

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验五** 汇编语言与C语言混合编程

**实验时间： 2016-4-19，14：30-17：30 实验地点： 南一楼804室102号实验台**

**指导教师： 张勇**

**专业班级：计算机科学与技术201409班**

**学 号： U2014414808 姓 名： 王林**

**同组学生： 无 报告日期： 2016年 4 月 19日**

**原创性声明**

  本人郑重声明：本报告的内容由本人独立完成，有关观点、方法、数据和文献等的引用已经在文中指出。除文中已经注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品或成果，不存在剽窃、抄袭行为。特此声明！

学生签名： 王林

日期： 2016.4.19

成绩评定

|  |  |  |
| --- | --- | --- |
| 实验完成质量得分（70分）（实验步骤清晰详细深入，实验记录真实完整等） | 报告撰写质量得分（30分）（报告规范、完整、通顺、详实等） | 总成绩（100分） |
|  |  |  |

指导教师签字：

日期：

**目录**

[实验五 汇编语言与C语言混合编程 1](#_Toc448848738)

[一、实验目的与要求 1](#_Toc448848739)

[二、实验内容 1](#_Toc448848740)

[三、实验过程 1](#_Toc448848741)

[5.1任务1 1](#_Toc448848742)

[5.1.1设计思想及存储单元分配 1](#_Toc448848743)

[5.1.2流程图 1](#_Toc448848744)

[5.1.3源程序 2](#_Toc448848745)

[5.1.4实验步骤 4](#_Toc448848746)

[5.2.5实验记录 4](#_Toc448848747)

[5.2任务2 4](#_Toc448848748)

[5.2.1设计思想及存储单元分配 5](#_Toc448848749)

[5.2.2流程图 5](#_Toc448848750)

[5.2.3源程序 5](#_Toc448848751)

[5.2.4实验步骤 7](#_Toc448848752)

[5.2.5实验记录 8](#_Toc448848753)

[四、体会 9](#_Toc448848754)

[五、参考文献 9](#_Toc448848755)

实验五 汇编语言与C语言混合编程

一、实验目的与要求

(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。

二、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。

三、实验过程

5.1任务1

在C 语言程序中嵌入汇编语言指令语句序列。

5.1.1设计思想及存储单元分配

设计思想：主体采用C语言编程，输入和输出部分采用C语言编程，然后搜索学生以及计算平均成绩用汇编编写，将汇编代码块嵌入到C语言。

内存分配：在C语言中嵌入汇编程序。在C语言中定义一个结构体，即学生信息的结构，包含一个储存姓名的长度为12的字符数组，包含一个大小为4的short型数组，用于储存语文、数学、英语以及平均成绩。然后在C程序中创建一个大小为1000的结构数组，用于保存1000个学生的信息。

然后再C语言中输入查找的学生姓名，通过lea取结构数组和字符数组的偏移地址，并在汇编代码中进行操作。

5.1.2流程图

无（流程图和实验二相同）

5.1.3源程序

【test1.c】

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

struct stuinfo {

char name[12]; //储存姓名

short score[4]; //储存成绩

};

int main() {

struct stuinfo info[1000];

char strn[12]; //输入的姓名

int i, j; //计数器

short label; //标记变量

char op;

//初始化学生表信息

for (i = 0; i < 997; i++) {

strcpy\_s(info[i].name, 12, "aaaaaaaa\_a");

memset(info[i].score, 0, 8);

}

//初始化三个学生信息用于测试

strcpy\_s(info[997].name, 12, "zhangsan\_a");

info[997].score[0] = 86; info[997].score[1] = 82; info[997].score[2] = 88; info[997].score[3] = 0;

strcpy\_s(info[998].name, 12, "zhangsan\_b");

info[998].score[0] = 48; info[998].score[1] = 79; info[998].score[2] = 46; info[998].score[3] = 0;

strcpy\_s(info[999].name, 12, "zhangsan\_c");

info[999].score[0] = 96; info[999].score[1] = 92; info[999].score[2] = 98; info[999].score[3] = 0;

do{

printf("Inlut the name of the student:");

gets\_s(strn, 12);

\_asm {

mov ecx, 1000

lea edi, info[0];

LOAPI:

lea esi, strn

mov edx, 12

LOAPJ :

mov al, byte ptr[esi]

mov bl, byte ptr[edi]

cmp al, bl

jnz NEXT

cmp al, '\0'

jz FIGURE

dec edx

inc edi

inc esi

jmp LOAPJ

NEXT :

dec ecx

jz REINPUT

add edx, 8

add edi, edx

jmp LOAPI

FIGURE :

add edi, edx

mov al, byte ptr[edi]

mov ah, 0

add ax, ax

add ax, ax

mov dl, byte ptr[edi + 2]

mov dh, 0

add ax, dx

add ax, dx

mov dl, byte ptr[edi + 4]

mov dh, 0

add ax, dx

mov bl, 7

div bl

mov ah, 0

mov word ptr[edi + 6], ax

}

for (i = 0; i < 1000; i++){

if (!strcmp(info[i].name, strn)){

printf("%s\t", info[i].name);

for (j = 0; j < 4; j++){

printf("%d\t", info[i].score[j]);

