

Pintos Introduction

[CSE4070]

Instructors

Prof. Prof. Sungyong Park

Teaching Assistants

Hongsu Byun

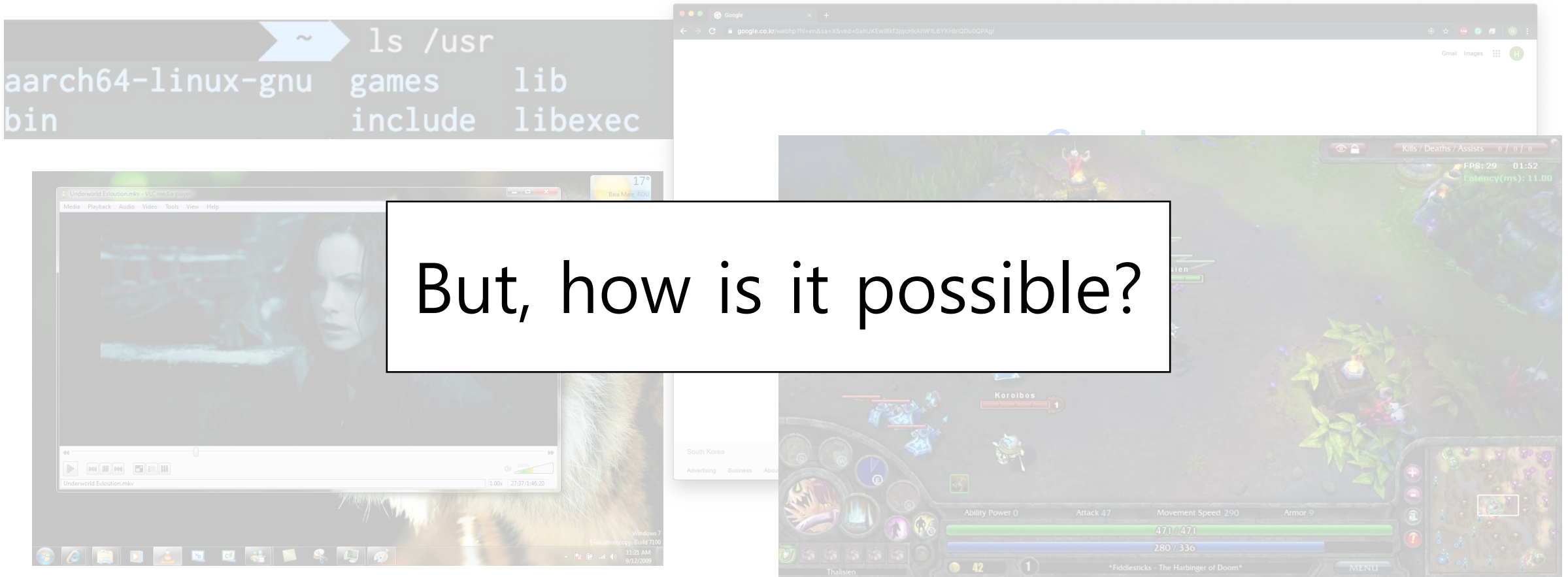
Fall 2021

Contents

1. **Operating Systems**
2. **What is Pintos?**
3. **Virtual Machine**
4. **Structures of Pintos**
5. **From Booting to Power Off**
6. **Project Schedule**

Operating Systems

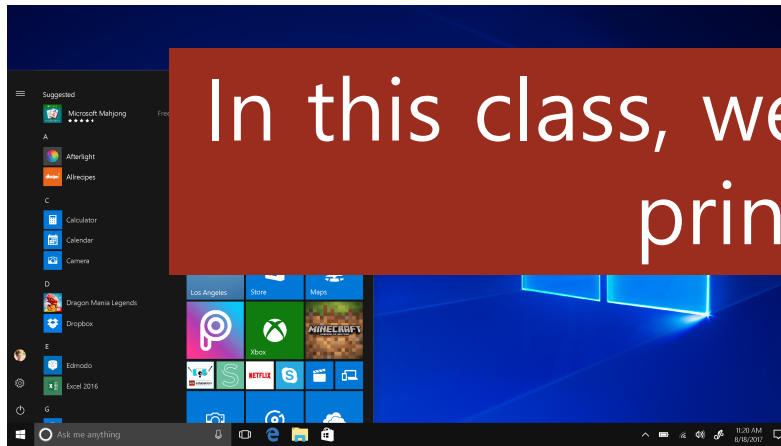
- We usually double click icon to launch applications on Windows or type commands on Linux and use the applications or see the result.



