

Report

数据科学与计算机学院 18级超算方向 田蕊 tianr6@mail2.sysu.edu.cn

一、实验题目

监控程序控制用户程序的执行

二、实验目的

1. 了解监控程序执行用户程序的主要工作
2. 了解一种用户程序的格式与运行要求
3. 加深对监控程序概念的理解
4. 掌握加载用户程序方法
5. 掌握几个BIOS调用和简单的磁盘空间管理

三、实验要求

1. 知道引导扇区程序实现用户程序加载的意义
2. 掌握COM/BIN等一种可执行的用户程序格式与运行要求
3. 将自己实验一的引导扇区程序修改为3-4个不同版本的COM格式程序，每个程序缩小显示区域，在屏幕特定区域显示，用以测试监控程序，在1.44MB软驱映像中存储这些程序。
4. 重写1.44MB软驱引导程序，利用BIOS调用，实现一个能执行BIN格式用户程序的监控程序。
5. 设计一种简单命令，实现用命令交互执行在1.44MB软驱映像中存储几个用户程序。
6. 编写实验报告，描述实验工作的过程和必要的细节，如截屏或录屏，以证实实验工作的真实性[2]

四、实验方案

(一)、硬件或虚拟机配置方法

在上一个实验的基础上，本实验不需要对虚拟机进行新的配置，只要将写入程序的软盘挂载在虚拟机上运行即可。

(二)、软件工具与作用

- | | |
|-----------------|------------------------|
| • 写字板 | 编辑汇编语言源文件 |
| • nasm | 编译汇编语言源文件 |
| • VM VirtualBox | 虚拟机运行编写好的程序 |
| • winhex | 对软盘进行编辑，将编译好的bin文件写入软盘 |
| • Typora | 编辑实验报告 |

(三)、方案的思想

本实验根据输入的指令的不同，监控程序调用不同的程序执行，核心思想就是在输入指令后，监控程序跳转到相应的空间获取程序的内容并进行执行。在实现的过程中利用中断实现提示语句的输出和磁盘读取的实现。在执行完目标程序之后再跳转回监控程序实现循环执行。

(四)、相关原理

BIOS中断

BIOS中断服务程序实质上是微机系统中软件与硬件之间的一个可编程接口，主要用于程序软件功能与微机硬件之间 接。度

实际上是一些对端口的输入输出操作，PC的每个端口都实现特定的功能，我们完全可以不调用BIOS提供的中断而直接用输入输出指令对这些端口进行操作，从而问可以实现象调用BIOS中断一样的功能，但这是一个前提是你答必须对这些端口有详细的了解。反过来说，PC的中断系统的一大好处就是能够让程序员无须了解系统底层回层答的硬件知识的而能够编程，从这点看，中断有点象我们平时所说的“封装”，BIOS中断服务为我们“封装”了许多系统底层的细节。[1]

常用的BIOS调用

功能	中断号	功能号
插入空行上滚显示页窗口	10H	06H
以电传方式显示单个字符	10H	0EH
显示字符串	10H	13H
复位磁盘系统	13H	00H
读扇区	13H	02H
读下一个按键	16H	00H

在这个程序中我们主要用到的中断是10H——显示服务和13H——磁盘服务，同时还用到了16H——键盘服务。BIOS的中断很多，我们这里着重介绍我们用到的内容。

10H——显示服务

汇编中的10H中断是由BIOS对显示器和屏幕所提供的服务程序。使用int 10h服务程序时，必须先指定ah寄存器为以下显示服务编号之一，以指定需要调用的功用。

功能06H

功能描述：初始化屏幕或滚屏

入口参数：

AH = 06H——向上滚屏

AL = 滚动行数(0——清窗口)

BH = 空白区域的缺省属性

(CH、CL) = 窗口的左上角位置(Y坐标，X坐标)

(DH、DL) = 窗口的右下角位置(Y坐标，X坐标)

出口参数：无[3]

