《汇编语言程序设计实验》

第五次上机实验任务

实验名称：实验五 汇编语言与C语言混合编程。

1. 实验目的和要求

(1) 掌握汇编语言程序与C语言程序混合编程的方法；

(2) 熟悉C编译器的基本优化方法;

(3) 了解C语言编译器的命名方法，主、子程序之间参数传递的机制。

1. 实验内容

任务1：在C 语言程序中嵌入汇编语言指令语句序列

对于实验二的程序进行改造，主控程序、以及输入输出等功能用C语言实现，学生姓名搜索和成绩计算用C程序中嵌入汇编指令语句序列的方式实现。

任务2：在C语言程序中调用 汇编语言实现的函数

对于实验二的程序进行改造，主控程序、以及输入输出等功能用C语言实现，学生姓名搜索和成绩计算用独立的汇编语言子程序的方式实现； 在C语言程序中调用汇编语言子程序。

**要求：**

(1)在不同的C语言开发环境中实现与汇编语言程序的混合编程，其操作方法有可能是不同的。请大家选择自己熟悉的C语言开发环境并查找相关的资料完成本实验。

1. 在实验报告中，比较详细的给出你的开发环境及其实现方法。
2. 观察C语言编译器中对各种符号的命名规则（指编译器内部可以识别的命名规则，比如，符号名前面是否加下划线“\_”，等），主、子程序之间参数传递的机制，通过堆栈传递参数后堆栈空间回收的方法。
3. 对混合编程形成的执行程序，用调试工具观察由C语言形成的程序代码与由汇编语言形成的程序代码之间的相互关系，包括段、偏移的值，汇编指令访问C的变量时是如何翻译的，等。
4. 请尝试在C语言源程序中不合理地嵌入汇编语言的指令语句，达到破坏C语言程序的正确性的目的。比如，在连续的几条C语言语句中间加入一条修改AX寄存器的汇编指令语句，而AX的内容在此处本不该被修改，这样就可观察到破坏C语言程序正确性的效果（该项实验表明：在C语言程序中，若不考虑上下语句翻译成怎样的机器码而随意嵌入汇编指令语句时，有可能存在出错的风险）。
5. 观察C编译器的优化策略对代码的影响。
6. 通过调试混合编程的程序，体会与纯粹汇编语言编写的程序的调试过程的差异。

（8）通过本次实验，希望大家明白：不同的编程语言是可以协同解决一个问题的，而且可以利用不同语言的特点来更好地解决问题；利用汇编语言的知识，能够更好地理解高级语言的内部处理原理与策略，为编写更好的C语言程序、用好C编译器提供支持。

**参考文献：**

[1]许向阳，《80X86汇编语言程序设计上机指南》“第十一章 汇编语言程序与C程序的连接”。

附件资料说明：

1. 第十一章 汇编语言程序与C程序的连接.doc
2. C 程序调用 汇编语言函数 操作说明.doc

操作步骤与多模块编程时建立一个项目（或工程，PROJECT）相同，只是选择模块时除了C语言的模块，还有汇编语言的模块（源程序或OBJ文件）。

1. BC3.1 或 TC3.0下混合编程的方法与上述“【1】【2】”中介绍的类似，但还需要注意一些区别。首先是C编译器需要外挂一个汇编程序以帮助处理汇编语言的语句，一般会使用TASM，它与MASM基本一致，语法规则只有少许差异。**教学网站中可以下载一个带DOS虚拟机的完整BC31软件，包括C编译器，TASM，TD等**。其次，命名规则有点不同，这里不需要在ASM关键词前加下划线。另外，若想C编译器对嵌入的汇编语言语句不报错，还需要在C语言程序开始之处加上说明语句#pragma inline，例如：

**#include <stdio.h>**

**#pragma inline**

**int main()**

**{ int count;**

**asm mov count,5**

**printf("count=%d",count);**

**return 0;**

**}**