



MIT6.828-神级OS课程-要是早遇到，我还会是这种 five 系列

一丁点儿

不会编程的“司机”不是优秀的电子学生

+ 关注他

2,385 人赞同了该文章

MIT6.828 Operating System Engineering

花了两个月多的时间，终于把 MIT6.828 的课程跟完了。刚接触这个课程时，就有一种感觉：哇，好神奇！这个课程真的是循序渐进，一步一步在你的脑海里建立起 OS 的大厦框架。从 Lab1 的手把手教学到 Lab6 的放任你自由，很好地对学生的学习了进行了引导。学完这个课程后，你也就拥有了一个属于你自己的完整的内核雏形！

当然这个课程需要一些基础，如汇编语言、操作系统、计算机组成原理以及体系结构等。毕竟操作系统是一门比较底层且考察综合性计算机知识的课程。但这都不是重点，我一直认为，以边学边做的方式来学习一门课程是最好的方式之一（为什么是之一呢，怕被喷...嘻嘻）。只要能坚持下来，最后你会体会到，你已经能从一个与之前完全不同的角度理解计算机！现在微内核概念这么火，鸿蒙就被宣传为微内核全场景 OS，难道不想从原理上真正理解一下微内核 OS 吗？没错，6.828 这个课程就是引导我们实现了一个微内核操作系统。

没想到这个内容能被这么多人看到，还挺激动的。不过点赞收藏 1:5 是什么情况，大家不要收藏退出一气呵成啊~如果真的对计算机底层比较感兴趣的话，一定要去 try 一下啊。

1. 简介

课程评价：神级课程——要是早遇到，我还会是这种 five（废物）系列

▲ 赞同 2385



● 140 条评论

➤ 分享

♥ 喜欢

★ 收藏

📄 申请转载



我的实现环境：

- 虚拟机 (VMware Workstation 和 VirtualBox 均可)
- Ubuntu16.04
- qemu, 最好使用 MIT 给的 patch 版本 (Lab6 用到 patch 版本的qemu 较多)。安装方法也可参考[Tools Used in 6.828 链接](#)
- 工具链 [Tools Used in 6.828](#)

环境搭建网络上教程很多，这里我就不再赘述，大家可自行百度（搜索以下关键词），包括：

- VMware 或 VirtualBox 安装
- Ubuntu虚拟机
- qemu安装：英文教程参考上文给出的[工具链的链接](#)，中文教程可参考网络上的博文（[QEMU 模拟器安装](#)）

2. xv6

学习6.828时，你会发现经常遇到 xv6 和 JOS这两个名词，不明白它们两者之间的关系，在完成实验时思路就会不是很清晰。xv6 是一个类Unix的教学操作系统，而 JOS 是在xv6的基础上改写，让我们能在其上进行实验的 OS。所以实际上，当我们遇到不会实现的问题时，可以去参考 xv6 相应部分的源码。

Homework 实现：完整实现代码。 [Github_SmallPond/MIT6.828_OS](#)

- [MIT6.828 Homework Shell MIT 6.828](#)
- [MIT6.828 HW2 Boot xv6 MIT6.828](#)
- [MIT6.828 HW3 XV6 System calls](#)
- [MIT6.828 HW4 xv6 lazy page allocation](#)
- [MIT6.828 HW5 xv6 CPU alarm](#)
- [MIT6.828 HW6 Threads and Locking](#)
- [MIT6.828 HW7 xv6 locking](#)
- [MIT6.828 HW8 User-level threads](#)
- [MIT6.828 HW9 barriers](#)
- [MIT6.828 HW10 Bigger file for xv6](#)
- [MIT6.828 HW11 xv6 log](#)

