

实验 2：加载用户程序的监控程序

姓名：姜洋帆 学号：17341068

院系：数据科学与计算机学院 专业：17 级计算机科学（大数据）

【实验题目】加载用户程序的监控程序

【实验目的】

了解 BIOS 中断，学习监控程序如何获取计算机硬件系统的控制权、如何控制用户程序的执行，并开发出最原始的操作系统。

【实验要求】

设计四个有输出的用户可执行程序，分别在屏幕 1/4 区域动态输出字符，如将用字符‘A’从屏幕左边某行位置 45 度角下斜射出，保持一个可观察的适当速度直线运动，碰到屏幕相应 1/4 区域的边后产生反射，改变方向运动，如此类推，不断运动；在此基础上，增加你的个性扩展，如同时控制两个运动的轨迹，或炫酷动态变色，个性画面，如此等等，自由不限。还要在屏幕某个区域特别的方式显示你的学号姓名等个人信息。

【实验方案】

一、硬件及虚拟机配置：Lenovo PC ； Oracle Virtual Box

二、软件工具及作用：

Notepad++ ：写代码

Nasm：编译 asm 文件到 bin 文件

Sublime ：查看、编辑二进制文件，将编译完成的程序机器码写入软盘

C 程序：自己编写的一个 C 语言程序，用来将编译生成的二进制文件填充为 1.44M 大小，并将后缀改为.img，生成 1.44M 的软盘

三、程序功能

Myos1 作为引导，键入 A 执行程序 A，功能与实验一相同，显示区域限制在左上角 1/4 区域内，程序内部设一个时钟，达到时间后返回引导程序，键入 B 执行程序 B，同样显示一个字符移动并反弹，取消了控制字符长度的功能，保留变色，显示区域为右上角 1/4 个区域。第三个程序是显示一个字符串（个人信息）并改变前景和背景颜色，一共改变十次，然后结束程序跳回引导。第四个程序为

发射一个字符（屏幕上仅保留一个字符在移动），并显示字符串。字符每次碰撞到上下边界时会改变中间字符串的颜色，同时字符自身的颜色也在不断改变。

四、程序设计

在 myos 显示字符之前进行清屏，显示字符后加入键入选择模块和跳转模块，其中第二个程序的跳转中将 c1 置为 3，将第二段程序的 bin 文件放入虚拟软盘的第三个扇区，也就是 400h 开始。

在原有的 stoneM 代码里面添加 loop 循环指令和跳转模块回跳到 myos

第三个程序（b.asm）是在 stoneM 基础上把显示的字符改成 B，并且换掉字符颜色。

第四个是编写的新的程序(try.asm)，显示出一个菱形并且用 10h 中断的 06 号功能上滚和 07 号功能下滚让它上下运动，并且设置循环让他运动五个来回。之后输入字符‘m’退出程序回到引导程序。

五、主要代码

Myos1 引导程序：

；程序源代码（myos1.asm）

```
org 7c00h          ; BIOS 将把引导扇区加载到 0:7C00h 处，并开始执行
OffSetOfUserPrg1 equ 0A100h
OffSetOfUserPrg2 equ 0A200h
OffSetOfUserPrg3 equ 0A300h
OffSetOfUserPrg4 equ 0A400h
```

Start:

```
mov ax, cs          ; 置其他段寄存器值与 CS 相同
mov ds, ax          ; 数据段
mov bp, Message     ; BP=当前串的偏移地址
mov ax, ds           ; ES:BP = 串地址
mov es, ax           ; 置 ES=DS
mov cx, 44           ; CX = 串长 (=9)
mov ax, 1301h        ; AH = 13h（功能号）、AL = 01h（光标置于串尾）
mov bx, 0007h        ; 页号为 0(BH = 0) 黑底白字(BL = 07h)
mov dh, 0            ; 行号=0
mov dl, 0            ; 列号=0
int 10h              ; BIOS 的 10h 功能：显示一行字符
;press a key
mov ah,0
int 16h

cmp al,'a'
je LoadnEx1
cmp al,'b'
je LoadnEx2
cmp al,'c'
je LoadnEx3
```

```

    cmp al,'d'
    je LoadnEx4

```

LoadnEx1:

```

;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,2                 ;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

