Pintos Overview

—— Why, What and How

TA Session

TA : zhongyinmin
Email : zhongyinmin@pku.edu.cn
Github : PKUFlyingPig

PKUOS

Welcome to the World of Operating System

Some announcements:

➢ Lab 0 Code will due next Thursday 11:59 pm

➢ Lab 0 Design Doc will due next Sunday 11:59 pm

Educational OS Project Zoo

JOS ➡️ xv6

IA32
MIT6.828

RISCV32
MIT6.S081

Nachos ➡️ Pintos

MIPS
Old CS162

IA32
CS162

Welcome to the World of Operating System

Q : Why Pintos ?

Design and Implementation

➢ OSDI, NSDI, PLDI … …

➢ Your design matters !!

➢ Talk is cheap, show me the code

➢ Write 2000+ LOC in a 10000+ LOC codebase

Welcome to the World of Operating System

Q : Why Pintos ?

You will learn by Read The Code

➢ important skill both in production and research

➢ learn from good coding style

➢ some tools may help you

Q : Why Pintos ?

You will learn by Design The Code

- ➤ think tenth, code once

- ➤ design doc template may help you

- ➤ not Pintos, but Your Pintos

Welcome to the World of Operating System

Q : Why Pintos ?

You will learn by Write The Code

- ➢ maybe your first time writing 2000+ LOC

- ➢ tricky multi-threading synchronization

- ➢ test-driven development

Welcome to the World of Operating System

Q : Why Pintos ?
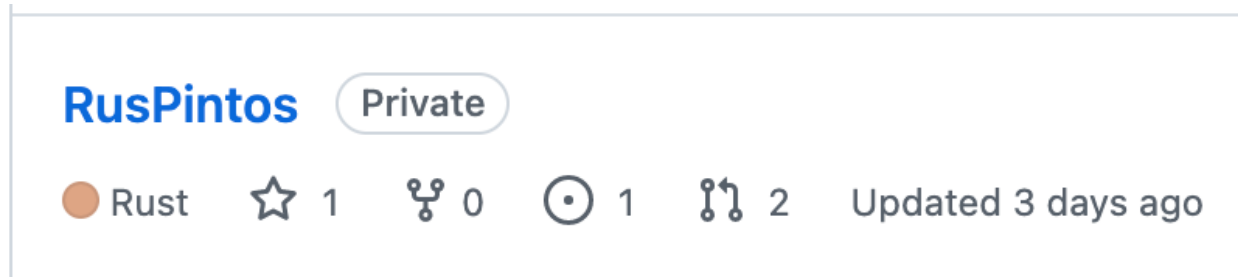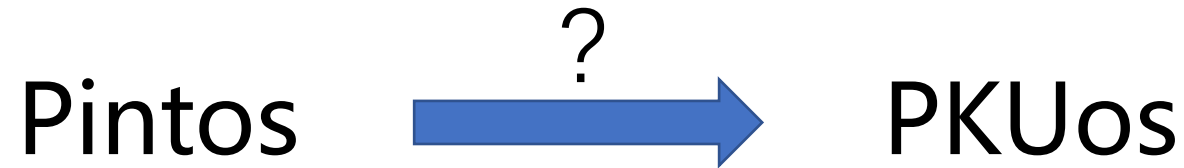
You w... ...ode

➢ You will

➢ start ea...

深呼吸

FOOD GROUPS

PKUOS

Welcome to the World of Operating System

Q : Why not Pintos ?

➢ IA32 architecture : CISC ISA, historical legacy

Pintos $\xrightarrow{\text{?}}$ PKUos

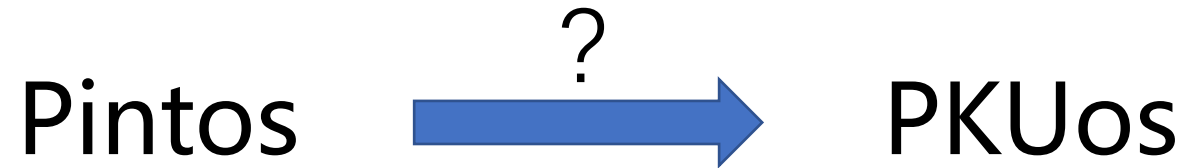**RusPintos** （Private）

🔴 Rust   ☆ 1   ⑂ 0   ⊙ 1   ⫝̸ 2   Updated 3 days ago

Pintos reimplemented in Rust

based on RISCV64.

