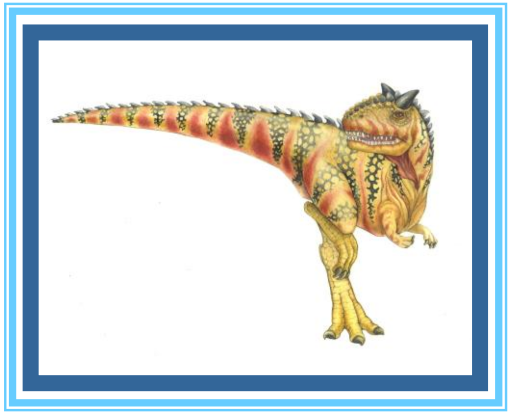
Chapter 0: Abstract



Introduction

- Lecturer

  - Gunhyuk Park (Office: EECS Building C–201)
  - Contact: 2261, maharaga@gist.ac.kr
  - TA: Minwook Lee, minwook-lee@gist.ac.kr

- Time

  - Mon/Wed 9:00 AM – 10:15 AM
  - Office Hour: Wed 3:00 PM – 5:00 PM (by Zoom)
  - Introduction to PintOS Projects: Irregularly, not confirmed yet (by Zoom)

- Reference

  - “Operating System Concepts”, A. Silberschatz, P. B. Galvin, and G. Gagne <- **Read the book!**

- Evaluation

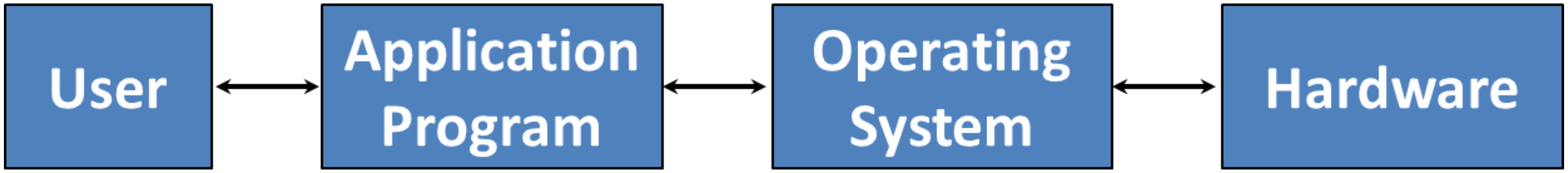
  Participation (10%), Project (20%), Midterm (30%), Final (40%)
  You should participate in the class at least 66%
  Your final grade is given by normalized absolute score

Course Overview

- W1: Introduction to Operating Systems
- W2: Operating System Structure & GitHub Introduction <- project team building
- W3: Process <- 1st project
- W4: Threads
- W5: CPU Scheduling <- 1st due, 2nd project
- W6: Process Synchronization
- W7: Deadlock and Starvation <- 2nd project due
- W8: Midterm Exam
- W9: Main Memory Management <- 3rd project
- W10: Virtual Memory Management
- W11: File-System Interface <- 3rd project due
- W12: File-System Implementation <- 4th project
- W13: Mass-Storage Structure
- W14: I/O Systems <- 4th due
- W15: Protection & Security
- W16: Final Exam

What is Operating System?

- A software that acts as an intermediary between a user of a computer and its hardware



Keyboard Input
Voice Input
Mouse Click
Touch Screen Tapping

Sound Feedback
Video Output
Application Animation

What a User Sees

Keyboard Input for SNS:
Define Text Variable
Read Keyboard Input in the Variable
Send Text Pointer to Graphics UI Buffer

Return Memory Address for Text Variable
Return the Text in Kernel Buffer
when the Application Asks for it

What a Front-end Programmer Sees

Keyboard Input for SNS:
Request Memory for Text Variable (Memory)
Checks Buffer (Keyboard)

Find an Empty Memory
with Sufficient Length and Return it
Detecting Key Inputs and Stores them in Buffer

