I/O Devices and Operations 1

Minsoo Ryu

Real-Time Computing and Communications Lab.
Hanyang University

msryu@hanyang.ac.kr

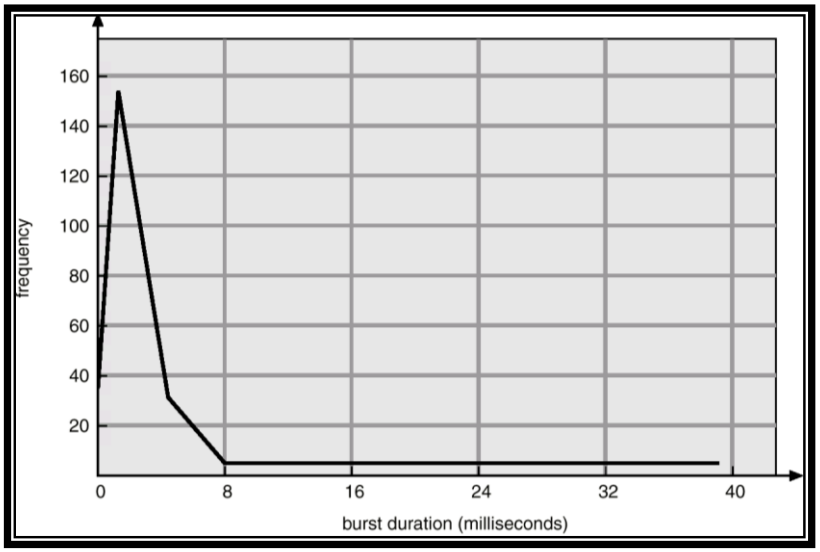
Topics Covered

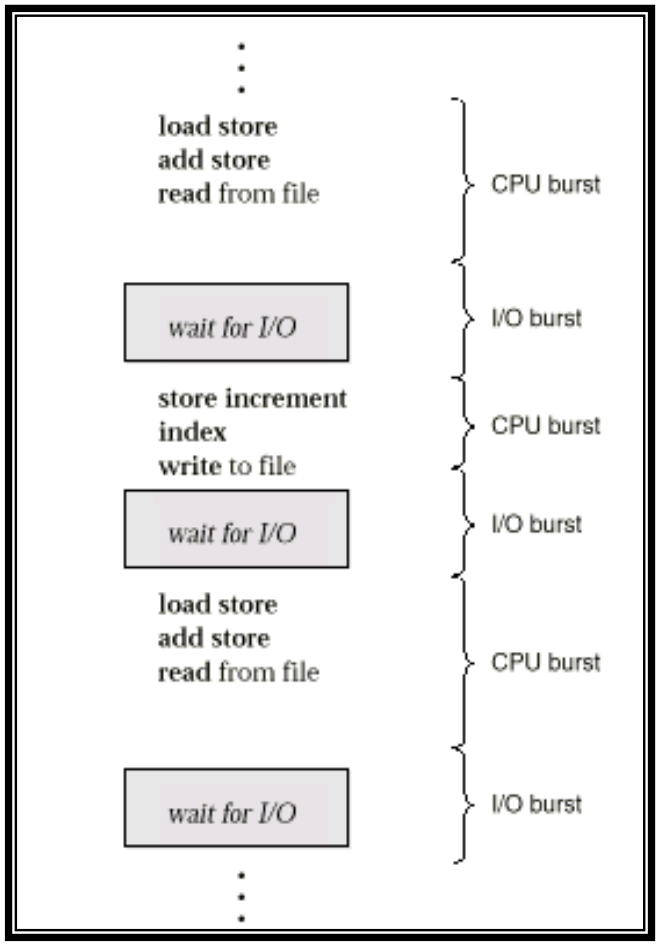
- ❒ Character Devices
- ❒ Block Devices
- ❒ Network Devices
- ❒ Clocks and Timers

CPU Execution and I/O

프로세스 실행될 때
CPU burst : Instruction 처리
I/O burst : I/O 작업 수행

- The two main jobs of a computer
  - CPU execution
  - Input/Output
- Usually, the main job is I/O
  - The CPU execution is incidental





