Report

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

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百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。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,因此最大识能访问 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

```
ah, 0
mov
                    ;调用中断获取字符放在AL中,输入指令
int
     16h
mov ah,0eh
mov b1,0
int 10h
           ; 调用10H号中断,显示输入的内容
    al, 'A' ;如果用户输入的是A,就调用第一个程序
cmp
je
   Load1Ex
    аl, 'в'
           ;如果用户输入的是B,就调用第二个程序
cmp
je Load2Ex
    al, 'C' ;如果用户输入的是C,就调用第三个程序
cmp
    Load3Ex
je
cmp al, 'D'
           ;如果用户输入的是D,就调用第四个程序
jе
     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 ;设置段地址(不能直接mov es,段地址) mov es,ax mov bx, OffSetOfUserPrg1 ;偏移地址; 存放数据的内存偏移地址 ; 功能号 mov al,1 ;扇区数 mov d1,0 ;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 ;起始扇区号 ; 起始编号为1 mov c1,3int 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

2020/5/5 14:03	Assembler Source	6 KB
2020/5/5 14:04	Assembler Source	6 KB
2020/5/5 14:05	Assembler Source	6 KB
2020/5/5 14:06	Assembler Source	6 KB
2020/5/5 17:43	Assembler Source	6 KB
	2020/5/5 14:04 2020/5/5 14:05 2020/5/5 14:06	2020/5/5 14:04 Assembler Source 2020/5/5 14:05 Assembler Source 2020/5/5 14:06 Assembler Source

编写好的汇编语言源程序

(二) 、利用nasm编译汇编语言源程序

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

- J			
HEX auxeer a	2020/5/5 20:01	BIN 文件	1 KB
HEX b	2020/5/5 20:01	BIN 文件	1 KB
HEX C	2020/5/5 20:01	BIN 文件	1 KB
HEX d	2020/5/5 20:01	BIN 文件	1 KB
myboot2	2020/5/5 20:01	BIN 文件	1 KB

(三)、将程序写入软盘的相应位置

首先用winhex打开编译好的各个bin文件和软盘映像文件

testOS.img	a.bin b.bin c.bin	d.bin myboot2.bin		
Offset	0 1 2 3	4 5 6 7 8		ANSI ASCII
