

西南林业大学
本科毕业（设计）论文
(二〇一八届)

题 目：____操作系统探索_____

分院系部：____大数据与智能工程学院_____

专 业：____计算机科学与技术专业_____

姓 名：____尹志成_____

导师姓名：____王晓林_____

导师职称：____讲师_____

二〇一八年六月

操作系统探索

尹志成

(西南林业大学 大数据与智能工程学院, 云南昆明 650224)

摘 要： 操作系统最初的诞生是为了搭配进行简单繁重的数字运算机，但随着时代的演进，计算机不仅作为处理各种运算的机器，其附加价值也越来越被人们看重，跟着计算机的发展，操作系统的使命也在一代代的改变，（约两百字）

关键词： 操作系统

Operating system exploration

Zach Yin

School of Big Data and Intelligence Engineering
Southwest Forestry University
Kunming 650224, Yunnan, China

Abstract: 英文摘要

Key words: Operate System

目 录

1 绪论	1
2 思路	2
2.1 操作系统探究	2
2.2 空白操作系统的启动	2
2.3 丰富操作系统内容	2
2.4 实现对外接口及安全防护	2
3 操作系统探索	3
4 空白操作系统的启动	4
4.1 操作系统启动流程	4
4.2 制作 MBR (ipl09.nas)	4
4.3 制作空白操作系统	7
5 编写操作系统内核	8
5.1 内存管理	8
5.2 输入输出	8
5.3 多进程	8
5.4 分时	8
6 实现对外兼容及安全防护	9
7 另一章	10
7.1 图片与表格	10
7.1.1 图片示例	10
7.1.2 表格示例	10
8 又一章标题	11

参考文献	12
指导教师简介	12
致 谢	14
A 我也不知道为什么要写附录	15
B 主要程序代码	16

插图目录

7-1 图片示例	10
--------------------	----

表格目录

7-1 表格示例	10
--------------------	----

1 绪论

在数字时代，操作系统的重要性不言而喻，它作为计算机软硬件之间的桥梁，存在于日常生活的每一个角落，而研究一个只有庞大的公司聚集成百上千的高级工程师才能完成的操作系统对于学生而言是一个几乎不可能完成的挑战，但是克服难关是锻炼技术的必经之路^[1]，所以研究并完成一个基本满足日常功能需求的操作系统作为此次的目标，并以此为跳板对操作系统进行更深一步的探究。

2 思路

此次的思路由四部分组成：1、操作系统探究 2、空白操作系统的启动 3、编写操作系统内核 4、实现对外兼容及安全防护

2.1 操作系统探究

从历史上计算机操作系统的发展联系到人们的日常生活，寻求符合操作系统发展且适应用户使用的特征要素。

2.2 空白操作系统的启动

利用汇编语言及操作系统相关知识探究操作系统如何从电气设备到软件代码的衔接

2.3 丰富操作系统内容

从内存管理，输入输出，多进程，分时四个模块丰富操作系统的内容

2.4 实现对外接口及安全防护

从接口设计及安全防护的角度完善操作系统

3 操作系统探索

4 空白操作系统的启动

4.1 操作系统启动流程

按下电源键之后启动计算机，启动过程分为四个阶段^[2]：

BIOS -> MBR -> VBR -> 操作系统

1、在 BIOS 完成 POST（硬件自检，Power-On Self Test）并选择启动顺序（Boot Sequence）把控制权转交给排在第一位的储存设备

2、计算机读取该设备的 MBR（第一个扇区，最前面的 512 个字节），在此装入 ZOS 的启动程序 IPL（Initial Program Loader）程序 ipl09.nas

ipl09.nas 指明了操作系统的位置，主分区第一个扇区的物理位置（柱面、磁头、扇区号等等）

3、计算机根据 VBR（Volume boot record）指引得到操作系统在这个分区里的位置继而加载操作系统

4、控制权转交给操作系统后，操作系统的 Kernel 被载入内存

4.2 制作 MBR（ipl09.nas）

MBR 负责指出操作系统的位置，主分区第一个扇区的物理位置（柱面、磁头、扇区号等等）

```
1  ORG                                0x7c00                                ; 指明程序装载地址

1  DB                                " zbote "                                ; 启动扇区名称 (8字节)
2  DW                                512                                ;
   ↳ 每个扇区 (sector) 大小 (必须512字节)

3  DB                                1                                ;
   ↳ 簇 (cluster) 大小 (必须为1个扇区)
```

