Pintos Overview

—— Why, What and How

TA Session

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

Some announcements:

- Lab 0 Code will due next Thursday 11:59 pm
- Lab 0 Design Doc will due next Sunday 11:59 pm
- We have updated the Lab0 Design Doc on Tuesday (removed two questions)

Educational OS Project Zoo

JOS

IA32
MIT6.828

xv6

RISCV32
MIT6.S081

Nachos

MIPS
Old CS162

Pintos

IA32
CS162

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

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

Q : Why Pintos ?

You will learn by Write The Code

- maybe your first time writing 2000+L C code
- tricky multi-threading synchronization
- test-driven development

Q : Why Pintos ?

You will learn by Debug The Code

- You will live in the GDB
- start early, start early, start early

PKUOS

Welcome to the World of Operating System

Q : Why not Pintos ?

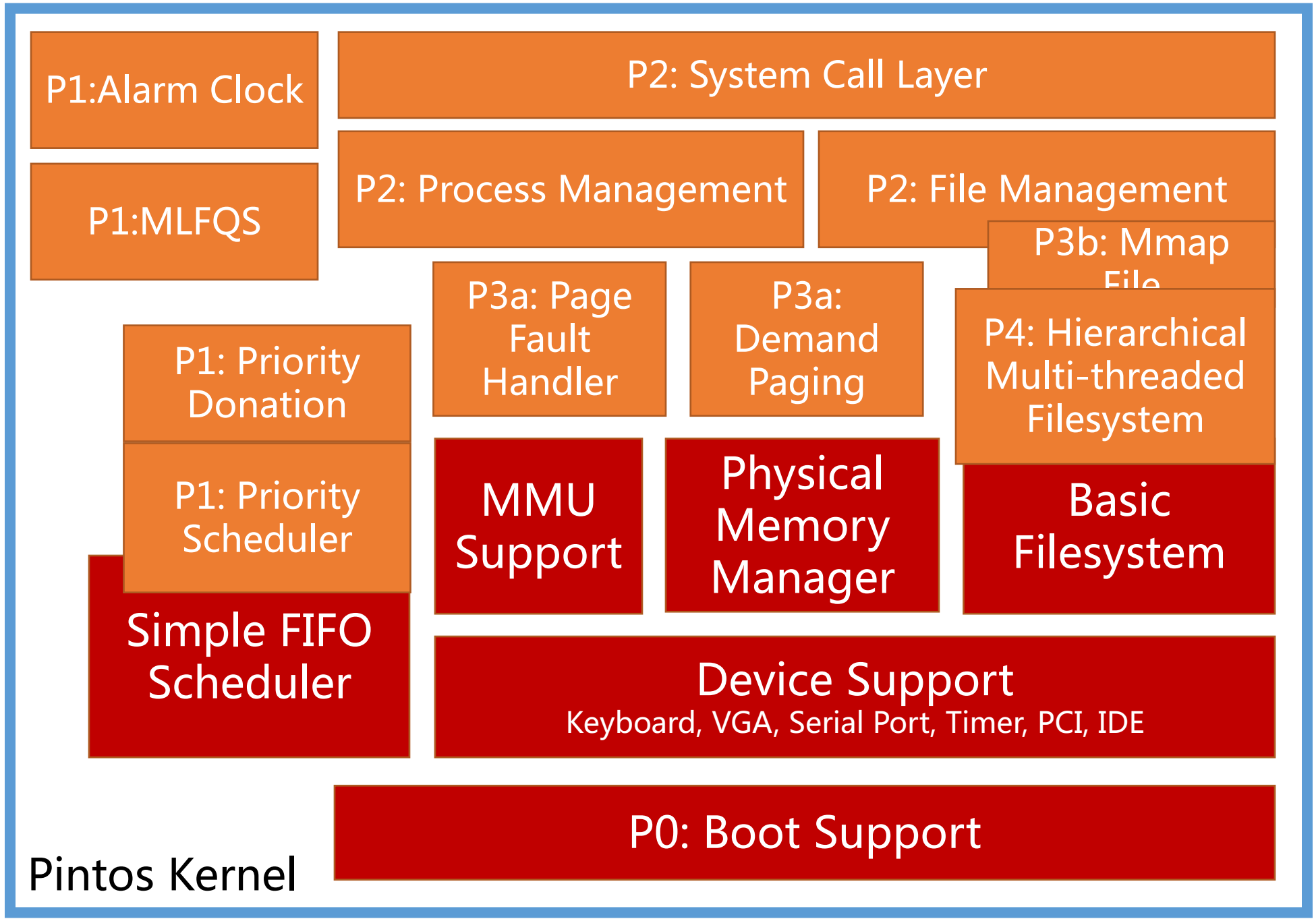
- IA32 architecture : CISC ISA, historical legacy