Q : Why not Pintos ?

➢ IA32 architecture : CISC ISA, historical legacy

Pintos ──?──▶ PKUos
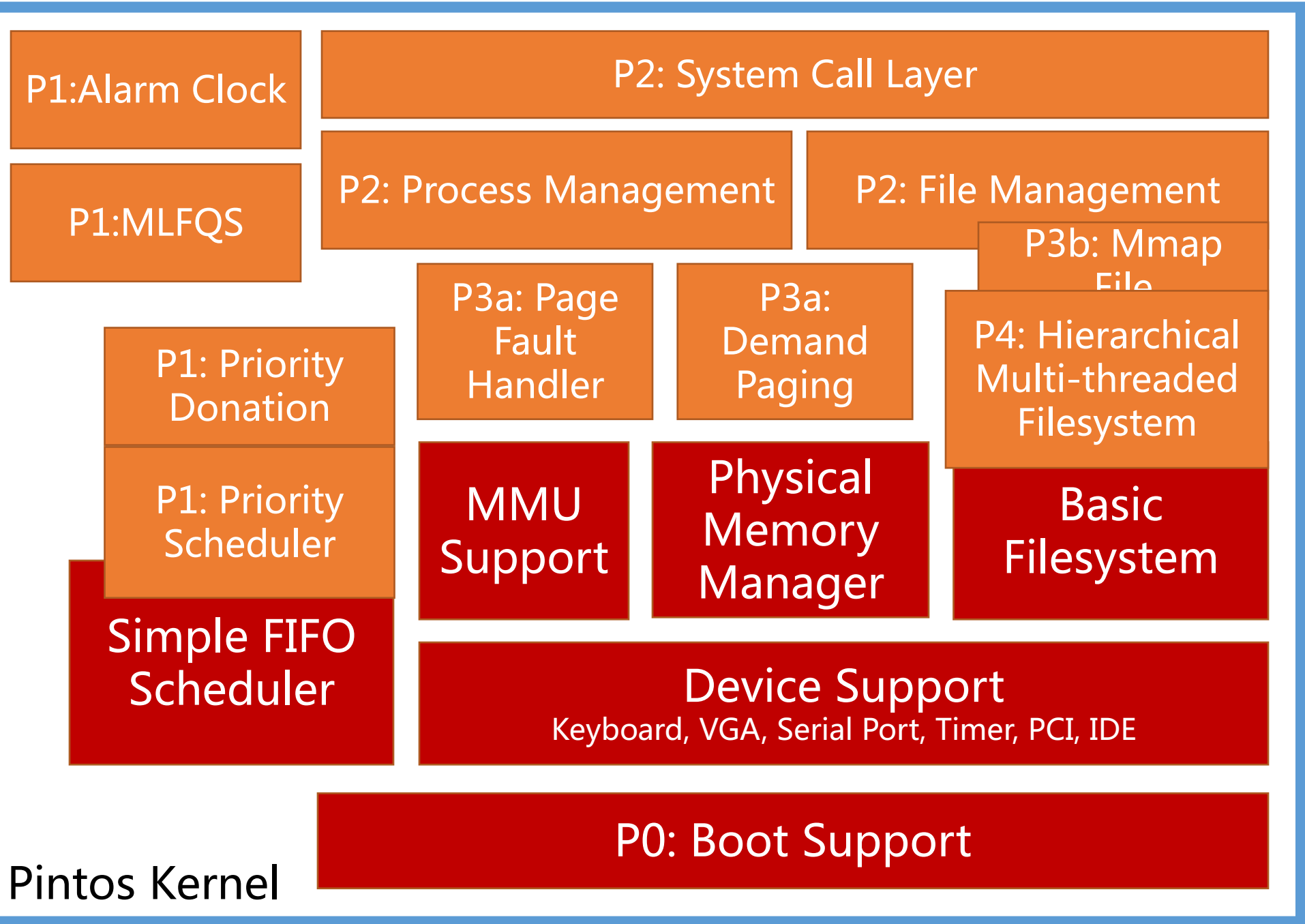
➢ time consuming : 100 hours +++

optional lab4, long long long lab document, per-lab TA session

Welcome to the World of Operating System

Q : So ... what will you do?