00000000	■4 06 B0 00 B5	35 00 B1 00 B6 1	.8 B2 4F B7 0F CD 10 8C C8 8E D8 BD 28 7D 8C D8 8E C0 B9 32 00 B8 01 🊪 ° μ ± ¶ °ο·ί ŒÈŽØ≒(}ŒØŽÀ¹2 ,
00000020	13 BB 07 00 B	36 00 B2 00 CD 1	.0 BD 7F 7D B9 1A 00 B6 01 B2 00 CD 10 BD 99 7D B9 1A 00 B6 02 B2 00 » ¶ ° í ¾ } ¹ ¶ ° í	1-4™}1 ¶ 2
00000040	CD 10 BD B3 71	7D B9 1A 00 B6 0		
00000060	B3 00 CD 10 30	3C 41 74 0E 3C 4	12 74 36 3C 43 74 5E 3C 44 0F 84 84 00 B8 5A 7D 89 C5 B9 25 00 B8 01 8 1 <at ""<="" <bt6<ct^<d="" td=""><td>,Z}%å¹%,</td></at>	,Z}%å ¹ %,
00000080	13 BB 0C 00 B2	32 00 B6 05 CD 1		¶µ±íé
000000A0	5E 04 B8 5A 71	7D 89 C5 B9 25 0	00 B8 01 13 BB 0C 00 B2 28 B6 05 CD 10 8C C8 8E C0 BB 00 85 B4 02 B0 ^ ,Z}%Å1% , » 2(¶ f	
00000000	01 B2 00 B6 00	00 B5 00 B1 04 C	2D 13 E9 32 08 B8 5A 7D 89 C5 B9 25 00 B8 01 13 BB 0C 00 B2 00 B6 0F = ¶ μ ± Í é2 ¸Z}\Åå%	
000000E0	CD 10 8C C8 8F	BE CO BB 00 89 B		
00000100	25 00 B8 01 13	13 BB 0C 00 B2 2		°°¶µ±
00000120	06 CD 13 E9 DA	DA OF EB FE 48 6	55 6C 6C 6F 2C 20 50 6C 65 61 73 65 20 73 65 6C 65 63 74 20 74 68 65 Í éÚ ëþHello, Please	select the
00000140	20 70 72 6F 6	57 72 61 6D 20 7	9 6F 75 20 77 61 6E 74 20 74 6F 20 72 75 6E 2E 20 48 65 6C 6C 6F 2C program you want to r	un. Hello,
00000160	20 4D 79 4F 73	73 20 69 73 20 6	5C 6F 61 64 69 6E 67 20 75 73 65 72 20 70 72 6F 67 72 61 6D 20 20 50 MyOs is loading user	program P
00000180	72 67 31 4E 63	51 6D 65 3A 41 2	0 20 53 65 63 74 6F 72 4E 75 6D 3A 20 33 20 20 50 72 67 32 4E 61 6D rg1Name:A SectorNum:	3 Prg2Nam
000001A0	65 3A 42 20 20	20 53 65 63 74 6	F 72 4E 75 6D 3A 20 34 20 20 50 72 67 33 4E 61 6D 65 3A 43 20 20 53 e:B SectorNum: 4 Prg	3Name:C S
000001C0	65 63 74 6F 72	72 4E 75 6D 3A 2	0 35 20 20 50 72 67 34 4E 61 6D 65 3A 44 20 20 53 65 63 74 6F 72 4E ectorNum: 5 Prg4Name:	D SectorN