- Lab1 启动PC
 - [LAB 1 Part1 PC Bootstrap](#)
 - [LAB 1 Part2 The Boot Loader](#)
 - [LAB1 Part3 The Kernel](#)
- Lab2 内存管理
 - [LAB2 Part1 Physical Page Management](#)
 - [LAB2 Part2 Virtual Memory](#)
 - [LAB2 Part3 Kernel Address Space\(内核地址空间\)](#)
- Lab3 用户级环境 (用户进程)
 - [LAB3 User-Level Environments PartA User Environments and Exception Handling](#)
 - [LAB3 User-Level Environments PartB Page Faults, Breakpoints Exceptions, and System Calls](#)
- Lab4 抢占式多任务
 - [LAB4 Preemptive Multitasking PartA Multiprocessor Support and Cooperative Multitasking](#)
 - [LAB4 Preemptive Multitasking PartB Copy-on-Write Fork](#)
 - [LAB4 Preemptive Multitasking PartC Preemptive Multitasking and IPC](#)
- Lab5 文件系统, Spawn and Shell
 - [Lab5 File system, Spawn and Shell](#)
- Lab6 网卡驱动
 - [Lab6 Network Driver](#)

4. 参考文献

1. [MIT 6.828 JOS 操作系统学习笔记/fatsheep9146](#), 刚入门时参考, 包括环境搭建。博文写得十分详细, 可惜貌似只写到了 Lab2。
2. [clpsz/mit-jos-2014](#), 此大神放出了自己到 Lab4 的代码, 其文档提及了一些细节, 很有帮助。
3. [Unknown Unknown](#), 过程较详细 (相当于对官方文档做了一遍翻译), 英语不好可以参考这边, 但我还是建议以官方文档为主, 毕竟英语还是要学好呀。这位大神做到了LAB5, 但不包括HW。
4. [bysui的博客](#), 这位大神完成了全部的实验, 但是我在后面才发现这么好的资源!

我的课程能顺利完成, 少不了各位大佬记录下的实验过程, 由衷感谢以上各位大神。同时希望我也能帮到后来的学习者~

编辑于 09-24

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

操作系统

微内核

操作系统内核

文章被以下专栏收录



操作系统与硬件

计算机硬件与操作系统的知识记录

关注专栏

推荐阅读

**开源大佬：微软将放弃
Windows转向Linux，原因...**

马超

发表于金融科技趋...

**MIT 6.828：实现操作
Lab1：快来引导一个P...**

anari...

发表

140 条评论

⇌ 切换为时间排序

写下你的评论...



精选评论 (3)

▲ 赞同 2385 ▼

● 140 条评论

➤ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...



雪化晴散

2019-08-27

哇，刚打算复习一遍的，真巧，感谢楼主的资料，更棒

👍 赞 💬 查看回复



粗茶泡饭

2019-10-24

目前正在跟6.S081，对比过6.828，作业确实变了很多，讲义没变，6.828上来就各种汇编，倒是和6.004衔接的很好，不过感觉难上手，6.S081一开始是让你熟悉和实现systemcall，感觉好上手一些，不过刚开始还是挺懵的，不熟悉read write pipe

👍 3

评论 (140)



maclinux

2019-08-26

收藏从未停止，学习从没开始

👍 215



一丁点儿 (作者) 回复 maclinux

2019-08-26

这个课程开始学习就停不下来[惊喜]

👍 10



OFFLINE 回复 maclinux

2019-08-27

刚刚点了收藏... 被你说中了... 也许收藏是为了下辈子吧.

👍 6

展开其他 1 条回复



吴雨儿

2019-08-26

请问需要什么前置知识不，我看xv6上来第一个实验就是实现一个shell?

👍 1



一丁点儿 (作者) 回复 吴雨儿

2019-08-26

做这个lab确实是需要一些计算机专业基础知识的，操作系统，汇编，计组等。不用懂太多，可以边做边学的，不会做的时候可以参考别人写的实验过程的。

👍 2



一丁点儿 (作者) 回复 吴雨儿

2019-08-26

shell的框架这个课程已经提供了源码，你只需要理解然后填写部分关键代码就好了





万盛中路摇滚乐队

2019-08-26

这个学期开的是6.S081

👍 3



一丁点儿 (作者) 回复 万盛中路摇滚乐队

2019-08-26

喔，我做的2018年的

👍 赞



万盛中路摇滚乐队 回复 一丁点儿 (作者)

2019-08-26

我在等今年的。 - - ..

