

# Neo0S

一个基于 *Rust* 的教学操作系统

学 校：南开大学  
指导教师：宫晓利  
队伍名称：能润就行

参赛成员：穆禹宸 袁贞芷 李潇逸  
选 题：PROJECT-0

# 目 录

---

CONTANTS

1

概述

2

问题分析

3

NeoOS简述

4

课程设计



# 概述

# 概述

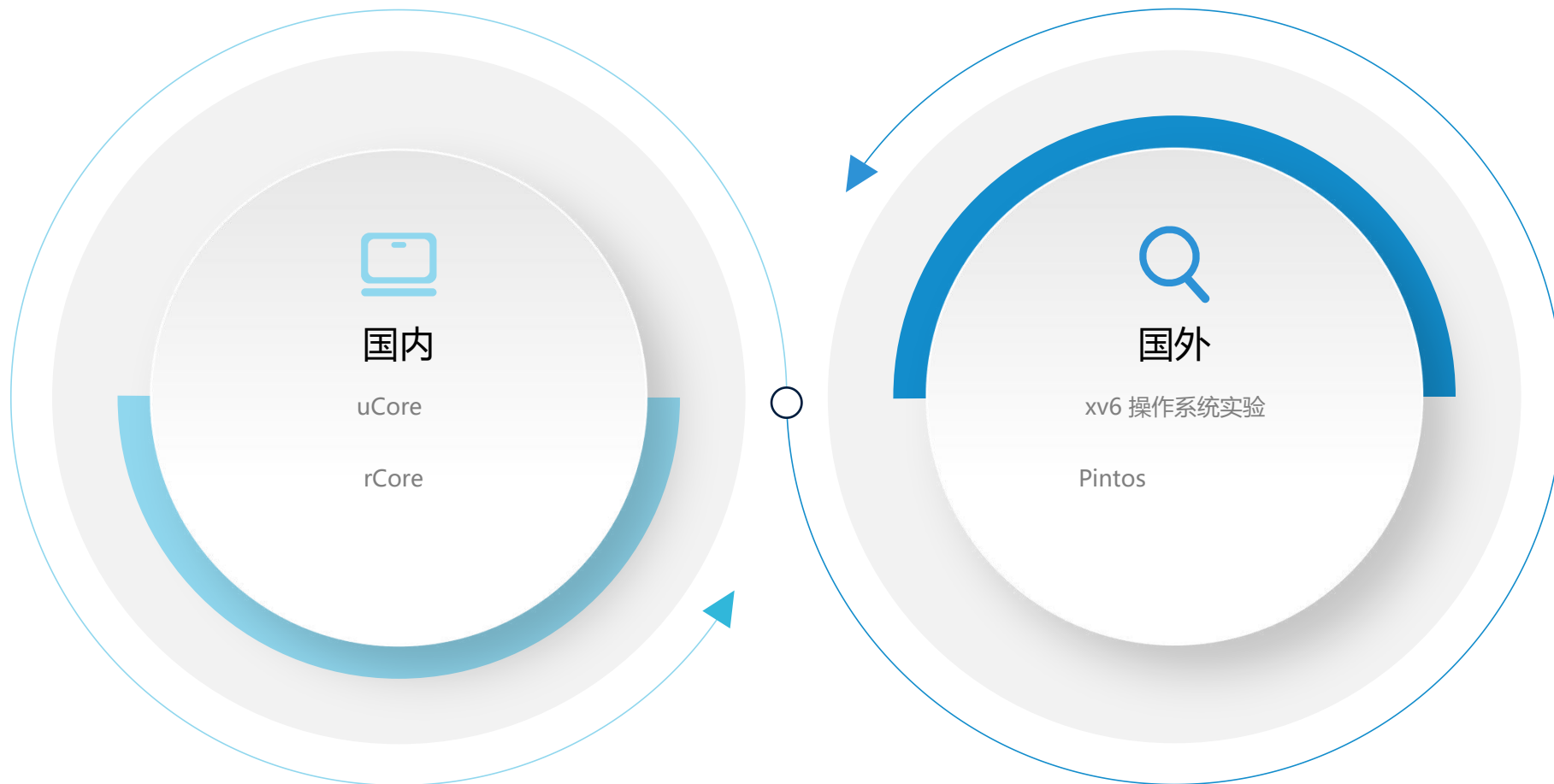
操作系统是计算机系统中最基础也是最重要的一个部分，其主要工作就是管理计算机系统的硬件资源、提供用户和应用程序的接口，以及提供各种服务。不论是对用户还是设计人员都是极为重要的。因此，操作系统是计算机科学领域中非常重要的一门课程。