Pintos ? PKUos

- time consuming : 100 hours +++

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

Q : So … what will I do?



P0: Getting Real

P1: Threading

P2: User Programs

P3: Virtual Memory

P4: File System

Students Create

Support Code

Typical workflow:

Labx released
on the PKU Course Website

Read through the lab document

TA session

Read through the lab document
and design doc carefully

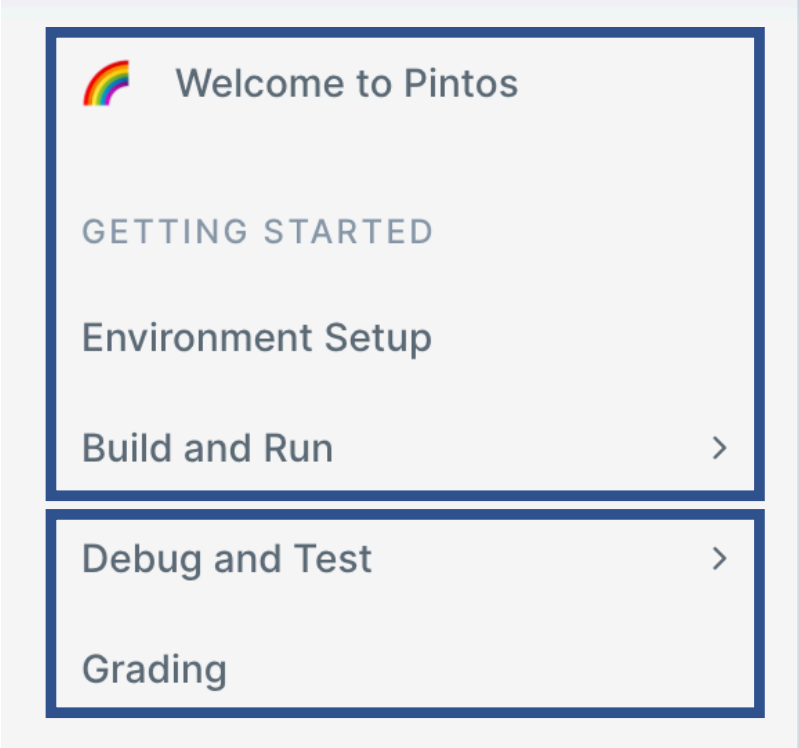
Design your data structures
and interfaces

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



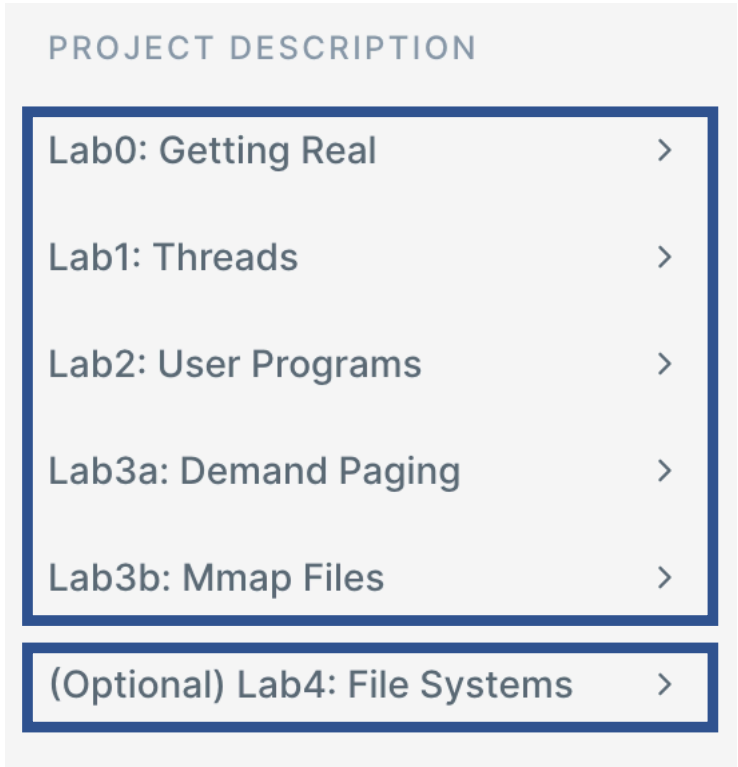
You have read. If not, get hurry!