说明：我们这里用到这个功能是为了每运行一次程序之后进行一次清屏。所以在我们的程序中AL置0，bh置0fh。因为是全屏清屏，所以设置ch = 0, cl = 0, dh = 24, dl = 79。设置好参数之后调用中断实现清屏

功能13H

功能描述：在Teletype模式下显示字符串

入口参数：

AH = 13H

BH = 页码

BL = 属性(若AL=00H或 01H)

CX = 显示字符串长度(

DH、DL) = 坐标(行、列)

ES:BP = 显示字符串的地址

AL = 显示输出方式

0——字符串中只含显示字符，其显示属性在BL中。显示后，光标位置不变

1——字符串中只含显示字符，其显示属性在BL中。显示后，光标位置改变

2——字符串中含显示字符和显示属性。显示后，光标位置不变

3——字符串中含显示字符和显示属性。显示后，光标位置改变

出口参数：无[3]

说明：我们这里多次用到这个功能，是为了实现输出提示语句，所以设置ax = 1301h, bx = 0007h, dh = 0, dl = 0。设置好参数之后调用中断实现显示输出提示语句

13H——磁盘服务

BIOS Int 13H 调用是 BIOS 提供的磁盘基本输入输出中断调用，它可以完成磁盘(包括硬盘和软盘)的复位，读写，校验，定位，诊断，格式化等功能。它使用的就是 CHS 寻址方式，因此最大能访问 8 GB 左右的硬盘(本文中如不作特殊说明，均以 1M = 1048576 字节为单位)。[4]

功能02H

功能描述：读扇区

入口参数：

AH = 02H

AL = 扇区数

CH = 柱面

CL = 扇区

DH = 磁头

DL = 驱动器，00H~7FH：软盘；80H~0FFH：硬盘

ES:BX = 缓冲区的地址

出口参数：CF = 0——操作成功，AH = 00H，AL = 传输的扇区数，否则，AH = 状态代码

说明：我们这里面用到这个功能来读扇区中的程序，其中程序一保存在第三扇区，程序二保存在第四扇区，程序三保存在第五扇区，程序四保存在第六扇区。所以不同的调用处cl取值分别为3, 4, 5, 6。ah = 2, al = 1,其余的寄存器全部取0。在子程序中返回监控程序时需要将cl设置为1。

16H——键盘服务

INT 16H的功能是键盘服务，具体运用如下：00H、10H——从键盘读入字符03H——设置重复率01H、11H——读取键盘状态04H——设置键盘点击02H,12H——读取键盘标志05H——字符及其扫描码进栈[5]

功能00H

功能描述：从键盘读入字符

入口参数：AH = 00H——读键盘

出口参数：AH = 键盘的扫描码

AL = 字符的ASCII码

说明：这里主要的应用是从键盘键入指令根据指令进行程序的调用，所以只要将ah设置为0，然后调用中断即可，al中保存的就是我们输入的字符。

(五)、程序流程

监控程序

1. 设置偏移
2. 实现清屏
3. 调用中断输出提示语句
4. 用户键入指令
5. 根据指令调用相应的子程序
6. 子程序运行结束返回监控程序的最开始

子程序

1. 设置偏移
2. 调用中断输出学号姓名
3. 判断字符的运动方向
4. 决定字符内容和颜色
5. 利用显存输出字符
6. 输出结束后返回监控程序最开始

(六)、程序关键模块

清屏

```
clear:
    mov ah,06h    ;清屏
    mov al,0
    mov ch,0      ;清屏左上角位置
    mov cl,0
    mov dh,24     ;清屏右下角位置
    mov dl,79
    mov bh,0fh
    int 10h
```

判断并调用子程序

```
    mov     ah, 0
    int     16h                      ;调用中断获取字符放在AL中,输入指令

    mov     ah,0eh
    mov     bl,0
    int     10h                      ;调用10H号中断,显示输入的内容

    cmp     al, 'A' ;如果用户输入的是A,就调用第一个程序
    je      Load1Ex
    cmp     al, 'B' ;如果用户输入的是B,就调用第二个程序
    je      Load2Ex
    cmp     al, 'C' ;如果用户输入的是C,就调用第三个程序
    je      Load3Ex
    cmp     al, 'D' ;如果用户输入的是D,就调用第四个程序
    je      Load4Ex
```