```

LoadnEx2:

```

;读软盘或硬盘上的若干物理扇区到内存的 ES:BX 处:
...
mov bx, OffsetOfUserPrg2 ;偏移地址; 存放数据的内存偏移地址
...
jmp OffsetOfUserPrg2

```

LoadnEx3:

```

;读软盘或硬盘上的若干物理扇区到内存的 ES:BX 处:
...
mov bx, OffsetOfUserPrg3 ;偏移地址; 存放数据的内存偏移地址
...
jmp OffsetOfUserPrg3

```

LoadnEx4:

```

...
mov bx, OffsetOfUserPrg4 ;偏移地址; 存放数据的内存偏移地址
...
jmp OffsetOfUserPrg4

```

AfterRun:

```

    jmp $                ;无限循环

```

Message:

```

    db 'Hello, MyOs is loading user program...'

```

MessageLength equ (\$-Message)

```

    times 510-($-$$) db 0
    db 0x55,0xaa

```

程序中返回引导程序的模块:

return:

```

    mov ah,0 ; 功能号
    int 16h ; 调用 16H 号中断
    ; 按任一键: 退出返回 DOS
    jmp 07c00h

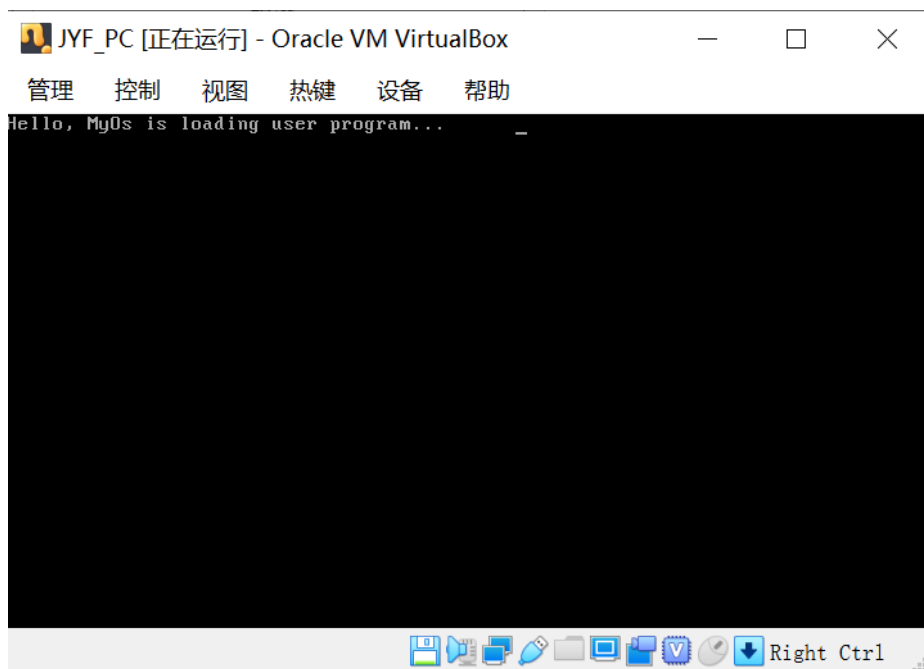
```