4 空白操作系统的启动

```

4  DW          1                                ;
   ↳ FAT起始位置（一般为第一个扇区）
5  DB          2                                ;
   ↳ FAT个数（必须为2）
6  DW        224                                ;
   ↳ 根目录大小（一般为224项）
7  DW        2880                               ;
   ↳ 该磁盘大小（必须为2880扇区1440*1024/512）
8  DB          0xf0                             ;
   ↳ 磁盘类型（必须为0xf0）
9  DW          9                                ;
   ↳ FAT的长度（必须是9扇区）
10 DW          18                               ;
   ↳ 一个磁道（track）有几个扇区（必须为18）
11 DW          2                                ;
   ↳ 磁头数（必须是2）
12 DD          0                                ;
   ↳ 不使用分区，必须是0
13 DD        2880                               ; 重写一次磁盘大小
14 DB          0,0,0x29                         ; 意义不明（固定）
15 DD        0xffffffff                         ; （可能是）卷标号码
16 DB          "    ZOS    "                   ;
   ↳ 磁盘的名称（必须为11字符，不足填空格）
17 DB          "FAT12    "                     ;
   ↳ 磁盘格式名称（必须是8字符，不足填空格）
18 RESB       18                               ; 先空出18字节

1  MOV        CH,0                             ; 柱面0
2  MOV        DH,0                             ; 磁头0
3  MOV        CL,2                             ; 扇区2

```

4 空白操作系统的启动

```
1 readfast:          ; 使用AL尽量一次性读取数据 从此开始
2 ; ES: 读取地址, CH: 柱面, DH: 磁头, CL: 扇区, BX: 读取扇区数
3
4     MOV             AX, ES                                ;
5     ↪ < 通过ES计算AL的最大值 >
6     SHL             AX, 3                                ;
7     ↪ 将AX除以32, 将结果存入AH (SHL是左移位指令)
8     AND             AH, 0x7f                              ;
9     ↪ AH是AH除以128所得的余数 (512*128=64K)
10    MOV             AL, 128                               ;
11    ↪ AL = 128 - AH;
12    ↪ AH是AH除以128所得的余数 (512*128=64K)
13    SUB             AL, AH
14
15    MOV             AH, BL                                ;
16    ↪ < 通过BX计算AL的最大值并存入AH >
17    CMP             BH, 0                                ;
18    ↪ if (BH != 0) { AH = 18; }
19    JE             .skip1
20    MOV             AH, 18
21
22 next:
23
24    POP             AX
25    POP             CX
26    POP             DX
27    POP             BX                                    ;
28    ↪ 将ES的内容存入BX
29    SHR             BX, 5                                ;
30    ↪ 将BX由16字节为单位转换为512字节为单位
31    MOV             AH, 0
32    ADD             BX, AX                                ;
33    ↪ BX += AL;
```

```

22      SHL                BX, 5                      ;
      ↪ 将BX由512字节为单位转换为16字节为单位
23      MOV                ES, BX                      ;
      ↪ 相当于EX += AL * 0x20;
24      POP                BX
25      SUB                BX, AX
26      JZ                  .ret
27      ADD                CL, AL                      ;
      ↪ 将CL加上AL
28      CMP                CL, 18                      ;
      ↪ 将CL与18比较
29      JBE                readfast                    ; CL
      ↪ <= 18则跳转至readfast
30      MOV                CL, 1
31      ADD                DH, 1
32      CMP                DH, 2
33      JB                  readfast                    ; DH <
      ↪ 2则跳转至readfast
34      MOV                DH, 0
35      ADD                CH, 1
36      JMP                readfast

```

4.3 制作空白操作系统

首个操作系统为测试操作，目的是测试 MBR 可以成功启动操作系统

```

1  fin:
2  [ ] HLT
3  [ ] JMP fin

```

5 编写操作系统内核

从内存管理，输入输出，多进程，分时四个模块丰富操作系统的内容

5.1 内存管理

5.2 输入输出

5.3 多进程

5.4 分时

6 实现对外兼容及安全防护

从接口设计及安全防护的角度完善操作系统

7 另一章

7.1 图片与表格

如果需要插入图片与表格的话，可以参考下面的简单例子。

7.1.1 图片示例

下面是插入图片的示例：

图 7-1 图片示例

7.1.2 表格示例

下面是一个表格的例子：

Hello	world	Hello, world!
hline Hello	world	Hello, world!
hline		

表 7-1 表格示例

8 又一章标题

接着写吧接着写吧接着写吧接着写吧

参考文献

- [1] 川合秀实, **2012-08**.
- [2] 阮一峰, 如何变得有思想, **2014**.

指导教师简介

王晓林，男，49岁，硕士，讲师，毕业于英国格林尼治大学，分布式计算系统专业。现任西南林业大学计信学院教师。执教Linux、操作系统、网络技术等方面的课程，有丰富的Linux教学和系统管理经验。

致 谢

感谢，

附录 A 我也不知道为什么要写附录

可以参考模版目录中的 `appendix.tex` 文件来写。

附录 B 主要程序代码