일반적으로 CPU burst는 실행 간격 짧음, I/O 작업은 실행 시간이 더 김 → 사용자가 느끼는 컴퓨터의 성능은 I/O에 의해 결정 됨

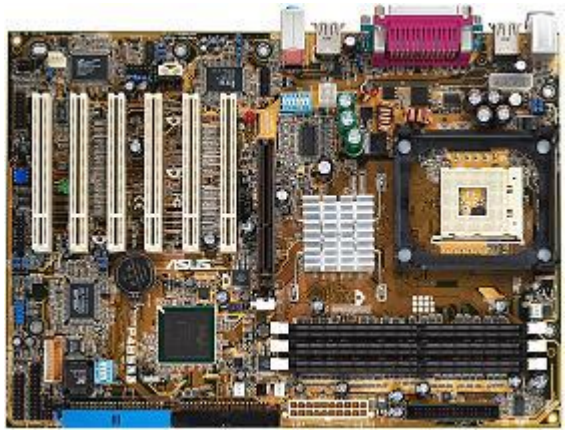
Variety of I/O Devices

- A computer is equipped with many I/O devices
  - VGA card, network card, disk controller, …

wifi

Network card

Main board



Video card

print

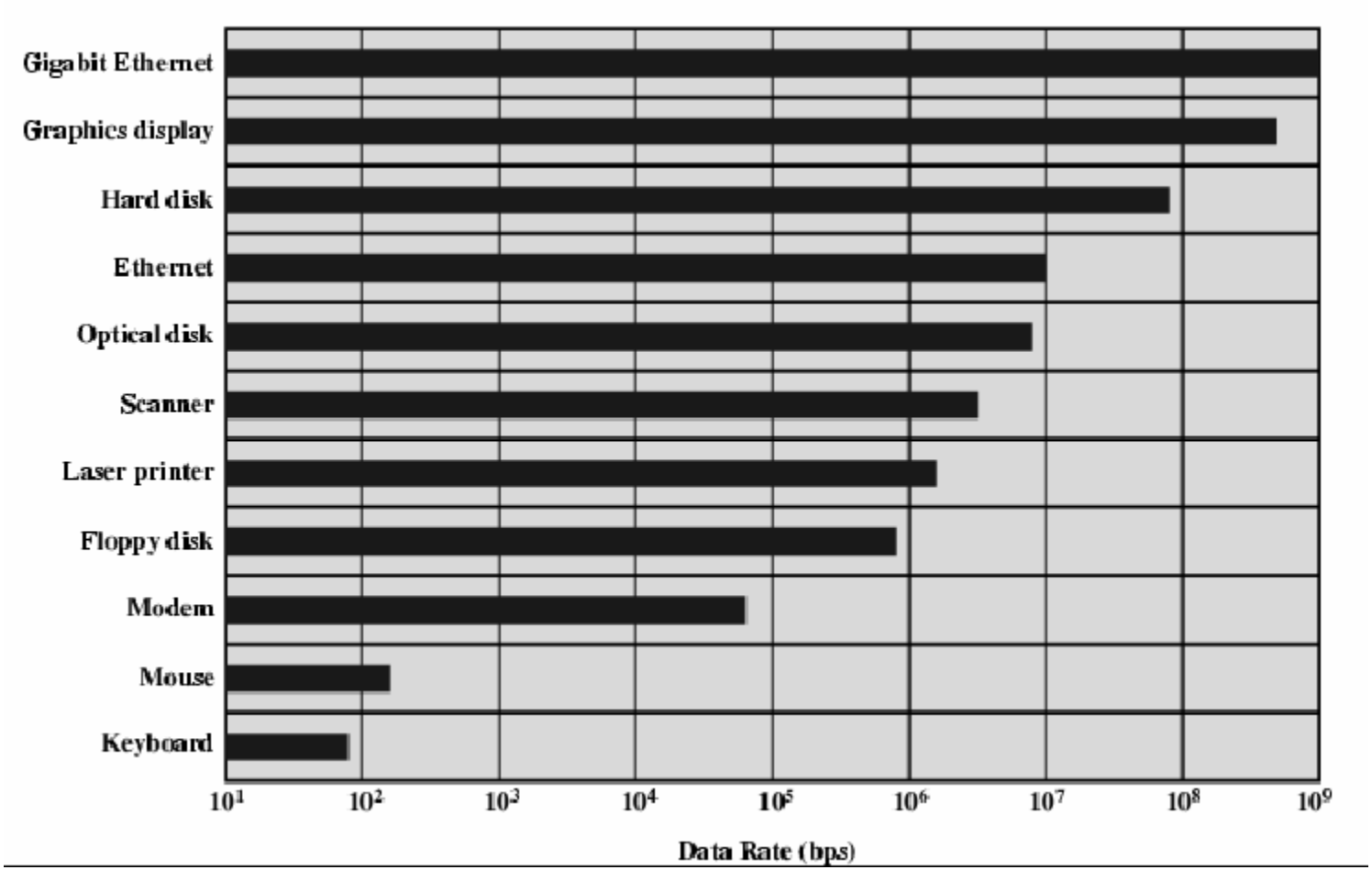
Solid state drive

Hard disk

USB

Sound card

Typical I/O Data Rates



Types of I/O Devices

- **Character devices (mouse, terminal, etc)**
  - **Commands include *get* and *put***

byte (char) 단위

- **Block devices (disk drive, flash drive, etc)**
  - **Commands include *read, write*, and *seek***
  - **Raw I/O or file-system access**
  - **Memory-mapped file I/O access possible**
    - mmap( ) function: convenient programming interface
    - mmap is faster than read, because of single copy operation

block 단위
(byte들의 묶음)

- **Network devices**
  - **NIC (network interface card)**

packet 단위
(byte의 stream)

Character Device: Mouse (1)

- **Brief history**
  - **First mechanical mouse with a roller ball**
    - Bill English at Xerox PARC in the early 1970s
  - **Introduced by Apple Macintosh in 1984**
    - They have helped to completely redefine the way we use computers since then
  - **Became the PC-human interface of choice quickly when Windows 3.1 made Graphical User Interface (GUI) a standard**
  - **Optical Mouse**
