

实验一

数据科学与计算机学院 计算机类 18 级 18340064 黄思蓉

一、 实验题目

编写一个引导扇区程序

二、 实验要求

接管裸机的控制权：

■ 搭建和应用实验环境

虚拟机安装，生成一个基本配置的虚拟机 XXXPC 和多个 1.44MB 容量的虚拟软盘，将其中一个虚拟软盘用 DOS 格式化为 DOS 引导盘，用 WinHex 工具将其中一个虚拟软盘的首扇区填满你的个人信息。

■ 接管裸机的控制权

设计 IBM_PC 的一个引导扇区程序，程序功能是：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。将这个程序的机器码放进放进第三张虚拟软盘的首扇区，并用此软盘引导你的 XXXPC，直到成功。

三、 实验方案

- 1 安装 VMware，在 VMware 中创建一个新的虚拟机，软盘的映像文件配置为用 asm 文件生成的 img 文件
- 2 安装 nasm 编译器，编译 asm 文件
- 3 学习汇编语言，编写引导程序代码

四、 实验过程

1 编写引导程序

老师提供的代码其实已经实现了以字符‘A’作运动轨迹，改变一下 ds 和 es 的指向就可以编译成功：

```
start:
    mov ax,0x7c00
    mov ds,ax                ; DS = CS,指向数据段
    mov ax,0B800h
    mov es,ax                ; ES = B800h, 指向显示器
```

所以在老师的代码的基础上稍作修改：char 是要显示的字符串“learnos”，number 是学号“18340064”，name 是要显示的名字“wongsiyoung”，字符后面是控制字符显示的形态和颜色

```
char db 'l',42h,'e',21h,'a',14h,'r',42h,'n',21h,'o',14h,'s',42h,'0',0 ;要显示的字符
number db '1',89h,'8',89h,'3',89h,'4',89h,'0',89h,'0',89h,'6',89h,'4',89h
name db 'w',89h,'o',89h,'n',89h,'g',89h,'s',89h,'i',89h,'y',89h,'o',89h,'u',89h,'n',89h,'g',89h
```

下图是稍微修改显示字符的控制代码，避免“learnos”字符串轨迹运动时吃掉了个人信息字符串的显示

```

show:
    mov ax,word[x];ax=8
    mov bx,80
    mul bx;8*80=行*列?
    add ax,word[y];640+列=641
    mov bx,2
    mul bx;ax=2*641=1282
    mov bx,ax

    mov dx,[es:bx]
    cmp dl,' '
    jnz loop1          ;避开显示的信息

    mov di,[cnt]
    mov al,byte[char+di]    ; 要显示的字符
    mov ah,byte[char+di+1]  ; 设置字符属性

    cmp al,'0'
    jz s1

    add di,2
    mov [cnt],di

con:
    mov [es:bx],ax        ; 送入显示器
    jmp loop1

s1:
    mov word[cnt],0
    mov al,byte[char]
    mov ah,byte[char+1]
    jmp con

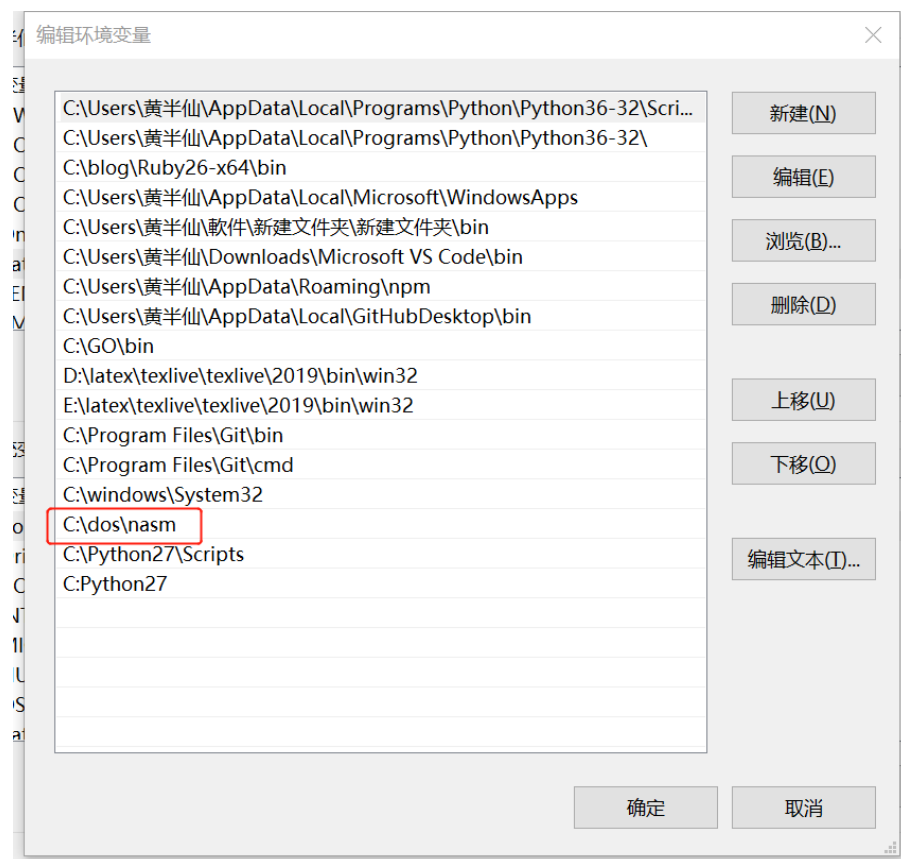
```

