中山大學
SUN YAT-SEN UNIVERSITY

《操作系统实验》

实验报告

（实验一）

学 院 名 称 ：	数据科学与计算机学院
专业（班级）：	16 计科 2 班
学 生 姓 名 ：	杨志成
学　　号 ：	16337281
时　　间 ：	2018　年　3　月　10　日

成 绩 ：

实验一 ： 接管裸机的控制权

一. 实验目的

(1) 掌握基础汇编语言，并灵活运用；
(2) 掌握虚拟机运行方式；
(3) 掌握如何取得一个逻辑的控制权；
(4) 学会运用基础汇编语言编写简单代码；
(5) 掌握如何在逻辑上显示信息。

二. 实验要求

(1)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。

(2)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。

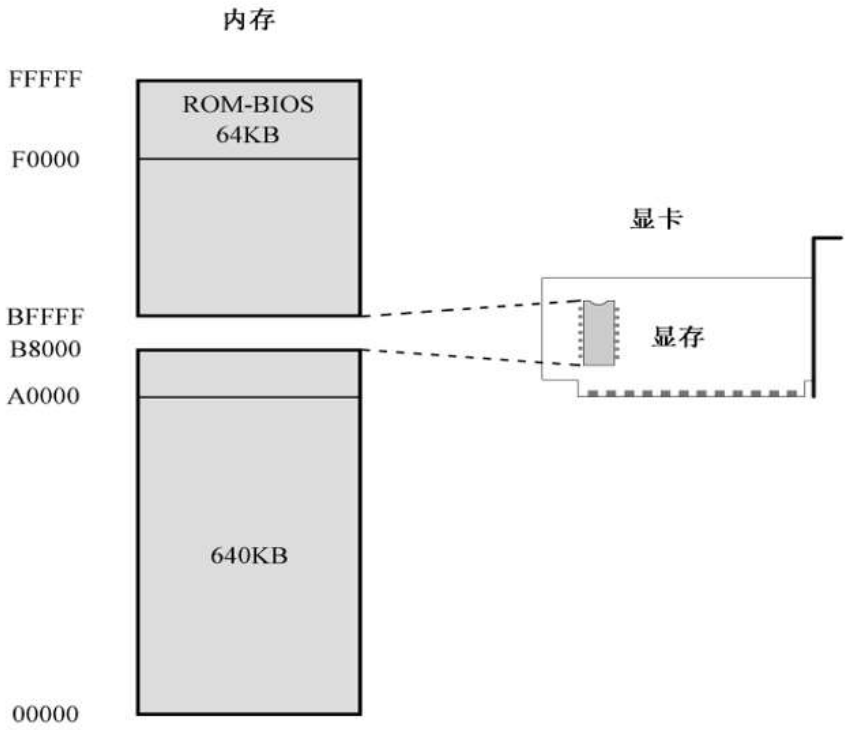
三、实验方案

本次试验中，我采取了循序渐进的思路，先实现了比较简单的版本，然后慢慢增加程序的功能，最后达到实验目的。比如，一开始我只实现了单个的字母 A 在屏幕上跳跃的版本，之后我又实现了不断变色的版本，到最后，我把学号和姓名作为信息显示到了屏幕上并不断弹射，效果如下：