实现时, 设置为满足一定条件触发调用 return, 返回引导程序

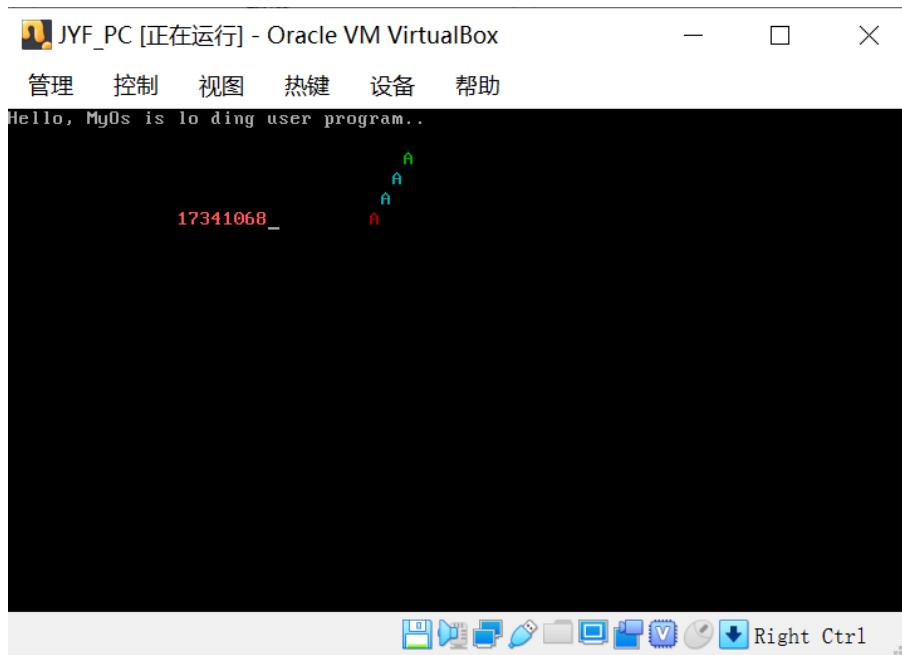
【实验过程】

功能展示

引导程序界面，打印信息，在键盘键入 ABCD 四个不同字母后，会分别进入第 1、2、3、4 个程序显示不同信息。不同程序分别显示在屏幕的左上、右上左下、右下四个 1/4 区域内。



键入 A，开始执行第一个程序，在左上 1/4 个区域内显示



JYF_PC [正在运行] - Oracle VM VirtualBox

管理 控制 视图 热键 设备 帮助

Hello, MyOs is loading user program...

ASCII art graphic of a person with the text 'jyf 17341068' in the center.

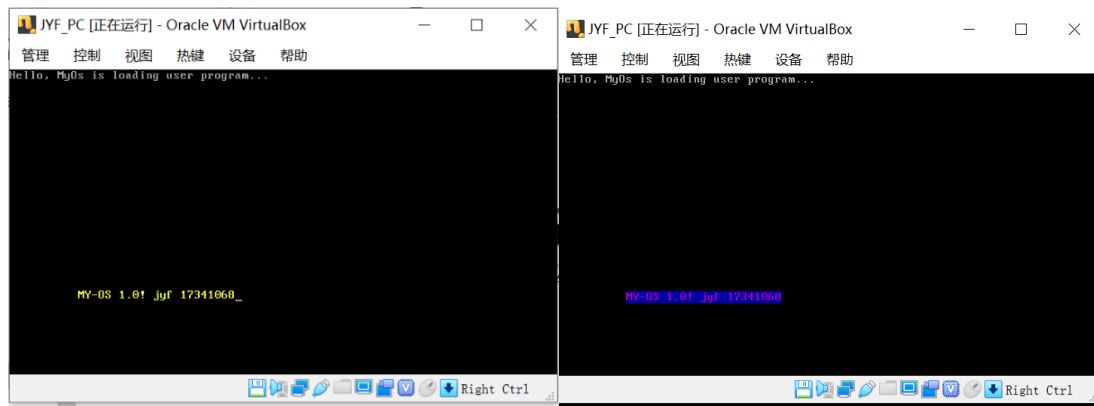
Right Ctrl

The image displays two side-by-side screenshots of a Windows Virtual Machine (VM) running in Oracle VM VirtualBox. Both windows have a title bar that reads "JYF_PC [正在运行] - Oracle VM VirtualBox".