000001E0	75 6D 3A 20 36	36 20 20 00 00 0	00 00 00 00 00 00 00 00 00 00 00 00 00	υa

myboot2.bin即监控程序

testOS.img	a.bin	b.	bin	c.b	in	d.bi	n n	nybo	ot2.b	in																							
Offset	0	1	2	3	4	5	6	7	8	9	A	В	С	D	Ε	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	ANSI ASCII
00000000	8C	C8	8E	C0	8E	D8	8E	C0	В8	C8	82	89	C5	в9	11	00	В8	01	13	BB	0C	00	B2	00	В6	05	CD	10	В8	00	В8	8E	œèžàžøžà¸è,%å¹ , » º¶í , ž
00000020	C0	C6	06	E2	82	41	FF	0E	D9	82	75	FA	C7	06	D9	82	50	C3	FF	0E	DB	82	75	EE	C7	06	D9	82	50	C3	C7	06	ÀÆ â,Aÿ Ů,uúÇ Ù,PÃÿ Û,uîÇ Ù,PÃÇ
00000040	DB	82	44	02	B0	01	3A	06	DD	82	74	1E	в0	02	3A	06	DD	82	74	52	B0	03	3A	06	DD	82	0F	84	83	00	B0	04	Û,D ° : Ý,t ° : Ý,tR° : Ý, "f °
00000060	3A	06	DD	82	0F	84	В3	00	EB	FE	FF	06	DE	82	FF	06	E0	82	8B	1E	DE	82	в8	0F	00	29	D8	74	0E	8B	1E	E0	: Ý, "° ëþÿ Þ,ÿ à,< Þ,)Øt < à
00000080	82	В8	28	00	29	D8	74	10	E9	CA	00	C7	06	DE	82	0D	00	C6	06	DD	82	02	EB	E5									, ()Øt éÊÇÞ, ÆÝ, ëåÇà,&Æ
000000A0	DD	82	04	E9	AF	00	FF	0E	DE	82	FF	06	Ε0	82	8B	1E	E0	82	В8	28	00	29	D8	74									Ý, é ÿ Þ,ÿ à,< à, ()Øt < Þ, ,
00000000	29	D8	74	10	E9	8E	00	C7	06	E0	82	26	00	C6	06	DD	82	03	EB	E5	C7	06	DE	82	07	00	C6	06	DD	82	01	EB)Øt éŽ Ç à,& Æ Ý, ëåÇ Þ, Æ Ý, ë
000000E0	74	FF	0E	DE	82	FF	0E	E0	82	8B	1E	DE	82	в8	05	00	29	D8	74	0D	8B	1E	E0	82	В8	FF	FF	29	D8	74	0F	EB	tỷ Þ,ỷ à,< Þ,)Øt < à,,ÿÿ)Øt ë
00000100	54	C7	06	DE	82	07	00	C6	06	DD	82	04	EB	E6	C7	06	E0	82	01	00	C6	06	DD	82	02	EB	3A	FF	06	DE	82	FF	TÇ Þ, Æ Ý, ëæÇ à, Æ Ý, ë:ÿ Þ,ÿ
00000120	0E	E0	82	8B	1E	E0	82	B8	FF	FF	29	D8	74	0D	8B	1E	DE	82	В8	0F	00	29	D8	74	0F	EB	1A	C7	06	E0	82	01	à, < à, , ÿÿ)Øt < Þ, ,)Øt ë Ç à,