# 问题分析

# 操作系统课程



着眼于操作系统整体功能设计与实现，尤其重视进程调度、内存分配等内容

# 问题

01

理论知识丰富，课程实践脱节

缺乏同步、文件系统、权限管理等方面讲解

**理论与实践不匹配**

02

无64位操作平台

仅使用legacy

较少多核CPU启动教学

**uCore过于陈旧**

03

出现bug难以定位

缺乏好用的包管理器

**C语言带来的内存问题**



# Neo0S 简述

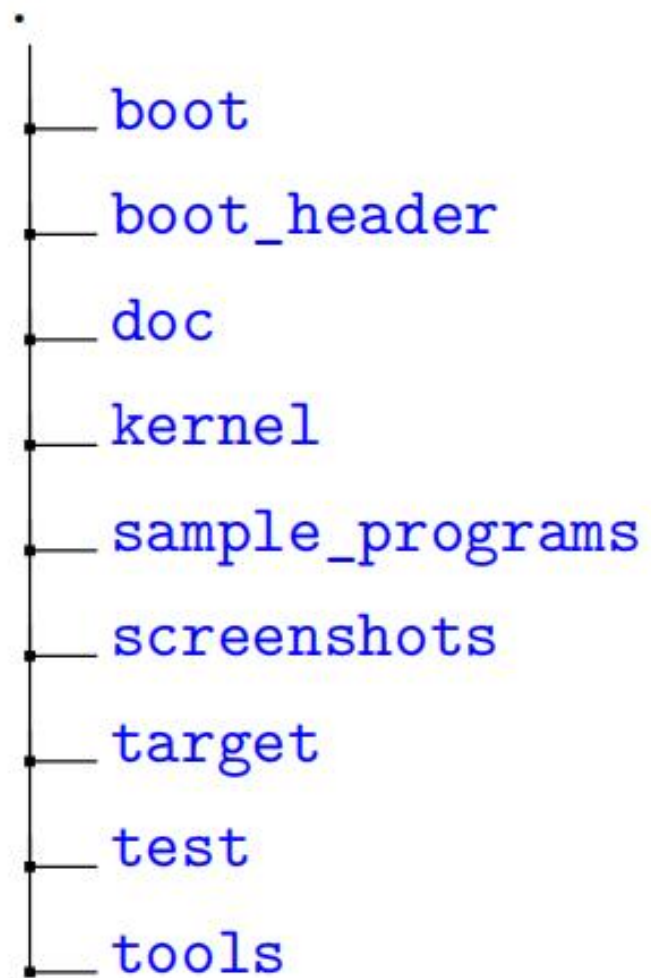


# NeoOS 概况

NeoOS 是一个操作系统课程实验项目，旨在为高校的计算机专业提供高效、可靠的操作系统实验。该项目的目标是设计和实现一个功能齐全的操作系统，具有虚拟内存管理、进程和线程管理、文件系统、网络栈等关键功能。通过参与 NeoOS 项目，学生们可以深入了解操作系统的内部工作原理，学习并应用各种操作系统核心概念和技术。