2 安装 nasm，编译 asm 程序

直接进入 nasm 官网下载安装包



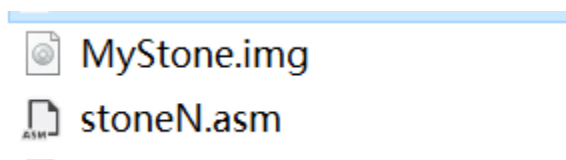
设置环境变量就可以全局使用了：



在 cmd 中使用以下命令编译刚刚编写的汇编代码：（也可以通过命令：nasm stonrN.asm -o MyStone.bin,生成二进制文件）

```
C:\dos\asm>nasm stoneN.asm -o MyStone.img
C:\dos\asm>
```

就可以看到在文件夹下就新增了 MyStone.img 文件：



使用资料书《x86 汇编语言-从实模式到保护模式》中提供的软件 HexViewer 查看 MyStone.img 文件

文件(F)

```

00000000 E9 41 00 50 C3 52 03 01 07 00 00 00 00 00 6C 42 |.A.P.R.....1B
00000010 65 21 61 14 72 42 6E 21 4F 14 53 42 30 00 31 89 e!a.rBn!O.SB0.1.
00000020 38 89 33 89 34 89 30 89 30 89 36 89 34 89 77 89 8.3.4.0.0.6.4.w.
00000030 6F 89 6E 89 67 89 73 89 69 89 79 89 6F 89 75 89 o.n.g.s.i.y.o.u.
00000040 6E 89 67 89 B8 C0 07 8E D8 B8 00 B8 8E C0 FC BF n.g.....
00000050 08 00 BE 1E 00 B9 08 00 F3 A5 BF 28 00 BE 2E 00 .....(....
00000060 B9 0B 00 F3 A5 FF 0E 03 00 75 FA C7 06 03 00 50 .....u....P
00000070 C3 FF 0E 05 00 75 EE C7 06 05 00 52 03 B0 01 3A .....u....R...:
00000080 06 07 00 74 1E B0 02 3A 06 07 00 74 53 B0 03 3A ...t...:tS...:
00000090 06 07 00 0F 84 85 00 B0 04 3A 06 07 00 0F 84 B5 .....
000000A0 00 EB FE FF 06 08 00 FF 06 0A 00 8B 1E 08 00 B8 .....
000000B0 19 00 29 D8 74 0E 8B 1E 0A 00 B8 50 00 29 D8 74 ..).t.....P.)t
000000C0 11 E9 CC 00 C7 06 08 00 17 00 C6 06 07 00 02 E9 .....
000000D0 BE 00 C7 06 0A 00 4E 00 C6 06 07 00 04 E9 B0 00 .....N.....
000000E0 FF 0E 08 00 FF 06 0A 00 8B 1E 0A 00 B8 50 00 29 .....P.)
000000F0 D8 74 0E 8B 1E 08 00 B8 FF FF 29 D8 74 11 E9 8F .t.....).t...
00000100 00 C7 06 0A 00 4E 00 C6 06 07 00 03 E9 81 00 C7 .....N.....
00000110 06 08 00 01 00 C6 06 07 00 01 EB 74 FF 0E 08 00 .....t.....
00000120 FF 0E 0A 00 8B 1E 08 00 B8 FF FF 29 D8 74 0D 8B .....).t..
00000130 1E 0A 00 B8 FF FF 29 D8 74 0F EB 54 C7 06 08 00 .....).t..T...
00000140 01 00 C6 06 07 00 04 EB 47 C7 06 0A 00 01 00 C6 .....G.....
00000150 06 07 00 02 EB 3A FF 06 08 00 FF 0E 0A 00 8B 1E .....:.....
00000160 0A 00 B8 FF FF 29 D8 74 0D 8B 1E 08 00 B8 19 00 .....).t.....
00000170 29 D8 74 0F EB 1A C7 06 0A 00 01 00 C6 06 07 00 ).t.....
00000180 01 EB 0D C7 06 08 00 17 00 C6 06 07 00 03 EB 00 .....
00000190 A1 08 00 BB 50 00 F7 E3 03 06 0A 00 BB 02 00 F7 ....P.....
000001A0 E3 89 C3 26 8B 17 80 FA 20 0F 85 B8 FE 8B 3E 0C ...&....>.
000001B0 00 8A 85 0E 00 8A A5 0F 00 3C 30 74 0D 83 C7 02 .....<0t....
000001C0 89 3E 0C 00 26 89 07 E9 9B FE C7 06 0C 00 00 00 .>..&.....
000001D0 A0 0E 00 8A 26 0F 00 EB EB EB FE 00 00 00 00 00 ...&.....
000001E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....U.
00000200