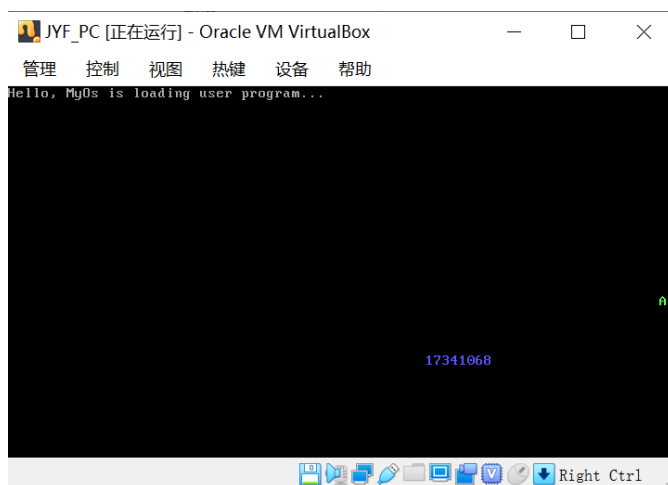
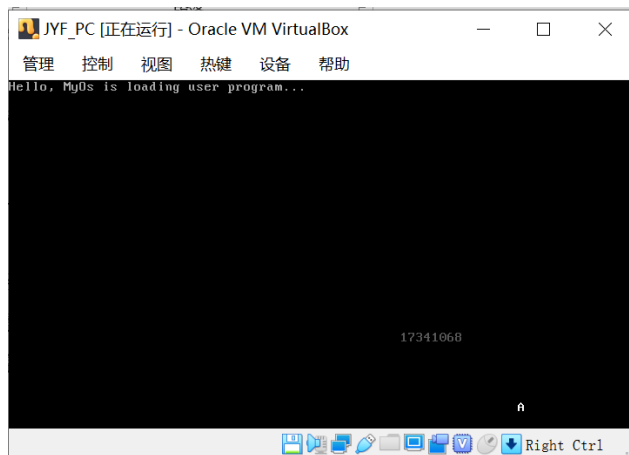
The left window shows a terminal window with a black background. The text "Hello, MyOS is loading user program..." is displayed in white. Below this, a prompt "MY-OS 1.0! juf 17341068_" is visible in white. The window has a menu bar with the following items: "管理", "控制", "视图", "热键", "设备", and "帮助".

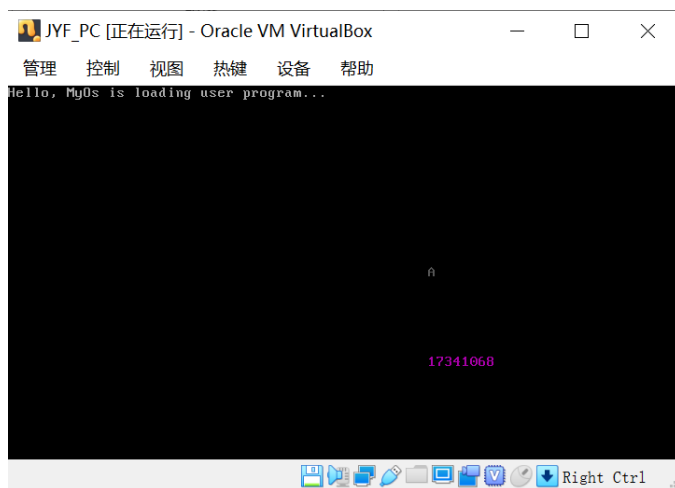
The right window shows a similar terminal window. The text "Hello, MyOS is loading user program..." is displayed in white. Below this, a prompt "MY-OS 1.0! juf 17341068" is visible in white. The window has a menu bar with the following items: "管理", "控制", "视图", "热键", "设备", and "帮助".

At the bottom of each window, a Windows taskbar is visible, showing various icons and the text "Right Ctrl".



键入 D 进入第 4 个程序，显示区域限制在右下方，功能为一个字符在区域内不断移动、反弹，中间显示一个字符串，字母每在上下边框反弹一次，中间的字符就改变一次颜色，改变 10 次颜色后退出，返回引导程序。





中间文件：

Study > 大二下 > Operation System > Exp#2 > asm&img

搜索"asm&img"

名称	修改日期	类型	大小
myos1	2019/3/15 14:35	文件	1 KB
myos1.asm	2019/3/15 14:23	Assembler Source	4 KB
myos1.img	2019/3/15 14:43	光盘映像文件	1,440 KB
p1	2019/3/15 14:35	文件	1 KB
p1.asm	2019/3/15 14:34	Assembler Source	5 KB
p1.img	2019/3/15 14:35	光盘映像文件	1,440 KB
p2	2019/3/15 14:38	文件	1 KB
p2.asm	2019/3/15 14:38	Assembler Source	4 KB
p2.img	2019/3/15 14:38	光盘映像文件	1,440 KB
p3	2019/3/15 14:41	文件	1 KB
p3.asm	2019/3/15 14:23	Assembler Source	1 KB
p3.img	2019/3/15 14:41	光盘映像文件	1,440 KB
p4	2019/3/15 14:42	文件	1 KB
p4.asm	2019/3/15 14:22	Assembler Source	5 KB
p4.img	2019/3/15 14:42	光盘映像文件	1,440 KB

p1-p4 为 4 个程序，在写入装载引导的软盘之前，都进行过独立的测试。附件内的程序文件 org 值以已改为引导程序设定的偏移量，若要测试，请手动将 org 偏移值修改为 07c00h，重新编译制作软盘映像。