}

printf("\n");

}

}

//汇编代码模块

\_asm{

REINPUT:

}

printf("Input Y to exit,N continue...");

op = getchar(); getchar();

} while (op == 'Y' || op == 'y');

for (i = 0; i < 1000; i++) {

printf("%s\t", info[i].name);

for (j = 0; j < 4; j++){

printf("%d\t", info[i].score[j]);

}

printf("\n");

}

getchar();

return 0;

}

5.1.4实验步骤

1、编辑C语言源代码，定义好数据结构，写好输入以及输出部分。

2、进行汇编部分代码的嵌入，通过网上查阅资料以及阅读实验上机教材指导书，学习有关汇编嵌入C语言的相关知识，然后进行汇编代码块的编写。

3、对.c文件进行编译、链接、生成可执行程序，并检测程序是否正确。

4、分析总结本次学习遇到的困难，为今后的学习打下基础。

5.2.5实验记录

1、实验环境条件：core i5 2.8GHz 2G内存 位命令行窗口；Sublime Text 2；VS2013。

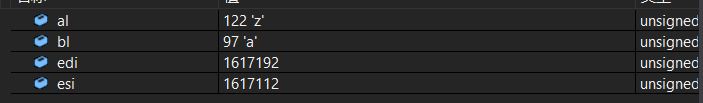
2、VS2013提供了C语言和汇编语言混合编程的环境，比较方便，汇编代码块通过

\_asm{

}写在括号中，写完直接编译即可。

3、VS2013可识别符号名前加一个‘\_’或者两个’\_’，‘\_asm’会被识别成汇编代码块。

4、由C语言进入汇编代码部分后，内存中可以发现寄存器的值发生改变。



**图 5.1 汇编部分内存示意图**

5、嵌入汇编代码后调试时会显示对应寄存器的变化。并且显示对应的入栈、出栈操作。

5.2任务2

在C语言程序中调用 汇编语言实现的函数。

5.2.1设计思想及存储单元分配

设计思想：主体部分采用C语言编程，将输入和输出部分用c语言编程，定义好数据结构以及内存分配，作为第一个.c文件。另外姓名搜索和成绩计算采用汇编编程，生成.asm文件，将asm作为子程序供c语言调用。

内存分配：在C语言中嵌入汇编程序。在C语言中定义一个结构体，即学生信息的结构，包含一个储存姓名的长度为12的字符数组，包含一个大小为4的short型数组，用于储存语文、数学、英语以及平均成绩。然后在C程序中创建一个大小为1000的结构数组，用于保存1000个学生的信息。

然后将c语言定义的数据通过参数传递到汇编程序中，在c语言中定义一个类型为extern void的函数，参数为成绩信息结构数组的首地址和字符数组的首地址。在汇编代码中，将函数申明为public，并定义参数为dword型。

然后分别编译，生成obj文件，链接生成可执行文件。

5.2.2流程图

无（流程图和实验二相同）

5.2.3源程序

【test2.c】

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

struct stuinfo {

char name[12]; //储存姓名

short score[4]; //储存成绩

};

extern void dif(struct stuinfo \*, char \*);

int main() {

struct stuinfo info[1000];

char strn[12]; //输入的姓名

int i, j; //计数器

short label; //标记变量

char op;

//初始化学生表信息

for (i = 0; i < 997; i++) {

strcpy\_s(info[i].name, 12, "xxxxxxxx\_x");

memset(info[i].score, 0, 8);

}

strcpy\_s(info[997].name, 12, "zhangsan\_a");

info[997].score[0] = 86; info[997].score[1] = 82; info[997].score[2] = 88; info[997].score[3] = 0;

strcpy\_s(info[998].name, 12, "zhangsan\_b");

info[998].score[0] = 48; info[998].score[1] = 79; info[998].score[2] = 46; info[998].score[3] = 0;

strcpy\_s(info[999].name, 12, "zhangsan\_c");

info[999].score[0] = 96; info[999].score[1] = 92; info[999].score[2] = 98; info[999].score[3] = 0;

do{

printf("Inlut the name of the student:");

gets\_s(strn, 12);

dif(info, strn);

for (i = 0; i < 1000; i++){

if (!strcmp(info[i].name, strn)){

printf("%s\t", info[i].name);

for (j = 0; j < 4; j++){

printf("%d\t", info[i].score[j]);

}

printf("\n");

}

}

printf("Input Y to exit,N continue...");

op = getchar(); getchar();

} while (op == 'Y' || op == 'y');

for (i = 0; i < 1000; i++) {

printf("%s\t", info[i].name);

for (j = 0; j < 4; j++){

printf("%d\t", info[i].score[j]);

}

printf("\n");

}

getchar();

return 0;

}

【test2.asm】

.386

.model flat, c

.code

public dif

dif proc par1:dword,par2:dword

mov ecx, 1000

mov edi, par1;