00000140	00	C6	06	DD	82	01	EB	E6	C7	06	DE	82	0D	00	C6	06	DD	82	03	EB	00	FE	06	E2	82	8A	1E	E2	82	B0	5B	28	ÆΎ, ĕœÇ Þ, ÆΎ, ĕ þâ,Šâ,°[(
00000160	D8	74	10	FE	06	E3	82	8A	1E	E3	82	B0	20	28	D8	74	09	EB	0E	C6	06	E2	82	41	EB	E9	C6	06	E3	82	11	EB	Øt þã,Šã,° (Øt ë Æâ,AëéÆã, ë
00000180	00	31	C0	A1	DE	82	BB	50	00	F7	E3	03	06	E0	82	BB	02	00	F7	E3	89	C3	8A	26	E3	82	A0	E2	82	26	89	07	là;D,»P ÷ã à,» ÷ã‰ÃŠ&ã, â,&‰
000001A0	FF	06	E4	82	A1	E4	82	3D	00	01	0F	85	78	FE	8C	C8	8E	C0	BB	00	7C	B4	02	B0	01	B2	00	В6	00	B5	00	В1	ÿä,;ä,=xþŒÈŽÀ» ′°²¶μ±
000001C0	01	CD	13	E9	3A	F9	EB	FE	31	38	33	34	30	31	36	31	20	54	69	61	6E	20	52	75	69	50	C3	44	02	01	06	00	Í é:ùĕþ18340161 Tian RuiPÃD
000001E0	00	00	41	11	00	00																											A

第一个子程序

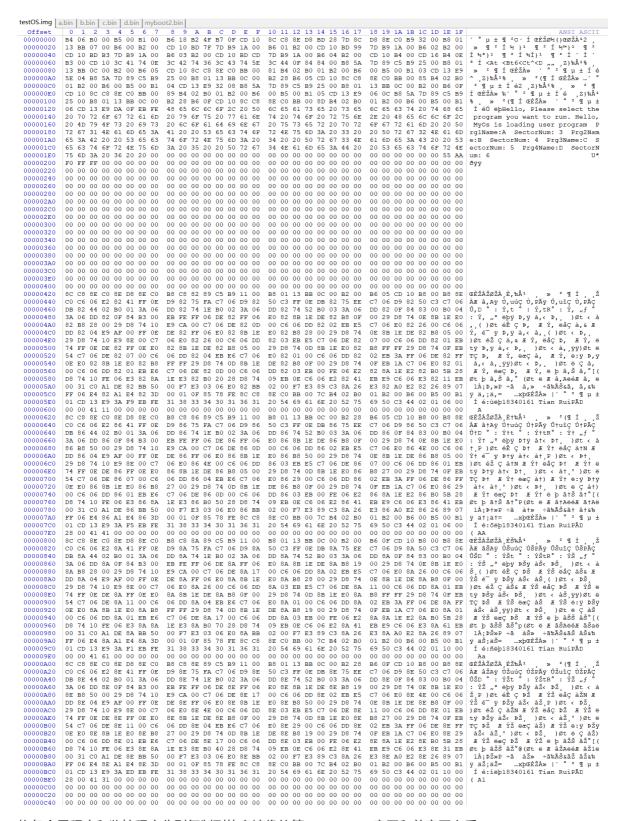
testOS.img a	.bin	b.b	oin]	c.b	n	d.bii	n n	nybo	ot2.b	oin																							
Offset	0	1	2 '	3	4	5	6	7	8	9	A	В	С	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1c	1D	1E	1F	ANSI ASCII
00000000	8C	C8	8E	C0	8E	D8	8E	C0																									œèžàžøžà,è†%å¹, »²(¶í, ¸ž
00000020		C6																															ÀÆ â†Ay ĎtuúÇ Ď†PÃY Û†uîÇ Ď†PÃÇ
00000040																																	Û†D ° : ݆t ° : ݆tR° : ݆ "f °
00000060																																	: Ýt "° ëþÿ Þtÿ àt< Þt,)Øt < à
00000080																																	t,P)Øt éÊÇÞt ÆÝt ëåÇ àtNÆ
0A00000A0																																	Ýt é ÿ Þtÿ àt< àt,P)Øt < Þt,
00000000) Øt éž ç à†N Æ Ý† ëåç Þ† Æ Ý† ë
000000E0																																	tÿ Þ†ÿ à†< Þ†,)Øt < à†,')Øt ë
00000100																																	TÇ Þ† Æ Ý† ëæÇ à†) Æ Ý† ë:ÿ Þ†ÿ
00000120																																	àt< àt,')Øt < Þt,)Øt ë Ç àt)