映像写入：

直接在 myos1 的软盘映像文件进行操作，将程序 1、2、3、4 分别写入第 2、3、4、5 个扇区，对应的启示地址为 513、1025、1537、2049 字节。先将 4 个程序正确设定内存的偏移值，编译成功后将对应的数据复制即可。

D:\Study\大二下\Operation System\Exp#2\asm&img\myos1.img - Sublime T... — □ ×

File Edit Selection Find View Goto Tools Project Preferences Help

EXP1 × EXP1.img × JVF.img × myos1.img ×

```
1 8cc8 8ed8 bd90 7c8c d88e c0b9 2c00 b801
2 13bb 0700 b600 b200 cd10 b400 cd16 3c61
3 740c 3c62 7420 3c63 7434 3c64 7448 8cc8
4 8ec0 bb00 a1b4 02b0 01b2 00b6 00b5 00b1
5 02cd 13e9 ba24 8cc8 8ec0 bb00 a2b4 02b0
6 01b2 00b6 00b5 00b1 03cd 13e9 a225 8cc8
7 8ec0 bb00 a3b4 02b0 01b2 00b6 00b5 00b1
8 04cd 13e9 8a26 8cc8 8ec0 bb00 a4b4 02b0
9 01b2 00b6 00b5 00b1 05cd 13e9 7227 ebfe
10 4865 6c6c 6f2c 204d 794f 7320 6973 206c
11 6f61 6469 6e67 2075 7365 7220 7072 6f67
12 7261 6d2e 2e2e 0000 0000 0000 0000 0000
13 0000 0000 0000 0000 0000 0000 0000 0000
14 0000 0000 0000 0000 0000 0000 0000 0000
15 0000 0000 0000 0000 0000 0000 0000 0000
16 0000 0000 0000 0000 0000 0000 0000 0000
17 0000 0000 0000 0000 0000 0000 0000 0000
18 0000 0000 0000 0000 0000 0000 0000 0000
19 0000 0000 0000 0000 0000 0000 0000 0000
20 0000 0000 0000 0000 0000 0000 0000 0000
21 0000 0000 0000 0000 0000 0000 0000 0000
22 0000 0000 0000 0000 0000 0000 0000 0000
23 0000 0000 0000 0000 0000 0000 0000 0000
24 0000 0000 0000 0000 0000 0000 0000 0000
25 0000 0000 0000 0000 0000 0000 0000 0000
26 0000 0000 0000 0000 0000 0000 0000 0000
27 0000 0000 0000 0000 0000 0000 0000 0000
28 0000 0000 0000 0000 0000 0000 0000 0000
29 0000 0000 0000 0000 0000 0000 0000 0000
30 0000 0000 0000 0000 0000 0000 0000 0000
31 0000 0000 0000 0000 0000 0000 0000 0000
32 0000 0000 0000 0000 0000 0000 0000 55aa
33 b403 8cc8 8ec0 8ed8 8ec0 b800 b88e e8c6
```

Line 1, Column 1 Tab Size: 4 Plain Text

D:\Study\大二下\Operation System\Exp#2\asm&img\myos1.img - Sublime ...