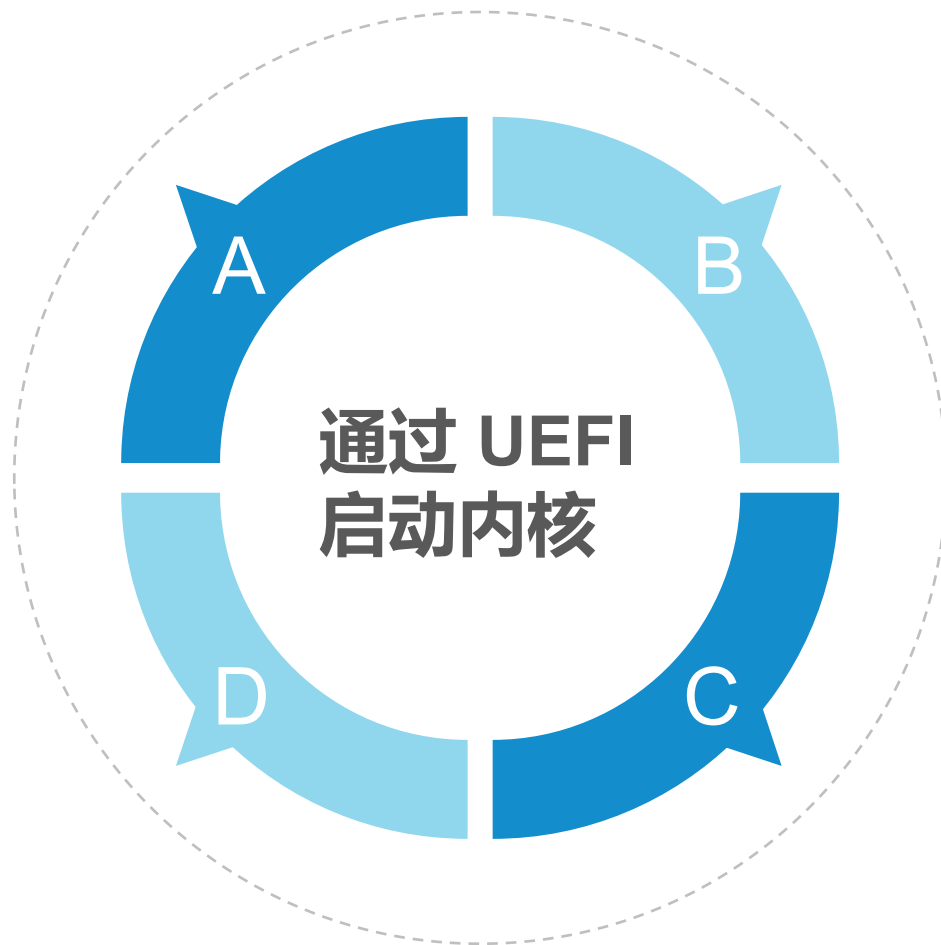
# 仓库布局



# BOOT

从磁盘读取内核镜像

读取物理内存信息，  
并构建页帧分配器



构建初步的页表

对内核各个 section、  
内核栈、BIOS 等物  
理地址进行映射

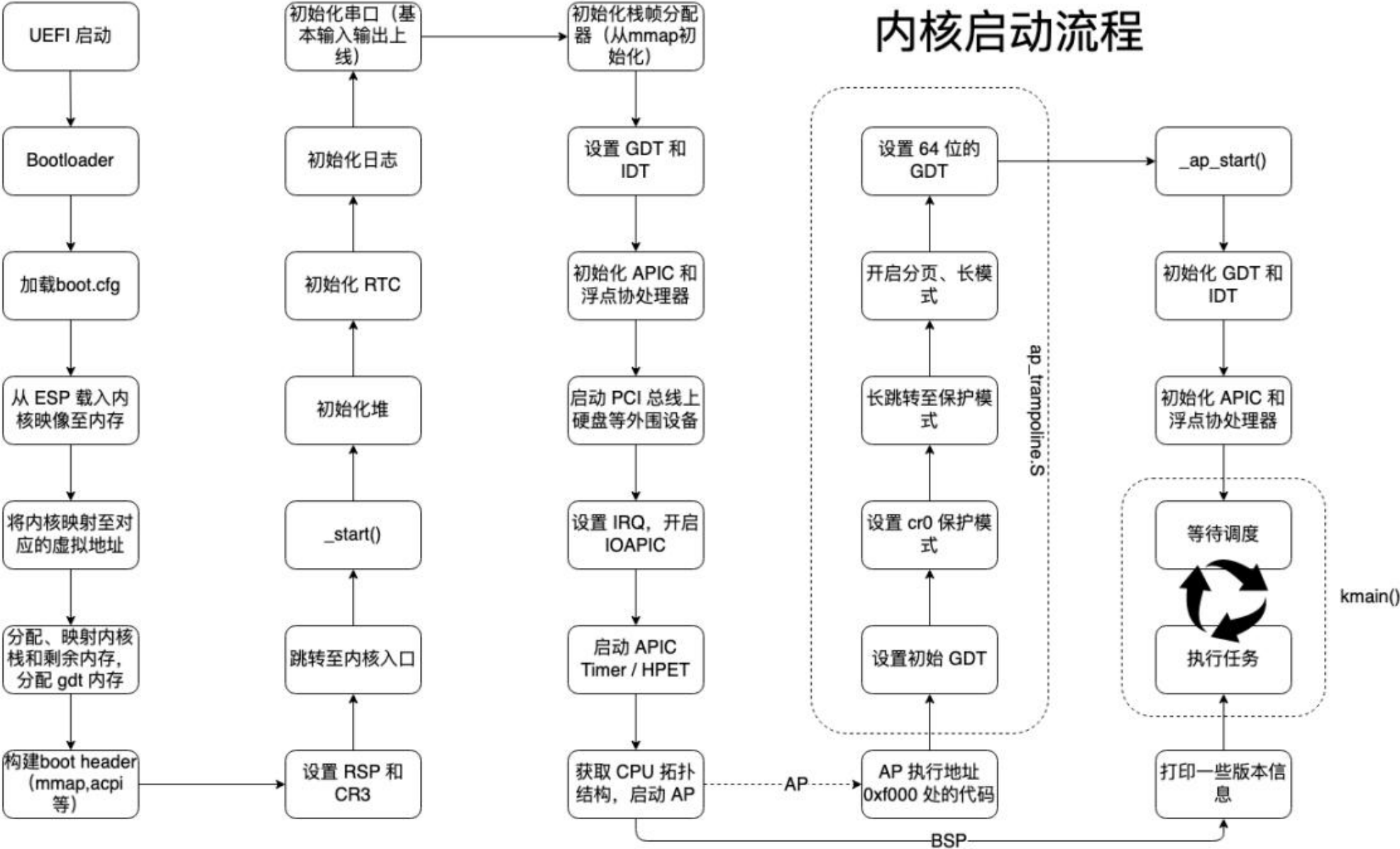
