

Pintos Overview



— Why, What and How

TA Session



TA: 吴永彤 <https://jokerwyt.github.io>
wuyongtong@stu.pku.edu.cn

PPT Credit: 向昱行 仲殷旻

Deadlines

Two deadlines: code dealines & design doc deadlines.
Check <https://pku-os.github.io//sp26/> frequently.

1 week for lab0

2 week for lab1,2,3a,3b



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

➤ Talk is cheap, show me the code

➤ Your design matters !!

➤ 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**



Q : Why Pintos ?

You will learn by **Write The Code**

- maybe your first time writing 2000+ LOC
- tricky multi-threading synchronization
- test-driven development



Me, debugging

Q : Why

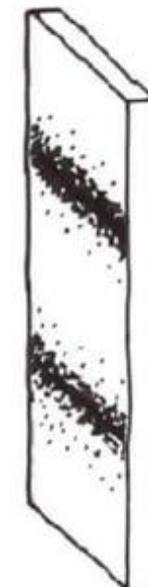
You will

➤ You will

➤ start early



Code



Welcome to

Q : Why **not** Pintos ?

- IA32 architecture : CISC ISA, historical legacy

Pintos  Tacos



TACOS
RUST OS NEVER GETS RUSTY.

Pintos reimplemented **in Rust**
based on **RISCV64**.



AKA OS
Welcome to the World of Operating System

Q : Why **not** Pintos ?