File Edit Selection Find View Goto Tools Project Preferences Help

EXP1 EXP1.img JVF.img myos1.img

```
32 0000 0000 0000 0000 0000 0000 0000 55aa
33 b403 8cc8 8ec0 8ed8 8ec0 b800 b88e e8c6
34 06f3 a241 ff0e d8a2 75fa ff0e daa2 75f4
35 c706 d8a2 50c3 c706 daa2 4402 b001 3a06
36 dca2 741e b002 3a06 dca2 745e b003 3a06
37 dca2 0f84 9300 b004 3a06 dca2 0f84 c600
38 ebfe e836 01ff 06df a2ff 06e1 a28b 1edf
39 a2b8 0d00 29d8 740e 8b1e e1a2 b828 0029
40 d874 19e9 dd00 ff0e dda2 0f84 82da c706
41 dfa2 0b00 c606 dca2 02e9 c700 c706 e1a2
42 2600 c606 dca2 04e9 b900 e8ee 00ff 0edf
43 a2ff 06e1 a28b 1ee1 a2b8 2800 29d8 740e
44 8b1e dfa2 b8ff ff29 d874 11e9 9500 c706
45 e1a2 2600 c606 dca2 03e9 8700 c706 dfa2
46 0100 c606 dca2 01eb 7ae8 af00 ff0e dfa2
47 ff0e e1a2 8b1e dfa2 b8ff ff29 d874 0d8b
48 1ee1 a2b8 ffff 29d8 740f eb57 c706 dfa2
49 0100 c606 dca2 04eb 4ac7 06e1 a201 00c6
50 06dc a202 eb3d e872 00ff 06df a2ff 0ee1
51 a28b 1ee1 a2b8 ffff 29d8 740d 8b1e dfa2
52 b80d 0029 d874 0feb 1ac7 06e1 a201 00c6
53 06dc a201 eb0d c706 dfa2 0b00 c606 dca2
54 03eb 00e8 2200 a1df a2bb 5000 f7e3 0306
55 e1a2 bb02 00f7 e389 c580 c401 a0f3 a265
56 8946 00e9 9efe ebfe bdf4 a2b9 0800 b801
57 13bb 0c00 b20f b605 cd10 c3a1 eba2 a3ef
58 a2a1 eda2 a3f1 a2a1 e7a2 a3eb a2a1 e9a2
59 a3ed a2a1 e3a2 a3e7 a2a1 e5a2 a3e9 a2a1
60 dfa2 a3e3 a2a1 e1a2 a3e5 a2a1 efa2 bb50
61 00f7 e303 06f1 a2bb 0200 f7e3 89c5 b400
62 a0f3 a265 8946 00c3 50c3 4402 0105 0007
63 0000 0007 0000 0007 0000 0007 0000 0007
64 0000 0041 3137 3334 3130 3638 0000 0000
65 b403 8cc8 8ec0 8ed8 8ec0 b800 b88e e8c6
66 0694 a341 ff0e 85a3 75fa c706 85a3 50c3
67 ff0e 87a3 75ee c706 85a3 50c3 c706 87a3
68 4402 b001 3a06 89a3 741e b002 3a06 89a3
```