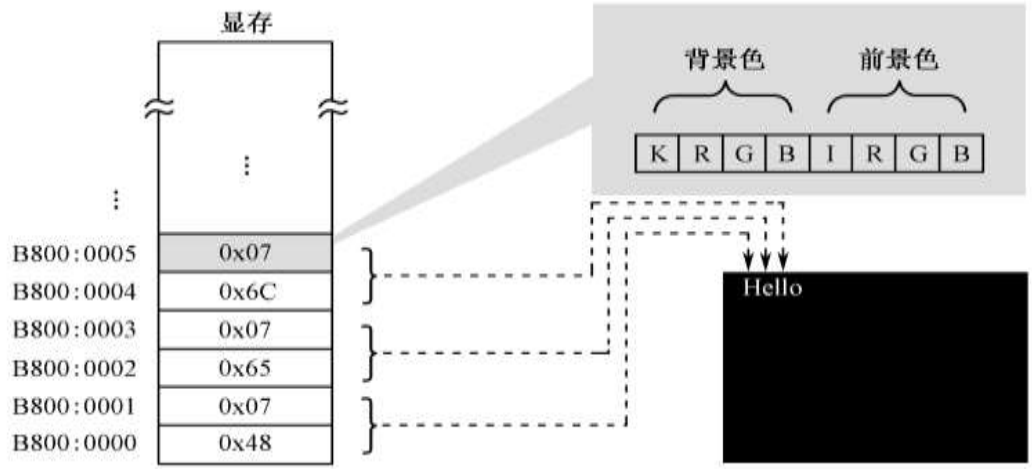
(1)基础原理

字符显示原理：

在 8086 中，可访问的显存空间大小为 1M，地址范围在 B8000~BFFFF



我们可以通过改变显存中存储的信息，改变屏幕的显示内容，比如字符的背景色和颜色，以及控制字符是否闪烁等等。



软盘工作原理：

软盘是个人计算机中最早使用的可移介质。软盘的读写是通过软盘驱动器完成的。软盘驱动器设计能接收可移动式软盘，目前常用的就是容量为 1.44MB 的 3.5 英寸软盘。软盘存取速度慢，容量也小，但可装可卸、携带方便。当计算机电源被打开时，它会进行加点自检（POST），然后寻找启动盘，如果是选择从软盘启动，计算机就会检查软盘的 0 面 0 磁道 1 扇区，如果发现它以 0xAA55 结束，则 BIOS 认为它是一个引导扇区，也就是我们说的 Boot Sector。当然，一个正确的 Boot Sector 除了以 0xAA55 结束之外，还应该包含一段少于 512 的执行码。一旦 BIOS 发现了 Boot Sector，就会将这 512B 的内容装载到内存的 0000:7C00 处，然后跳转到 0000:7C00 处，将控制器彻底交给这段引导代码。到此为止，计算机不再由 BIOS 中固有的程序来控制，而变成由操作系统的一部分来控制。