Typical workflow:

Lab released
on the Course Website

Read through the lab document

TA session

Read through the lab document
and design doc carefully

Design your data structures
and interfaces

Write and Debug code
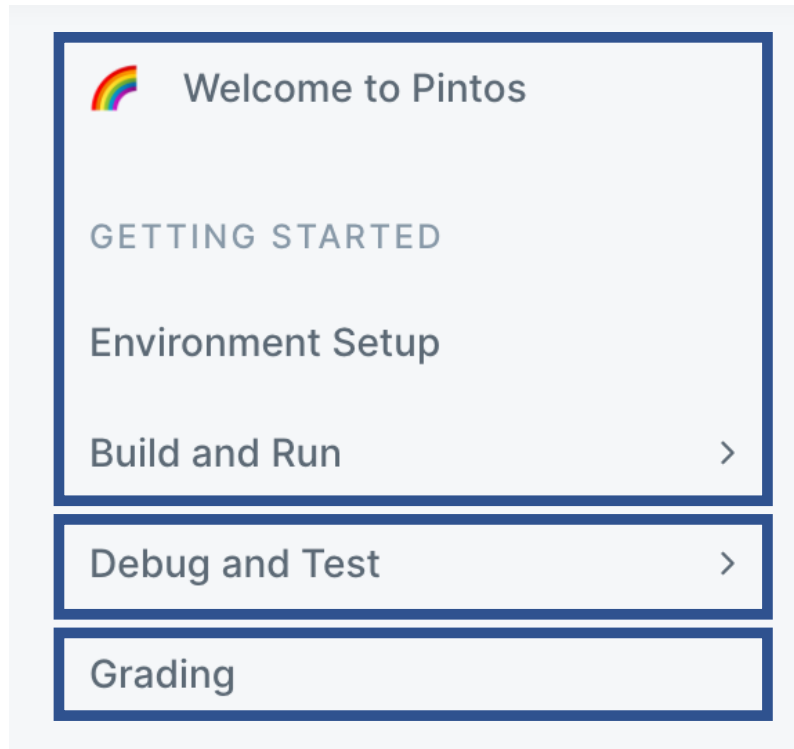Pass all the test cases
Submit code zip before code DDL

Answer the questions in design doc
submit it before design doc DDL

Q :How to survive? PintosBook long, but helpful

Set up you local development environment.

Look through it and look back if needed.

Important, read it carefully.

Q :How to survive? PintosBook

PROJECT DESCRIPTION

Lab0: Getting Real	>
Lab1: Threads	>
Lab2: User Programs	>
Lab3a: Demand Paging	>
Lab3b: Mmap Files	>
(Optional) Lab4: File Systems	>

Look through it before each TA Session.

Read it carefully during implementation.

Optional but rewarding Lab4.

PKUOS

Welcome to the World of Operating System

Q :How to survive? PintosBook

PKUOS - Pintos

Pintos source browser for PKU Operating System course

| **Main Page** | **Data Structures** ▾ | **Files** ▾ |

File List

Here is a list of all files with brief descriptions:

▼ 📁 src
  ▶ 📁 devices
  ▶ 📁 examples
  ▶ 📁 filesys
  ▶ 📁 lib
  ▶ 📁 tests
  ▶ 📁 threads
  ▶ 📁 userprog
  ▶ 📁 utils

cts going.

ters.

[IntrList].

t List, 2000.

PKUOS

Welcome to the World of Operating System

Q :How to survive? Your kind TA fat, but helpful

Learn to ask questions.

