

学号： 2018091611031

姓名： 张晓桐

专业方向： 系统与技术

单位名称： 电子科技大学

毕设课题名称： x86模拟器上的

32位多任务图形窗口简易操作系统

校外导师： 李忠

校内导师： 邱元杰

**信息与软件工程学院**

**毕业设计（顶岗实习）**

**初期报告**

目录

[第一章 研究现状及发展态势 3](#_Toc92235722)

[1.1研究现状 3](#_Toc92235723)

[1.2发展态势 3](#_Toc92235724)

[第二章 选题依据及意义 3](#_Toc92235725)

[2.1 选题依据 3](#_Toc92235726)

[2.2选题意义 3](#_Toc92235727)

[第三章 课题研究内容 3](#_Toc92235728)

[第四章 关键问题，最终目标，拟采取的主要理论、技术路线和实施方案等 4](#_Toc92235729)

[4.1拟解决的关键问题 4](#_Toc92235730)

[4.2最终目标 4](#_Toc92235731)

[4.3拟采取的主要理论、技术路线和实施方案等 4](#_Toc92235732)

[4.4 起步工作 5](#_Toc92235733)

[第五章 课题特色或创新点 6](#_Toc92235734)

第一章 研究现状及发展态势

1.1研究现状

操作系统是软件工程专业的基础课，初次学习通常侧重在理论，而一个真正的操作系统是庞大、复杂的，手写一个操作系统这件事显得非常神秘，也许让一个本科生望而却步了。

有一些优秀的大学设计了完整的操作系统的实验，例如MIT6.828，WISC-xv6,清华ucore，哈工大linux-0.11,多数以挖空的形式给出任务，整个实验就是完整的操作系统，通常是命令行形式。也有一些关于手写操作系统的书籍，例如于渊的《Orange’S:一个操作系统的实现》，我没看过，据说很好，川合秀实的《30天自制操作系统》，对图形界面的设计比较详细，但其用到了很多作者自制的工具，对原理的剖析较弱。还有很多剖析Linux源码的书籍，我手上有《Linux内核设计与实现》，里面介绍了一些数据结构和功能的实现，可以参考。

和服务器、商城、博客项目相比，手写一个操作系统，尽管是很简单的操作系统，从“练手”的角度讲，往往受众更少。但与其相关的开源项目和工具是十分丰富的。

1.2发展态势

x86架构基本被英特尔和amd把持，[生态链](https://www.zhihu.com/search?q=%E7%94%9F%E6%80%81%E9%93%BE&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A%221651783268%22%7D)很强大，在未来一段时间仍是主流。课题选择的i386/i486是intel早期的cpu,尽管已经过时了，但仍然适合学习。

随着信息的联通和学习资料的丰富，越来越多人开始练习操作系统的lab了，例如MIT6.828很多人都在写。

第二章 选题依据及意义

2.1 选题依据

操作系统实验设计计算机组成原理、汇编语言、C语言和若干种设计思想，和专业内容强相关，同时也是很好的工程训练。因为关于i386架构的CPU的参考资料较多，各种攻击

2.2选题意义

学习操作系统，可以串联起从微小的集成电路到浩瀚的互联网和分布式集群，实现一个简单的操作系统，特别是具有图形界面的，十分常有成就感的。借由本课题，学习汇编语言，了解BIOS程序和CPU架构，学习中断、时钟、图形界面和多任务的实现。

完成操作系统项目的过程中会遇到很多问题，是对编程、排错和解决问题基本功的训练。同时涉足嵌入式开发，为求职带来更多可能。

第三章 课题研究内容

在windows或者其它操作系统上编写源代码，并编译生成机器语言文件，对机器语言文件进行加工，生成软盘（或硬盘）映像文件。使用硬件模拟器运行这个映像文件，运行操作系统。

课题研究内容由有以下几个部分组成并层层递进：

汇编和CPU架构：汇编指令，合法的操作系统盘制作，编译和链接，实模式和保护模式

使用键盘鼠标、显示窗口：了解中断机制和显示原理，学习数据结构和算法优化键鼠功能

保护模式使用：存储器的保护，程序的动态加载和执行，任务和任务切换，内存管理

使用应用程序：命令行窗口和应用程序编写

第四章 关键问题，最终目标，拟采取的主要理论、技术路线和实施方案等

4.1拟解决的关键问题

工具链选择和使用

保护模式的学习

操作系统设计和实现

4.2最终目标

采用迭代开发，逐步从一个简单的汇编程序演变成一个完整的操作系统。开发工具在一次配置后应该使用方便，对系统中的功能借鉴不同的实现算法进行调优。

4.3拟采取的主要理论、技术路线和实施方案等

硬件模拟器Bochs: [bochs: The Open Source IA-32 Emulation Project (Home Page) (sourceforge.io)](https://bochs.sourceforge.io/)

硬件模拟器qemu:

汇编软件nasm:

交叉编译：ssh连接centos系统的服务器

构建软件：make bat脚本和shell脚本

版本控制：在本地以功能点为单位创建文件夹并上传至github

保护模式参考李忠老师的《x86汇编 从实模式到保护模式》

4.4 起步工作

task1 制作合法的系统盘，并显示文字

1.1 CPU加电后做什么

CPU加电后，电路进行初始化，然后会进行各种自查，最后，进入到“取指令->执行指令”这样的循环中（这样的循环是“自动的”，即由时钟驱动）。那么第一条指令就是访问系统盘相关的。接下来的一系列指令完成从系统盘加载第一个512字节内容，并检查这512字节是否合格，若合格则将其加载到内存0x7c00处，并修改取指相关的寄存器，跳到0x7c00处开始执行。

1.2合格的系统盘

前512字节称为引导扇区，字节从0编号，510号内容为0xaa，511号内容为0x55。引导扇区被加载到0x7c00后，CPU从0x7c00处开始执行，所以引导扇区上写着一些指令，加载操作系统并执行操作系统的工作就在这一个扇区的内容内完成。nasm将汇编指令文件转化为纯二进制指令集合。使用dd软件制作虚拟软盘或者虚拟硬盘文件供硬件模拟器使用。

1.3 bochs和qemu

硬件模拟器。通过配置文件或命令行指定读取的虚拟盘的位置、格式。虚拟盘文件中的内容如何排列，不同的类型有不同的规定，比如.vhd就是一种可以被bochs使用的硬盘格式。模拟器读取什么格式的虚拟磁盘，虚拟磁盘文件就要遵循同样的标准制作，这就是“协议”。我选取了raw格式3.5英寸软盘，可以承载2880\*512字节数据，raw格式就是一个文件里的内容完全等价于一张3.5英寸软盘中的数据，raw，就是没有修饰的意思，除了磁盘中应该有的数据，虚拟磁盘文件中不多一点其他的东西。

1.4 makefile

随着系统的开发，制作系统映像的步骤会很多。比如编译、汇编、连接、向模拟器发出命令，甚至将部分文件交给有Linux系统的主机处理。这些任务之间有顺序和依赖关系，Make软件帮我们分析这些复杂的关系,只需对它说：make请帮我用bochs调试我的操作系统吧。make很聪明，会把所需的工作完成，然后最终执行“用bochs调试操作系统”这个目标。

task2 读取磁盘内容到指定内存位置

2.1 存放位置的协议

引导程序只有最多510字节，装不下一个操作系统。操作系统保存在磁盘中，应该由引导程序读出，放在内存的某个位置，适时跳转到操作系统中执行。制作映像文件时将系统写入到哪个扇区，引导程序读磁盘时就应该从那个扇区开始读。

2.2程序大小的协议

引导程序需要知道操作系统文件有多大，以便决定读多少扇区。操作系统可以在一开头就给出文件大小，这样引导程序就从这个约定的位置读出操作系统的大小，以便决定还要读多少个扇区。

2.3 使用bochs查看特定内存位置内容

task3加载操作系统后跳转执行

3.1分段和跳转指令

重定位的基础。

3.2重定位工作

二次加载的操作系统中会有一些标号，这些标号的汇编地址是相对于汇编文件头的偏移值。操作系统并不知道自己被加载到哪里，所以没办法在汇编的时候确定物理地址。物理地址则是运行时内存中的地址。所幸可以将操作系统的指令看作存在一个单独的段（一段内存的意思）中，就叫操作系统段吧！操作系统被加载到0x10000，操作系统段就从0x10000开始，操作系统被加载到0x20000,操作系统段就从0x20000开始。物理地址由段基址和段内偏移相加构成。实际编写的时候，为了支持复杂的功能，操作系统又由代码段、系统栈段、数据段等等构成，每个段都有其段基址。引导程序在将操作系统加载到内存中时，应负责修改操作系统中的各种段的段基址。

引导程序应该和操作系统达成协议：有关段的信息应该放在两者都知道的地方，在 CPU的控制权交给操作系统之前，引导程序将段的信息修改为与本词加载位置匹配的“正确的段的信息”。

3.3 跳转的实现方式

task4进入保护模式

4.1保护模式下的内存访问

4.2进入保护模式的细节

4.3查看保护模式下的寄存器和数据结构

task5 内存布局的设计

…

第五章 课题特色或创新点

既有从汇编和内存角度的原理分析，了解系统加载、保护模式和多任务的实现机制，又实现图形界面增添趣味。迭代开发，每个增量都在模拟器上运行，和挖空Oslab相比更加全面细致，使得开发的操作系统客观可感。

如果时间充裕，可为系统增加网络协议栈。

参考文献

[1].《x86汇编语言从实模式到保护模式》李忠 王晓波 余洁

[2].《30天自制操作系统》[日]川合秀实