- IA32 architecture : CISC ISA, historical legacy

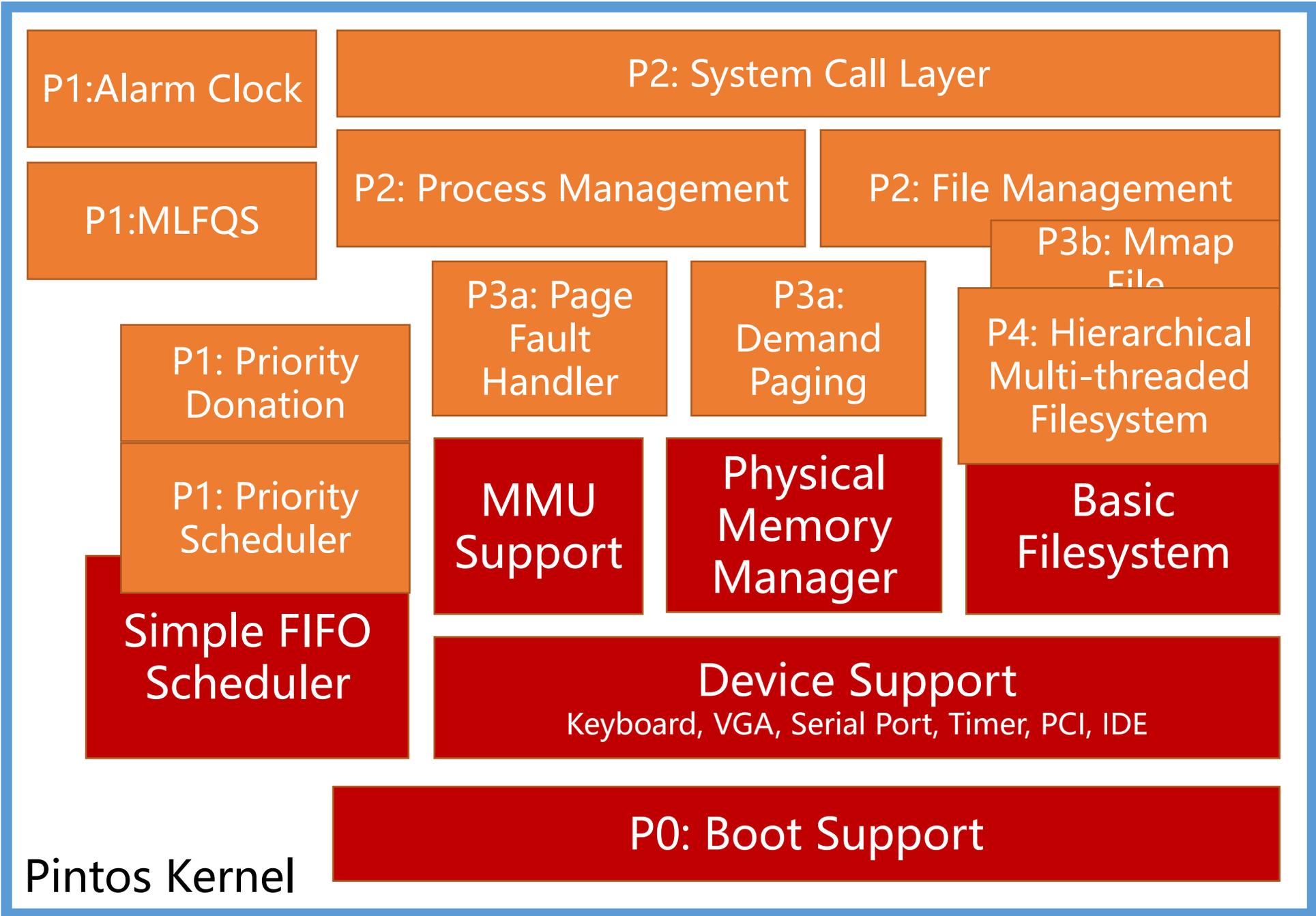
Pintos  Tacos

- time consuming : 100 hours + + +

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



Q : So ... what will you do?



- P0: Getting Real
- P1: Threading
- P2: User Programs
- P3: Virtual Memory
- P4: File System

Students Create

Support Code

Typical workflow:

Lab released
on the Course Website



Read through the lab document



TA session

We are here for Lab 0!

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

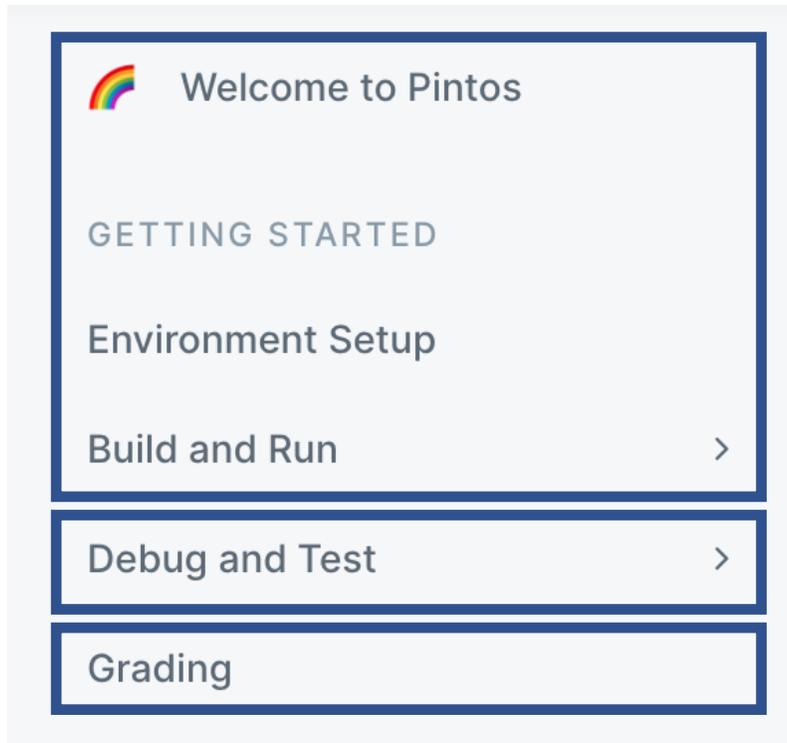


Answer the questions in design doc
submit it before design doc DDL



Q :How to survive?

PintosBook long, but helpful



Set up your 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.



Q: How to survive? PintosBook

APPENDIX

- Code Guide >
- 4.4BSD Scheduler
- C Standards
- Project Documentation
- Development Tools
- Bibliography
- Code Browser ↗

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	
▶ filesystem	
▶ lib	
▶ tests	
▶ threads	
▶ userprog	
▶ utils	



ts going.

ters.

e [IntrList].

t List, 2000.

Q :How to survive? Your kind TA

Learn to ask questions.

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

Office hour: Friday 10:30 a.m -11:30 a.m, Yan Yuan Building 818

Piazza: check <https://pku-os.github.io//sp26/>



But your TAs are not your ~~personal assistants~~ (or ~~Mr.Deepseek~~).



- "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 Friendly Manual)
- STFW (Search The Friendly Web)
- AI is helpful for reading docs.



Think twice, Ask once.