Line 1, Column 1 Tab Size: 4 Plain Text

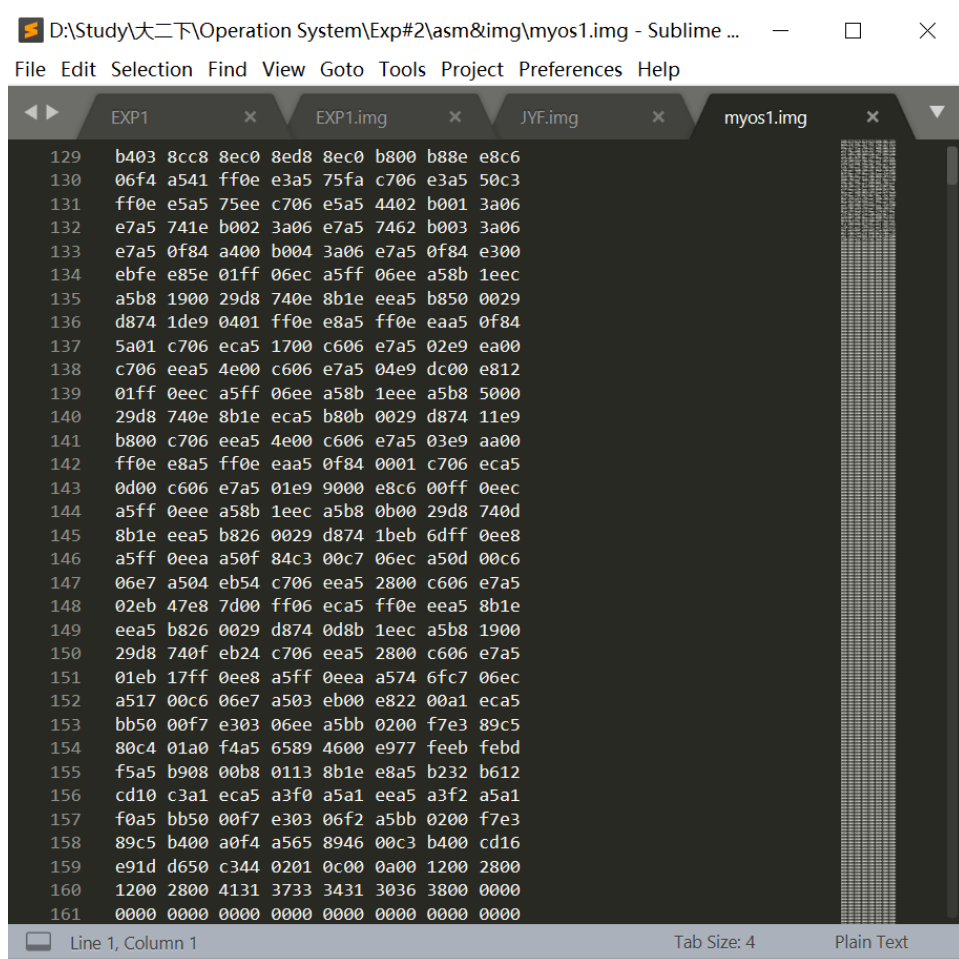
D:\Study\大二下\Operation System\Exp#2\asm&img\myos1.img - Sublime ...