You have read, but not fully understand.

PKUOS

Welcome to the World of Operating System

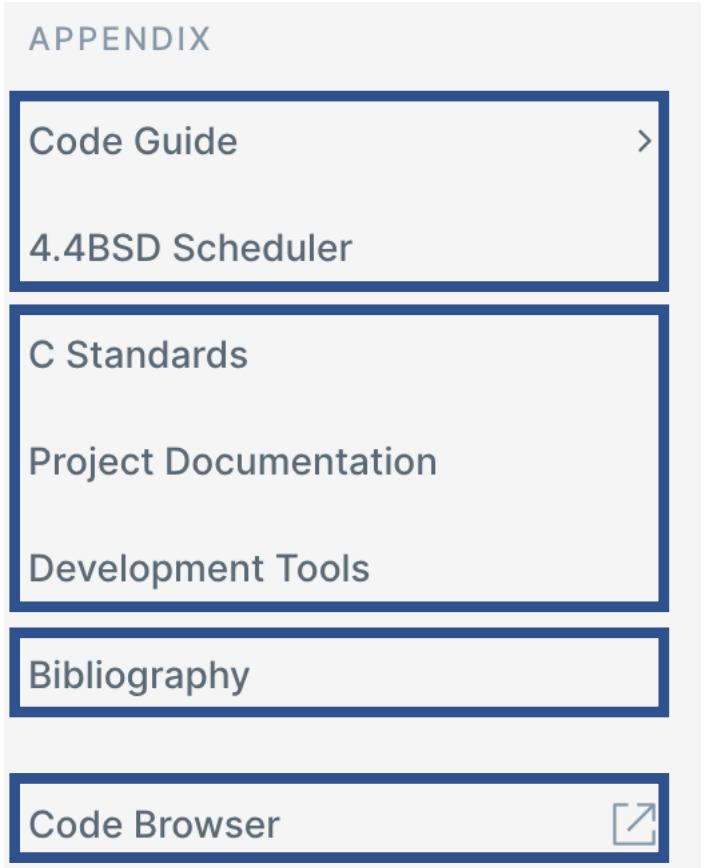
Q :How to survive? PintosBook



Look through before TA Session.
Read carefully during implementation.

Optional but rewarding Lab4.

Q :How to survive? PintosBook



Read when needed as the projects going.

Referenced in the previous chapters.

```
# Notify BIOS that boot failed.  See [IntrList].
int $0x18
```

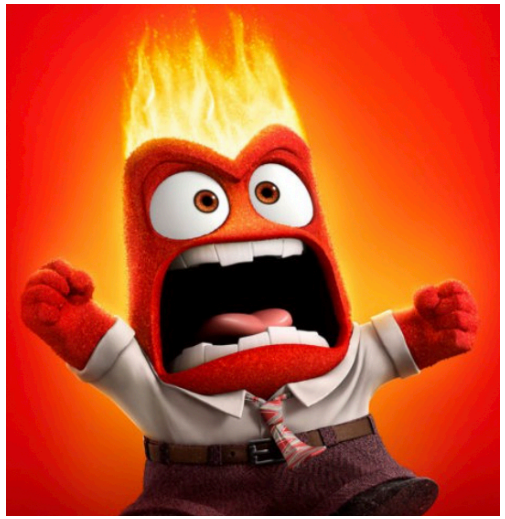
[IntrList]. R. Brown, Ralf Brown's Interrupt List, 2000.

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

Learn to ask questions.

Do not be shy, ask in the WeChat group, in class or personally.

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 !”

