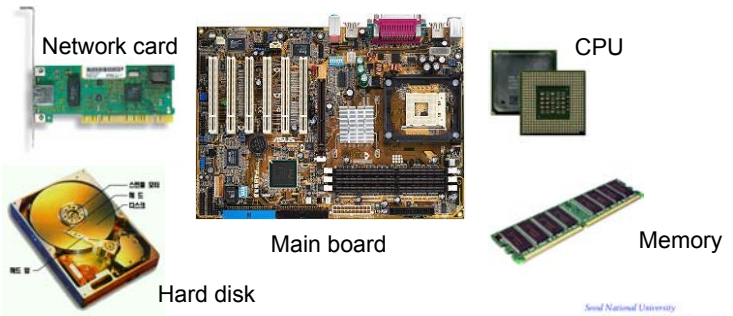
I/O Devices and Device Drivers (Topic 11)

홍 성 수

서울대학교 공과대학 전기 공학부
Real-Time Operating Systems Laboratory

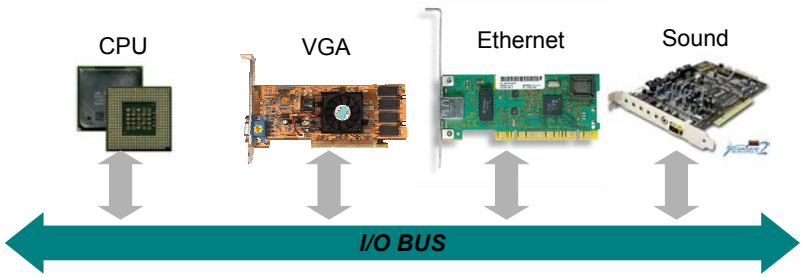
Inside Your PC

- Your PC is equipped with
  - CPU and memory
  - many I/O devices
    - VGA card, network card, disk controller, …



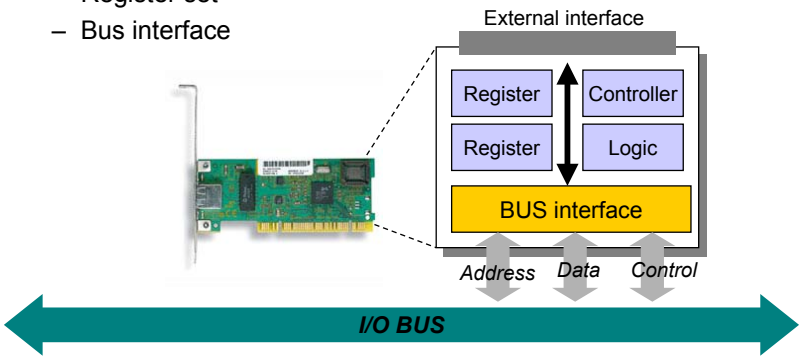
Device Connection

- Devices are attached to I/O bus
  - ISA, PCI, EISA, SCSI, …



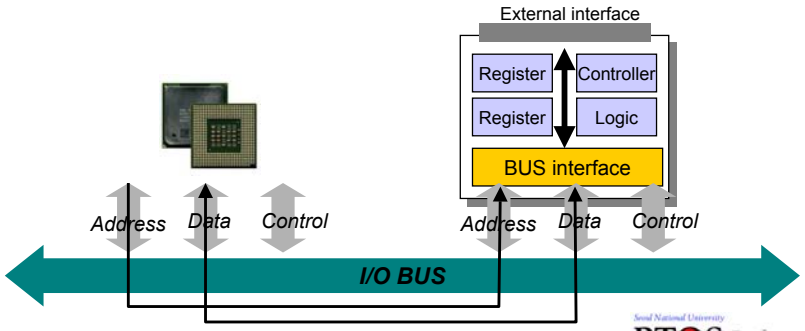
Inside I/O Device

- I/O device consists of
  - Controller, Logic
  - Register set
  - Bus interface



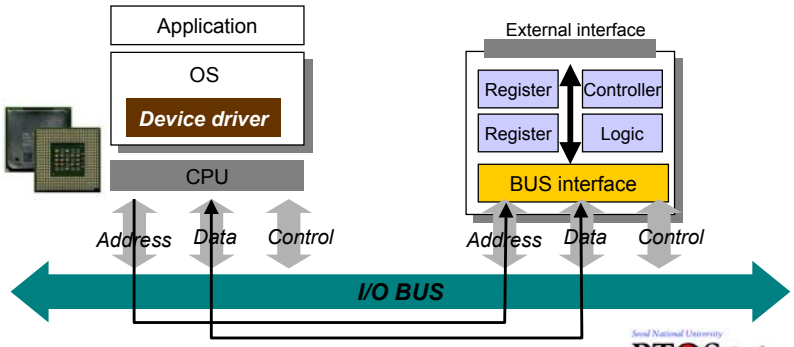
Device Control

- How CPU control device
  - CPU write register address on the bus
    - Address spaces are assigned to each device
  - Device write / read data on the bus



Device Driver

- Software layer between application and device
- Implements operations exposed to users
  - (Ex) open, close, read, write, ioctl, …



What is Device Driver?