Operating Systems

- What do we need to run applications?
- We need some intermediary between hardware and applications.
- It is operating system.

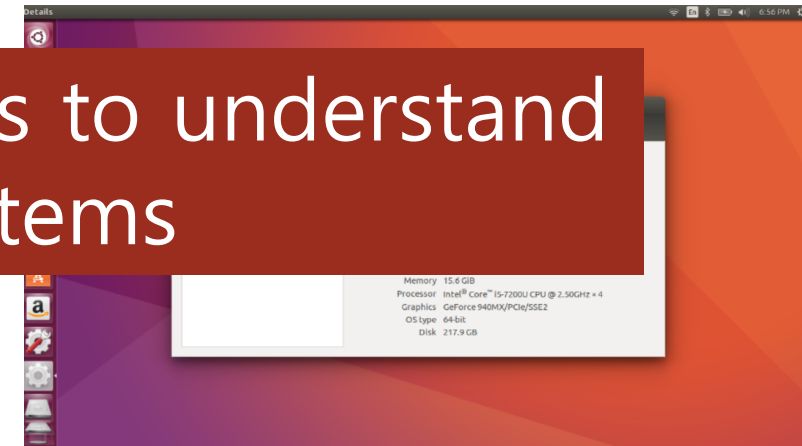
In this class, we will do Pintos projects to understand principles of operating systems



Windows



macOS



Linux

What is Pintos?

- Though existing OSes have rich functionalities, Pintos doesn't.
- Pintos is a simple operating system framework for the 80x86 architecture.
- It only supports basic functionalities and provides us some skeletons.
 - ✓ Kernel threads
 - ✓ Loading and running user programs
 - ✓ Simple file system
 - ✓ Simple memory management
- We will develop Pintos to be more complicated operating system.

What is Pintos?

- Pintos is the operating system like Windows, macOS and Linux.
- Thus we can install the Pintos on real machine which has 80x86 architecture.
- It means we need the machine that can interpret Intel's ISA. (Instruction Set Architecture)
- If you don't have **80x86 architecture** machine, you should buy one.
- Even if we already have it, Pintos is simple so that we can not do many things by using Pintos.
- It means that it is **not comfortable to develop** Pintos in the machine where Pintos is installed.
- **We will use virtual machine** for Pintos projects.

Virtual Machine (VM)

- Virtual machine is an emulation of a computer system.
- We can think of it literally as virtually constructed machine.
- We can install and run operating systems or applications on virtual machine.
- We will run Pintos on virtual machine.
- Pintos supports two softwares for virtual machine, Bochs and QEMU.
- We use QEMU as Pintos emulator.