    - Gary Gordon at Agilent Laboratories in 1999

Character Device: Mouse (2)

❒ Optical mouse

- **Tiny camera takes 1500-7080 images per second**
  - Camera = laser + a CMOS sensor
- **Images sent for analysis to a DSP operating typically at 18 MIPS**
- **DSP detects patterns in images and thus estimates motion**
- **Data ports are used for two-way communication**
- **Upon mouse movement, a 3/5-byte packet is sent to the port**
  - Typical description of the data
    - ✓ (x_s, y_s), (x_d, y_d), mouse-up/down
- **This data packet is decoded by the mouse driver and its internal co-ordinates are updated**

Character Device: Mouse (3)

- Data transfer with optical mouse
  - Sensors (CMOS)
  - Mouse Controller (DSP)
  - Communication link (Cable/Wireless)
  - Data interface (Serial, PS/2, USB)
  - Device driver
  - Application

Character Device: Terminal (1)

- Example
  - DEC VT100, Heathkit Z19

Character Device: Terminal (2)

- **Terminal = keyboard + display**
  - **Keyboard and display are handled independent in most systems (no automatic echo, full duplex serial link)**

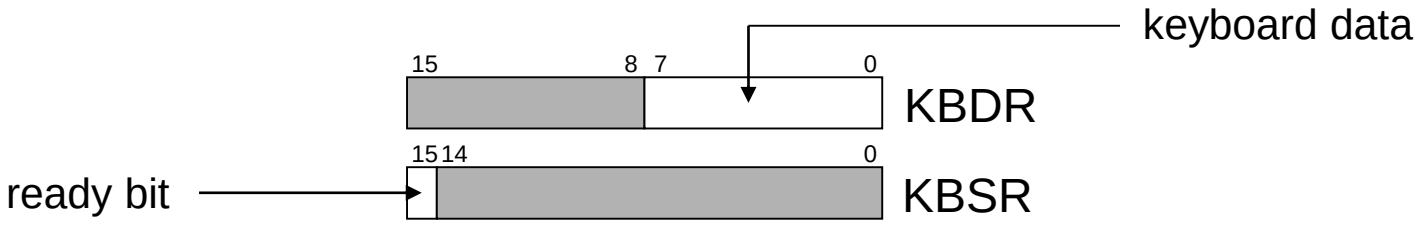
- **I/O registers are connected to host via serial line**
  - **Keyboard data/status registers**
    - KBDR (Keyboard Data Register), KBSR (Keyboard Status Register)
  - **Display data/status registers**
    - DDR (Display Data Register), DSR (Display Status Register)