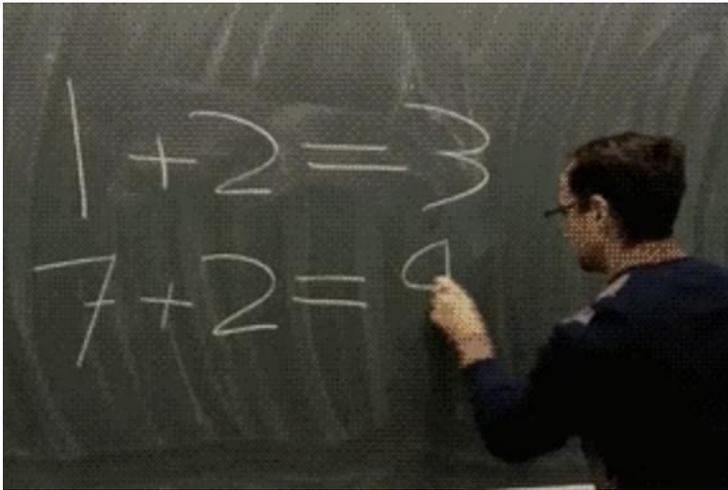
- "I encounter xxx under xxx condition."
- "Google says xxx, StackOverflow says xxx, Document says xxx, but yyy."
- "Hey, TA, I found xxx and I think you do not know about it !"



Q :How to survive?

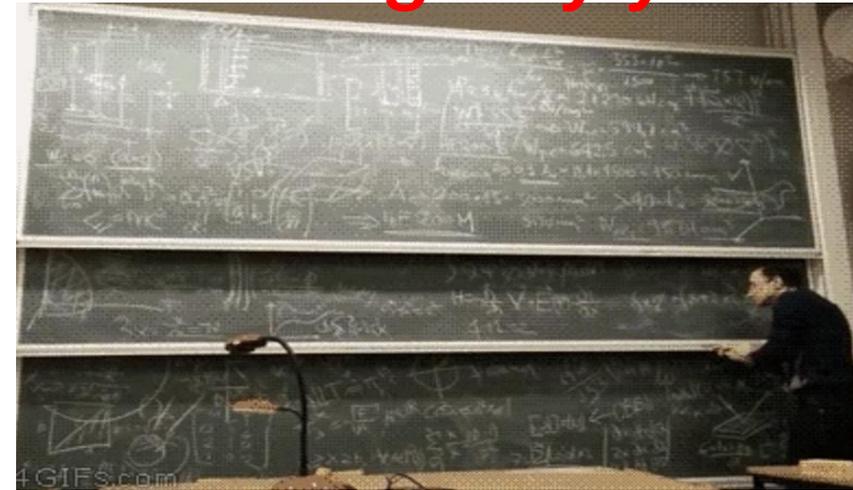
Good habits *awkward*, but *helpful*

Use **Version Control** tool — **Git**
Learn git by yourself!



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)