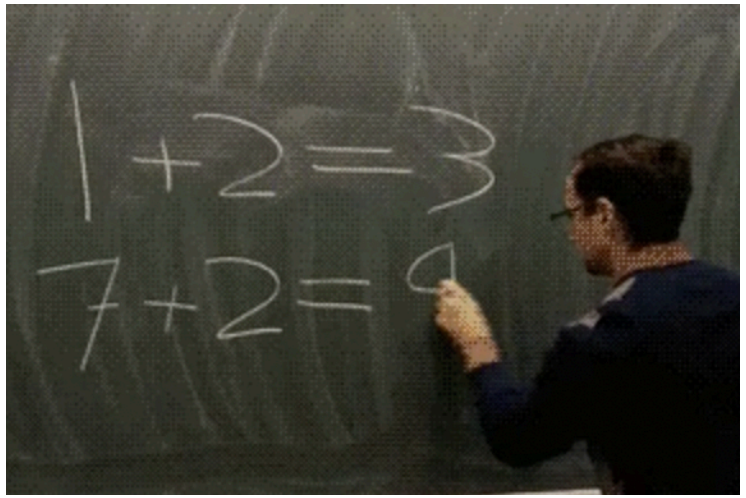
PKUOS

Welcome to the World of Operating System

Q :How to survive?

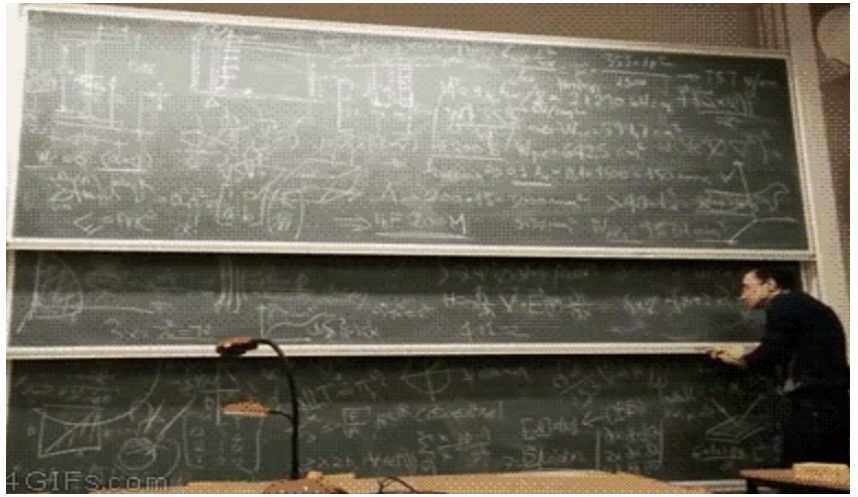
Good habits awkward, but helpful

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)