00000140																																	Æ Ýt ëæÇ Þt Æ Ýt ë þ âtš ât°[(
00000160	D8	74	10	FE	06	E3	86	8A																									Øt þ ㆊ ㆰP(Øt ë Æ â†AëéÆ ã†Aë
00000180	00	31	C0	A1	DE	86	BB	50																									là;Þ†»Þ ÷ā à†» ÷ā‱āš&↠â†&%
000001A0	FF	06	E4	86	A1	E4	86	3D																									ÿät;ät=xþŒÈŽÀ» ′°°¶µ±
000001C0	01	CD	13	E9	3A	F5	EB	FE	31	38	33	34	30	31	36	31	20	54	69	61	6E	20	52	75	69	50	C3	44	02	01	06	00	Í é:őëþ18340161 Tian RuiPÃD
000001E0	28	00	41	41	00	00																											(AA

第二个子程序

testOS.img a	.bin b	.bin	c.b	in	d.bii	n n	nybo	ot2.b	oin																							
Offset	0	1 2	3	4	5	6	7	8	9	A	В	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	ANSI ASCII
00000000	8C C	8E	C0	8E	D8	8E	C0	В8	C8	8A	89	C5	в9	11	00	В8	01	13	BB	0C	00	B2	00	В6	0F	CD	10	В8	00	В8	8E	
00000020	C0 C	6 06	E2	8A	41	FF	0E	D9	8A	75	FA	C7	06	D9	8A						8A										06	
00000040	DB 82	A 44	02	B0	01	3A	06	DD	8A	74	1E	B0	02	3A	06	DD	8A	74	52	B0	03	3A	06	DD	8A	0F	84	83	00	B0	04	
00000060	3A 0	6 DD	8A	0F	84	вз	00	EB	FE	FF	06	DE	8A	FF	06	E0	8A	8B	1E	DE	8A	В8	19								E0	
00000080	8A B	3 28	00	29	D8	74	10							8A							02											š,()Øt éÊÇÞŠ ÆÝŠ ëåÇ àŠ&Æ
000000A0	DD 82	A 04	E9	AF	00	FF	0E														29											ÝŠ é ÿ ÞŠÿ àŠ< àŠ,()Øt < ÞŠ,
000000C0	29 D	3 74	10	E9	8E	00	C7																								EB	
000000E0	74 F	F OE	DE	8A	FF	0E	E0							0F							1E										EB	
00000100	54 C													C7																	FF	
00000120	OE E	8 C	8B	1E	E0	8A	B8	FF	FF	29	D8	74	0D	8B	1E	DE	8A	В8	19	00	29	D8	74	0F								àš< àš,yy)øt < ⊅š,)øt ë Ç àš
00000140	00 C	6 06	DD	8A	01	EB	E6	C7						C6							FE										28	Æ ÝŠ ëœÇ ÞŠ Æ ÝŠ ë þ ⊊ ⊰[(
00000160	D8 7													D8							E2										EB	
