实验二 ：加载用户程序的监控程序

目录

一.实验目的

1．初步了解 BIOS 的作用，原理

2．掌握 BIOS 的中断调用方法

3．开发原始的操作系统

开发监控程序，以实现多道批处理。通过键盘输入控制用户程序执行。

二.实验要求

1. 设计四个（或更多）有输出的用户可执行程序

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。

2. 设计引导程序，用于指定运行这四个有输出的用户可执行程序之一

显示必要的提示信息后，从引导盘的特定扇区加载一个他人开发的 COM 格式的可执行程序到指定的内存位置，然后启动这个程序，实现操作系统执行用户程序这一项基本功能

3. 自行组织映像盘的空间存放四个用户可执行程序

三.实验方案：

1.方案思想

本实验我设计一个主引导程序，它作为监控程序运行。引导程序通过键盘输入选择执行的用户程序，在用户程序中通过特定输入返回引导程序。

2.相关原理：

【批处理系统 and 监控程序】

批处理系统又名批处理操作系统。批处理是指用户将一批作业提交给操作系统后就不再干预，由操作系统控制它们自动运行。这种采用批量处理作业技术的操作系统称为批处理操作系统。批处理操作系统分为单道批处理系统和多道批处理系统。批处理操作系统不具有交

互性，它是为了提高CPU的利用率而提出的一种操作系统。

监控程序是操作系统的最早期的形式

1.获取计算机硬件系统的控制权

2.提供计算机输入设备和输出的控制程序

3.控制用户程序的执行

【BIOS 中断服务调用】

BIOS中中断例程即BIOS中断服务程序

- 是微机系统软、硬件之间的一个可编程接口，用于程序软件功能与微机硬件实现的衔接。DOS/Windows操作系统对软、硬盘、光驱与键盘、显示器等外围设备的管理即建立在系统BIOS的基础上。程序员也可以通过 对INT 5、INT 13等终端的访问直接调用BIOS终端例程。

调用BIOS中断服务程序的方法

- 每个中断服务有特定的参数，一般使用指定的寄存器传递参数；
- 利用软中断指令调用
- BIOS中断调用的一般格式为：

```
mov ah,功能号
..... ; 设置各种入口参数
int 中断号
```

常用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

Int 10h：

BIOS 的10H提供了显示字符串的调用

BIOS的10H号调用功能与参数

显示字符串	10H	13H	AL：放置光标的方式 BL：字符属性字节 BH：显示页(0~3) DH：行位置(0~24) DL：列位置(0~79) CX：字符串的字节数 ES:BP：字符串的起始地址	AL=0 2 光标留在串头 AL=1 3 光标放到串尾 AL=0 1 串中只有字符 AL=2 3 串中字符和属性字节交替存储 BL：位 7 为 1 闪烁 位 6~4 为背景色 RGB 位 3 为 1 前景色高亮 位 2~0 为前景色 RGB

Int 13h:

BIOS的13h号中断可以实现读扇区到指定内存地址的功能。因为引导程序和用户程序的地址是可知的，配合上jmp指令就可以使得程序在引导程序中跳转到用户程序中，或者在用户程序中返回到引导程序中。这就可以实现引导程序对用户程序的控制。在分时系统中，将跳转返回引导程序的地址换成引导程序中间的地址并保存好数据就可以实现分时运行程序。

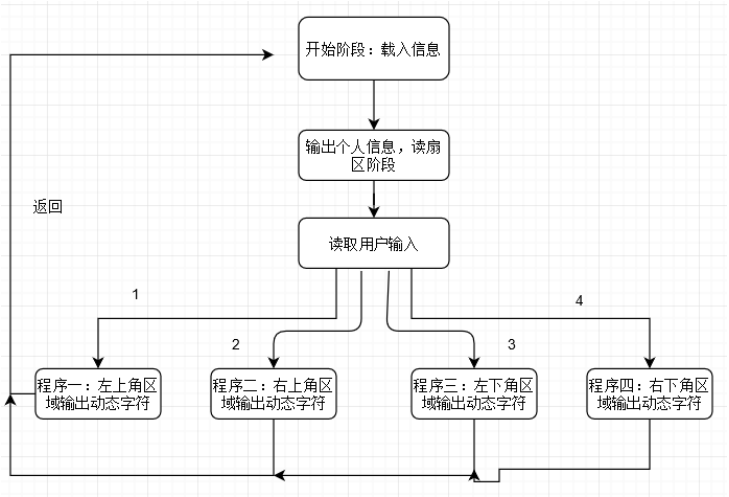
BIOS 的13H提供了磁盘读写的调用

BIOS的13H号调用功能与参数

读扇区	13H	02H	AL：扇区数(1~255) DL：驱动器号(0 和 1 表示软盘，80H 和 81H 等表示硬盘或 U 盘) DH：磁头号(0~15) CH：柱面号的低 8 位 CL：0~5 位为起始扇区号(1~63)，6~7 位为硬盘柱面号的高 2 位(总共 10 位柱面号，取值 0~1023) ES：BX：读入数据在内存中的存储地址	返回值： ■ 操作完成后 ES：BX 指向数据区域的起始地址 ■ 出错时置进位标志 CF=1，错误代码存放在寄存器 AH 中 ■ 成功时 CF=0、AL=0

