



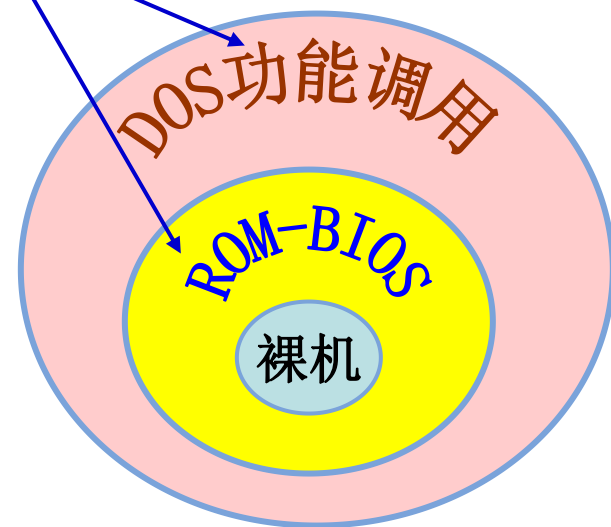
简单的系统功能调用



系统功能调用

- ✧ 程序员用汇编语言编程时需要利用 ROM-BIOS 和 操作系统 提供的 API 来实现如键盘输入和屏幕输出这类操作
- ✧ 系统功能调用是程序设计的一个重要方面

汇编语言程序



基本DOS输入输出功能调用

- ◇ DOS利用21H号中断提供给用户近百个系统功能，主要包括设备管理、目录管理和文件管理三个方面的功能。
- ◇ 本课程接下来学习以下常用的基本功能调用
 1. 单字符输入（1号调用）
 2. 单字符显示（2号调用）
 3. 显示字符串（9号调用）
 4. 字符串输入（10号调用）
 5. 结束调用（4CH号调用）

100



(2) 单字符输出 (2号DOS功能调用)


- (1) 功能调用号: **AH = 02H**
- (2) 入口参数: **DL = 欲显示字符的ASCII码**
- (3) 功能调用: **INT 21H**
- (4) 出口参数: 无




； 以下代码显示一个问号 “？”

mov dl,'?'	；设置入口参数
mov ah,02h	；设置功能调用号
int 21h	；功能调用

例：回车换行的子程序



```
crlf    proc
        push ax          ; 保护寄存器
        push dx
        mov dl,0dh       ; 回车功能的ASCII码是0DH
        mov ah,2
        int 21h
        mov dl,0ah       ; 换行功能的ASCII码是0AH
        mov ah,2
        int 21h
        pop dx           ; 恢复寄存器
        pop ax
        ret
crlf    endp
```



(3) 字符串输出 (9号DOS功能调用)

(1) 功能调用号: **AH = 09H**

(2) 入口参数:

DS:DX = 欲显示字符串在内存中的首地址 (逻辑地址形式: **DS** = 段地址, **DX** = 偏移地址)

内存中的字符串以**ASCII**码形式保存, 最后必须添加一个 “\$” 结尾 (并不显示)

(3) 功能调用: **INT 21H**

(4) 出口参数: 无

例：提示按任意键继续

； 在数据段定义要显示的字符串

```
msg      db 'Press any key to contiune...','$'
```

； 在代码段编写程序

```
mov ah,9
```

```
mov dx,offset msg
```

； 设置入口参数：

； **DS** = 数据段地址（假设已经设置好）

； **DX** = 字符串的偏移地址

```
int 21h
```


(4) 字符串输入 (10号DOS功能调用)

格式: LEA DX,缓冲区首偏移地址

MOV AH,10

INT 21H

功能: 从键盘上输入一字符串到用户定义的输入缓冲区中, 并送显示器显示。

缓冲区格式:

缓冲区最大容量 (1B) : 实际读入的字符数,
不包括回车 (1B) : 接收的字符串

```
max    db 10
```

```
length db ?
```

```
string db 10 dup(0)
```

(5) 程序结束 (4CH号DOS功能调用)

格式: MOV AH, 4CH
INT 21H

功能: 终止当前程序并返回调用程序。

4. ROM-BIOS输入输出功能调用

ROM-BIOS也以中断服务程序的形式，向程序员提供系统的基本输入输出程序。

ROM-BIOS功能更加基本，且与操作系统无关。

当DOS没有启动或不允许使用DOS功能调用时，可以使用ROM-BIOS功能调用。

例如输出一个字符：

(1) 功能调用号：**AH = 0EH**

(2) 入口参数：**AL = 欲显示字符的ASCII码**

通常**BX = 0**

(3) 功能调用：**INT 10H**

(4) 出口参数：无

系统功能调用步骤



通常按照如下4个步骤进行：

- (1) 在**AH**寄存器中设置系统功能调用号
- (2) 在指定寄存器中设置入口参数
- (3) 用中断调用指令（**INT i8**）执行功能调用
- (4) 根据出口参数分析功能调用执行情况

Syscall.asm





参考资料



简化段定义的源程序格式



```
.model small           ; 小型模式存储模式
.stack                ; 1KB空间堆栈段
.data ; 数据段
..... ; 数据定义
.code ; 代码段
start:  mov ax,@data   ; 起始点
        mov ds,ax      ; 设置DS
..... ; 程序代码
        mov ax,4c00h
        int 21h         ; 结束点, 返回DOS
..... ; 子程序代码
end start              ; 汇编结束
```



存储模式

存储模式	特点
TINY 微型模式	COM类型程序，只有一个小于64KB的逻辑段（MASM 6.x支持）
SMALL 小型模式	小应用程序，只有一个代码段和一个数据段（含堆栈段），每段不大于64KB
COMPACT 紧凑模式	代码少、数据多的程序，只有一个代码段，但有多数据段
MEDIUM 中型模式	代码多、数据少的程序，可有多代码段，只有一个数据段
LARGE 大型模式	大应用程序，可有多代码段和多个数据段（静态数据小于64KB）
HUGE 巨型模式	更大应用程序，可有多代码段和多个数据段（对静态数据没有限制）
FLAT 平展模式	32位应用程序，运行在32位80x86CPU和Windows 9x或NT环境



BIOS

- ◇ BIOS是英文Basic Input/Output System的缩写，即基本输入输出系统。
- ◇ BIOS是加载在电脑硬件系统上的最基本的软件代码，一般被存放在主板上的ROM之中，它保存着计算机最重要的基本输入输出程序、系统设置信息、开机自检程序和系统自启动程序，为计算机提供最底层、最直接的硬件设置和控制。



DOS

- ◇ DOS即Disk Operating System，意为“磁盘操作系统”。MS-DOS是1979年由微软公司为IBM个人电脑开发的，它是一个单用户单任务的操作系统。
- ◇ 从1981年MS-DOS1.0直到1995年MS-DOS7.1的15年间，DOS在IBM PC兼容机市场中占有举足轻重的地位。
- ◇ 直到1995年微软在推出Windows 95之后，宣布MS-DOS不再单独发布新版本。不过FreeDOS等与MS-DOS兼容的DOS则在继续发展着。