00000180	00 3													8A							C3			E3								là;Þš»P ÷ā àš» ÷ā%Āš&āš âš&%
000001A0	FF 0													8C							B4										B1	
000001C0	01 CI					EB	FE	31	38	33	34	30	31	36	31	20	54	69	61	6E	20	52	75	69	50	C3	44	02	01	10	00	Í é:ñëþ18340161 Tian RuiPÃD
000001E0	00 00) 41	61	00	00																											Aa

第三个子程序

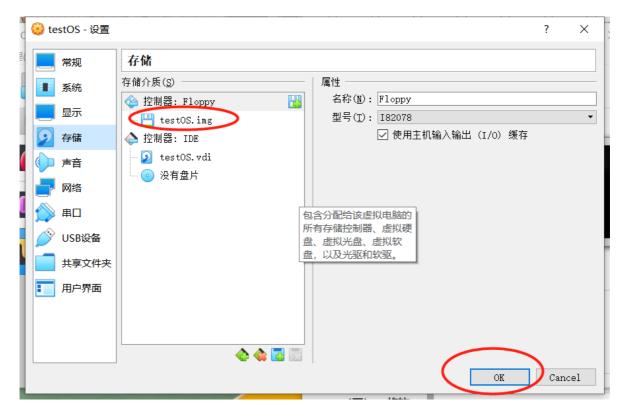
Offset 0 1 2 0000000 8c C8 8c 00000020 C0 C6 06 00000040 DB 8c 40 00000060 3a 06 DD 00000080 8c B8 50 00000000 DB 8c 04 00000000 74 FF 08 00000000 74 FF 08		6 7 8E CO	8 9 A B B8 C8 8E 89 C	C D E F 1	10 11 12 13 14 15 16 17	18 19 1A 1B 1C 1D 1E 1F	AMOT AGOTT
00000020 C0 C6 06 00000040 DB 8E 44 00000060 A 06 DD 00000080 8E B8 50 000000A0 DD 8E 04 000000C0 29 D8 74 000000E0 74 FF 0E		8E C0	DO CO OE OO C				ANSI ASCII
00000040 DB 8E 44 00000060 3A 06 DD 00000080 8E B8 50 00000000 DD 8E 04 00000000 29 B8 74 00000000 74 FF 0E	06 E2 8E 41		DO CO SE 89 C	C5 B9 11 00 B	38 01 13 BB 0C 00 B2 28	B6 OF CD 10 B8 00 B8 8E	
00000060 3A 06 DD 00000080 8E B8 50 000000A0 DD 8E 04 000000C0 29 D8 74 000000E0 74 FF 0E	44 02 B0 01	FF OE	D9 8E 75 FA C	C7 06 D9 8E 5	50 C3 FF 0E DB 8E 75 EE	C7 06 D9 8E 50 C3 C7 06	ÀÆ âŽAŸ ÙŽuúÇ ÙŽPß ÛŽuîÇ ÙŽPÃÇ
00000080 8E B8 50 000000A0 DD 8E 04 000000C0 29 D8 74 000000E0 74 FF 0E	J. DO 0.	3A 06	DD 8E 74 1E B		DD 8E 74 52 B0 03 3A 06	DD 8E 0F 84 83 00 B0 04	ÛŽD ° : ÝŽt ° : ÝŽtR° : ÝŽ "f °
000000A0 DD 8E 04 000000C0 29 D8 74 000000E0 74 FF 0E	DD 8E 0F 84		EB FE FF 06 D		EO 8E 8B 1E DE 8E B8 19		: ÝŽ "° ĕþÿ ÞŽÿ àŽ< ÞŽ,)Øt < à
000000C0 29 D8 74 000000E0 74 FF 0E	50 00 29 D8	74 10	E9 CA 00 C7 0	06 DE 8E 17 0	00 C6 06 DD 8E 02 EB E5		Ž,P)Øt éÊÇ ÞŽ Æ ÝŽ ëåÇ àŽN Æ