调用子程序

```
;读软盘或硬盘上的若干物理扇区到内存的ES:BX处:
mov ax,cs                ;段地址  ; 存放数据的内存基地址
mov es,ax                ;设置段地址（不能直接mov es,段地址）
mov bx, OffsetOfUserPrg1 ;偏移地址; 存放数据的内存偏移地址
mov ah,2                 ; 功能号
mov al,1                 ;扇区数
mov dl,0                 ;驱动器号 ; 软盘为0，硬盘和U盘为80H
mov dh,0                 ;磁头号 ; 起始编号为0
mov ch,0                 ;柱面号 ; 起始编号为0
mov cl,3                 ;起始扇区号 ; 起始编号为1
int 13H ;                 调用读磁盘BIOS的13h功能
; 用户程序a.com已加载到指定内存区域中
jmp OffsetOfUserPrg1
```

五、实验过程

(一)、编写汇编语言程序

根据老师上课讲述的内容编写汇编语言源程序并对程序进行检查调试

 a.asm	2020/5/5 14:03	Assembler Source	6 KB
 b.asm	2020/5/5 14:04	Assembler Source	6 KB
 c.asm	2020/5/5 14:05	Assembler Source	6 KB
 d.asm	2020/5/5 14:06	Assembler Source	6 KB
 myboot2.asm	2020/5/5 17:43	Assembler Source	6 KB

编写好的汇编语言源程序

(二)、利用nasm编译汇编语言源程序

```
nasm
Microsoft Windows [版本 10.0.18362.778]
(c) 2019 Microsoft Corporation。保留所有权利。

D:\NASM>E:

E:\>cd 操统实验\实验二

E:\操统实验\实验二>nasm myboot2.asm -o myboot2.bin

E:\操统实验\实验二>nasm a.asm -o a.bin

E:\操统实验\实验二>nasm b.asm -o b.bin

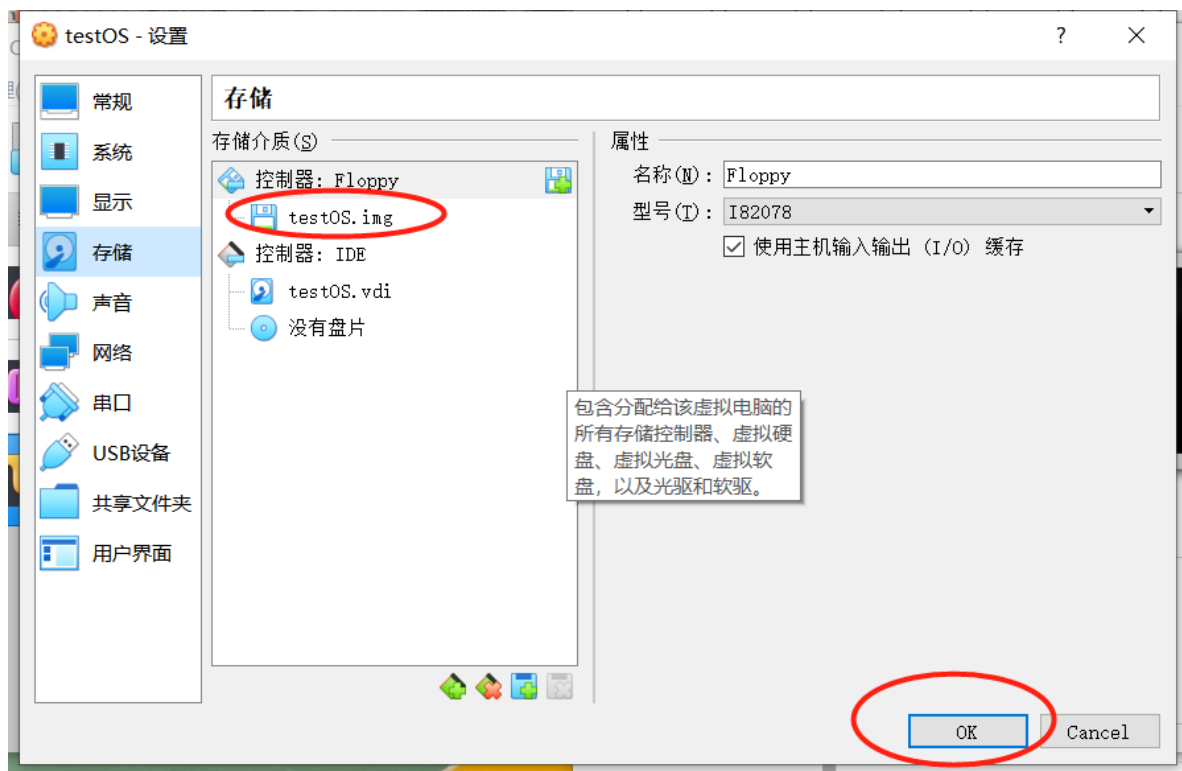
E:\操统实验\实验二>nasm c.asm -o c.bin

E:\操统实验\实验二>nasm d.asm -o d.bin
```