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



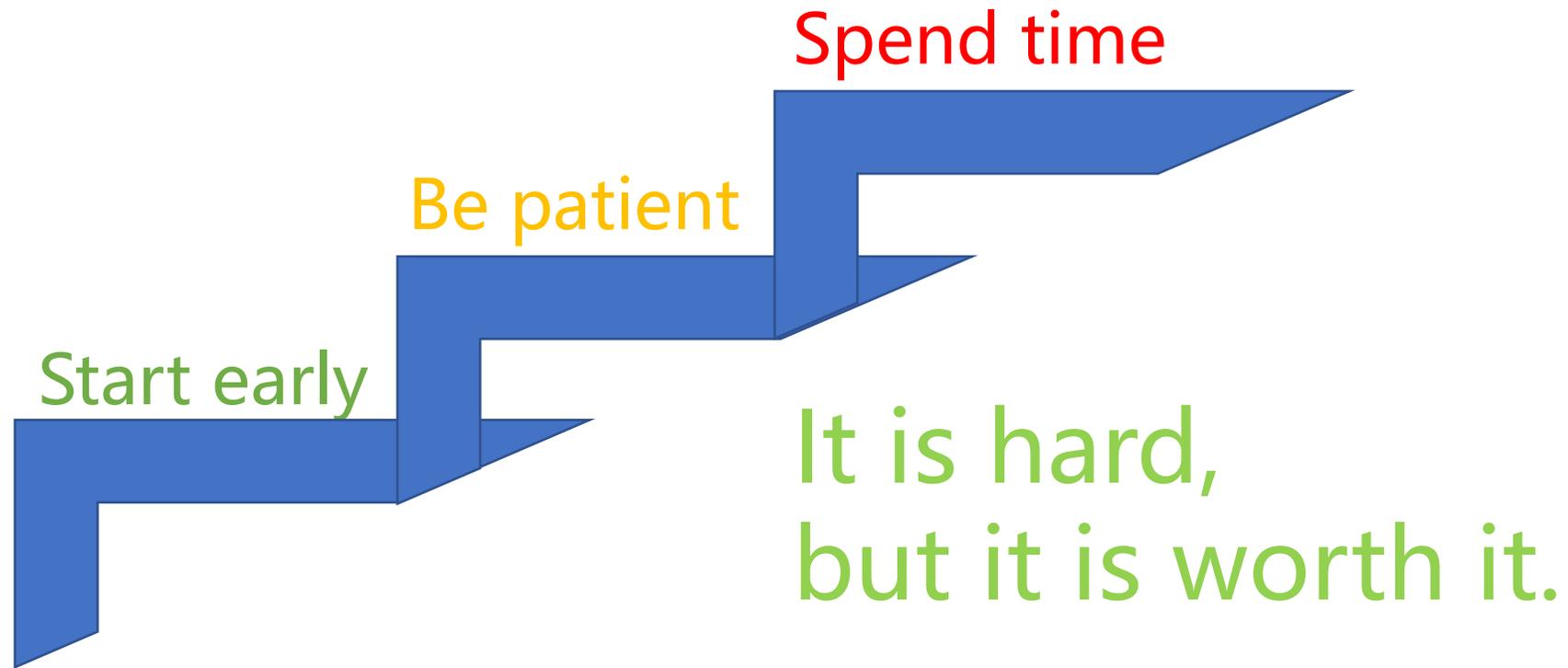
Q :How to survive?

Software Engineering Good Practice

Principles:

- Is your design & code self-explained for your classmates?
 - If not, improve it.
 - Do **NOT** add massive comments to rescue.
- Correctness > simplicity >> performance
- Clean mechanism & interface, fewer preconditions.
- Some bad examples:
 - 调用这个函数之前必须调用另一个函数
 - 存在无法从函数签名中推断的副作用

Q :How to survive?



Q : How about Claude Code

- You can use it.
- But at least design your OS by yourself!



Welcome to the World of Operating System

Lab0 FAQs



Welcome to the World of Operating System

In this Lab, you will be...

- Walking through the **booting** of Pintos
- Try your hands on **debugging** Pintos
- Write your **first line of code** in Pintos: a tiny shell

OS Booting sounds overwhelming?

- All essential information are provided in the PintosBook
- You don't need to master all details; you practice how you learn from new information!



Welcome to the World of Operating System

Tips

- [Use old VSCode:](#)
- https://code.visualstudio.com/updates/v1_98

Visual Studio Code ×

 容器未满足 VS Code 服务器的所有要求。

Warning: Missing GLIBC >= 2.28! from /lib/x86_64-linux-gnu/libc-2.27.so
Error: Missing required dependencies. Please refer to our FAQ
<https://aka.ms/vscode-remote/faq/old-linux> for additional information.

[了解详细信息](#) [关闭远程](#) [更多操作...](#)

Cutting through the **confusing** jargons...

Docker ENV

You will (assuming you use Docker for running Pintos):

- develop Pintos in a ubuntu:18.04 container (regardless of your host)
- cross-compile Pintos into i386 (Intel 80386) binaries (IA32, 4GB)
- execute Pintos with either QEMU or Bochs, kernel emulators
 - ...which come with the BIOS (Basic Input/Output System) firmware that loads a custom OS **bootloader**
 - ...which, as part of Pintos, locates and loads the actual **kernel**