Character Device: Terminal (3)

- **Keyboard input handling**
  - **One interrupt per character**
    - One character (8-bit data or control function) is sent at a time
    - ASCII encoding is used: ‘A’ is 0x65
  - **Slow speed**
    - 10-1800 characters per second
    - Measure: Baud rate (bits per second)

Character Device: Terminal (4)

- **When a character is typed:**
  - **Its ASCII code is placed in bits [7:0] of keyboard data register**
  - **The "ready bit" of keyboard status register is set to zero**
  - **Generate interrupt and disable keyboard**
    - Any typed characters will be ignored

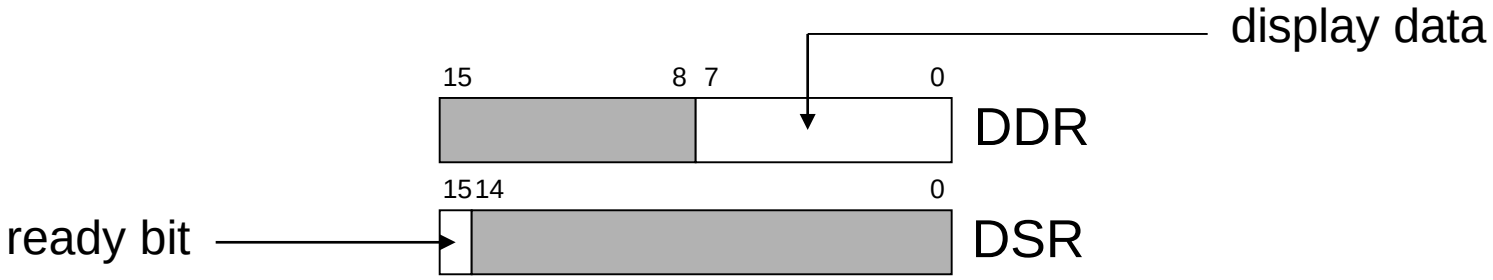


- **When KBDR is read:**
  - **The "ready bit" of KBSR is set to one**
  - **Enable keyboard**

Character Device: Terminal (5)

