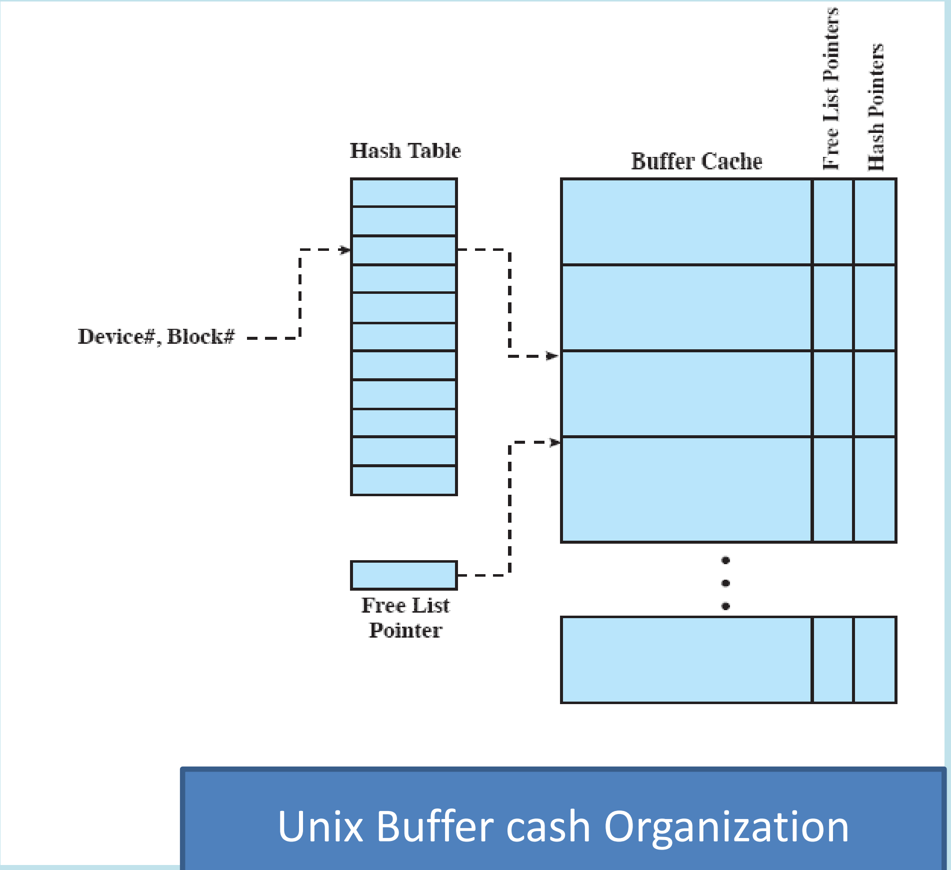
UNIX SVR4 I/O

- Each individual device is associated with a special file
- Two types of I/O
  - Buffered
  - Unbuffered



Buffer Cache

- Three lists are maintained
  - Free List
  - Device List
  - Driver I/O Queue



Unix Buffer cash Organization

Character Cache

- Used by character oriented devices
  - E.g. terminals and printers
- Either written by the I/O device and read by the process or vice versa
  - Producer/consumer model used

Unbuffered I/O

- Unbuffered I/O is simply DMA between device and process
  - Fastest method
  - Process is locked in main memory and can't be swapped out
  - Device is tied to process and unavailable for other processes

I/O for Device Types

Unix

Device I/O in UNIX

	Unbuffered I/O	Buffer Cache	Character Queue
Disk drive	X	X	
Tape drive	X	X	
Terminals			X
Communication lines			X
Printers	X		X

Thank you for my listening