```

可以看到共有 512 字节，且以 0x55aa 结尾，说明是可引导程序

```

;code ENDS
;      END start
times 510-($-$$) db 0
db 0x55,0xaa

```

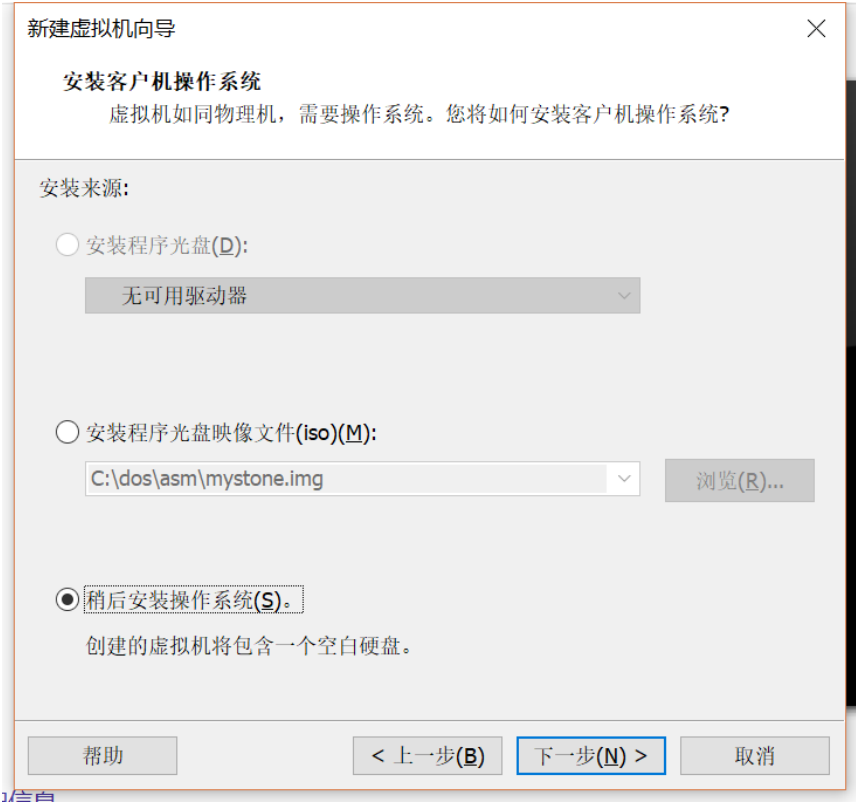
其实只要在代码最后中加这两行代码就行了，前提是前面的代码量不超过 510 字节。