Do not be shy, ask in class, in office hour or in the Piazza.

PKUOS

Welcome to the World of Operating System

But … … your TAs are not your personal assistants.

- "My program crashed."

- "What does this error mean?"

- "I failed xxx testcase."

- "My computer can not boot."

Think twice, Ask once.

> ➤ [How to ask questions the smart way.](#)

> ➤ RTFM (Read The Fucking Manual)

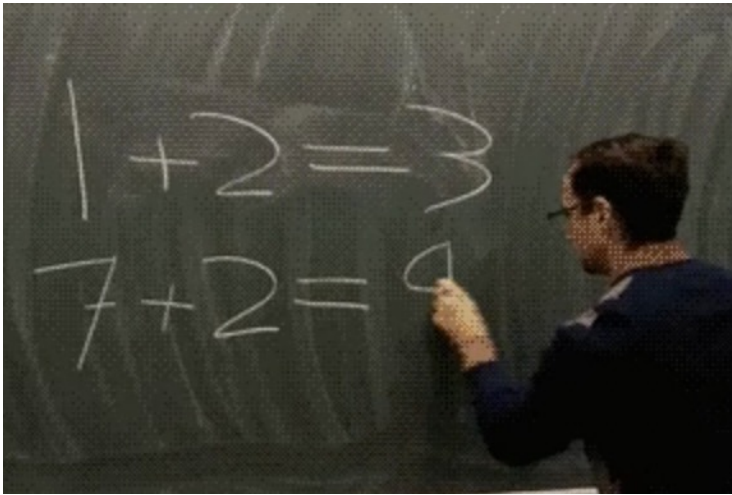> ➤ STFW (Search The Fucking Web)

Think twice, Ask once.

➢ "I encounter xxx under xxx condition."

➢ "Google says xxx, StackOverflow says xxx, Document says xxx, but yyy."

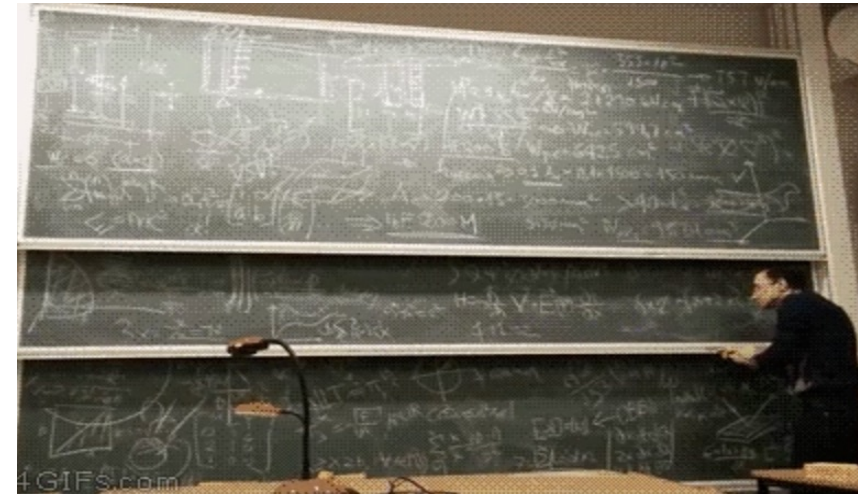➢ "Hey, fat TA, I found xxx and I think you do not know about it !"

Q :How to survive? Good habits awkward, but helpful

Use Version Control tool —— Git

Newly written code

A week later

The same code

How to write good commit message.

Q :How to survive?   Good habits

Write concise but good comments.

> ➤ Summarize the function in one sentence first.