- Software layer between application and device
- Implements operations exposed to users
  - (Ex) open, close, read, write, ioctl, …

Types of Linux Devices

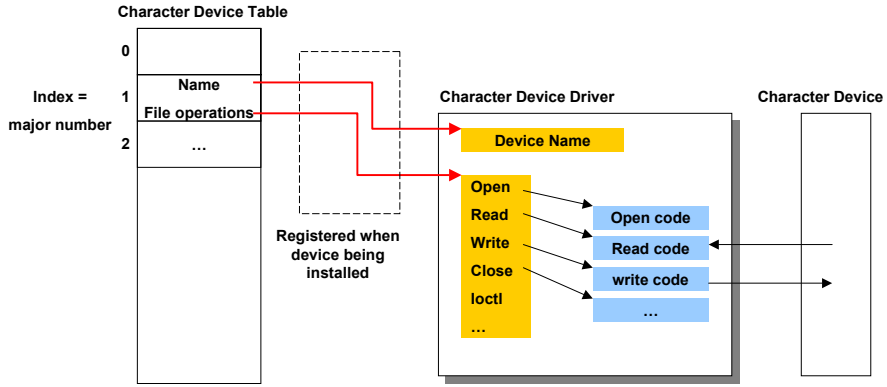
- Character Devices
  - Transfer unit: character (byte)
  - Can be accessed like a file (open, close, read, write, …)
  - (Ex) console, keyboard, mouse, … (/dev/tty1, /dev/lp1, …)
- Block Devices
  - Transfer unit: block (usually some kilobytes)
  - Can be accessed like a file (open, close, read, write, …)
  - External view to users is same as character devices
  - Internally, block buffer is used for efficiency (contrast to character devices)
  - (Ex) hard disk drive, CD-ROM drive, … (/dev/hda1, …)
- Network Interfaces
  - Can not be easily mapped to neither character or block devices
  - (Ex) eth0

Device Files

- Represent an I/O device
  - /dev/tty0 = first serial port
- Two device files can represent same I/O devices (but may be implemented in a different way)
  - /dev/psaux, /dev/psmouse = serial mouse
- Attributes
  - Type: block or character
  - Major number: specifies device driver
  - Minor number: argument to device driver, kernel don't care

Name	Type	Major	Minor	Desc
/dev/fd0	Block	2	0	Floppy Disk
/dev/had	Block	3	0	First IDE disk
/dev/hda2	Block	3	2	Second partition of first IDE disk
/dev/hdb	Block	3	64	Second IDE disk
/dev/console	Char	5	1	Console
/dev/null	char	1	3	Null device

Structure of Character Device Drivers



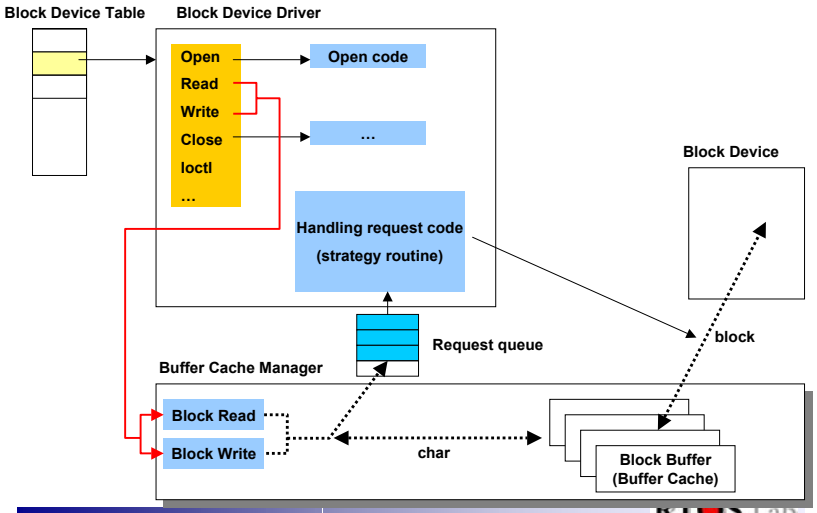
Character Device Drivers

- To applications:
  - Device file = Regular file
- To file system:
  - Regular file: read from or write to disk drive
  - Device file: invoke device driver operations
    - Open: initialize device
    - Read: device data → user buffer
    - Write: user buffer → device
    - Close: called when removed from device table
    - Ioctl: device specific control
      - (Ex) baud rate change, access permission set,

Block Device Drivers

- Difference with Character Device Drivers
  - Actual transfer unit: *block* (some kilobytes)
  - <Device Driver : Device> transfer unit cannot be character.
- How to implement character oriented read/write?
  - Kernel provides general *block conscious* read/write functions
    - translate character I/O to block I/O
  - Device driver only need to implement *handle block request*
    - sometimes called as 'strategy routine'