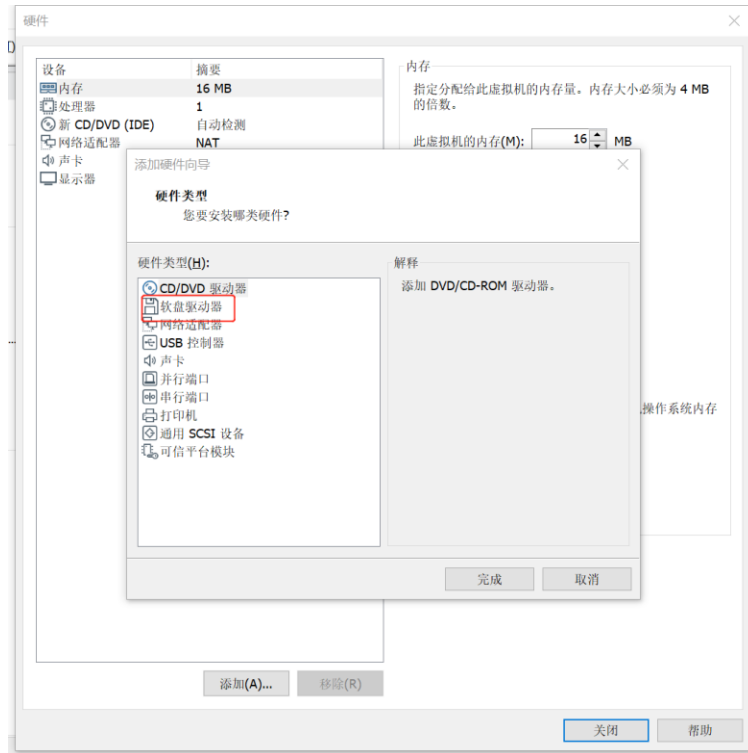
3 创建虚拟机

一样地从官网获取 VMware 安装包，然后在本机上安装就可以了

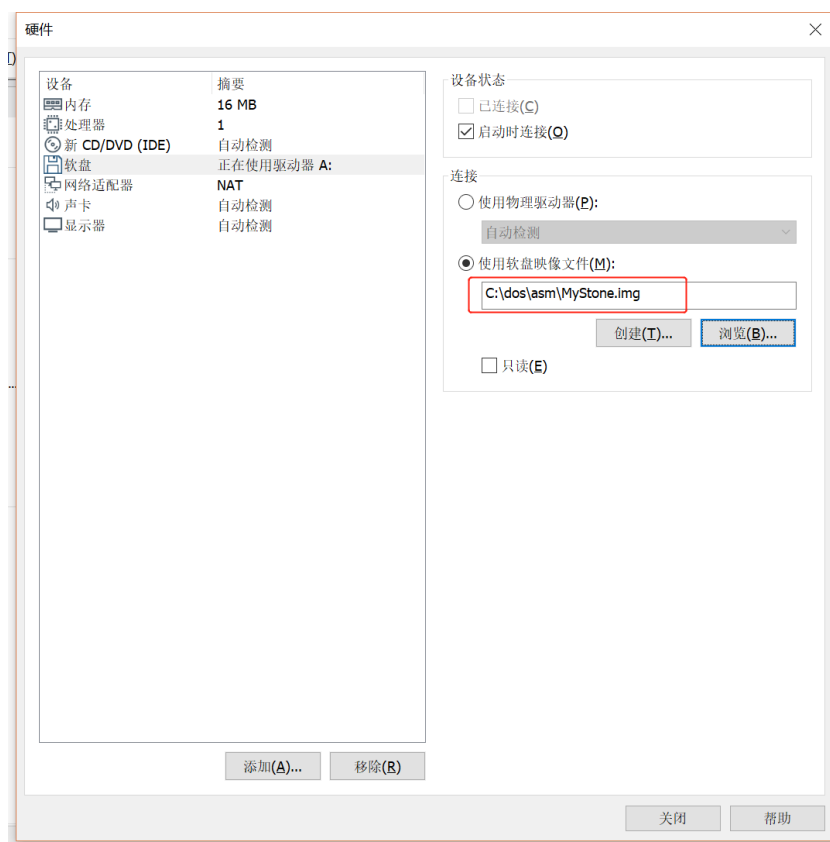
名称	修改日期	类型	大小
VMware VIX	2019/3/31 13:06	文件夹	
VMware Workstation	2019/3/31 13:06	文件夹	
VMware-workstation-full-15.0.4-12990004....	2019/3/31 12:49	应用程序	523,571 KB
VMware-workstation-full-15.0.4-12990004....	2020/2/19 11:02	360压缩 ZIP 文件	471,328 KB