👍 赞

[查看全部 16 条回复](#)

ever

2019-08-26

弱弱问下有中文么？

👍 赞



一丁点儿 (作者) 回复 ever

2019-08-26

Github上应该是有人翻译过的。

👍 赞



黄叶军

2019-08-26



👍 赞



Aegis Liang

2019-08-27

我记得这个课程中间缺了几集，请问今年这个课程重新录制了么？也是未开始的人问...

👍 赞



一丁点儿 (作者) 回复 Aegis Liang

2019-08-27

录制？ 你是指视频吗？ 我找到的是全损音质全损画质[飙泪笑]， 所以没有看视频

👍 1



我记得你的双眼 回复 一丁点儿 (作者)

2019-08-27

需要看视频么？ 还是直接跟着Schedule看上面的资料就好？





小子丁点儿

答主做了多久？

赞

一丁点儿 (作者) 回复 杀手小顾
两个半月呢

2019-08-27

赞



Harris 回复 一丁点儿 (作者)

2019-11-09

这么长时间啊，大概多少小时呢[好奇]

赞

展开其他 1 条回复



魂之挽歌

2019-08-27

这个不错哎。似乎没有视频或者教案，就是把lab刷一遍嘛？

赞

一丁点儿 (作者) 回复 魂之挽歌
是的，有个xv6的讲义，官方的schedule也是很详细啦

2019-08-27

赞



魂之挽歌 回复 一丁点儿 (作者)

2019-12-04

这个刷完了，很棒！有没有类似的课程，讲编译器的呢[惊喜]

赞

查看全部 6 条回复



杀手小顾

2019-08-27

厉害。

赞



雪化晴散

2019-08-27

哇，刚打算复习一遍的，真巧，感谢楼主的资料，更棒

赞

一丁点儿 (作者) 回复 雪化晴散

2019-08-27

我看了一下，这个课程实现的是 32 位内核，然而在体系结构上，32 位和 64 位区别还是有一些的。

建议题主试着实现一下 64 位，因为即使是手机，现在也已经使用了 64 位系统，虽然是 ARM。

而 PC 和服务器的早已使用 64 位。

根据我当年的学习经验，64 位最好直接看 Intel 官方手册，网上资料很少，而且不准确。

👍 1

一丁点儿 (作者) 回复 pretty kernel

2019-08-28

感谢您的意见~ 不过我做这个课程只是为了了解OS的工作流程等，比如OS基本原理，微内核，文件系统，硬件驱动等。并不是为了真正实现出一个完全可用的OS。

👍 1



Kousaka Chihaya

2019-08-29

我当年做的时候，lab2的一个bug导致lab5调了一个星期.....

👍 1

一丁点儿 (作者) 回复 Kousaka Chihaya

2019-08-29

我是lab4的Bug，导致lab5调了一个星期😓

👍 1



Eamon

2019-08-30

6.824也推荐

👍 赞



高原

2019-09-02

种草

👍 赞



吕不为

2019-09-05

看到了我的博客...

👍 赞

一丁点儿 (作者) 回复 吕不为

2019-09-05

你目前还在读哪本书呢？

咱在在门门没上OS课之前遇到什么bug吗？比如memset script有问题，这就不就一丁点代码，正确答案会导致内核初始化第一次env_create时，加载用户binary，memset出triple fault。这bug我已经掉三天头发了。不知道是不是之前lab导致的，好伤

👍 赞

一丁点儿 (作者) 回复 youcant

2019-09-07

没有遇到你这个问题哎，官方测试代码应该是不会出现问题的。

👍 赞



youcant 回复 一丁点儿 (作者)

2019-09-07

那应该是我之前代码有问题 我太难了

👍 赞

[查看全部 16 条回复](#)



黑黑黑

2019-09-07

请问这门课多了一个engineering后和操作系统有什么区别呢？没学过操作系统可以上吗？

👍 赞

一丁点儿 (作者) 回复 黑黑黑

2019-09-07

就是操作系统课程。这个课就是MIT的OS课程，他们的学生也是没上过OS的课的。不过做起来可能有点困难，边做边学就好。

👍 赞



黑黑黑 回复 一丁点儿 (作者)

2019-09-09

多谢！

👍 赞

1 2 3 下一页