File Edit Selection Find View Goto Tools Project Preferences Help

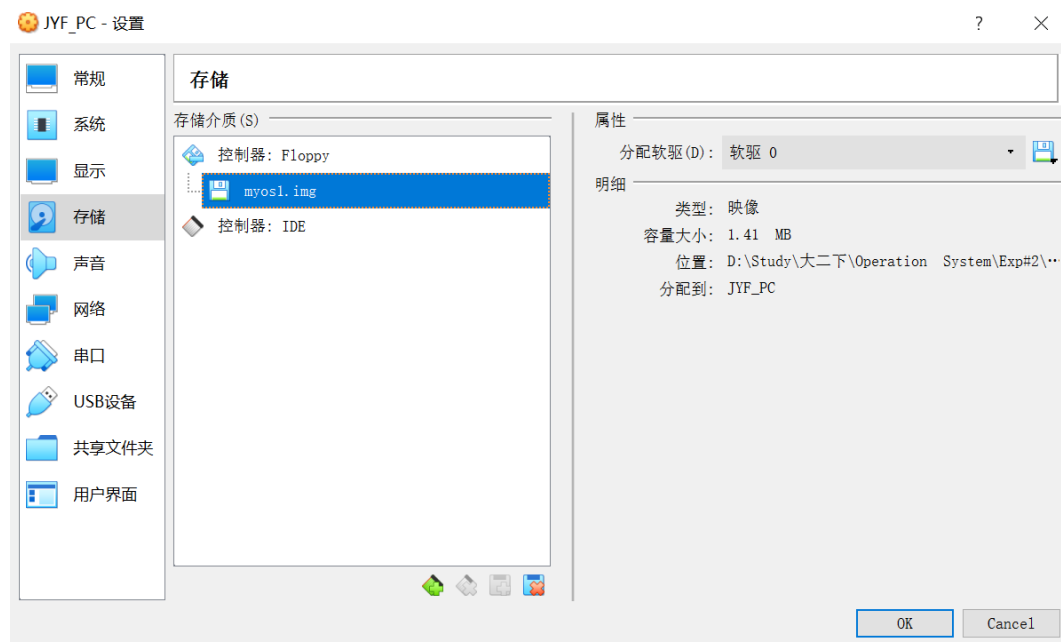
EXP1 EXP1.img JVF.img myos1.img

```
65 b403 8cc8 8ec0 8ed8 8ec0 b800 b88e e8c6
66 0694 a341 ff0e 85a3 75fa c706 85a3 50c3
67 ff0e 87a3 75ee c706 85a3 50c3 c706 87a3
68 4402 b001 3a06 89a3 741e b002 3a06 89a3
69 745b b003 3a06 89a3 0f84 8d00 b004 3a06
70 89a3 0f84 bd00 ebfe ff06 8ca3 ff06 8ea3
71 8b1e 8ca3 b80d 0029 d874 0e8b 1e8e a3b8
72 5000 29d8 7419 e9d4 00ff 0e8a a30f 847f
73 d9c7 068c a30b 00c6 0689 a302 e9be 00c7
74 068e a34e 00c6 0689 a304 e9b0 00ff 0e8c
75 a3ff 068e a3b8 1e8e a3b8 5000 29d8 740e
76 8b1e 8ca3 b8ff ff29 d874 11e9 8f00 c706
77 8ea3 4e00 c606 89a3 03e9 8100 c706 8ca3
78 0100 c606 89a3 01eb 74ff 0e8c a3ff 0e8e
79 a38b 1e8c a3b8 ffff 29d8 740d 8b1e 8ea3
80 b826 0029 d874 0feb 54c7 068c a301 00c6
81 0689 a304 eb47 c706 8ea3 2800 c606 89a3
82 02eb 3aff 068c a3ff 0e8e a38b 1e8e a3b8
83 2600 29d8 740d 8b1e 8ca3 b80d 0029 d874
84 0feb 1ac7 068e a328 00c6 0689 a301 eb0d
85 c706 8ca3 0b00 c606 89a3 03eb 00e8 2200
86 a18c a3bb 5000 f7e3 0306 8ea3 bb02 00f7
87 e389 c580 c401 a094 a365 8946 00e9 a4fe
88 ebfe bd95 a3b9 0c00 b801 13bb 0c00 b232
89 b607 cd10 c350 c344 0201 0500 0700 2800
90 0700 2800 416a 7966 2031 3733 3431 3036
91 3800 0000 0000 0000 0000 0000 0000 0000
92 0000 0000 0000 0000 0000 0000 0000 0000
93 0000 0000 0000 0000 0000 0000 0000 0000
94 0000 0000 0000 0000 0000 0000 0000 0000
95 0000 0000 0000 0000 0000 0000 0000 0000
96 0000 0000 0000 0000 0000 0000 0000 0000
97 8cc8 8ed8 8ec0 8ed0 b0c0 00ff 0e47 a375
98 faff 0e49 a375 f4c7 0647 a310 27c7 0649
99 a3b0 36bd 4da3 b917 00b8 0113 b20a b612
100 cd10 83c3 01ff 0e4b a375 d0b4 00cd 16b8
```

Line 1, Column 1 Tab Size: 4 Plain Text



虚拟机设置



【实验总结】

该实验是以实验一已经老师给出的样例代码为基础，需要编写一个引导程序，使得操作系统可以根据用户输入信息来选择加载内存中不同的程序运行。实验的关键在于理解监控程序运行的原理。在将所有程序编译出的二进制数据放入软盘后，机器从软盘启动，会根据汇编代码指定的 `org` 值，将程序数据放在内存的不同位置，然后默认从 `07c00h` 启动第一个程序，所以引导程序一定要将 `org` 设置为 `07c00h`。而剩余的待执行程序则根据汇编代码，放在内存的其他区域。通过 `jmp` 指令，可以从引导程序跳到相应的程序并执行。要注意，在设定不同程序的偏移地址时，地址要足够大，以确保能够容纳下整个程序，同时还要保证每个程序大小不超过 512 字节，这样才可以放在同一个扇区中，方便执行指令。

本次实验困难的地方，是最后将不同程序的机器码整合到同一个软盘映像中的步骤，本打算使用 `notepad++`，安装插件后进行二进制文件的操作，后来发现实现起来并不容易，对文件编辑操作的支持也并不友好，遂改为使用 `sublime` 修改二进制、磁盘映像文件。由于单独测试每个程序时的 `org` 偏移量与作为被引导的层序的偏移量不同，所以每次测试成功后都要修改 `org` 值重新编译。另外，设置偏移量也需要考虑清楚，不能随意设定一个值。自己设定的偏移量分别为 `0A100h`、`0A200h`、`0A300h`、`0A400h`。

、