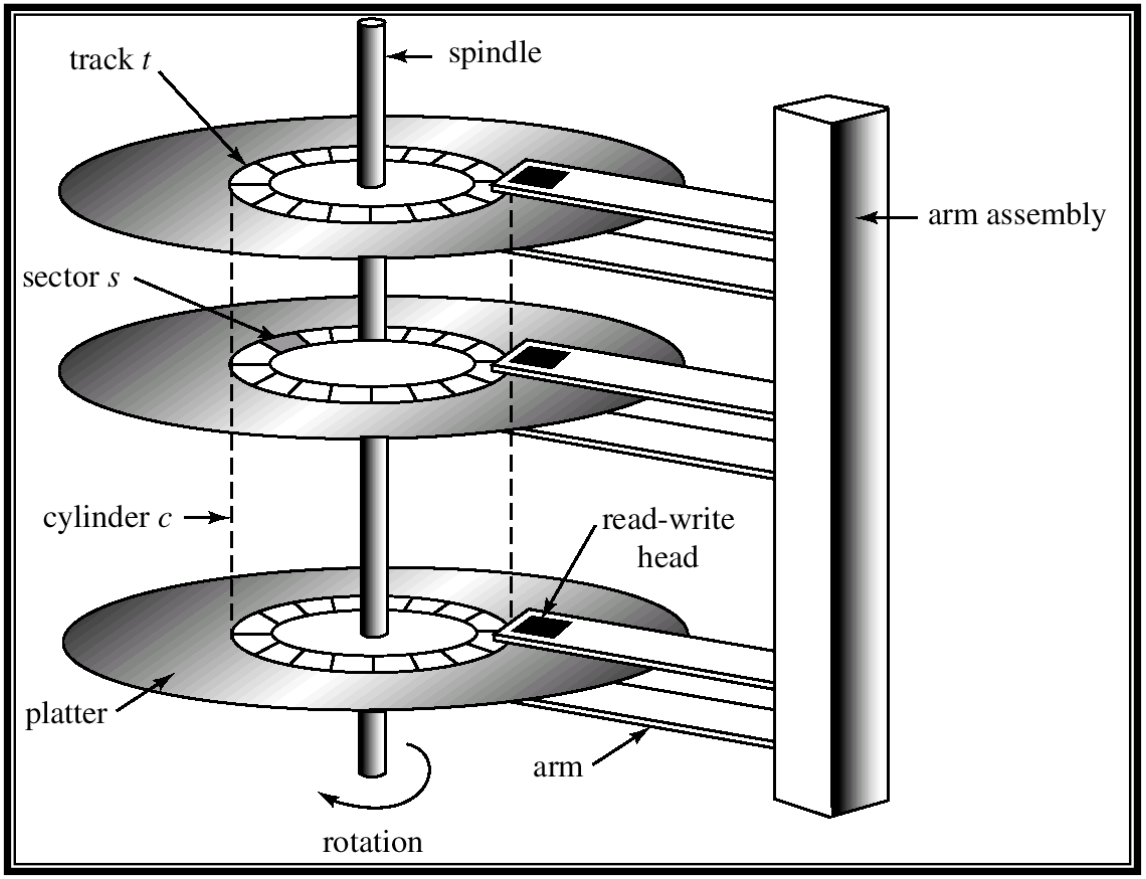
❒ Display output handling

- **When monitor is ready to display another character**
  - The “ready bit” of display status register is set to one



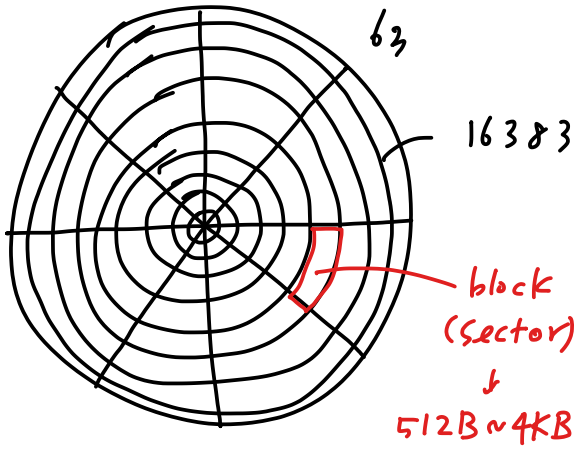
- **When data is written to DDR:**
  - The “ready bit” of DSR is set to zero
  - Character in DDR is displayed
  - Any other character data written to DDR is ignored

Block Device: Disk Drive (1)



Block Device: Disk Drive (2)

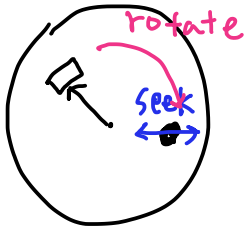
- **Example disk characteristics**
  - **2-6 heads (platters x 2)**
  - **Platter diameter between 0.8" and 8"**
  - **16,383 tracks (cylinders) per surface**
  - **63 blocks (sectors) per track**
  - **Block (sector) size of 512 to 4096 bytes**
    - 4KB physical emulated at 512-byte sectors
  - **Capacity ranges up to 4 TB**



Block Device: Disk Drive (3)

❒ Disk operation

- **Select desired read/write head**
- **Move heads to the correct track (“*seek*”)**
  - Seek time
- **Wait for disk to “rotate” desired block into position**
  - Rotational delay
- **Read from or write to the block**
  - Transfer latency