000000E0 74 FF 0E	04 E9 AF 00		DE 8E FF 06 E		EO 8E B8 50 00 29 D8 74		ÝŽ é ÿ ÞŽÿ àŽ< àŽ,P)Øt < ÞŽ,
	74 10 E9 8E		06 E0 8E 4E 0		BE 03 EB E5 C7 06 DE 8E) Øt éž ç àžN Æ Ýž ëåç Þž Æ ÝŽ ë
			8E 8B 1E DE 8				tÿ ÞŽÿ àŽ< ÞŽ,)Øt < àŽ,')Øt ë
	06 DE 8E 11	00 C6	06 DD 8E 04 E				TÇ ÞŽ Æ ÝŽ ëæÇ àŽ) Æ ÝŽ ë:ÿ ÞŽŸ
00000120 OE E0 8E	BE 8B 1E E	8E B8			DE 8E B8 19 00 29 D8 74	OF EB 1A C7 06 E0 8E 29	àž< àž,')Øt < Þž,)Øt ë Ç àž)
00000140 00 C6 06	06 DD 8E 01	EB E6			DD 8E 03 EB 00 FE 06 E2	8E 8A 1E E2 8E B0 5B 28	Æ ÝŽ ëœÇ ÞŽ Æ ÝŽ ë þ ⎊ ⎰[(
00000160 D8 74 10	10 FE 06 E3		1E E3 8E B0 4				Øt þ ㎊ ㎰0(Øt ë Æ âŽAëéÆ ãŽ1ë
	CO A1 DE 81	BB 50	00 F7 E3 03 0			E3 8E A0 E2 8E 26 89 07	là;Þž»P ÷ã àž» ÷ã‰ãš&ãž âž&‰
000001A0 FF 06 E4	E4 8E A1 E4	8E 3D	00 01 OF 85 7		BE CO BB 00 7C B4 02 B0	01 B2 00 B6 00 B5 00 B1	
000001C0 01 CD 13		EB FE	31 38 33 34 3	30 31 36 31 2	20 54 69 61 6E 20 52 75	69 50 C3 44 02 01 10 00	Í é:íëþ18340161 Tian RuiPÃD
000001E0 28 00 41	41 31 00 00						(A1



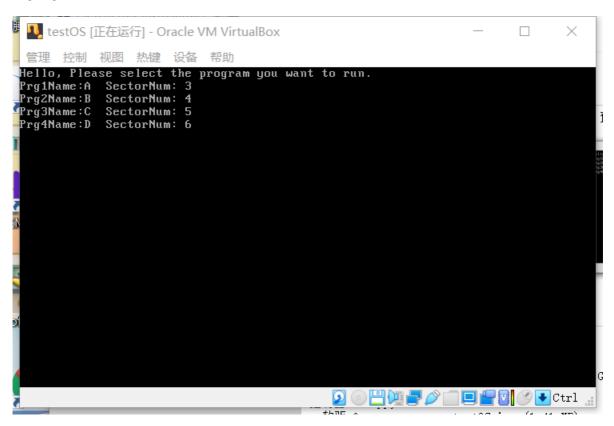
将各个子程序和监控程序分别复制到软盘镜像的第3,4,5,6扇区和首扇区之后

事实上子程序是可以放在任意扇区的,这里将第二扇区空出来的本意是想在第二扇区存储一个程序输出指令解释的程序,但是没有成功,所以就空出了一个扇区。

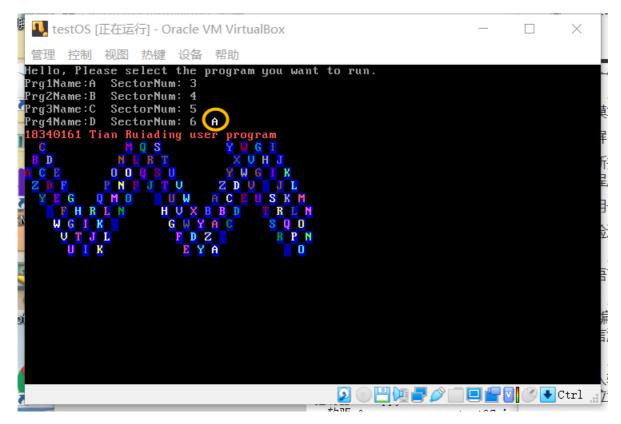
(四) 、将软盘挂载到虚拟机上



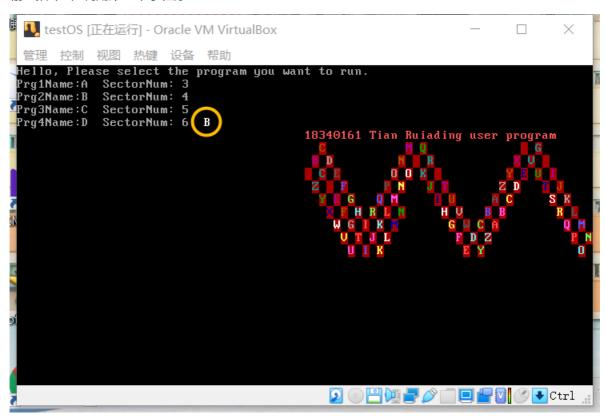
(五)、运行虚拟机



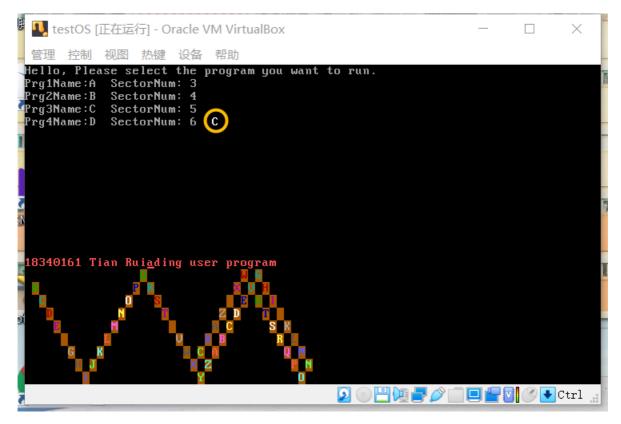
引导程序



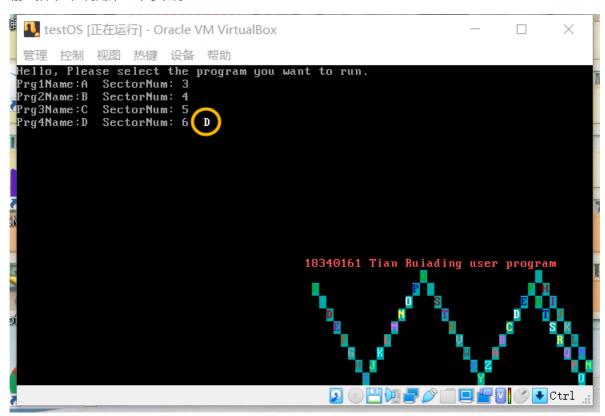
输入指令A,调用第一个子程序



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].<u>https://blog.csdn.net/hua19880705/article/details/8125706,2012-10-29</u>.

[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/jiaguoxinzhi/article/details/2949304?utm_medium=distribute.pc_relevant_t0.none-task-blog-Blog CommendFromMachineLearnPai2-1&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.