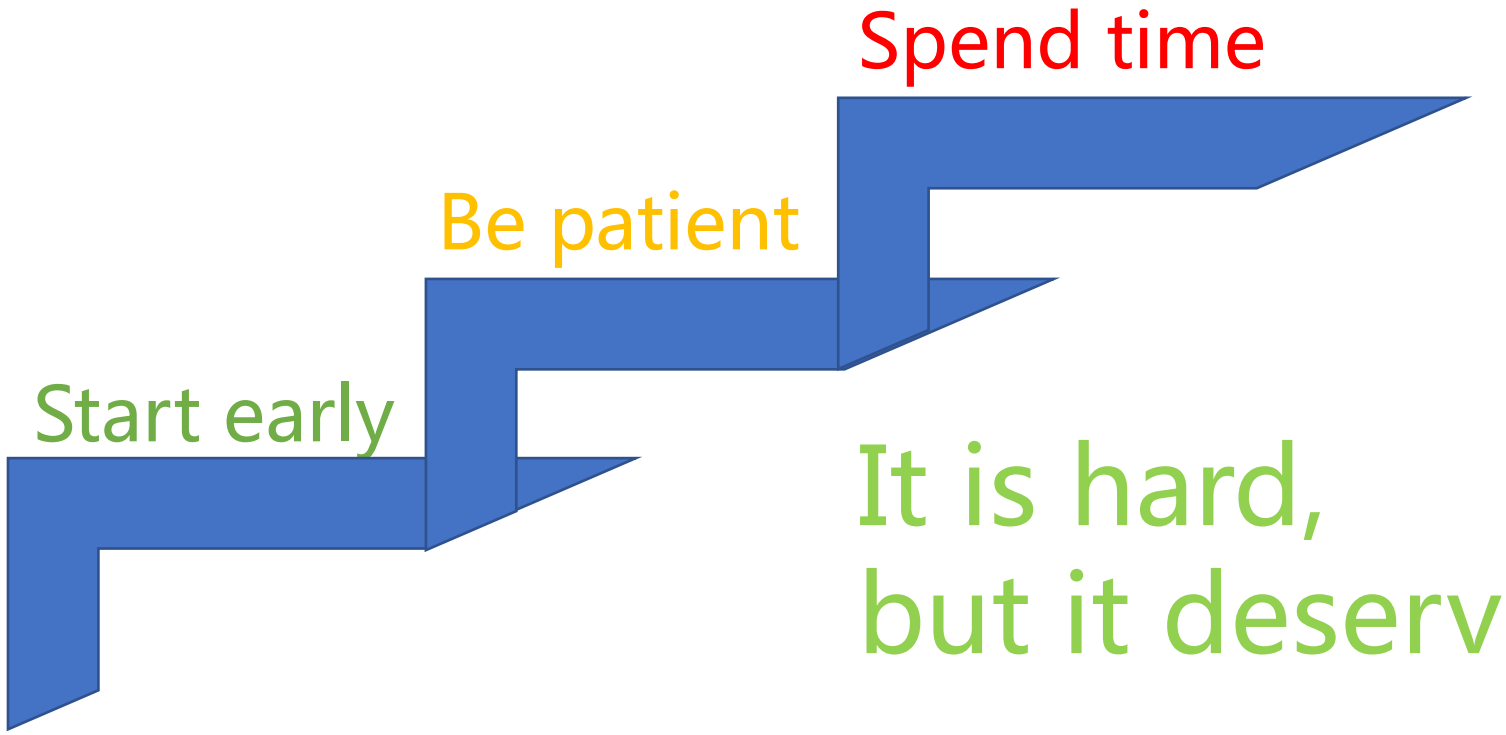
2 each, up to 10

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

Q :How to survive?

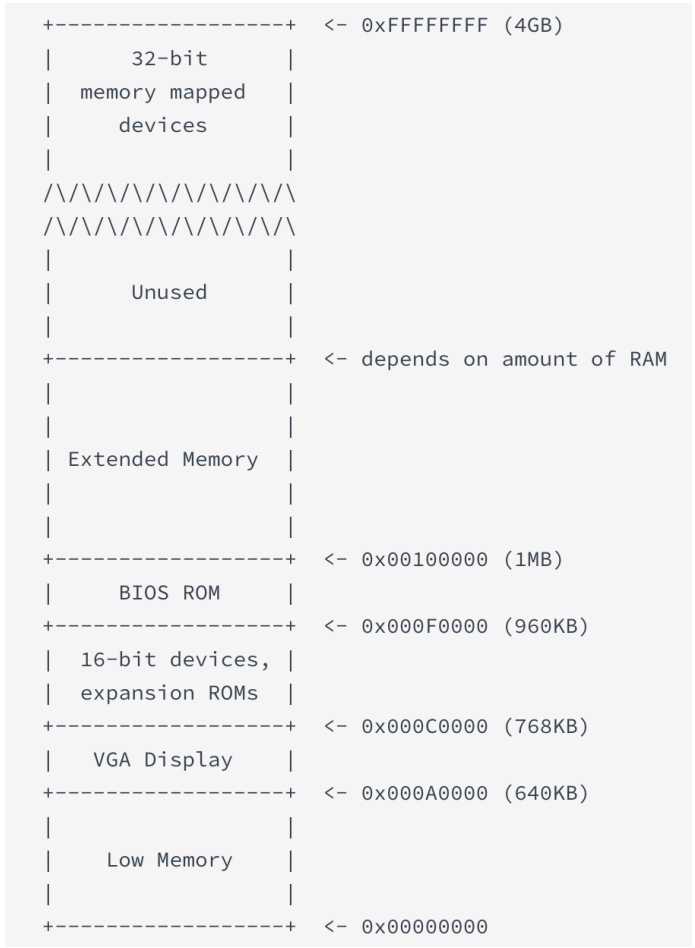


It is hard,
but it deserves.