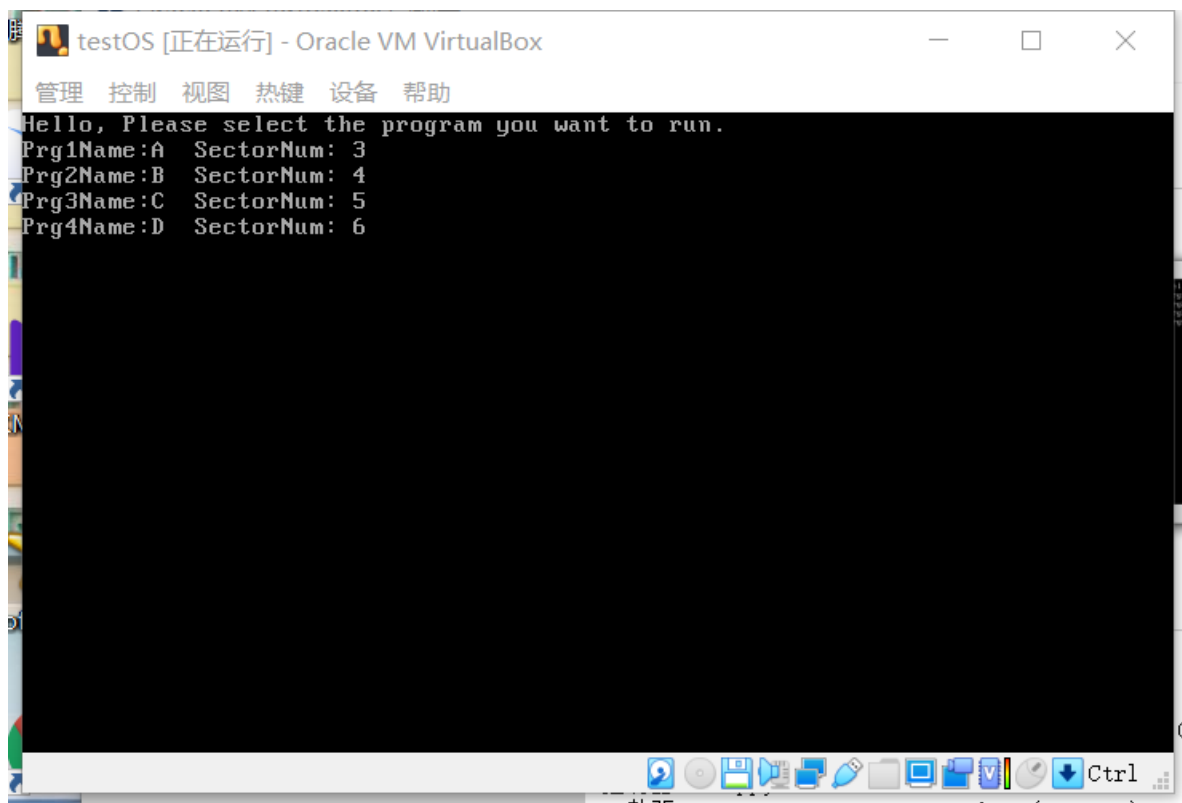
编译好后得到相应的.bin文件

 a	2020/5/5 20:01	BIN 文件	1 KB
 b	2020/5/5 20:01	BIN 文件	1 KB
 c	2020/5/5 20:01	BIN 文件	1 KB