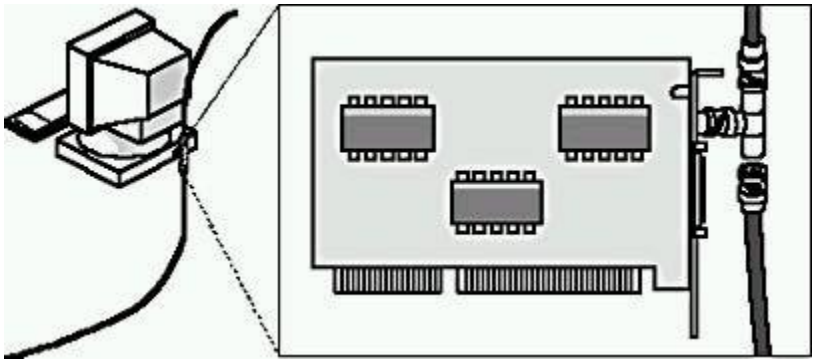
❒ Disk performance

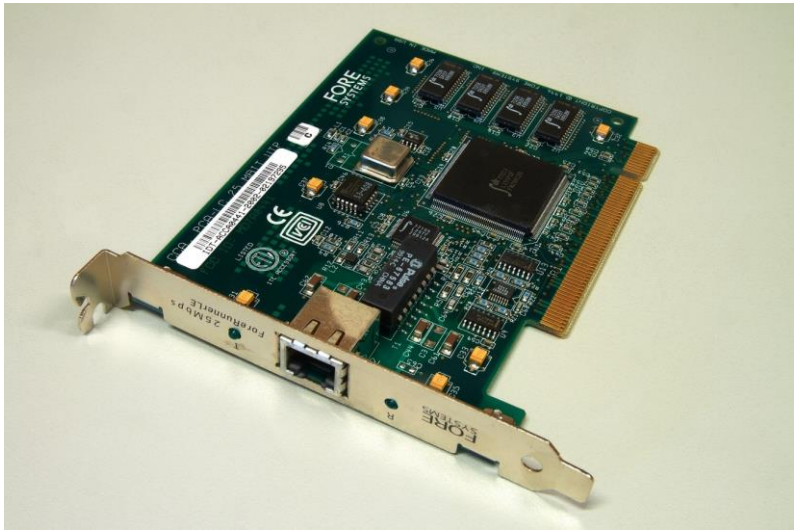
- **Seek time: 0-50 ms (average 10-20 ms)**
- **Rotational delay: 0-16 ms**
- **Typical drive spins at 3600-5400 RPM**