Pintos

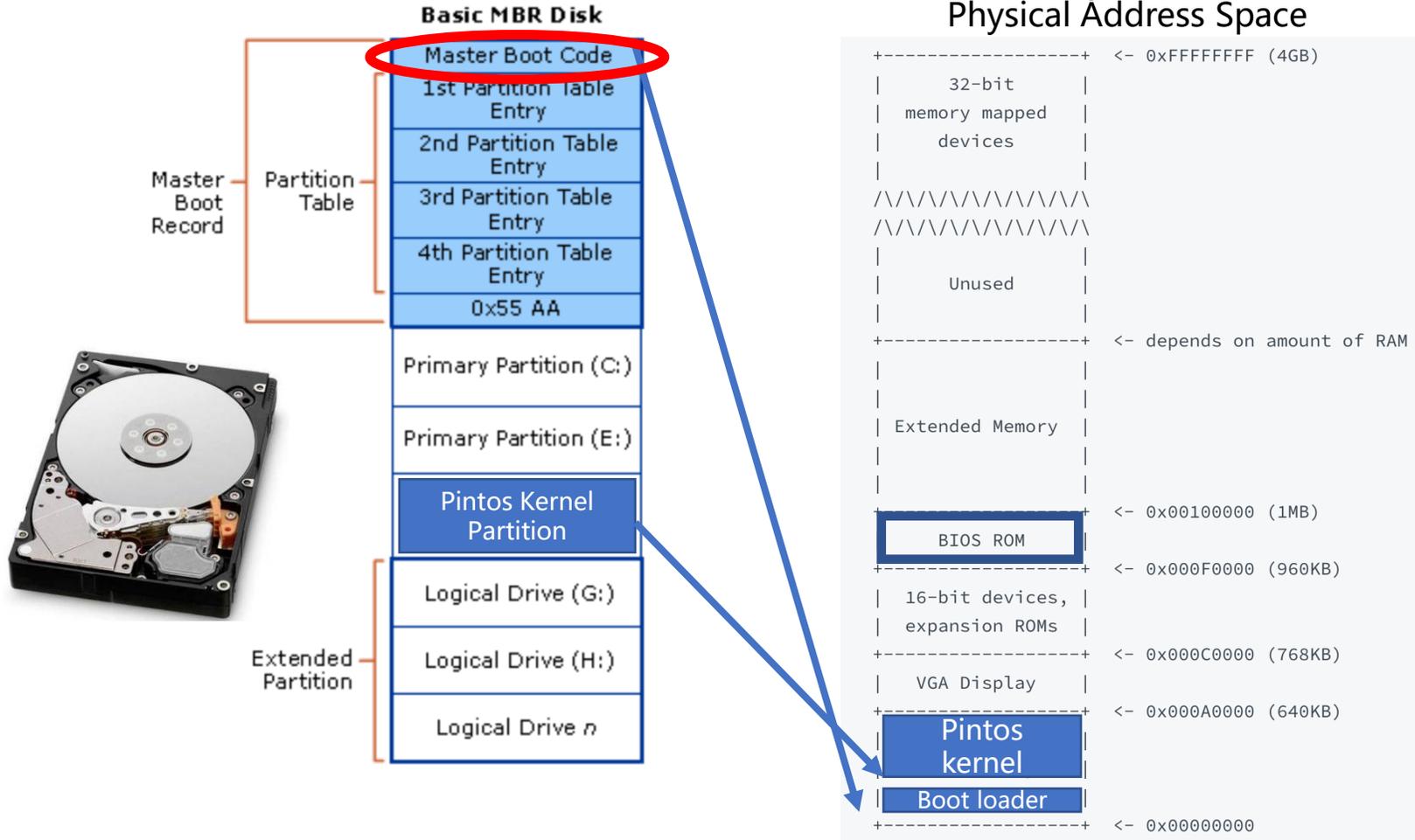
loader.S

start.S

- ...which is nothing but an i386 ELF executable, with .text, .data, and an entry point
 - ...which switches from real mode to protected mode, and calls pintos_init()

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 (GRand Unified Bootloader),
UEFI (Unified Extensible Firmware Interface), ...



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

$PAddr = SEG \ll 4 + \text{Operand}$

- 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



Welcome to the World of Operating System

And if you really want to know every detail...

- Dig through Makefiles for how the Bootloader (`loader.S`) and the kernel itself is linked and run
- `src/utlis/pintos` is how you will be running Pintos, which is actually a Perl script that you can try to read and understand