 d	2020/5/5 20:01	BIN 文件	1 KB
 myboot2	2020/5/5 20:01	BIN 文件	1 KB

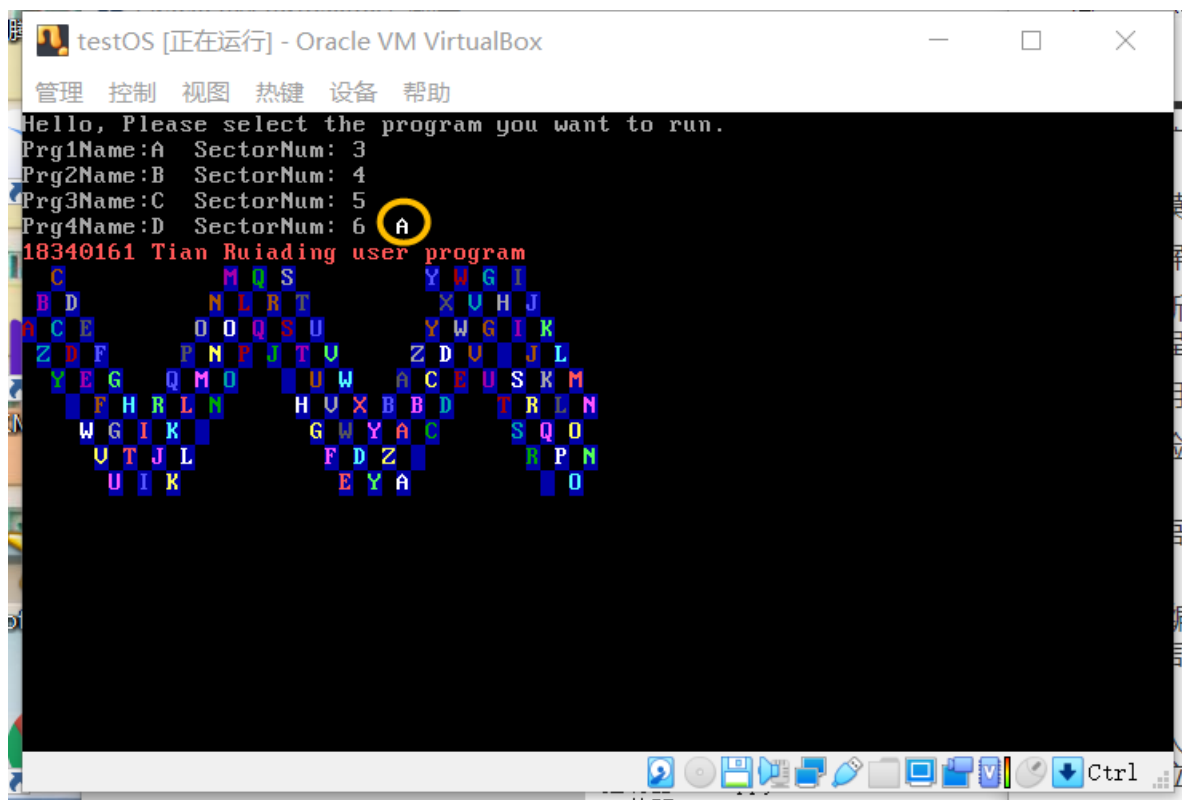
[illegible]



