Operating System

Lecture 12:

Input/Output

1

Today

- I/O Layering
- I/O Devices
- Device Drivers
- Device Controllers
- I/O Software

2

I/O Layering

- I/O Layers
  - User processes
  - Device-independent software
  - Device drivers
  - Device Controllers
  - Devices (HW)

I/O request

I/O reply

3

I/O Devices

- A computer system is composed of components
  - CPU, memory, I/O devices
  - Communication is over system bus
    - can be more than one

4

Device Controllers

- An I/O device is made up of
  - mechanical components (if any)
  - electronic components
    - adapter or controller
- OS interacts with the controller using registers
- Registers accessed via memory (memory-mapped) or I/O instructions

5

Device Drivers

- Encapsulates device-dependent code
- Generally, must implement a standard interface
- Code contains device-specific register for reads/writes

6

Device-Independent OS Software

- Kernel I/O subsystem
- Functionality that is general and independent of any specific I/O device

7

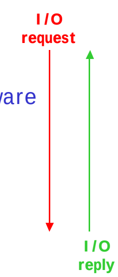
User-Level Software

- User-level buffering
  - e.g., stdio
- Spooling daemons
  - printer
  - network

8

I/O Layering

- I/O Layers
  - User processes
  - Device-independent software
  - Device drivers
  - Device Controllers
  - Devices (HW)



9

I/O Devices

- Block device
  - accesses information in addressable fixed size blocks
  - example
    - disk

10

I/O Devices

- Character device
  - accesses information in terms of streams of characters
  - examples
    - network, terminals, printers, mice

11

I/O Devices

- Other
  - memory-mapped
  - examples
    - graphics, video

12

I/O Devices

- Characteristics of I/O devices
  - character-stream or block
  - sequential or random-access
  - synchronous or asynchronous
  - sharable or dedicated
  - speed of operation
  - read-write, read only, or write only

13

Device Controllers

- Interrupts
- Direct Memory Access (DMA)

14

Interrupts

- When an I/O device has finished the work
  - it causes an interrupt
  - by asserting a signal on a bus line that is has been assigned to
  - this signal is detected by the interrupt controller, which decides what to do

15

Interrupts

- The interrupt controller
  - processes the interrupt immediately (if no other interrupts are pending)
  - to handle the interrupt, the controller
    - puts a number on the address lines
    - specifying which device wants attention
    - and asserts a signal that interrupts the CPU

16

Interrupts

- The interrupt signal causes the CPU to stop what it is doing
  - the number on the address lines is used as an index into a table called interrupt vector to bring…
  - … a new program counter, that gives the appropriate interrupt service procedure (interrupt handler)

17

Interrupts

- The interrupt service procedure
  - handles the interrupt
  - acknowledges the interrupt by writing a certain value to one of the interrupt controller's I/O ports
  - the I/O device is now free to generate a new interrupt

18

Direct Memory Access (DMA)

- Programmed I/O
  - CPU moves data word-by-word between device and memory
  - if device is slow, can be inefficient

19

Direct Memory Access (DMA)

- DMA
  - CPU sets up transfer of data between device and memory
  - CPU can do other work while transfer occurs
  - interrupt occurs when DMA transfer completes

20

Example: DMA from Disk

- Read from disk
  - first transfer to disk's controller buffer
  - Then DMA to memory

21

Kernel I/O Subsystem

- Concepts
  - Device independence
  - Uniform naming
  - Synchronous and Asynchronous
    - **blocking (S)**
    - **interrupt-driven (A)**
  - Shared and dedicated

22

Kernel I/O Subsystem

- Supervises
  - The management of the name space for files and devices
  - Access control to files and devices
  - Operation control
    - **e.g., a modem cannot seek**
  - File system space allocation
  - Device allocation

23

Kernel I/O Subsystem

- Also
  - Buffering, caching, spooling, and device reservation
  - I/O scheduling
  - Device status monitoring, error handling, and failure recovery
  - Device driver configuration and initialization

24

Disks

- Disk Interleaving
- Disk Scheduling

25

Disks

- Performance
  - Seek Time
  - Rotational latency
- Improve: access time and bandwidth
- Bandwidth = number of bytes transferred divided by total time between 1st and last request

26

Disk Interleaving

- A sector is read and it must be transferred to memory
- Disk head may pass beginning of next sector by the time transfer is done
- To improve performance, interleave sectors
  - single interleaving
  - double interleaving

27

Disk Scheduling

- I/O request
  - input or output?
  - disk address
  - memory address for transfer
  - number of bytes

28

Disk Scheduling

- If device driver and controller available
  - serve immediately
- else
  - request is placed in a queue

29

Disk Scheduling

- Scheduling means choosing from the pending I/O requests, which is going to be the next

30

Scheduling Algorithms

- FCFS (First Come First Served)
  - fair but not necessarily the fastest service
  - example:
    - **98-183-37-122-14-124-65-67**
    - **head starts at 53, will move 640 cylinders**
    - **problem: big swing! 122 -> 14 -> 124**

31

Scheduling Algorithms

- SSTF (Shortest Seek Time First)
  - can cause starvation
  - knowing the future would be better
  - not optimum
  - example:
    - **98-183-37-122-14-124-65-67**
    - **head starts at 53, will move 236 cylinders**

32

Scheduling Algorithms

- SCAN
  - elevator algorithm
  - moves all the way back and forth
  - example:
    - **98-183-37-122-14-124-65-67**
    - **head starts at 53, going downwards**
    - **head will move 236**

33

Scheduling Algorithms

- C-SCAN (Circular SCAN)
  - elevator algorithm
  - moves all the way back and forth
  - serves in one direction only
  - example:
    - **98-183-37-122-14-124-65-67**
    - **head starts at 53 going upwards**
    - **head will move 183 cylinders**

34

Scheduling Algorithms

- LOOK
  - elevator algorithm
  - moves back and forth
  - serves in both directions
  - example for LOOK:
    - **98-183-37-122-14-124-65-67**
    - **head starts at 53 going upwards**

35

Scheduling Algorithms

- C-LOOK
  - elevator algorithm
  - moves back and forth
  - Serves in one direction only
  - example for C-LOOK:
    - **98-183-37-122-14-124-65-67**
    - **head starts at 53 going upwards**
    - **head will move 153 cylinders**

36

Scheduling Algorithms

- Optimal will vary according to sequence of requests, but costs to determine optimal does not compensate for not using either SSTF or (C-)SCAN or (C-)LOOK

37

Scheduling Algorithms

- File system organization will influence the requests for disk service
  - reading a contiguous file: requests are close together
  - reading a linked or indexed file: requests may be scattered

38

Disk Scheduling

- Disk scheduling in modern systems can be implemented by the controller
- This is good for performance because disk has control over seek and rotational speed

39

Disk Scheduling

- However, it is interesting for the OS to have control because of
  - priorities
    - **demand paging over application I/O**
    - **writes over reads**
  - I/O ordering
    - **updating FS tables before starting writing a new file**

40

Performance

- Put directory info in the middle of the disk
- Cache directories and index blocks
- Best algorithm depends on FS implementation
- Good defaults
  - SSTF or (C-)LOOK or (C-)SCAN

41