LOAPI:

mov esi, par2

mov edx, 12

LOAPJ:

mov al, byte ptr[esi]

mov bl, byte ptr[edi]

cmp al, bl

jnz NEXT

cmp al, 0

jz FIGURE

dec edx

inc edi

inc esi

jmp LOAPJ

NEXT:

dec ecx

jz REINPUT

add edx, 8

add edi, edx

jmp LOAPI

FIGURE:

add edi, edx

mov al, byte ptr[edi]

mov ah, 0

add ax, ax

add ax, ax

mov dl, byte ptr[edi + 2]

mov dh, 0

add ax, dx

add ax, dx

mov dl, byte ptr[edi + 4]

mov dh, 0

add ax, dx

mov bl, 7

div bl

mov ah, 0

mov word ptr[edi + 6], ax

REINPUT:

ret

dif endp

end

5.2.4实验步骤

1、编辑C语言源代码，定义好数据结构，写好输入以及输出部分。

2、编写汇编部分代码，作为一个子程序。实现学生姓名搜索功能以及中找到学生后对学生平均成绩的计算功能。

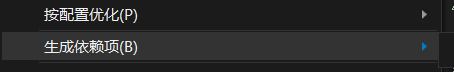
3、对.c文件和.asm文件进行编译生成对应的OBJ文件，然后链接、生成可执行程序，并检测程序是否正确。

4、分析总结本次学习遇到的困难，为今后的学习打下基础。

5.2.5实验记录

1、实验环境条件：core i5 2.8GHz 2G内存 位命令行窗口；Sublime Text 2；VS2013。

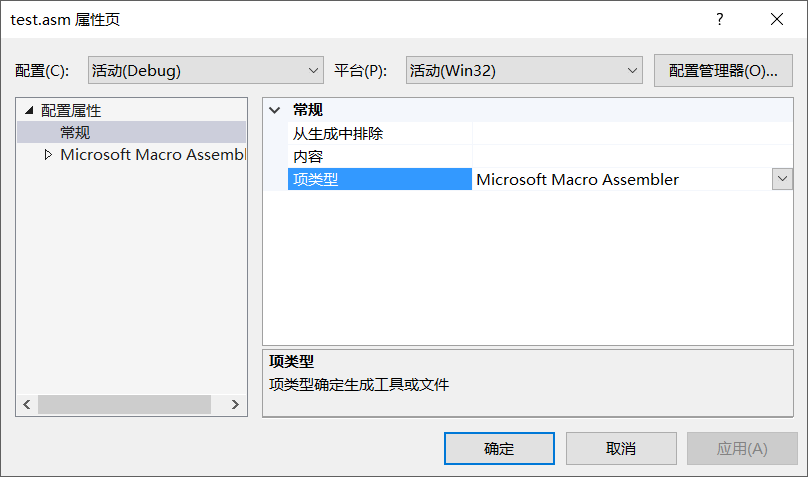
2、私用VS进行任务2时，要修改部分设置，如图操作：



在工程名上右键，选择生成依赖项。

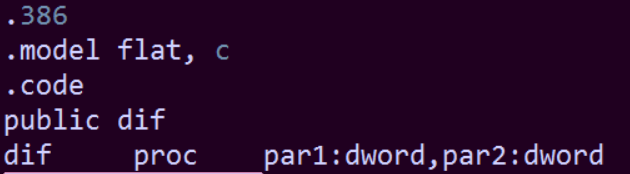


勾选上masm选项。



然后在文件右键-属性-项类型选择如图。

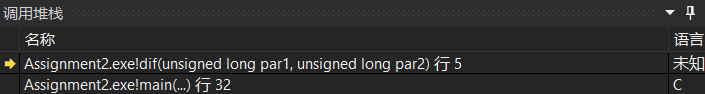
3、在.c文件中，要添加对子函数的引用声明：extern void dif(struct stuinfo \*, char \*);在.asm文件中，添加公开声明和子模块声明：



**图 5.2 子程序模块定义图解**

4、在.asm文件中，通过mov操作来对寄存器赋值，将传进来的参数赋值，即是取偏移地址操作，然后通过变址寻址操作实现内存访问。

5、调用汇编子程序时，有压栈操作，将下一条指令和返回指令压栈。



**图5.3 调用汇编子程序**

6、进入子程序后，可以查看寄存器的值。

7、对C语言进行不同等级的优化，反汇编看到的代码差异较大。

8、实验中对内存储存信息初始化要正确，其次，在汇编代码中，如果用到寄存器，最好先初始化为0，然后再复制和其他操作。

四、体会

本次上机实验中，用到的工具主要是VS2013。VS2013在本次实验中比较方便，可以直接对汇编文件进行编译。而且在本次实验中，比较容易调试。

通过本次实验，尅发现汇编语言也可以用在C语言中进行混合编程，从而使程序有更好地结构并且有更高的执行效率。可见汇编语言在其他语言中也能发挥很大的作用。不同语言可以协同解决一个问题，汇编潜入到C语言，可以编写更好地C语言的代码，也可以让我们更好的理解C语言内部处理机制。

但是，对于汇编内嵌到c语言和与c语言外链还不够熟悉，需要进一步练习。

五、参考文献

[序号] 作者.书名.版本(第×版).译者.出版地:出版者,出版年：起页-止页

[序号] URL: 网络地址，如URL:http://www.cbs.dtu.dk/services