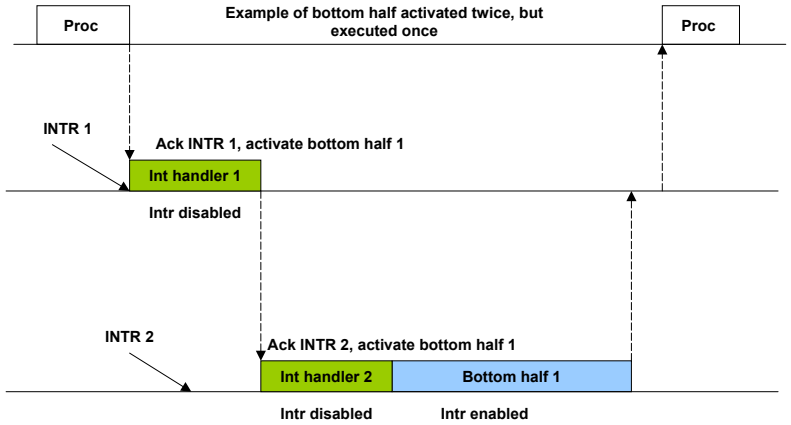
Structure of Block Device Drivers



Interrupt Handling in Device Drivers

- Register 'interrupt handling routine' when opening
- Kernel invokes registered handler when interrupt occurs
- Linux interrupt handler
  - Interrupt Handler
    - High priority function (such as acknowledging to PIC)
    - Run with *interrupt disabled*
    - Invoked every time interrupt occurred
    - Marks a bottom half as active
  - Bottom Half
    - Low priority function (such as transferring data from device)
    - Run with *interrupt enabled*
    - Execution may be deferred

Bottom Half



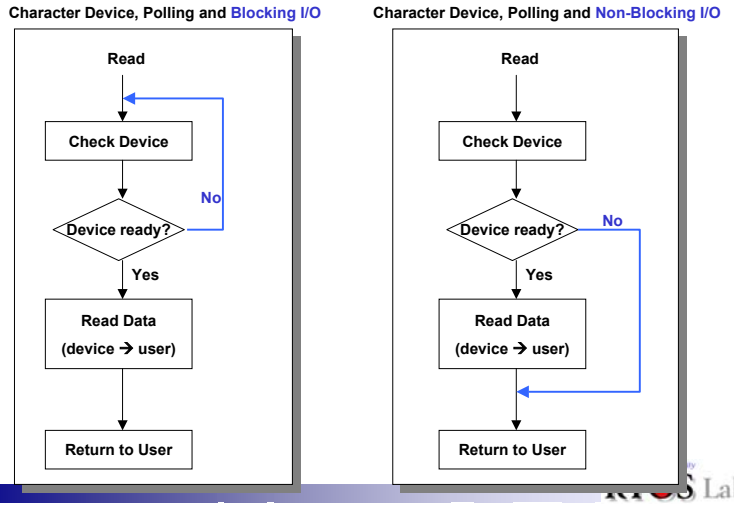
General Device Driver Routines (1)

- Open
  - Reads minor number
  - Initialize appropriate device
  - Initialize device driver internal data structures
  - If needed, change the file operation table (according to minor)
  - Increase usage counter
  - If needed, register interrupt handler, enable irq
- Close
  - Free dynamically allocated data structures
  - Decrease usage counter
  - Un-register the interrupt handler

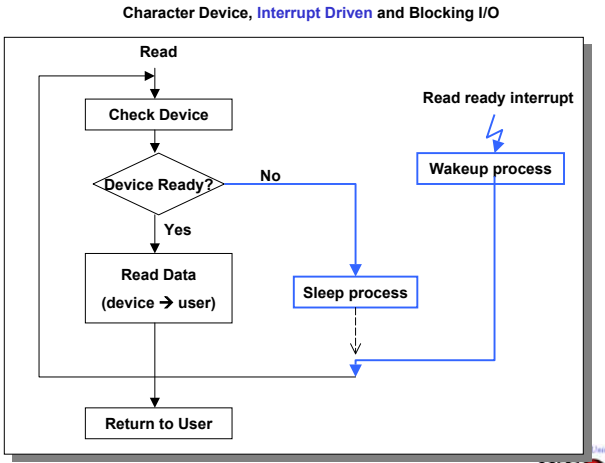
General Device Driver Routines (2)

- Read / Write
  - Polling vs Interrupt Driven
    - Polling
      - checks device if data can be read or written
      - Usually for character device
    - Interrupt Driven
      - sleep process until data can be read or written
      - Usually for character and block device
  - Blocking vs Non-Blocking
    - Blocking
      - If data cannot be read or written: wait until ready
    - Non-Blocking:
      - If data cannot be read or written: immediately returns

General Device Driver Routines (3)



General Device Driver Routines (4)



General Device Driver Routines (5)