> ➤ Pre-condition: input constraints (You may ASSERT these constraints)

> ➤ Post-condition: return value, exception (kernel panic)

missed comments (you can only omit the comment if the code is self-explained)

2 each, up to 10

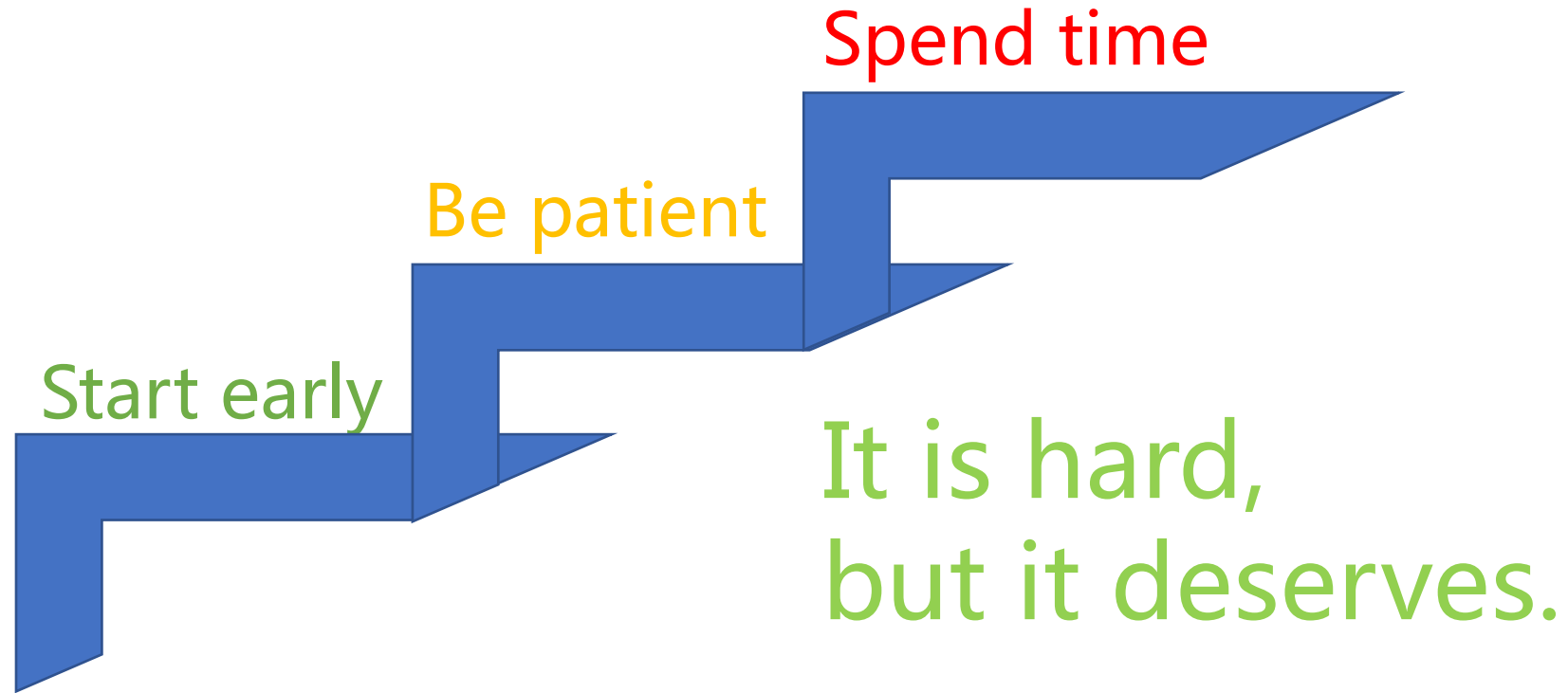Welcome to the World of Operating System

Q :How to survive?   Good habits

Module and Abstraction.

