开发独立内核的操作系统

- 引导程序与操作系统内核
- 汇编程序与C程序编译器的组合
- 操作系统内核
- 实验项目3说明

1 引导程序与操作系统内核

- 引导操作系统
  - 计算机硬件加载操作系统并执行，让操作系统接管硬件系统。
    - 能放进引导扇区，做成引导扇区程序
    - 不能放进引导，实际上操作系统功能多，程序规模大，执行代码不能直接放在一个引导扇区内容，
- 引导程序
  - 专门设计一个程序，放在引导扇区，用于加载操作系统执行代码并将控制权移交给操作系统
  - 设计和加载引导程序，第一个实验项目已解决
  - 设计和加载操作系统内核，第二个实验项目已解决。

2 C与汇编的交叉调用

- 操作系统为什么要用到汇编语言
  - 设置自身运行模式和环境，需要设置硬件寄存器
  - 设置I/O端口实现I/O操作
  - 初始化中断向量表和实现中断处理
  - 实现控制原语
- 操作系统为什么要用C语言
  - 适合构造复杂的数据结构和相关数据结构的管理
  - 实现复杂的功能或算法
- C与汇编的交叉调用是现有操作系统的开发方法

2.1 C与汇编混合编程例子

- 一个程序由汇编语言模块和C语言模块产生
- 任务：
  - 汇编程序调用c程序将一个字符串中的小写字母转换为大写，然后显示该字符串
- 本例程序由两个模块组成showstr.asm和upper.c
- 汇编语言为主入口模块
- 产生一个COM格式的执行程序
- 用TCC和TASM分别编译 (操作)

C程序源代码

```
1.  /*程序源代码（upper.c）*/
2.  char Message[10]="AaBbCcDdEe";
3.        /*变量_Message,初值为AaBbCcDdEe*/
4.  upper(){
5.    int i=0;
6.    while(Message[i]) {
7.          if (Message[i]>='a'&&Message[i]<='z')
8.              Message[i]=Message[i]+'A'-'a';
9.          i++;
10.    }
11. }
```

```
                                ;程序源代码（showstr.asm）
1.  extrn _upper:near           ;声明一个c程序函数upper
2.  extrn _Message:near         ;声明一个外部变量
3.  _TEXT segment byte public 'CODE‘
4.  assume cs:_TEXT
5.    org 100h
6.  start:  mov  ax,  cs
7.          mov  ds,  ax        ; DS = CS
8.          mov  es,  ax        ; ES = CS
9.          mov  ss,  ax        ; SS = CS
10.         mov  sp, 100h
11.         call near ptr _upper    ;调用C的函数
12.         mov bp, offset _Message ; BP=当前串的偏移地址
13.         mov ax, ds          ;BP = 串地址
14.         mov es, ax          ;置ES=DS
15.         mov cx, 10          ; CX = 串长（=10）
16.         mov  ax, 1301h      ;  AH = 13h（功能号） AL = 01h（光标置于串尾）
17.         mov bx, 0007h       ; 页号为0(BH = 0) 黑底白字(BL = 07h)
18.         mov  dh, 10         ; 行号=10
19.         mov dl, 10          ; 列号=10
20.         int10h              ; BIOS的10h功能：显示一行字符
21.         jmp $
22. _TEXT ends
23. end start
```

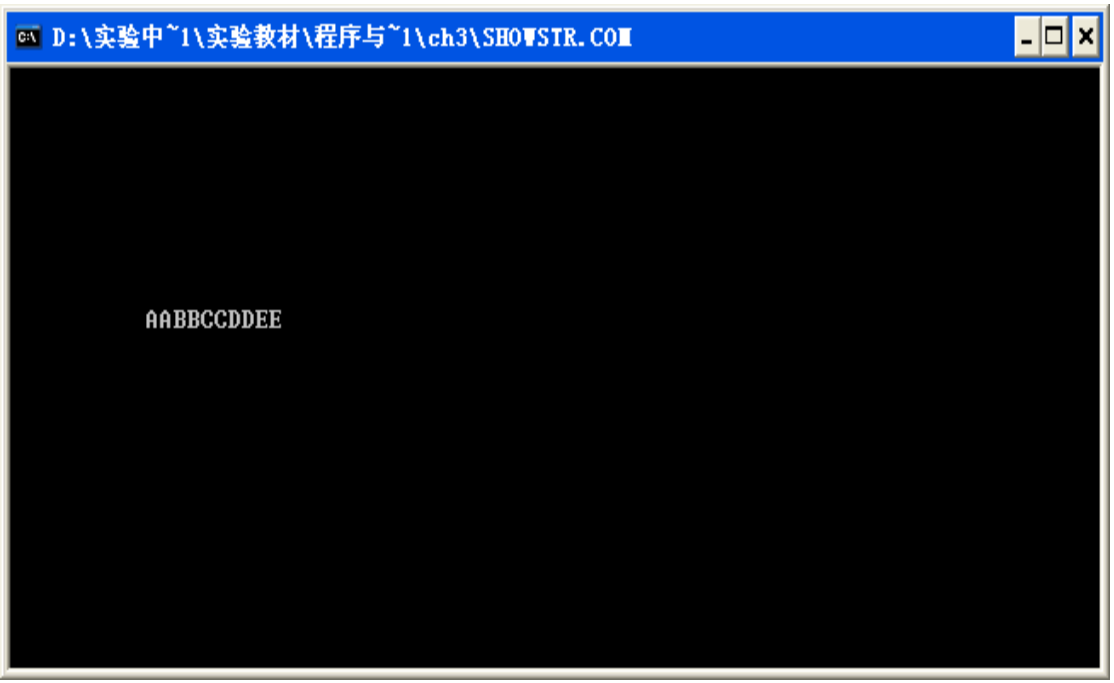
TCC与TASM环境使用(操作)