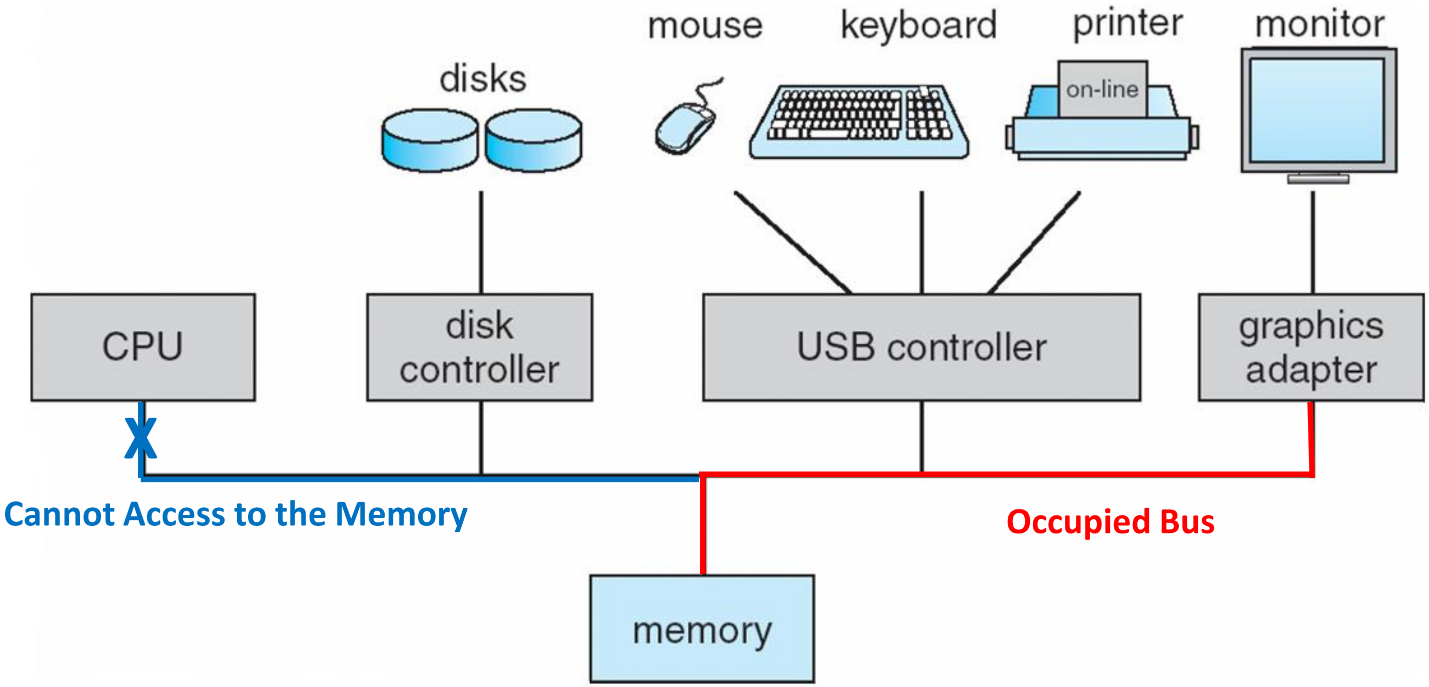
What a Back-end Programmer Sees

OS, Integration of Basics

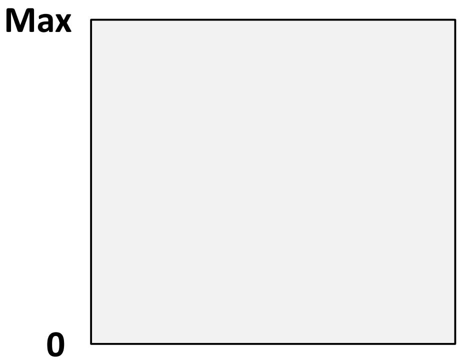
- Software
  - Computer Programming
    C language, basic data structures, pointer, system calls (write, open, …)
  - Object-Oriented Programming
    C++/Java, class, instance, graphical user interface
  - Data Structure and Algorithms
    Efficient way of solving problems, array, list, time complexity
- Hardware
  - Digital System Design
    Flip-flops, logic gates, boolean algebra, state machine, ALU
  - Computer Architecture
    von Neumann architecture, BUS, buffer, synchronous/asynchronous
- Operating system handles computer hardware with various software techniques for efficiency, robustness, and accuracy