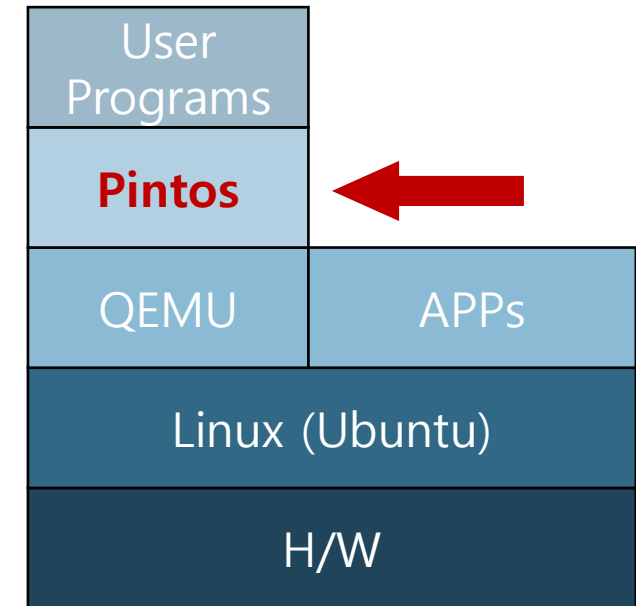


Structures of Pintos

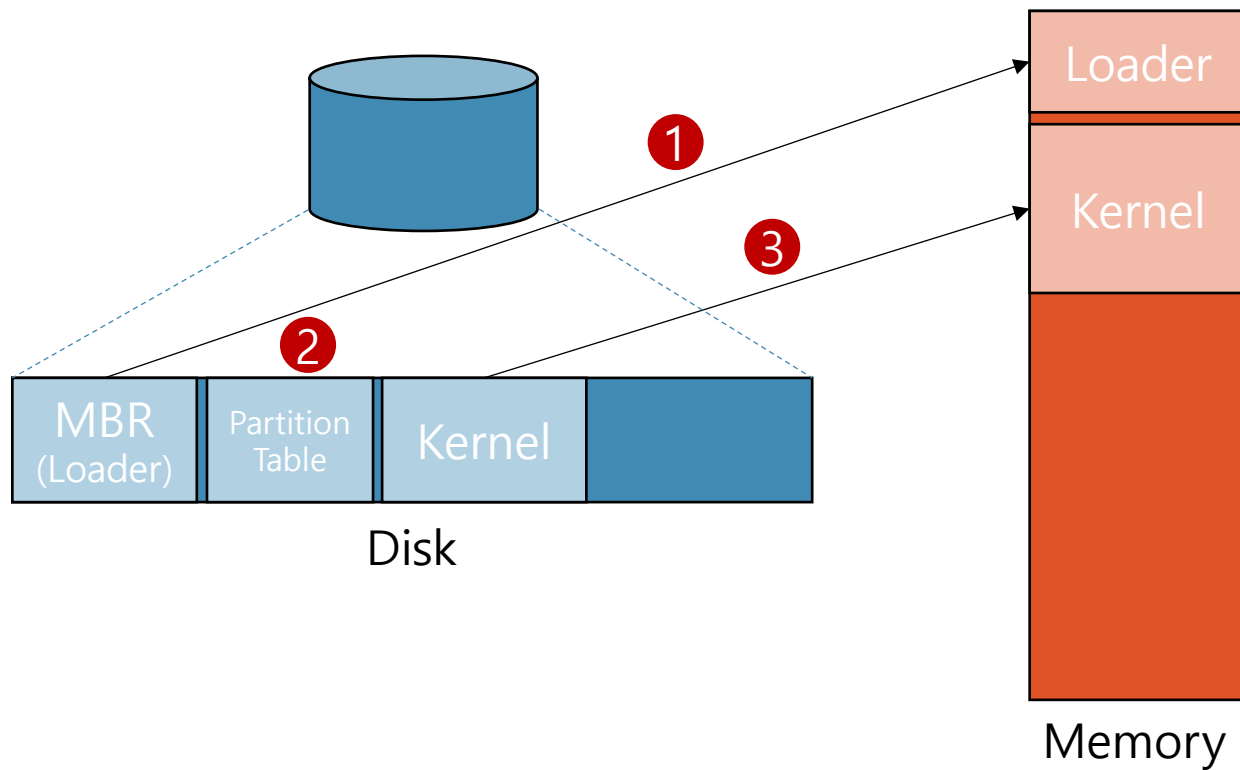
- Pintos can **boot, power off** and **run only one application** at once but you can not check the result of execution.
- The basic functionalities which Pintos provides.

Functionalities	Projects
Load and run user program	User Programs (1), (2) ✓
Kernel threads	Threads ✓
Simple file system	File Systems
Memory management	Virtual Memory

- We will do User Programs (1), (2) and Threads projects.



From Booting to Power Off



1. BIOS loads the loader into memory, which is stored in the first sector of the first disk.
2. Loader finds the kernel image by reading the partition table on each disk.
3. When the loader finds a bootable kernel partition, it reads the kernel into memory.
4. Loader's final job is to extract the entry point from the kernel image and transfer control to it.

From Booting to Power Off

- Kernel's entry point is `start()` in `'threads/starts.S'`
- It **obtains the machine's memory size**, by asking the BIOS for the PC's memory size.
- Basic page table is created, which maps virtual address **3GB ~ 3GB + 64MB** to physical address **0MB ~ 64MB**.
- Call **`main()`** in `'threads/init.c'`
- In `main()`, each kernel system is initialized such as **thread, memory and interrupt** system.
- If `'-q'` was specified on the kernel command line, `shutdown_power_off()` in `'devices/shutdown.c'` is called and Pintos is terminated.

