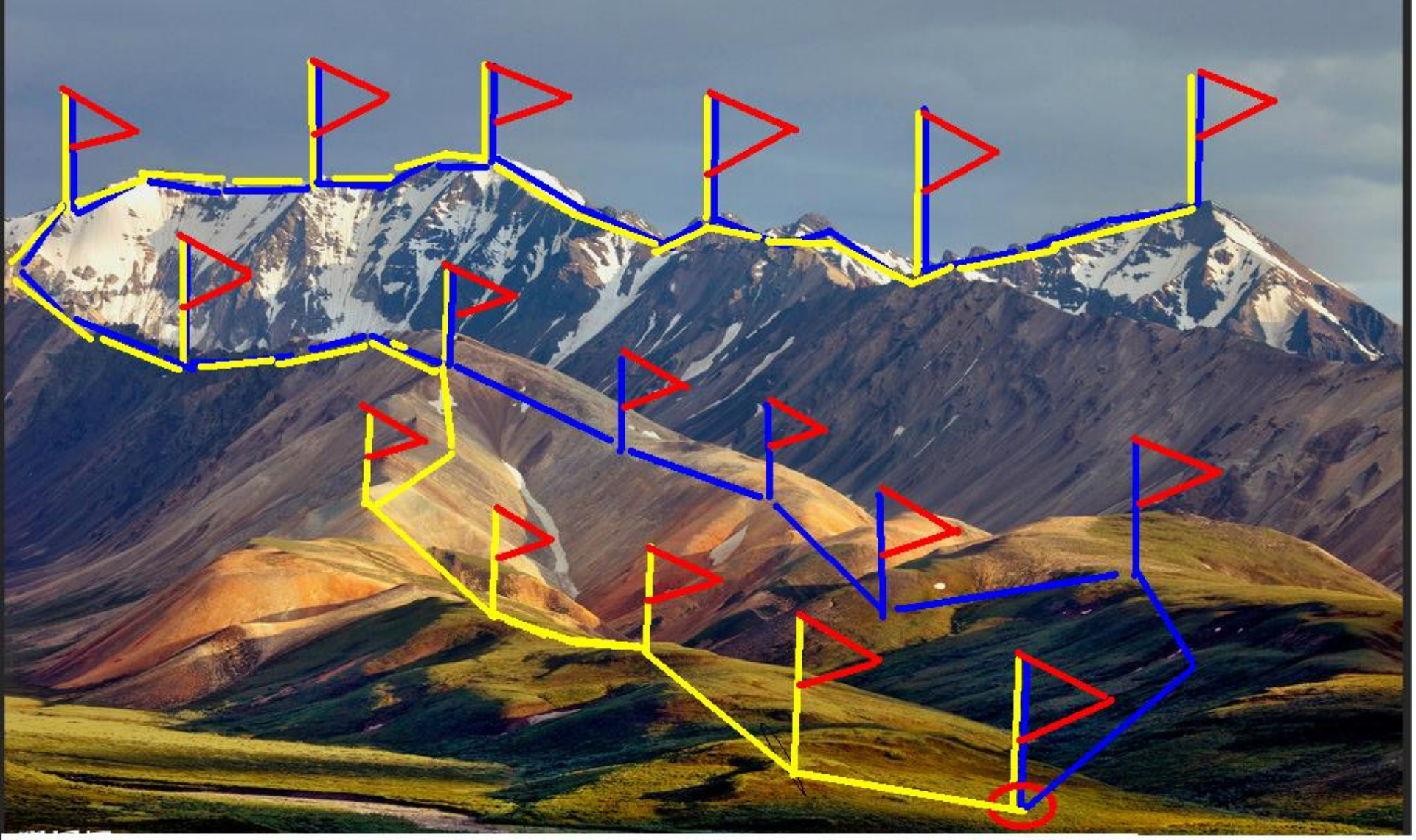


A线路

1裸机控制权与引导程序

2加载执行**COM**格式的批处理系统

3汇编与**C**语言实现独立内核

4中断处理与异步事件响应

5系统调用

6二状态进程模型实现

B线路

1裸机控制权与引导程序

2加载执行**COM**格式的批处理系统

3汇编与**C**语言实现独立内核

4在**FAT12**格式软盘中引导内核和加载用户程序

5中断处理与系统调用实现

6二状态进程模型实现

1加载用户程序的监控程序

- BIOS
- 开发最原始的操作系统
- 实验项目2说明

1. BIOS调 用

- BIOS调用
  - BIOS是英文"Basic Input Output System"的缩略语，直译过来后中文名称就是"基本输入输出系统"。其实，它是一组固化到计算机内主板上一个ROM芯片上的程序，它保存着计算机最重要的基本输入输出的程序、系统设置信息、开机后自检程序和系统自启动程序。其主要功能是为计算机提供最底层的、最直接的硬件设置和控制。
- BIOS芯片中主要存放：
  - **自诊断程序**：通过读取CMOSRAM中的内容识别硬件配置，并对其进行自检和初始化；
  - **CMOS设置程序**：引导过程中，用特殊热键启动，进行设置后，存入CMOS RAM中；
  - **系统自举装载程序**：在自检成功后将磁盘相对0道0扇区上的引导程序装入内存，让其运行以装入DOS系统；
  - **主要I/O设备的驱动程序和中断服务**：由于BIOS直接和系统硬件资源打交道，因此总是针对某一类型的硬件系统，而各种硬件系统又各有不同，所以存在各种不同种类的BIOS，随着硬件技术的发展，同一种BIOS也先后出现了不同的版本，新版本的BIOS比起老版本来说，功能更强。

1 BIOS (续)