# KERNEL



NeoOS 的核心



# 内核启动流程



# KERNEL 功能

01

## 堆初始化

指定一块 bss 区域作为初始堆的起始地址，该区域通过 ld 脚本实现，该脚本制定了 bss 所在的虚拟地址。

02

## 物理内存初始化

将 bootloader 读取到的物理内存信息进行存储，之后将这些信息传递给物理页帧分配器用于给之后的内存分配做准备。

03

## 终端初始化

## CPU初始化

初始化各个核心的中断处理器，并启动浮点单元。

04

## PCI总线初始化

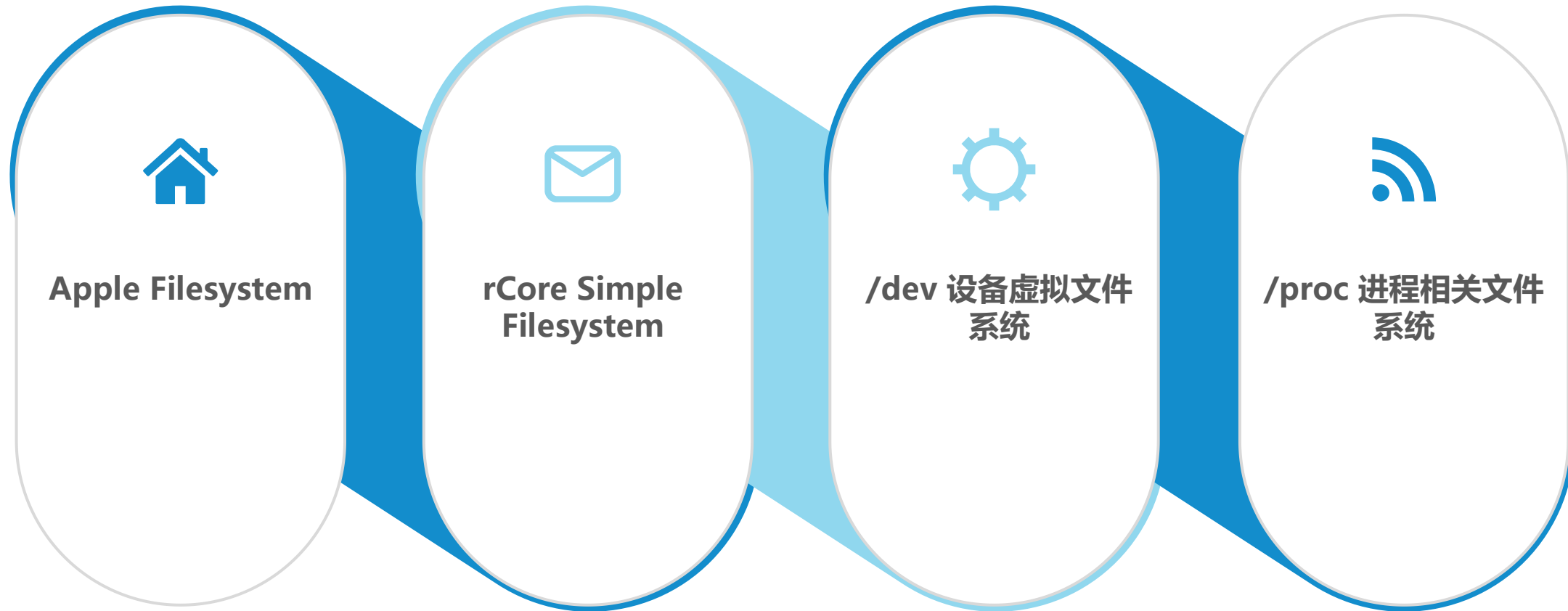
链接了网卡、SATA 设备，主要功能是探测总线上的设备，之后初始化这些设备并分配 MMIO、DMA 等用于访问它们的内存区域。