在本次实验中，我们正是运用这一原理来接管裸机的控制权的。

(2)实验工具和环境

实验支撑环境

硬件：个人计算机

主机操作系统：Windows/Linux/Mac OS/其它

虚拟机软件：VMware/VirtualPC/Bochs/其它

PC 虚拟机裸机/DOS 虚拟机/其它

实验开发工具

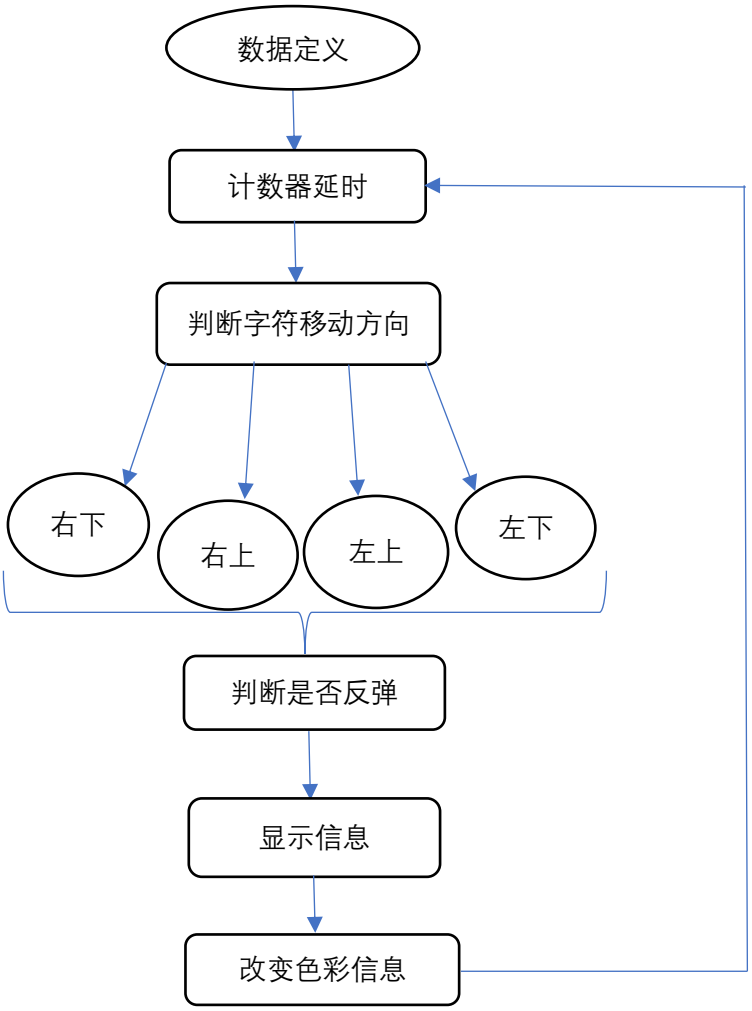
汇编语言工具：x86 汇编语言

高级语言工具：标准 c 语言

磁盘映像文件浏览编辑工具

调试工具：Bochs

(3)程序流程



(4)程序模块功能

该程序分为几大模块：

1. 文字移动模块（斜线四个方向）
2. 画框延时模块
3. 判断是否越界（反弹模块）
4. 字符信息显示模块

(5)代码文档组成

本次汇编语言实验虽然并不是我第一次接触汇编语言，但是依然遇到了许多困难，这里先从本次汇编代码的具体组成讲起：

1．org 指令：

ORG 是 Origin 的缩写：起始地址，源。在汇编语言源程序的开始通常都用一条 ORG 伪指令来实现规定程序的起始地址。

由于 8086 在加载软盘第一扇区的的程序时，会把内存加载到 0x7c00 的偏移地址中，因此我们在此处需要使用 org 命令

```
11      org 0x7c00
12      mov ax,cs
13      mov ds,ax                       ; DS = CS
14      mov ax,0B800h                   ; 文本窗口显存起始地址
15      mov es,ax                       ; es = B800h
```

2．延时程序

此处的代码相当于两个嵌套在一起的循环，就可以实现延时的功能。这时你可能会问：为什么不用一个循环来写呢？因为 8086 限制的立即数大小小于 50000*580，因此使用嵌套循环更为合理

```
5     delay equ 50000                   ; 计时器延迟计数,用于控制画框的速度
6     ddelay equ 580                    ; 计时器延迟计数,用于控制画框的速度

17    loop1:
18        dec word[count]               ; 递减计数变量
19        jnz loop1                     ; >0: 跳转;
20        mov word[count],delay
21        dec word[dcount]                  ; 递减计数变量
22        jnz loop1
```

3．移动字符方向，并判断是否越界

在老师发给我们的代码中，我发现了这样的一个 bug：当字符运动到了四个边角时，会有越界的情况发生，最后字符就会离开屏幕，不再显示。对于这样的 bug，并不难解决，我在每次判断越界条件时，多增加了一次判断因素，如果字符运动到了四个角落就让他原路返回，这样，就完美的规避了这个 bug。

```
27      mov al,1
28      cmp al,byte[rdul]
29      jz  DnRt
30      mov al,2
31      cmp al,byte[rdul]
32      jz  UpRt
33      mov al,3
34      cmp al,byte[rdul]
35      jz  UpLt
36      mov al,4
37      cmp al,byte[rdul]
38      jz  DnLt
39
40  DnRt:
41        inc word[x]
42        inc word[y]
43        mov bx,word[x]
44        mov ax,wide
45        sub ax,bx
46        jz  dr2ur
47        mov bx,word[y]
48        mov ax,len - str_len
49        sub ax,bx
50        jz  dr2dl
51        jmp show
52  dr2ur:
53      mov word[x],wide-2
54      cmp word[y],len - str_len
55      jz dr2ul
56      mov byte[rdul],Up_Rt
57      jmp show
58  dr2dl:
59      mov word[y],len-2-str_len
60      mov byte[rdul],Dn_Lt
61      jmp show
62  dr2ul:
63      mov word[y],len-2-str_len
64      mov byte[rdul],Up_Lt
65      jmp show
```