(五)、运行虚拟机



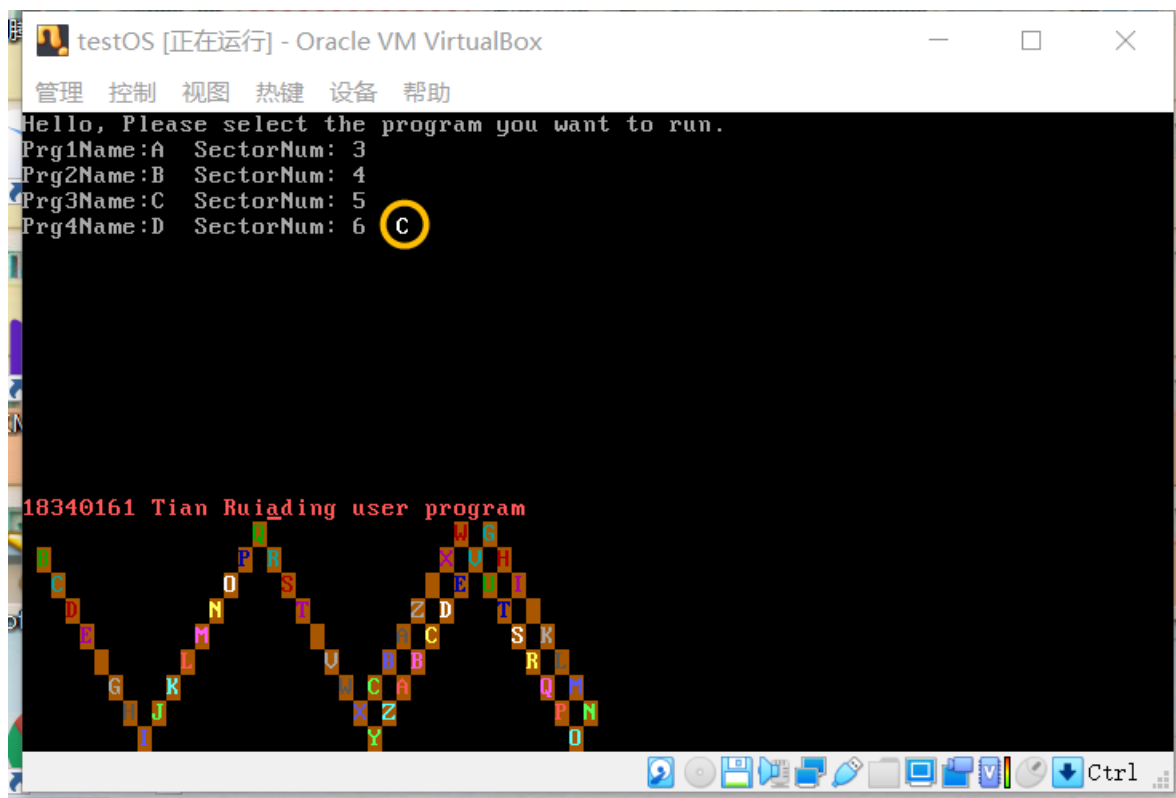
引导程序



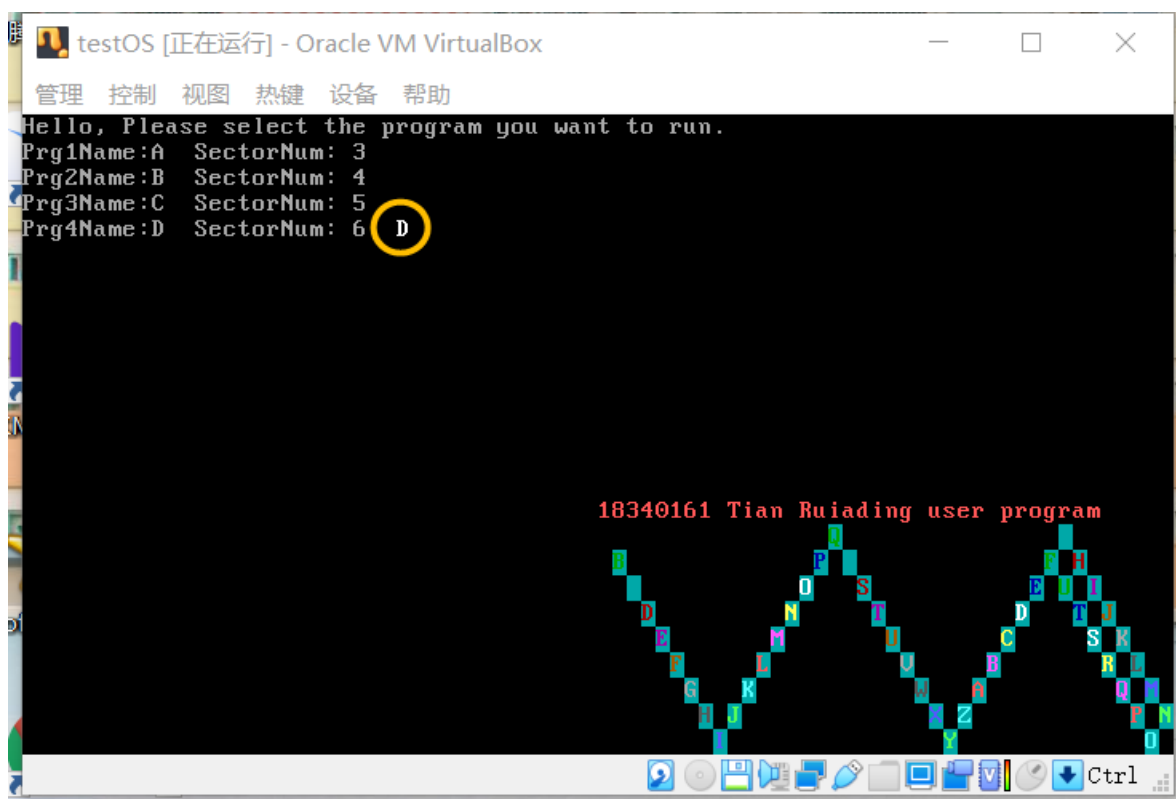
输入指令A，调用第一个子程序



输入指令B，调用第二个子程序



输入指令C，调用第三个子程序



输入指令D，调用第四个子程序

六、实验总结

(一)、心得体会

这次实验在上一次的基础上实现，中间涉及到了一些从来没有遇到过的知识，在刚刚开始实验的时候很没有头绪，之后反复看了老师给的程序实例，之后有反复看了网课录屏又自己摸索才终于有了一点点头绪，然后一点点实现下去。在这次实验过程中发现汇编语言debug是一个比较复杂的过程，不是十分的方便，自己也不是十分的熟练，希望以后自己可以在这个方面加强。另外这次实验觉得自己对org指令有了一个更加什么的理解，是一个进步，对winhex的使用也比以前熟练了很多。这次大部分的操作都是在Windows中的，希望下次可以在linux中实现相应的操作。

(二)、特色与不足

特色

1. 每一个子程序调用结束实现清屏操作，使得程序界面看起来十分整洁
2. 能够循环无限次调用子程序
3. 设计了程序组织表，方便用户的查看与分析

不足

1. 程序的信息表十分简短，不够详细
2. 没有足够的指令指引
3. 子程序十分简单
4. 没有对.com格式程序进行探讨

七、参考文献

-
- [1]暗影之王01.BIOS中断[EB/OL].<https://zhidao.baidu.com/question/71714006.html>,2016-03-11.
- [2]凌应标.16位A线路实验项目的目的与要求[Z].广州:SYSU University,2020.
- [3]hua19880705.汇编中的10H中断int 10h详细说明[EB/OL].<https://blog.csdn.net/hua19880705/article/details/8125706>,2012-10-29.
- [4]灵易联盟.扩展int 13H/调用规范 /大硬盘读写中断/FAT NTFS文件结构[EB/OL].https://blog.csdn.net/jiaguoxinzh/article/details/2949304?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1,2008-09-18.
- [5]qingkongyeyue.键盘I/O中断调用 (INT 16H) [EB/OL].<https://blog.csdn.net/qingkongyeyue/article/details/68490194>,2017-03-30.