- 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
```

1 BIOS（续）

- 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

BIOS的10H调用

- 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

BIOS的10H调用

- 显示字符串

mov ah,13H ; 功能号
mov al,1 ; 光标方式（光标放到串尾）
mov bl,属性 ; 背景与前景色
mov bh,0 ; 显示页号（第0页）
mov dh,行号
mov dl,列号
mov es,字符串段地址
mov bp,字符串偏移地址
mov cx,串长
int 10H ; 调用中断

BIOS的10H调用

- **显示字符**

以电传方式显示单个字符（从当前光标位置开始，且更新当前光标位置）：

```
mov ah,0e ; 功能号
mov al,字符 ; ASCII码
mov bl,0 ; BL必须设为0，显示页号（第0页）
int 10H ; 调用中断
```

- **读按键**

返回值：AL=ASCII码，AH=扩展码/扫描码

```
mov ah,0 ; 功能号
int 16H ; 调用中断
```

BIOS的13H调用

- 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

BIOS编程-例

- 自己编写的COM程序：显示1个字符串、当键盘用户按键则结束。
- 汇编代码（showstr.asm）：

```
org 100h ; 告诉编译器程序加载到100H处
mov ax,cs ; 通过AX中转, 将CS的值传送给DS和ES
mov ds,ax
mov es,ax
mov ss,ax