4．显示模块

在此模块中，我还加入了变色的方式，使得在字符串不断移动反弹的过程中，还可以变换颜色。

```
153  show:
154        mov ax,word[x]
155        mov bx,len
156        mul bx
157        add ax,word[y]
158        mov bx,2
159        mul bx
160        mov bx,ax
161        dec byte[color]
162        jz reset
163  here:
164        mov cx,str_len
165        mov si,inf
166  input_str:
167        mov ah,byte[color]
168        mov al,[si]
169        mov [es:bx],ax
170        inc si
171        inc bx
172        inc bx
173        loop input_str
174        jmp loop1
175
176  reset:
177        mov byte[color],07H
178        jmp here
```

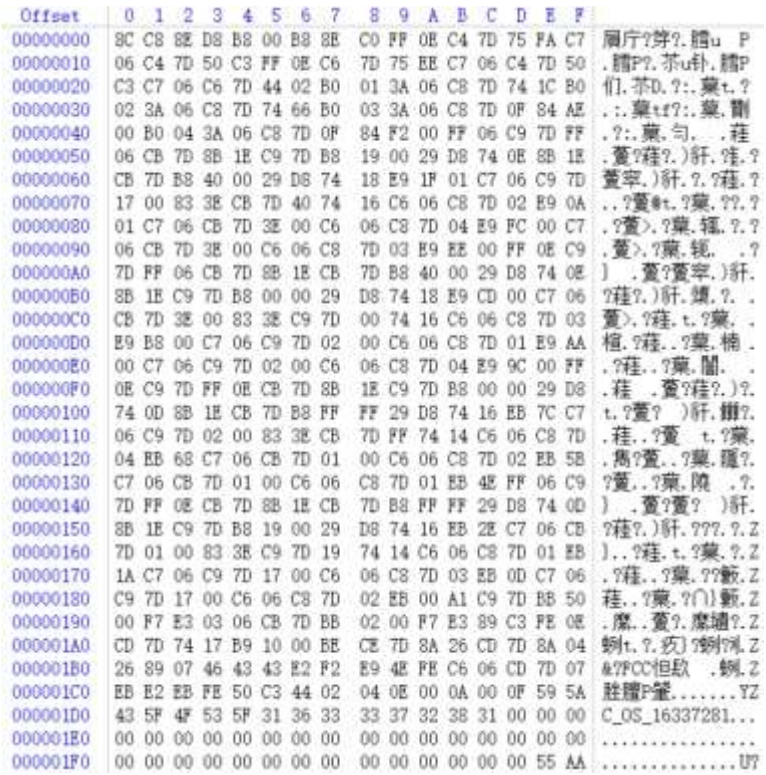
四．实验过程与实验结果：

（1）在 windows 中用命令行将 stone.asm 文件进行编译运行：

```
C:\Users\DELL\Desktop>nasm -f bin stone.asm -o stone.com
```

（2）在 winhex 中将该 stone.com 文件打开，把它复制进虚拟机第三个软盘中再把个人信息复制到第二个软盘中填满第一扇区，最后把 DOS 装进第一个软盘