Lab0 FAQs

Booting Pintos

Physical Address Space



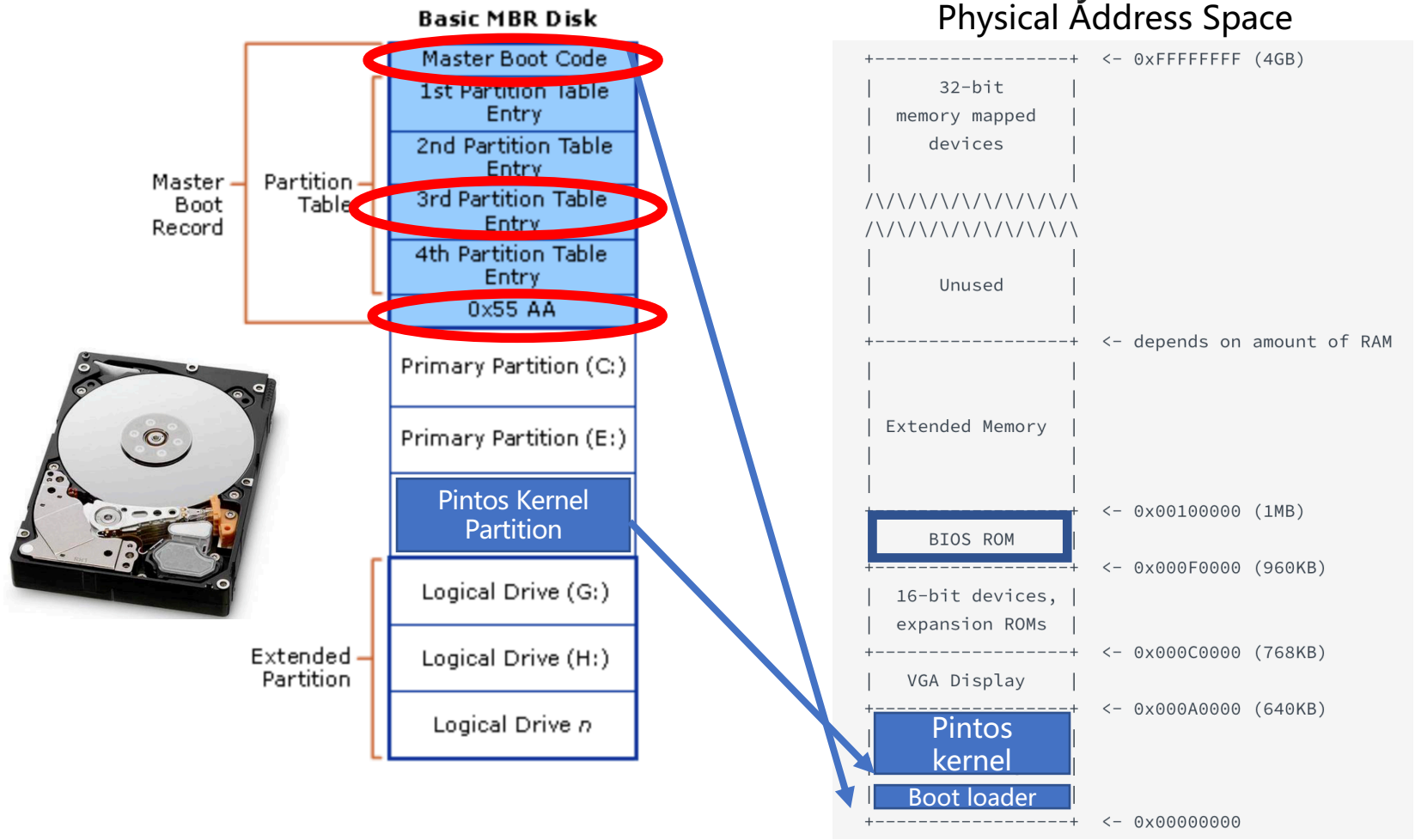
4GB physical memory == 4GB RAM ?

```
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.



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

Any Questions ?

https://www.wjx.cn/vj/hhnJxie.aspx

PKUOS

Welcome to the World of Operating System