- 环境
  - TCC编译器
  - TASM汇编器
  - TLINK链接器
- 使用TCC编译命令

  tcc -mt -c –o upper.obj upper.c >ccmsg.txt
- TASM汇编命令

  tasm showstr.asm showstr.obj > amsg.txt
- 链接命令

  tlink /3 /t showstr.obj upper.obj , showstr.com,,


D:\实验中~1\实验教材\程序与~1\ch3\SHOWSTR.COM
AABBCCDDEE

2.2.1 C与汇编混合编程基本问题

- 变量互相引用

  例如，showstr.asm中12行

  mov bp, offset _Message ; BP=当前串的偏移地址

  引用C模块中的字符串变量Message

- 过程互相调用

  例如， showstr.asm中11行

  call near ptr _upper

  引用C模块中的函数upper()

- 参数传递
- 链接

观察C程序的汇编代码

- 我们编写一个简单的C程序，编译产生的汇编代码（局部）：

```
int a=3,b=4,c;
void f(int,int );
cmain(){
    f(a,b);
}
void f(int u,int v) {
  c=u+v;
}
```

```
_TEXT   segment   byte public 'CODE'
_main   proc      near
    push  word ptr _DATA :_b
    push  word ptr _DATA :_a
    call  near ptr _f
    pop   cx
    pop   cx
    ret
_main   endp
_f  proc  near
    push  bp
    mov   bp,sp
    mov   ax,word ptr [bp+4]
    add   ax,word ptr [bp+6]
    mov   word ptr _DATA :_c,ax
    pop   bp
    ret
_f  endp
_TEXT   ends
```

C语言程序编译规则

1. C语言程序中，变量名汇编后前面加了下划线，如_a和_b；函数名也如此，如_f；
2. 参数传递时，用push word ptr _DATA :_b就是实参b压栈，而push word ptr _DATA :_a 就是实参a压栈，然后调用函数f(a,b)，之后两个POP指令是调用后清除栈中的参数。说明调用C函数时，参数按后面参数先进栈的顺序压栈。

更复杂的一个C程序的编译结果

```
/*;程序源代码（strcopy.c）*/
void myprint()          ;/*声明myprint */
void putch()           ;/*声明putch */
char str1[80]="AA\n";
main(){
   char str2[80]="BB\n";
   char str3[80];
   putch('A');
   myprint(str1);
   myprint(str2);
   myprint(str3);
   while (1);
}
void putch(char ch) {
    ch=ch+1;
}
void myprint(char* str) {
    *str=*str+1;
}
```

编译结果，老师加了备注;

符号与变量引用规则

- 引用C程序中的变量和函数名
  - 在编译后，前面都加了下划线，所以汇编程序中引用C程序中的变量和函数名时，要加下划线；
  - 例如，变量名Message和函数名upper
- 引用汇编程序中变量和函数名
  - 汇编程序中变量名和函数名前面要加下划线
  - C程序中才能引用时去掉下划线。

函数调用参数传递与栈操作

- 汇编模块中调用C模块中的函数
  - 调用前要用extrn声明C模块的函数
  - 根据C中函数原型，用栈传递参数，顺序后参先进栈
  - 调用C函数后，要将栈中参数弹出
  - 进栈出栈以字为单位
- C模块中调用汇编模块中的过程
  - 汇编模块的过程从栈中取得参数，不应出栈，顺序与C进栈对应

3. 编译器选择和测试

- 汇编语言选择
- C语言选择

2.4 操作系统内核

- 操作系统的执行代码格式选择
  - 纯机器码
    - BIN
    - COM
  - 浮动代码
    - EXE
    - ELF

2.5 内核开发参考原型

- TASM汇编语言模块
- C语言模块
- 注意事项

让Notepad++调用TCC直接编译源代码

- 启动Notepad++，按一下“F5”键，然后将下面的内容复制进去。

  cmd /k tcc –c "$(FULL_CURRENT_PATH)" & PAUSE & EXIT
- 输入完了,要点保存。
  这里给我们要创建的“编译”功能命个名，比如说我这里命名为:TCCompiling，同时指定一个快捷键，比如说我这里指定的是Ctrl+F7
- 如果没有意外的话你就可以开始使用Notepad++写C代码了，写完想编译时只需简单的按一下Ctrl+F7即可

2.6 实验项目3说明

- 用C和汇编实现操作系统内核
- 增加批处理能力
  - 提供用户返回内核的一种解决方案
  - 一条在内核的C模块中实现
    - 在磁盘上建立一个表，记录用户程序的存储安排
    - 可以在控制台查到用户程序的信息，如程序名、字节数、在磁盘映像文件中的位置等
    - 设计一种命令，命令中可加载多个用户程序，依次执行，并能在控制台发出命令
    - 在引导系统前，将一组命令存放在磁盘映像中，系统可以解释执行