05

## ACPI初始化

06

# 文件系统与虚拟文件系统

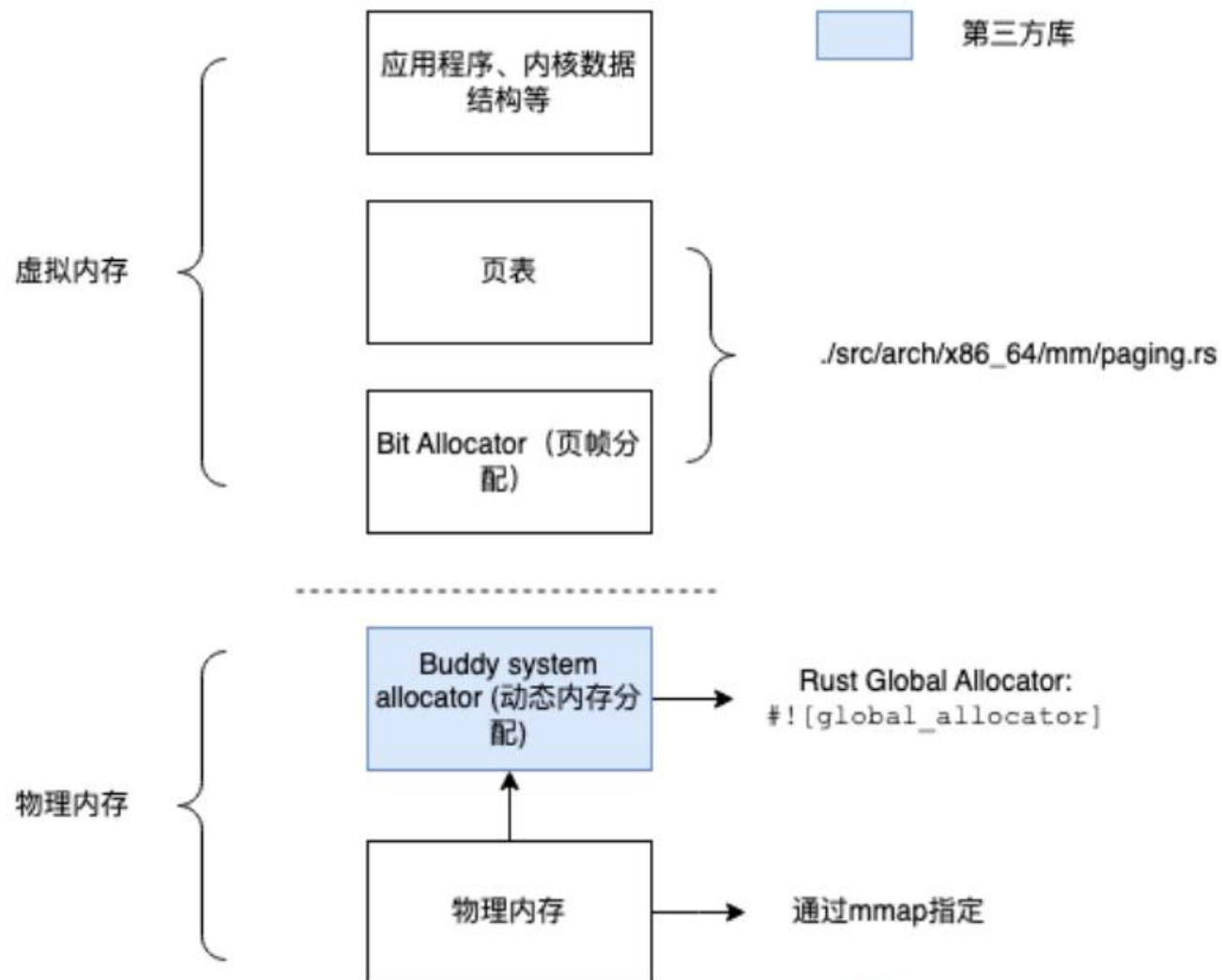


# 文件系统层次结构

内核/用户交互层
文件对象层 (kernel/src/fs/file.rs)
应用层 (kernel/src/fs/apfs)
驱动层抽象 (kernel/src/drivers/block.rs)
裸驱动层 (isomorphic driver)
硬件层 (PCI-e Bus)



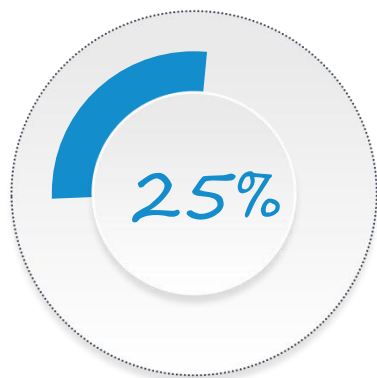
# 虚拟内存管理





# 课程设计

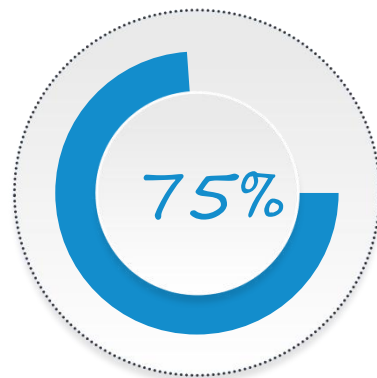
# 课程框架



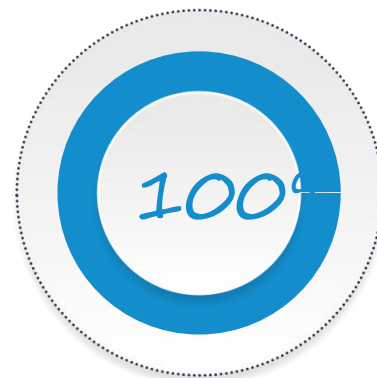
环境配置



内核设计



进程切换



文件系统

## 环境配置



The diagram features a central circle with the text '环境配置' (Environment Configuration). To its right are three horizontal bars, each representing a step. The top bar is dark blue with a globe icon and the text 'APFS的安装' (01). The middle bar is light blue with a Wi-Fi icon and the text 'RUST工具链' (02). The bottom bar is dark blue with a laptop icon and the text '必备软件包' (03). Dotted lines connect the central circle to each of the three bars.