; 显示字符串1 "MY-OS 1.0"（开始）
mov ah,13h ; 功能号
mov al,1 ; 光标放到串尾
mov bl,0ah ; 亮绿
mov bh,0 ; 第0页
mov dh,0ah ; 第10行
mov dl,23h ; 第35列
mov bp,str1 ; BP=串地址
mov cx,9 ; 串长为9个字符
int 10h ; 调用10H号中断
; 显示字符串1（结束）
mov ah,0 ; 功能号
int 16h ; 调用16H号中断
; 按任一键：退出返回DOS
mov ax,4c00h
int 21h
str1 db “MY-OS 1.0!”
```

1.2开发最原始的操作系统

- 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。

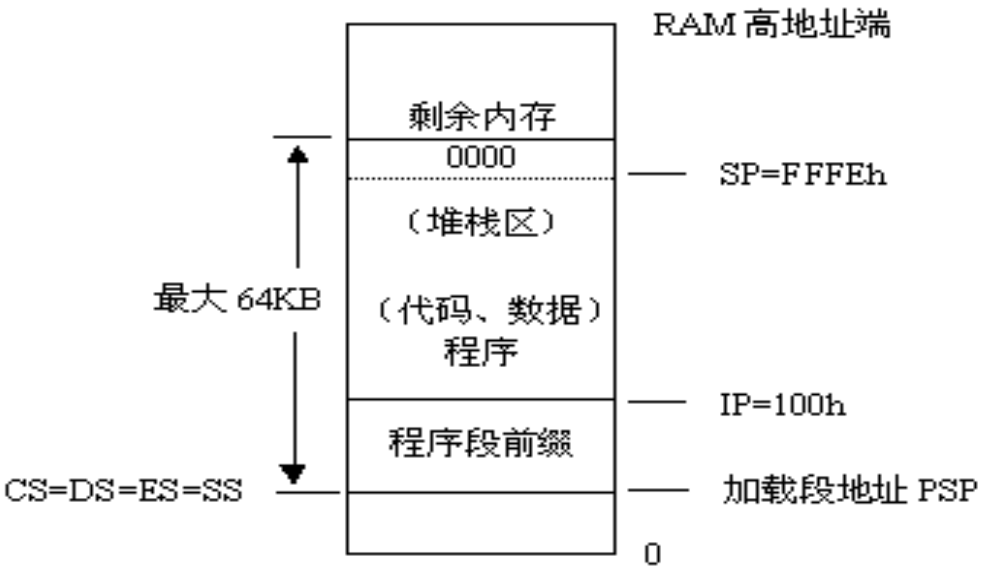
- 监控程序是操作系统的最早期的形式
  - 获取计算机硬件系统的控制权
  - 提供计算机输入设备和输出的控制程序
  - 控制用户程序的执行
- 如果我们要为IBM_PC开发监控程序级的操作系统，那么应该怎样完成任务？
  - 因为该机器的BIOS能够控制输入设备和输出设备，所以我们的工作主要实现控制用户程序的执行这一项。

1.3 用户程序的格式

- 监控程序如何获取计算机硬件系统的控制权
  - 如果程序不超过512字节，则可以引导扇区程序的方法实现
  - 如果程序超过512字节，则要分解为引导扇区部分和其它功能部分
- 监控程序如何控制用户程序的执行
  - 用户程序的格式
    - COM
    - BIN
  - 加载用户程序
    - 磁盘存储方法
    - 磁盘读取控制编程

1.3.1 COM格式

- COM（**com**mand file，命令文件）是CP/M和DOS的一种原始二进制可执行格式，以.com为扩展名。COM文件非常简单，没有文件头、没有元数据，只有代码和数据。
- COM文件会被装载到当前段的0x100（256）处，不能重新定位。由于不能分段，所以COM文件的大小必须≤64KB-256B，且不能有独立的数据段和堆栈段，程序的所有代码和数据都必须位于一个段中。
- 另外，在Windows操作系统的64位版本中，不再支持COM程序的运行。
- DOS加载COM程序的内存映象图



1. 3.2 PSP

- DOS装载可执行文件（.COM/.EXE）自动构造，返回DOS后自动释放该空间

偏移量	内容
00~01H(2B)	INT 20H指令
02~03H(2B)	存储器中段的大小
04~09H(6B)	保留
0A~0DH(4B)	中止 中断向量INT 22H
0E~11H(4B)	断点 中断向量INT 23H
12~15H(4B)	错误 中断向量INT 24H
16~2BH(22B)	DOS保留
2C~2DH(2B)	环境变量区段地址
2E~4FH(34B)	DOS运行区
50~51H(2B)	INT 21H指令

1. 3.3返回DOS

- 利用4ch功能调用返回DOS（.COM/.EXE）

  DOS功能调用的4ch子功能（返回DOS）：

  入口参数：AH＝4ch，AL＝返回数码

  产生终止程序执行返回操作系统的指令代码,它的可选参数是一个返回的数码，通常用0表示没有错误。例如程序结束点放置对应的代码是：

  ```
  mov ax,4c00h
  int 21h
  ```

1.4程序执行前存放安排

- 运行前，用户程序a.com存放在哪里，怎样存放？
- 一种简单的磁盘存储组织：
  - 1.44MB的软盘共2880个扇区，布局如图2-2.0扇区：

图2-2 1.44MB的软盘布局图

内存安排

- 目前我们使用IBM_PC基本内存640k，实模式的段+段内偏移的方式访问内在单元。
- BIOS装入引导扇区程序时，优先使用最低端的一个64k段作为引导扇区程序的存放区域，偏移量7c00h。
- 为了简单起见，我们也把用户程序a.com存放在同一个段中，就放在偏移量A100h开始的一个区域吧。

1.6 监控程序基本型

- 程序
  - 本程序按引导扇区程序要求设计
  - 功能：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。
  - 程序利用BIOS调用显示字符串和读磁盘扇区加载用户程序
  - 涉及BIOS的10H和13H号调用

```
;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 A100h
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, 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, 0                       ; 行号=0
        mov     dl, 0                                   ; 列号=0
        int     10h                                     ; BIOS的10h功能：显示一行字符
LoadnEx:
   ;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
AfterRun:
    jmp $               ;无限循环
Message:
    db 'Hello, MyOs is loading user program A.COM…'
MessageLength  equ ($-Message)
    times 510-($-$$) db 0
    db 0x55,0xaa
```

实验项目2：加载用户程序的监控程序

- 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。

- 修改参考原型代码，允许键盘输入，用于指定运行这四个有输出的用户可执行程序之一，要确保系统执行代码不超过512字节，以便放在引导扇区
- 自行组织映像盘的空间存放四个用户可执行程序