3.程序流程：

a 引导程序



如图所示：程序的流程为：

1，显示个人信息。

2，读扇区，并写入内存。

3，输入字符选择程序

4，进入对应的子程序

5，通过键盘输入返回引导程序

4.程序关键模块：

显示个人信息模块：(调用10h中断)

```
Start:
    mov ax, cs          ; 置其他段寄存器值与CS相同
    mov ds, ax          ; 数据段
    mov es, ax        ; 置ES=DS
    mov bp, Message       ; BP=当前串的偏移地址
    mov cx, MessageLength  ; 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, 0000h         ; 行号=0, 列号=0
    int 10h           ; BIOS的10h功能：显示一行字符
```

加载四个用户子程序：(调用13h中断)

```
;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,4                   ;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
```

输入[1 / 2 / 3 / 4]选择子程序：(16h中断)

```
31      mov ah,0
32      int 16H
33      cmp al,49
34      jz OffSetOfUserPrg1
35      cmp al,50
36      jz OffSetOfUserPrg2
37      cmp al,51
38      jz OffSetOfUserPrg3
39      cmp al,52
40      jz OffSetOfUserPrg4
```

用户程序按‘q’键返回引导程序：

```
    mov ah,01h
    int 16h
    jnz quit
    jmp loop1

quit:
    mov ah,00h
    int 16h
    cmp al,'q'
    jz 7c00h
    jmp loop1
```

四.实验过程与结果

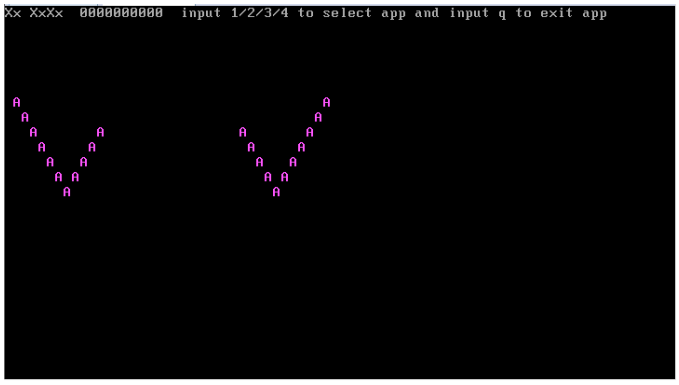
1.Notepad++编写汇编程序

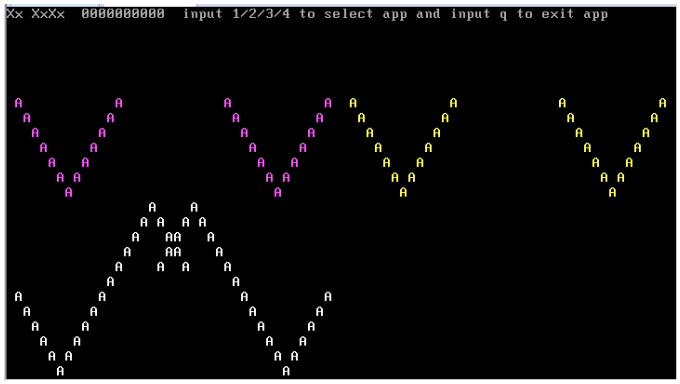
2.用nasm编译汇编程序生成二进制文件

3.用winhex将二进制文件写入映像文件，用户程序文件分别存于到2到5号扇区

4.将映像文件作为虚拟机软盘

【运行结果】





遇到的问题：使用16h中断01号功能时，输入会留在缓存中，导致返回后的输入受影响

解决情况：在调用16h中断01号功能后调用16h中断00号功能读出缓存中的内容。

五.实验总结：

本次实验是从实验一修改而来，本实验主要分为两个部分，主引导程序和用户程序。主引导程序同时作为监控程序，可算作粗糙的操作系统内核，主要负责加载控制用户功能。用户程序则实现功能部分，这部分并不重要，主要是为了配合反映内核效果。

这一次的实验我学会了很多知识，例如BIOS中断服务的调用，引导程序实现控制功能的方法。

之前一直不懂监控程序的用途，为什么叫监控程序，如何实现监控，现在觉得有了个思路，就是通过监控程序来调用其他用户程序。每次执行完程序又返回到监控程序，当然这中间涉及到了中断服务程序和跳转指令。

参考文献：

1．《80x86 汇编语言程序设计教程》 （杨季文）
2．中断表 （csdn）