打开 VMware，新建一个虚拟机，以下是简要步骤截图：（一般只用点击下一步就可以了，不用过多设置）



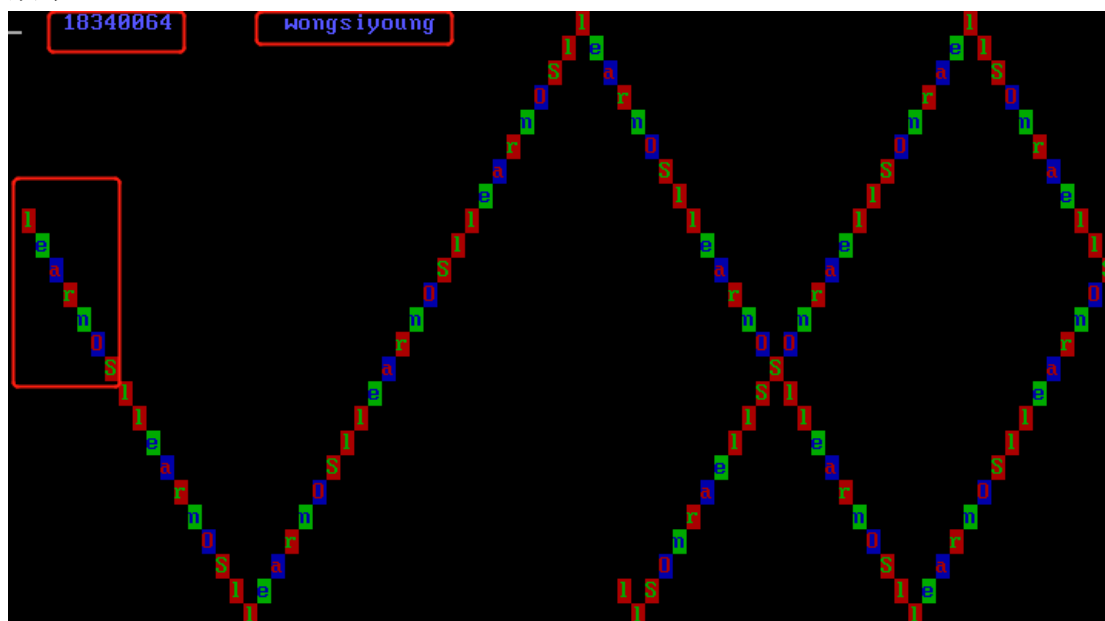


这里添加软盘，以便配置刚刚生成的 MyStone.img 文件：



创建虚拟机完毕，开机即可。

效果：



五、 实验总结

实验成功之后，把过程写进实验报告，回看实验报告的流程看似很简单，但其中有很多曲折。

一开始对于实验内容其实是很模糊的，主要也是因为缺乏汇编知识。在学习计算机组成原理的时候接触的是 mips 指令，对于 x86 是不了解的，所以一开始在看到老师的代码的时候，一来看不懂，二来直接对老师提供的代码编译会报错。

所以一开始是先看了王爽的《汇编语言（第3版）》才对汇编语言有所了解，接着也看了李忠的《x86 汇编语言-从实模式到保护模式》，再回看老师提供的代码就看懂了。编译不成功的，其实只用把代码中的：

```
start:
    ;xor ax,ax                ; AX = 0
    mov ax,cs                 ; ES = 0
    mov es,ax                 ; DS = CS
    mov ds,ax                 ; ES = CS
    mov es,ax                 ; ES = CS
    mov ax,0B800h             ; 
    mov gs,ax                 ; GS = B800h
    mov byte[char], 'A'
```

改成：


```

start:
    mov ax,0x7c00
    mov ds,ax                ; DS = CS,指向数据段
    mov ax,0B800h
    mov es,ax                ; ES = B800h, 指向显示器

```

就可以了！

不过因为王爽的汇编语言书是用 masm 的，而李忠的汇编语言书用的 nasm，一开始并不知道两者的区别，后来实践的时候就知道了……所以因为先看的王爽的汇编书，而用 masm 只能编译成.obj 和.exe 文件，无法直接在虚拟机中配置使用，在这里折腾了挺久的，后来使用 nasm，可以很方便的编译成 img 或 bin 文件就解决问题了。

在完成了添加个人信息和字符的一些个性变化后，尝试过写如同时控制两个运动的轨迹，写了之后感觉逻辑上是可以的，但是代码量太长了，超过了 512 字节，还在想着改进办法……

最后总的来说，当启动虚拟机后，看到画面的运动轨迹还是很有成就感的！另外也要加紧汇编语言的学习！

六、 参考资料

1. 王爽著.《汇编语言（第3版）》.清华大学出版社》.2003年9月
2. 李忠著.《x86 汇编语言-从实模式到保护模式》.电子工业出版社，2013年1月