此处贴上第三张软盘的图：



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	8C	C8	8E	D8	B8	00	B8	8E	C0	FF	0E	C4	7D	75	FA	C7	屑庁?笋?.膻u P
00000010	06	C4	7D	50	C3	FF	0E	C6	7D	75	EE	C7	06	C4	7D	50	.膻P?.苯u卦.膻P
00000020	C3	C7	06	C6	7D	44	02	B0	01	3A	06	C8	7D	74	1C	B0	们.苯D.?:.糞t.?
00000030	02	3A	06	C8	7D	74	66	B0	03	3A	06	C8	7D	0F	84	AE	.:.糞tf?:.糞.鬱
00000040	00	B0	04	3A	06	C8	7D	0F	84	F2	00	FF	06	C9	7D	FF	.?:.糞.匀. .莊
00000050	06	CB	7D	8B	1E	C9	7D	B8	19	00	29	D8	74	0E	8B	1E	.蔓?莊?.)肝.?生.?
00000060	CB	7D	B8	40	00	29	D8	74	18	E9	1F	01	C7	06	C9	7D	蔓宰.)肝.?.?莊.?
00000070	17	00	83	3E	CB	7D	40	74	16	C6	06	C8	7D	02	E9	0A	..?蔓@t.?糞.??.?
00000080	01	C7	06	CB	7D	3E	00	C6	06	C8	7D	04	E9	FC	00	C7	.?蔓>.?糞.辕.?.?
00000090	06	CB	7D	3E	00	C6	06	C8	7D	03	E9	EE	00	FF	0E	C9	.蔓>.?糞.钱. .?
000000A0	7D	FF	06	CB	7D	8B	1E	CB	7D	B8	40	00	29	D8	74	0E	] .蔓?蔓宰.)肝.
000000B0	8B	1E	C9	7D	B8	00	00	29	D8	74	18	E9	CD	00	C7	06	?莊?.)肝.鳙.?. .
000000C0	CB	7D	3E	00	83	3E	C9	7D	00	74	16	C6	06	C8	7D	03	蔓>.?莊.t.?糞. .
000000D0	E9	B8	00	C7	06	C9	7D	02	00	C6	06	C8	7D	01	E9	AA	楦.?莊..?糞.楠 .
000000E0	00	C7	06	C9	7D	02	00	C6	06	C8	7D	04	E9	9C	00	FF	.?莊..?糞.闇. .
000000F0	0E	C9	7D	FF	0E	CB	7D	8B	1E	C9	7D	B8	00	00	29	D8	.莊 .蔓?莊?.)?.
00000100	74	0D	8B	1E	CB	7D	B8	FF	FF	29	D8	74	16	EB	7C	C7	t.?蔓?)肝.鐳?.
00000110	06	C9	7D	02	00	83	3E	CB	7D	FF	74	14	C6	06	C8	7D	.莊..?蔓 t.?糞.
00000120	04	EB	68	C7	06	CB	7D	01	00	C6	06	C8	7D	02	EB	5B	.馮?蔓..?糞.躍?.
00000130	C7	06	CB	7D	01	00	C6	06	C8	7D	01	EB	4E	FF	06	C9	?蔓..?糞.隗 .?.
00000140	7D	FF	0E	CB	7D	8B	1E	CB	7D	B8	FF	FF	29	D8	74	0D	] .蔓?蔓?)肝.
00000150	8B	1E	C9	7D	B8	19	00	29	D8	74	16	EB	2E	C7	06	CB	?莊?.)肝.???.?.Z
00000160	7D	01	00	83	3E	C9	7D	19	74	14	C6	06	C8	7D	01	EB	]..?莊.t.?糞.?.Z
00000170	1A	C7	06	C9	7D	17	00	C6	06	C8	7D	03	EB	0D	C7	06	.?莊..?糞.??鲅.Z
00000180	C9	7D	17	00	C6	06	C8	7D	02	EB	00	A1	C9	7D	BB	50	莊..?糞.?门)鲅.Z
00000190	00	F7	E3	03	06	CB	7D	BB	02	00	F7	E3	89	C3	FE	0E	.麋..蔓?.麋墙?.Z
000001A0	CD	7D	74	17	B9	10	00	BE	CE	7D	8A	26	CD	7D	8A	04	蜊t.?.疚]?蜊?.Z
000001B0	26	89	07	46	43	43	E2	F2	E9	4E	FE	C6	06	CD	7D	07	&?FCC但欸 .蜊.Z
000001C0	EB	E2	EB	FE	50	C3	44	02	04	0E	00	0A	00	0F	59	5A	胜膻P鲞.....YZ
000001D0	43	5F	4F	53	5F	31	36	33	33	37	32	38	31	00	00	00	C_OS_16337281...
000001E0	00	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	55	AA	.....U?