하드디스크의 분당 회전 속도

Network Interfaces

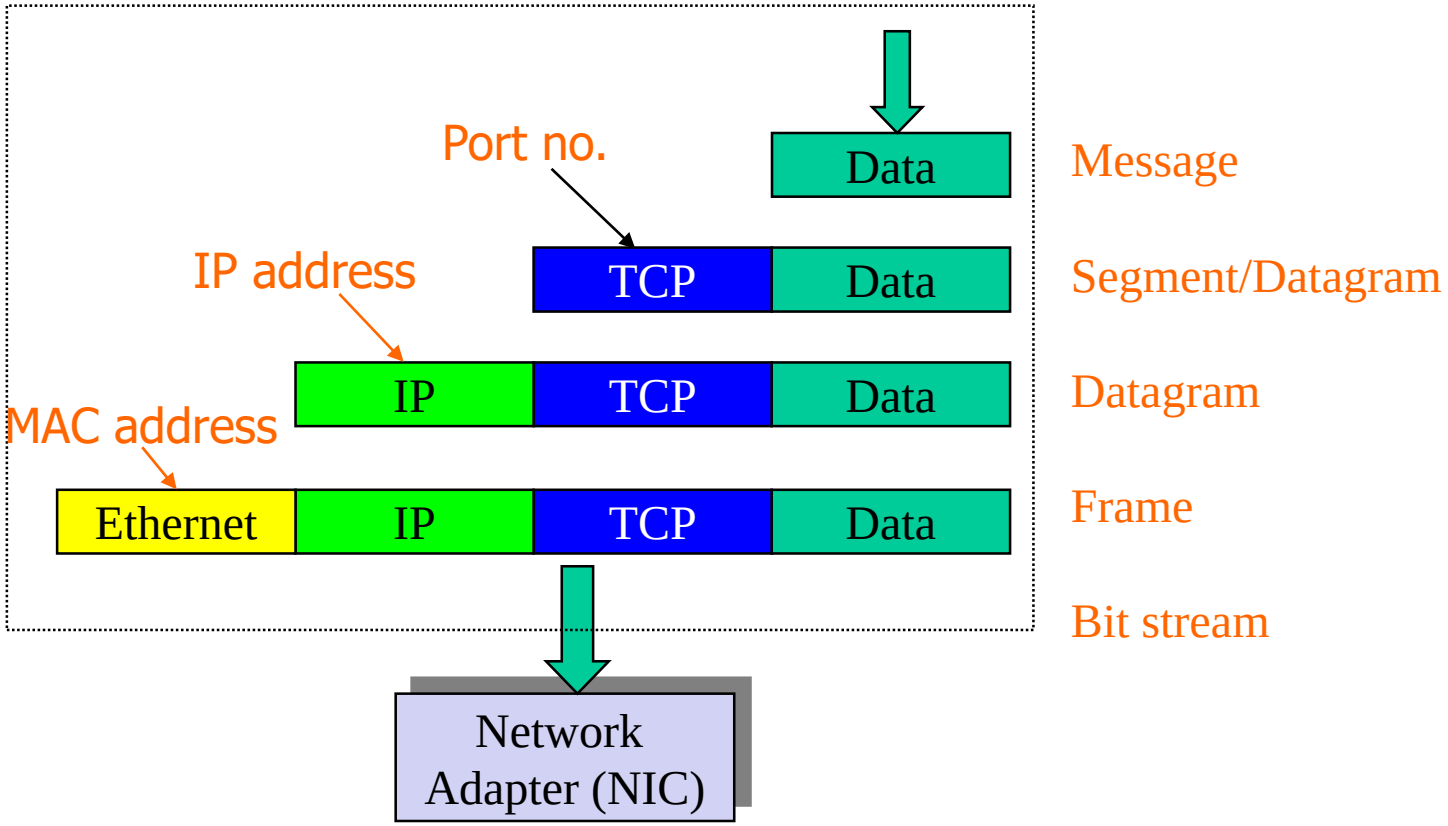
- Network interfaces
  - UART and RS-232, USB, ethernet, etc