➢ A function should (only) do one thing   clean

➢ A function more than 100 LOC   warning

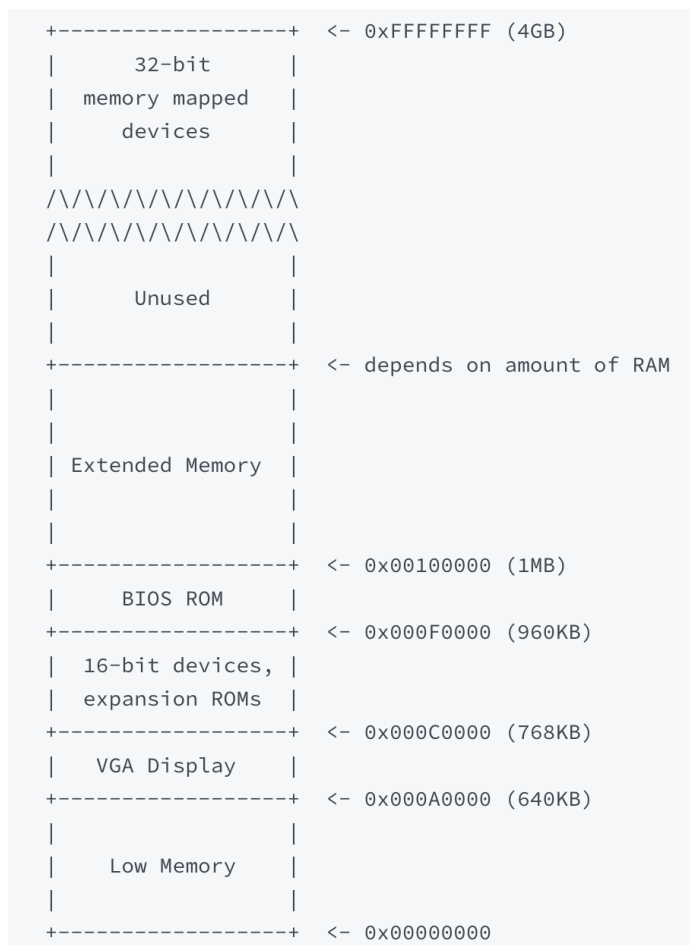➢ A function more than 200 LOC   Something may go wrong

Welcome to the World of Operating System

Q :How to survive?

Spend time

Be patient

Start early

It is hard,
but it deserves.

PKUOS

Welcome to the World of Operating System

Lab0 FAQs

Booting Pintos

Physical Address Space

```
+------------------+ <- 0xFFFFFFFF (4GB)
|    32-bit        |
|  memory mapped   |
|    devices       |
|                  |
/\/\/\/\/\/\/\/\/\/\
/\/\/\/\/\/\/\/\/\/\
|                  |
|    Unused        |
|                  |
+------------------+ <- depends on amount of RAM
|                  |
|                  |
| Extended Memory  |
|                  |
|                  |
+------------------+ <- 0x00100000 (1MB)
|    BIOS ROM      |
+------------------+ <- 0x000F0000 (960KB)
| 16-bit devices,  |
| expansion ROMs   |
+------------------+ <- 0x000C0000 (768KB)
|   VGA Display    |
+------------------+ <- 0x000A0000 (640KB)
|                  |
|   Low Memory     |
|                  |
+------------------+ <- 0x00000000
```

## 4GB physical address space == 4GB RAM ?

```
Pintos hda1
Loading...........
Kernel command line:
Pintos booting with 3,968 kB RAM...
367 pages available in kernel pool.
367 pages available in user pool.
Calibrating timer...  32,716,800 loops/s.
Boot complete.
```

You can even set the RAM size in pintos options.

Booting Pintos

This MBR code is usually referred to as a boot loader.