（3）打开虚拟机运行，显示效果

这是正在将字符串进行弹射的效果：

这是字符不断弹射，最后串铺满屏幕的结果：

字符串“YZC_OS_16337281”在屏幕上不断弹射并且会不断变色。证明实验结果是正确的

五、实验总结：

此次试验虽然不是我第一次使用汇编语言，但还是遇到了很多问题，比如 nasm 格式的汇编语言不是十分熟悉，因为上学期我们所接触的汇编语言使用的是 masm 格式，所以一开始遇到了一点困难。但通过不懈的努力，还是克服了过去。下面列举一些我在此次实验中**遇到的问题与思路**：

（1）如何使用虚拟机？

由于并没有熟练地使用过虚拟机，我一开始也是遇到了一些操作上的问题，比如如何创建一个裸机等等，但这些问题都在我的探索下迎刃而解了。

（2）实验课代码文件有 bug

实验课的代码文件格式是 masm，但我把用 nasm 格式会变后发现代码报错，我只好实用 nasm 格式去修改代码。在修改的过程中，我发现 stone.asm 在判断边界条件时疏忽了一个问题，那就是当字符跳动到四个角落是，字符会飞到窗口外面，越弹越远，不会回到窗口上。

虽然这并不是有什么严重影响的 bug，但站在严谨的角度上，我觉得这个问题不可忽略，于是着手修改了这个代码，修改此代码并不麻烦，只需要在判断条件时，多一次判断即可，具体修改如下：

```
dr2ur:
    mov word[x],wide-2
    cmp word[y],len - str_len
    jz dr2ul
    mov byte[rdul],Up_Rt
    jmp show
dr2dl:
    mov word[y],len-2-str_len
    mov byte[rdul],Dn_Lt
    jmp show
dr2ul:
    mov word[y],len-2-str_len
    mov byte[rdul],Up_Lt
    jmp show
```

该代码经修改后，当字符碰到边界时，会原路返回，而非一开始的走到屏幕之外，就这样，我成功的规避了一个 bug。

（3）有关代码风格

接触半年汇编语言以来，虽然已经了解了汇编语言各个指令的格式及功能，但是，这半年来，我所写的汇编程序一般不会超过 50 行，因此并没有考虑到代码风格的问题。但在此次试验中，代码有大概 200 行，代码风格不再是一个可以忽视的问题，优良的代码风格不仅看上去使得代码更加美观，还可以让代码更容易维护和修改。这个问题我在接下来的操作系统实验中也会更加注意。

（4）本次实验的循序渐进的方法

本次实验我并没有一下子就实现了我想要的结果，而是慢慢的实现一些基本功能，最后打到我的预期目的。第一步我学会了在裸机显示信息，第二步，我实现了单个字符的跳动，第三步我实现了整个字符串的跳动，最后我把变色的功能也添加了上去。完成了自己的第一次实验。

（5）代码超过 512 字节怎么办？

在实现最总版本的过程中，我遇到了这样的一个问题，过于麻烦的代码导致编译后的.com 文件的 16 进制格式超过了 512，而我们只应该使用软盘上的第一个扇区，此后，我简化了我的汇编代码，成功将.com 文件变小，达到了我的目的。

实验感想：

1、网络真的很强大，用在学习上将是一个非常高效的助手。几乎所有的资料都能够在网上找到。从虚拟机的安装，到的各种基本命令操作这些都能在网上找到。也因为这样，整个课程设计下来，我浏览的相关网页已经超过了 100 个(不完全统计)。当然网上的东西很乱很杂，自己要能够学会筛选。

2、不能决定对或错的，有个很简单的方法就是去尝试。就所以要非常的谨慎，尽量少出差错，节省时间。多找个几个参照资料，相互比较，慢慢研究，最后才能事半功倍。

3、同学间的讨论，这是很重要的。老师毕竟比较忙。对于实验最大的讨论伴侣应该是同学了。能和学长学姐讨论当然再好不过了，没有这个机会的话，和自己班上同学讨论也是能够受益匪浅的。大家都在研究同样的问题，讨论起来，更能够把思路理清楚，相互帮助，可以大大提高效率。

4、敢于攻坚，越是难的问题，越是要有挑战的心理。这样就能够达到废寝忘食的境界。当然这也是不提倡熬夜的，毕竟有了精力才能够打持久战。但是做课设一定要有状态，能够在吃饭，睡觉，上厕所都想着要解决的问题，这样你不成功都难。

操作系统作为计算机专业的一门核心课程，我应该认真对待，不仅仅要在理论课上弄明白很多东西，还需要在实验课上践行自己所学到的理论，就如同老师第一节实验课的 PPT 中所写，“**纸上得来终觉浅，绝知此事要躬行**”!