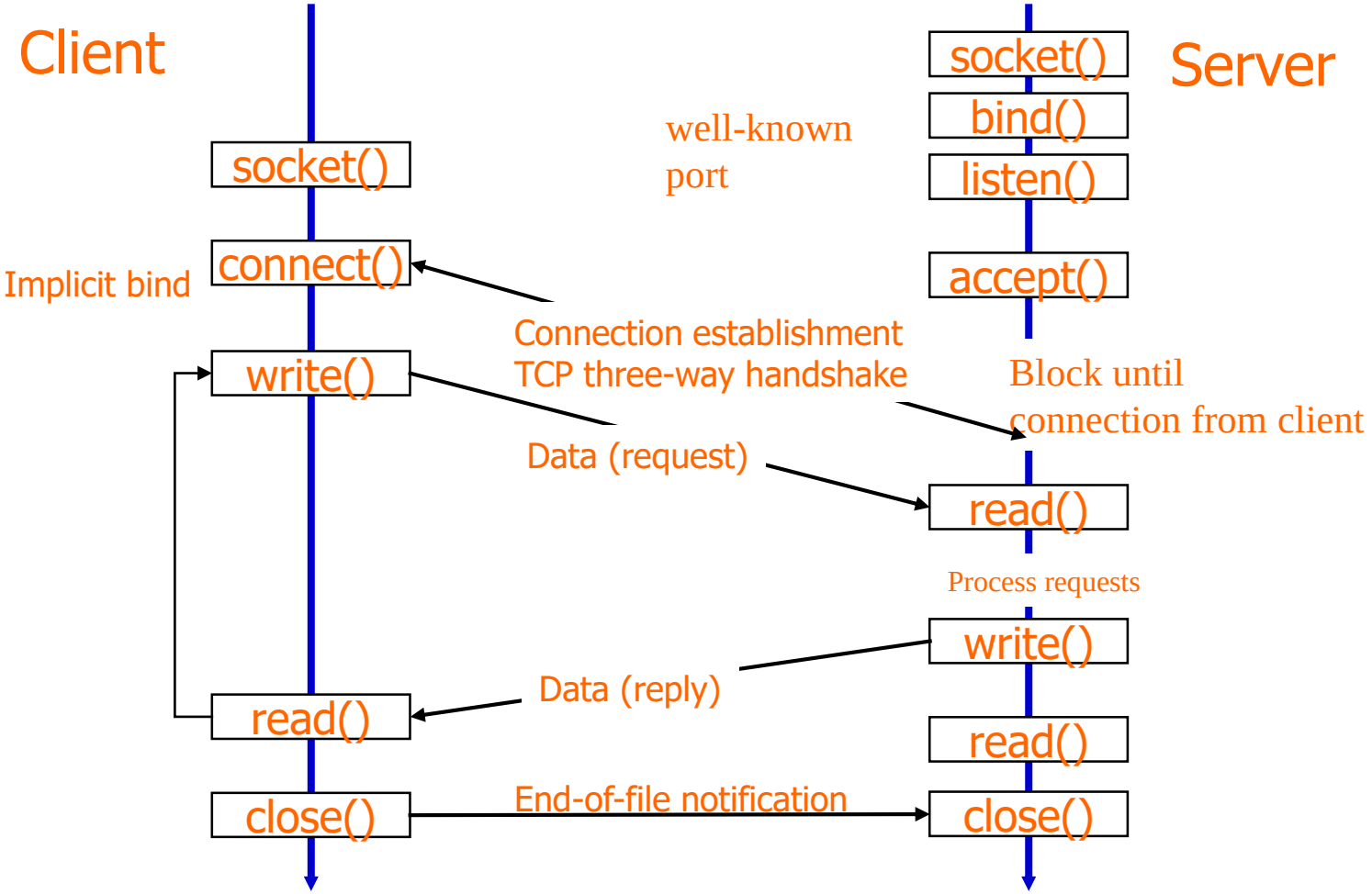


Layering (Protocol Stack)

- Network I/O is more complex than other types of I/O
  - Complex network protocols and layered processing



TCP Socket Programming Flow



Clocks and Timers

- Provide three basic functions
  - Give the current time
  - Give the elapsed time
  - Set a timer to trigger operation X at time Y
- Programmable interval timer
  - The hardware for last two functions
  - Wait and generate an interrupt
  - CPU scheduler uses this
- Unfortunately, the system calls for timer functions are not standardized across operating systems
  - alarm() in UNIX
  - timer_create() and timer_settime() in POSIX

Unix Time Values

- **Two different time values**
  - **Calendar time**
    - Counts the number of seconds since the Epoch (Jan 1, 1970)
    - UTC (coordinated Universal Time, Greenwich Mean Time)
  - **Process time**
    - Measures the CPU time used by a process
    - Process time is measured in clock ticks
      - ✓ Usually, 50, 60, or 100 ticks per second
    - CPU time = User CPU time + System CPU time

- **Lack of timer support on Unix**
  - **We can get around by using alarm( ) system call**
    - It generates a SIGALRM signal after the number of real-time seconds (specified by seconds) has elapsed
  - **`ioctl` on UNIX covers odd aspects of I/O such as clocks and timers (catchall for I/O operations)**

Linux

- The timer interrupt is set to a default frequency
  - “linux/param.h”
  - 100 Hz for most hardware platforms

- Jiffies
  - The number of clock ticks since the operating system was booted
  - When the timer interrupt occurs, the jiffies value is incremented

thank you!