Computer System Organization

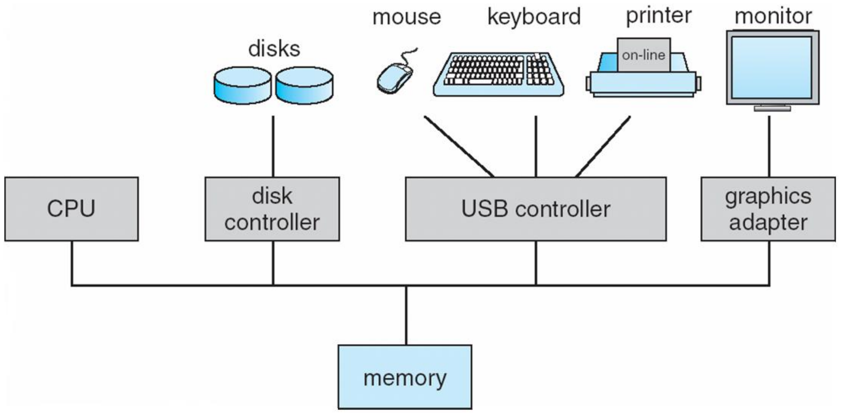
- Computer-system operation
  - One or more CPUs and device controllers are connected through a common bus that provides access to shared memory
  - Concurrent execution of applications in CPUs and devices competes for memory cycles



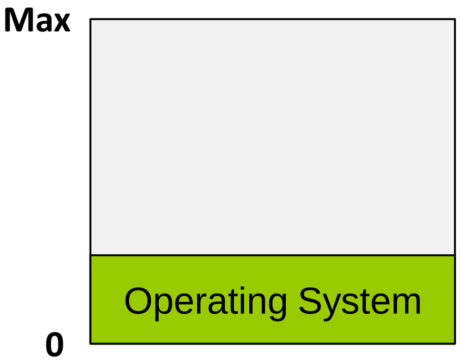
Operating System Abstract



Main Memory Before Booting

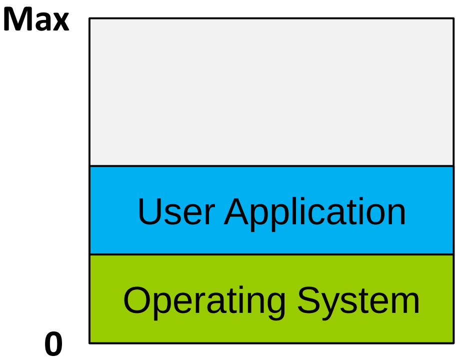


- Booting from Power Off
  - Bootstrap from ROM or EEPROM
  - Checks peripheral devices and main system
  - Runs an operating system stored in a disk



Main Memory After Booting

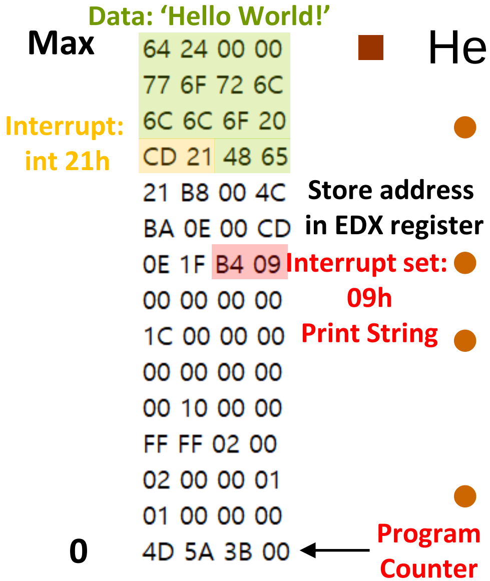
Operating System Abstract (2)



Main Memory
After Running an Application

```
#include <stdio.h>

int main() {
  printf("Hello World!\n");
  return 0;
}
```



Data: 'Hello World!'