Basic MBR Disk

Physical Address Space
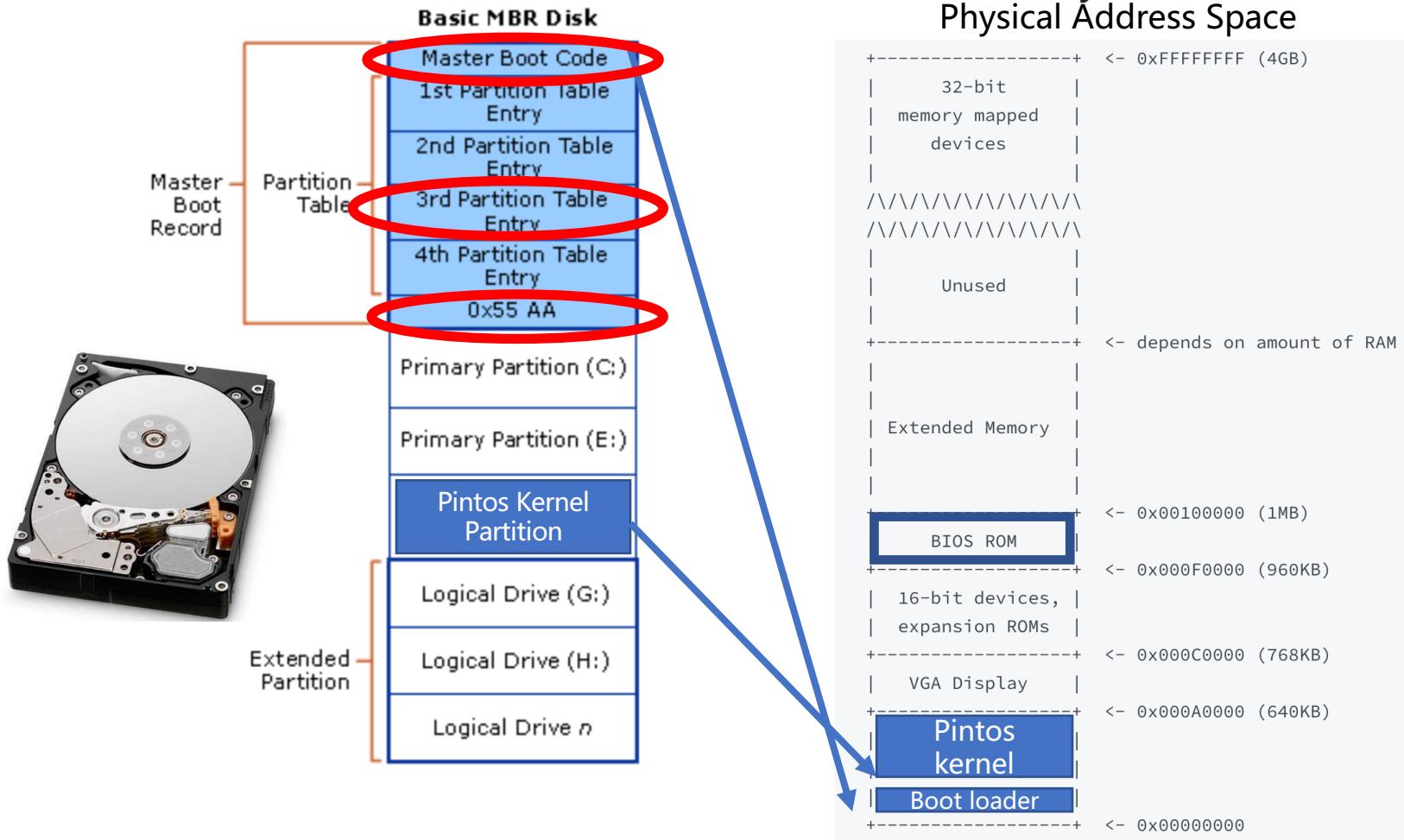
Hard-wired by the hardware

The real-world booting process can be much more **complicated**

GRUB, UEFI, … …

PKUOS

Welcome to the World of Operating System

X86 Mode (history legacy)

X86 Real Mode

Enabled in start.S →

X86 Protected Mode

- 16-bit Instructions and Registers

    AX, BX, CX, DX, SI, DI, BP, SP

- 20-bit Memory Address Space (Up to 1MB)

    16-bit segment registers

    CS, DS, SS, ES, FS, GS

    PAddr = SEG << 4 + Operand

- 32-bit Instructions and Registers

    EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP

- 32-bit Memory Address Space (Up to 4GB)

    Reserved segment registers, but for protection

    Address translation enabled

Conclusion

➢ Why Pintos?

- Design and Implementation
- Read, Design, Write, Debug the code

➢ What will you do in the projects?

- Projects Map
- Typical workflow

➢ How to survive the projects?

- PintosBook
- Ask questions
- Good habits
- Good attitude

➢ Lab0 FAQs: Booting Pintos, X86 mode