APFS的安装

01

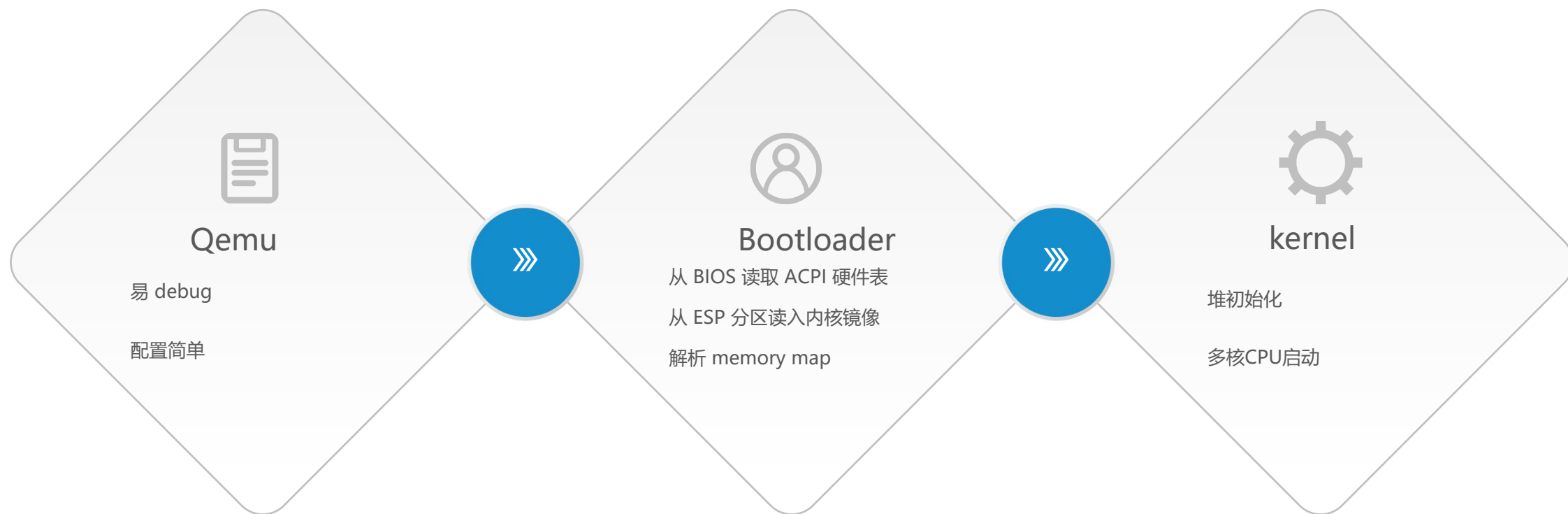
RUST工具链

02

必备软件包

03

# Lab 2



# Lab 3

## 进程与线程

### 目的

熟悉进程与线程的概念和它们之间的关系  
理解并编写一个简单的线程调度器

练习与解答



# Lab 4

1

## 目的

了解基本的文件系统系统调用的实现

了解一个Linux APFS文件系统的设计与实现

2

## Linux APFS

用于在Linux系统上支持APFS文件系统

支持快照

3

## mount

Linux系统中用于挂载文件系统的命令

4

## 练习

复习与总结

# 总结与实现意义





# 未来展望

01

实现更多的多核调度算法

02

完善网络栈的更多技术细节

03

进一步丰富文件系统,  
使其支持更多系统

04

加入足够的测试样例

05

设计更多有区分度的实验

06

考虑设计课程网站

# 感谢观看

学 校：南开大学  
指导教师：官晓利  
队伍名称：能润就行

参赛成员：穆禹宸 袁贞芷 李潇逸  
选 题：PROJECT-0