Max

Interrupt: int 21h

Store address in EDX register

Interrupt set: 09h Print String

```
64 24 00 00
77 6F 72 6C
6C 6C 6F 20
CD 21 48 65
21 B8 00 4C
BA 0E 00 CD
0E 1F B4 09
00 00 00 00
1C 00 00 00
00 00 00 00
00 10 00 00
FF FF 02 00
02 00 00 01
01 00 00 00
4D 5A 3B 00
```

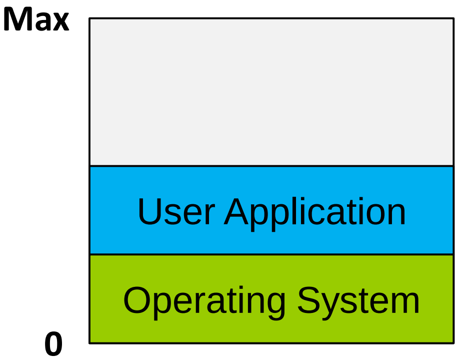
0

Program Counter

Hello World Application in Hex Code

- Hello World Application
  - A user generates a code and let a computer compiles it to generate a program in machine language
  - Program: HelloWorld.exe in a disk
  - Process: Corresponding hex code is loaded in main memory
  - These hex codes are generated differently by your compiler and CPU

Operating System Abstract (3)



Max

User Application

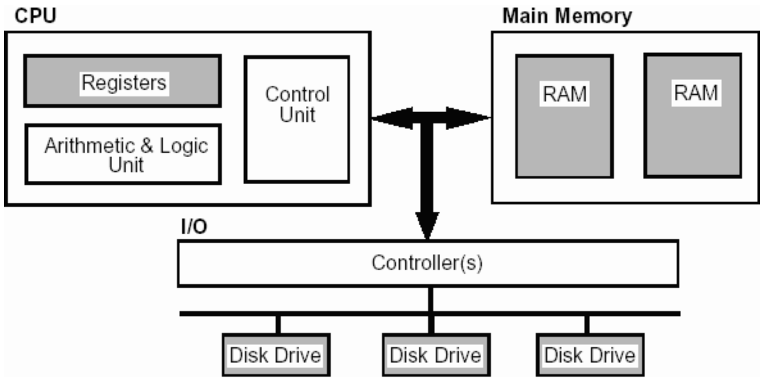
Operating System

0

Main Memory
After Running an Application

```
           Data: 'Hello World!'
Max        64 24 00 00
           77 6F 72 6C
Interrupt: 6C 6C 6F 20
  int 21h  CD 21 48 65
           21 B8 00 4C  Store address
           BA 0E 00 CD  in EDX register
           0E 1F B4 09 Interrupt set:
           00 00 00 00        09h
           1C 00 00 00  Print String
           00 00 00 00
           00 10 00 00
           FF FF 02 00
           02 00 00 01
           01 00 00 00
0          4D 5A 3B 00 ←  Program
                          Counter
```

Hello World Application
in Hex Code



- In 32-bit system, on demands, 4 bytes of data are loaded at the program counter in main memory to the instruction register
- Each instruction varies from 1 byte to several bytes

So… PintOS

- What’s PintOS?

  - Educational operating system for x86 architecture

  - Developed by Ben Pfaff at 2004

  - Built on a x86 simulator (QEMU, bochs)