Conclusion

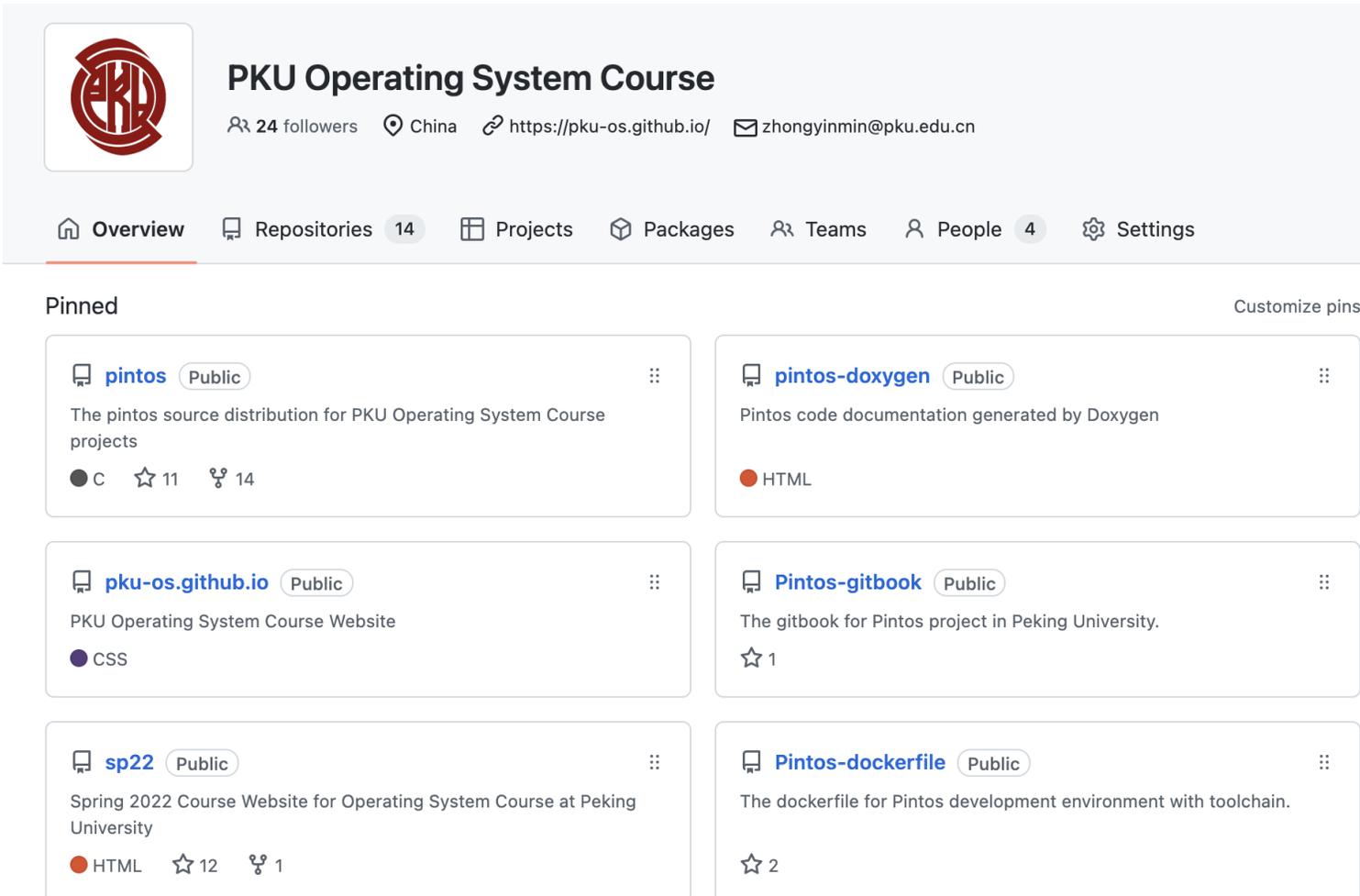
- Why Pintos?
 - Design and Implementation
 - Read, Design, Write, Debug the code
- What will you do in the projects?
 - Projects Map
 - Typical workflow
- How to survive the projects?
 - PintosBook
 - Ask questions
 - Good habits
 - Good attitude
- Lab0 FAQs: Booting Pintos, X86 mode





<https://github.com/PKU-OS>

Learn it,
Master it,
Love it,
and Join us.



PKU Operating System Course
24 followers China <https://pku-os.github.io/> zhongyinmin@pku.edu.cn

Overview Repositories 14 Projects Packages Teams People 4 Settings

Pinned Customize pins

- pintos** Public
The pintos source distribution for PKU Operating System Course projects
C 11 stars 14 forks
- pintos-doxygen** Public
Pintos code documentation generated by Doxygen
HTML
- pku-os.github.io** Public
PKU Operating System Course Website
CSS
- Pintos-gitbook** Public
The gitbook for Pintos project in Peking University.
1 star
- sp22** Public
Spring 2022 Course Website for Operating System Course at Peking University
HTML 12 stars 1 fork
- Pintos-dockerfile** Public
The dockerfile for Pintos development environment with toolchain.
2 stars