

第7章 在C/C++中使用汇编

罗文坚
中国科大 计算机学院

<http://staff.ustc.edu.cn/~wjluo/mcps/>

1

本章内容

- 16位DOS应用程序中使用汇编语言与C/C++语言
- 32位应用程序中使用汇编语言与Visual C/C++语言
- 汇编和C++混合目标码

2

本章内容

- 16位DOS应用程序中使用汇编语言与C/C++语言
- 32位应用程序中使用汇编语言与Visual C/C++语言
- 汇编和C++混合目标码

3

Visual C++内嵌汇编的格式

- 两种方式:
 1. 所有汇编代码都放在__asm块中。
 2. 每条指令前加__asm。

```
__asm {  
    statement-1  
    statement-2  
    .....  
    statement-n  
}
```

```
__asm statement-1  
__asm statement-2  
.....  
__asm statement-n
```

4

Visual C++内嵌汇编的格式

- 注释的风格:
 1. 注释可以放在汇编语言块中任何语句的后面。
 2. 使用汇编语言格式的注释或C/C++格式的注释都是可以的。
 3. Visual C++建议尽量避免使用汇编风格的注释, 因为汇编风格的注释有可能与C宏相冲突。
- 例:
 - mov esi, buf ;初始化esi寄存器
 - Mov esi, buf //初始化esi寄存器
 - Mov esi, buf /*初始化esi寄存器*/

5

Visual C++内嵌汇编的基本规则

- 内嵌汇编代码可以做:
 1. 使用CPU指令集中的指令。
 2. 使用寄存器操作数。
 3. 通过名字引用函数参数。
 4. 可以引用在汇编语句块外面声明的代码标号和变量。
 5. 使用汇编风格或C风格的数值表示法。例, 0A26H或0xA26等价。
 6. 在语句中使用PTR操作符。例, inc BYTE PTR[esi]。
 7. 使用EVEN和ALIGN伪指令。

6

Visual C++内嵌汇编的基本规则

- 编写内嵌汇编代码时不能使用的特性：
 1. 使用数据定义伪指令，如DB、DW。
 2. 使用汇编语言表达式的运算符（PTR、LENGTH、SIZE和TYPE除外）。
 3. 使用宏指令，包括MACRO、ENDM等。
 4. 引用段名（但可以用段寄存器作为操作数）。

7

Visual C++内嵌汇编的基本规则

- 寄存器的值：
 - 在汇编语句块的开始处，不能对寄存器的值做任何假设，因为汇编语句块前面执行的语句可能已经对寄存器进行了修改。
 - Microsoft Visual C++的关键字__fastcall导致编译器使用寄存器来传递参数。为了避免寄存器冲突，不要一起使用__fastcall和__asm。
- 长度、类型和大小
 - 在内嵌汇编中可使用LENGTH、SIZE和TYPE操作符。

8

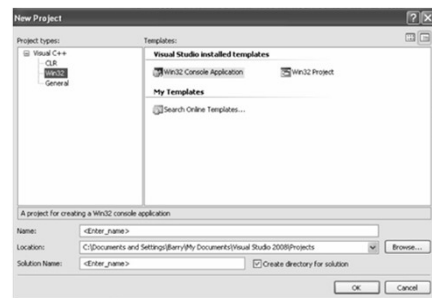
32位应用程序中使用汇编语言与Visual C/C++语言

- 使用控制台I/O访问键盘和显示器
- 直接访问I/O端口
- 开发Windows的Visual C++应用程序

9

使用控制台I/O访问键盘和显示器

- 例，以二进制到十六进制之间的任意进制显示1~100之间的数字。



10

使用控制台I/O访问键盘和显示器

- 例，以二进制到十六进制之间的任意进制显示1~100之间的数字。

```
#include "stdafx.h"
#include <conio.h>

char *buffer = "Enter a number between 0 and 1000: ";
char *buffer1 = "Base: ";
int a, b = 0;

void disp(int base, int data);
```

11

使用控制台I/O访问键盘和显示器

- 例，以二进制到十六进制之间的任意进制显示1~100之间的数字。

```
int _tmain(int argc, _TCHAR* argv[])
{
    int i;
    _cputs(buffer);
    a = _getche();
    while ( a >= '0' && a <= '9' )
    {
        _asm sub a, 30h;
        b = b * 10 + a;
        a = _getche();
    }
}
```

12

使用控制台I/O访问键盘和显示器

- 例，以二进制到十六进制之间的任意进制显示1~100之间的数字。

```
_putch(10);
_putch(10);
_putch(13);
for ( i = 2; i < 17; i++ )
{
    _cputs(buffer1);
    disps(10,i );
    _putch(' ');
    _putch('=');
    _putch(' ');
    disps(i, b);
    _putch(10);
    _putch(13);
}
getche();
return 0;
}
```

13

使用控制台I/O访问键盘和显示器

- 例，以二进制到十六进制之间的任意进制显示1~100之间的数字。

```
void disps(int base, int data)
{
    int temp;
    _asm
    {
        mov  eax, data
        mov  ebx, base
        push ebx
        disps1:
        mov  edx,0
        div  ebx
        push edx
        cmp  eax,0
        jne  disps1
        disps2:
        pop  edx
        cmp  ebx,edx
        je   disps4
        add  dl,30h
        cmp  dl,39h
        jbe  disps3
        add  dl,7
        disps3:
        mov  temp,edx
    }
    _putch(temp);
    _asm jmp disps2;
disps4:;
}
```

14

直接访问I/O端口

- 如果要读写I/O端口，可以使用控制台I/O命令。
 - 例，_inp(port), _outp(port, byte_data)。
- 如果使用Window XP、Windows Vista这样的Windows环境，I/O端口是不能直接访问的。
 - 这些系统下访问I/O端口的唯一办法是开发内核驱动。

15

本章内容

- 16位DOS应用程序中使用汇编语言与C/C++语言
- 32位应用程序中使用汇编语言与Visual C/C++语言
- 汇编和C++混合目标码

16

汇编和C++混合目标码

- 用Visual C++链接汇编语言
- 在C/C++中添加新的汇编语言指令

用Visual C++链接汇编语言

- 汇编模块

```
.586
.model flat, C
.stack 1024
.code
public Reverse
Reverse proc uses esi, \
arraychar:ptr
    mov esi,arraychar
    mov eax,0
    push eax
    .repeat
        mov al,[esi]
        push eax
        inc esi
    .until byte ptr [esi] == 0
    mov esi,arraychar
    .while eax != 0
        pop 'eax
        mov [esi],al
        inc esi
    .endw
    Ret
Reverse     endp
End
```

17

18

用Visual C++链接汇编语言

- C++模块

```
#include <stdio.h>
#include <conio.h>

extern "C" void Reverse(char *);

char chararray[17] = "So what is this?";

int main(int argc, char* argv[])
{
    printf ("%s \n", chararray);
    Reverse (char array);
    printf ("%s\n", chararray);
    getche();
    return 0;
}
```

19

本章小结

- 内嵌汇编
 - 了解基本格式和要求
- 混合编程
 - 了解编程格式和链接过程

20

作业

- （补充题7.1）编写一个短的可由C++使用的汇编语言模块，该程序将一个数字变为其绝对值。程序名为Abs，假定数字是一个32位有符号整数。

21