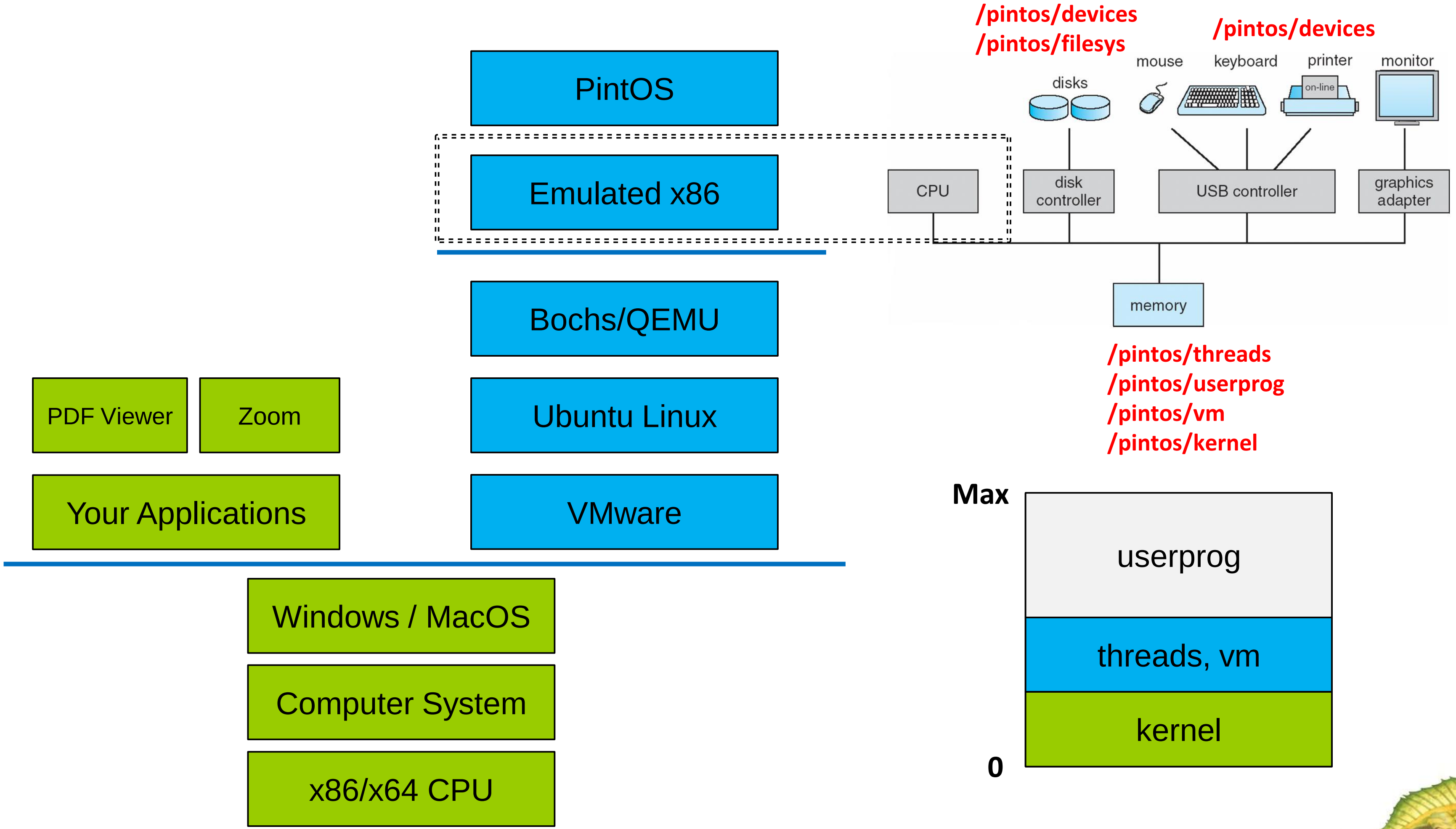
- Why PintOS?

  - You can learn concepts and mechanisms of OS components in the course, but it does not mean you ‘understand’ how it works

    -> The same concept can be differently implemented (same policy, different mechanism)

  - PintOS is a small, simple, short, and easy-to-understand operating system

PintOS Structure



PintOS

Emulated x86

Bochs/QEMU

Ubuntu Linux

VMware

PDF Viewer

Zoom

Your Applications

Windows / MacOS

Computer System

x86/x64 CPU

/pintos/devices
/pintos/filesys

/pintos/devices

/pintos/threads
/pintos/userprog
/pintos/vm
/pintos/kernel

Max

userprog

threads, vm

kernel

0

PintOS Projects

- # of team members: 2-3,

- Four projects will be announced:
  1. Pintos Installation and Alarm Clock
  2. Process Schedulers
  3. System Calls
  4. Virtual Memory

- In the project announcing week, TA will give an introductory talk with you
  - You can ask TA for details of projects
  - Participation is optional
  - Let’s set the meeting time slot with TA

- Keep in mind
  - Do not copy any of codes; Once your plagiarism is detected, you get F