实验三

(1/2)子程序设计一

一. 基础性实验

1. 实验目的

- 1. 掌握主程序和子程序之间的调用关系及其调用方法.
- 2. 掌握子程序调用过程中近程调用与远程调用的区别。
- 3. 掌握子程序设计方法,能合理划分子程序.
- 4. 掌握汇编子程序的定义/调用/返回/参数传递等有关问题的实现,以及运行过程中的堆栈和标志位变化情况.

2. 实验内容

用堆栈传送参数和参数表地址方式,编制键入 8-bit非压缩 BCD 码 加法并显示的程序.

3. 实验仪器设备

- 硬件环境: AMD Ryzen 9.
- 软件环境: Windows XP Pro SP3 (in VMware Workstation 16 Pro).
- 编辑程序: Visual Studio Code 1.62.3.
- 调试程序: DEBUG.EXE.
- 编译程序: MASM.EXE.
- 连接程序: LINK.EXE.

4. 实验步骤

- 1. 检查并修改原程序
- 2. 编译和连接
- 3. 用单步方式运行程序

debug 5-1.exe

-t

4. 检查并记录各寄存器和存储单元内容的变化

```
_ 🗆 ×
 C:\VINDOVS\system32\cmd.exe - debug 5-1.exe
                                                                               BP=0000 SI=0000 DI=0000 NU UP EI PL NZ NA PO NC
                                              DX = 0000
CS = 0B33
               BX =0000
ES =0B21
                                                                                                                                                 AX=0B31
DS=0B31
                                                                              BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
              BX = 0000
ES = 0B21
                                               DX = 0000
CS = 0B33
                                                                                              SI=0000 DI=0000
                                                                              BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
                                                                                                                                                    → 转到
                               CX=00CF DX=0000 SP=0000
SS=0B31 CS=0B33 IP=007D
SUB CX_CX
                                                                              BP=0000 SI=0000 DI=0000
NV UP EI PL NZ NA PO NC
AX=0B31
DS=0B31
               BX = 0008
ES = 0B21
                                          SUB
0B33:007D 2BC9
AX=0B31
DS=0B31
                              CX=0000 DX=0000 SP=0000
SS=0B31 CS=0B33 IP=007F
MOV AH,01
                                                                              BP=0000 SI=0000 DI=0000
NU UP EI PL ZR NA PE NC
AX=0B31 BX=0008
DS=0B31 ES=0B21
0B33:007F B401
```

5. 思考题

(1) 说明调用指令 CALL 和无条件转移指令 JMP 的区别及应用.

JMP 指令永久地改变程序计数器,不在堆栈中留下信息.

CALL 指令在堆栈中留下信息,这样可以恢复原来的程序执行顺序. CALL 是一条将程序控制权转移到子程序的指令,目的是回到主程序中.

- JMP 常用于形成循环和直接跳转.
- CMP 常用于函数的调用.

二. 加强型实验

1. 实验目的

进一步掌握主程序和子程序之间的调用关系及其调用方法.

2. 实验内容

编写程序求数据区中 10 个无符号数中的最大值和最小值,结果分别存入 MAX 和 MIN 单元中,要求最大值和最小值分别用子程序完成计算,主程序和子程序之间通过寄存器传递参数.

3. 实验仪器设备

- 硬件环境: AMD Ryzen 9.
- 软件环境: Windows XP Pro SP3 (in VMware Workstation 16 Pro).
- 编辑程序: Visual Studio Code 1.62.3.
- 调试程序: DEBUG.EXE.
- 编译程序: MASM.EXE.
- 连接程序: LINK.EXE.

4. 实验步骤

- 1. 检查并修改原程序
- 2. 编译和连接

```
C:\VINDOVS\system32\cmd.exe

Ø Warning Errors
Ø Severe Errors

C:\masm>echo DEBUG_link

C:\masm>link.exe 5-2.obj

Microsoft (R) Overlay Linker Version 3.65

Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [5-2.EXE]:
List File INUL.MAP1:
Libraries [.LIB]:
LINK: warning L4921: no stack segment

C:\masm>echo DEBUG_run

DEBUG_run

C:\masm>5-2.exe

MAX = 92

MIN = 10

C:\masm>pause

Press any key to continue . . .
```

3. 用单步方式运行程序

debug 5-2.exe

-t

4. 检查并记录各寄存器和存储单元内容的变化

```
_ 🗆 ×
             C:\VINDOVS\system32\cmd.exe - debug 5-2.exe
实验红具
            AX=0B31 BX=0000
DS=0B21 ES=0B21
0B33:0003 8ED8
                                          CX=00C7
SS=0B31
                                                         DX =0000
CS =0B33
                                                                         SP=0000
IP=0003
                                                                                        BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
                                                     MOU
                                                                  DS,AX
  DEV
            AX=0B31 BX=0000 CX=00C7 DX=0000 SP=0000
DS=0B31 ES=0B21 SS=0B31 CS=0B33 IP=0005
0B33:0005 8D160A00 LEA DX,[000A]
                                                                                        BP=0000 SI=0000 DI=0000
                                                                                          NU UP EI PL NZ NA PO NC
DS:000A=4D20
 Dev-C++
                                                                                                                                                       转到
                                         CX=00C7 DX=000A SP=0000
SS=0B31 CS=0B33 IP=0009
MOU AH,09
            AX=0B31 BX=0000
DS=0B31 ES=0B21
0B33:0009 B409
                                                                                        BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
  2
            AX=0931 BX=0000
DS=0B31 ES=0B21
0B33:000B CD21
                                          CX=00C7 DX=000A
SS=0B31 CS=0B33
INT 21
                                                                         SP=0000
IP=000B
                                                                                         BP=0000 SI=0000 DI=0000
                                                                                          NU UP EI PL NZ NA PO NC
                                                                                        BP=0000 SI=0000 DI=0000
NU UP DI PL NZ NA PO NC
            AX=0931 BX=0000
DS=0B31 ES=0B21
                                          CX=00C7 DX=000A
SS=0B31 CS=00A7
                                                                         SP=FFFA
IP=107C
            00A7:107C 90
```

(2/2)子程序设计二

一. 基础性实验

1. 实验目的

- 1. 强化主程序和子程序之间的调用关系及其调用方法.
- 2. 掌握嵌套子程序的设计以及调用方法.
- 3. 掌握递归调用子程序的方法.

2. 实验内容

试编制在 ARRAY1 无符号数组中,选出最大值及其所在的位置,然后以 10 进制数形式显示在 CRT 上的源程序.

3. 实验仪器设备

- 硬件环境: AMD Ryzen 9.
- 软件环境: Windows XP Pro SP3 (in VMware Workstation 16 Pro).
- 编辑程序: Visual Studio Code 1.62.3.
- 调试程序: DEBUG.EXE.
- 编译程序: MASM.EXE.
- 连接程序: LINK.EXE.

4. 实验步骤

- 1. 检查并修改原程序
- 2. 编译和连接

```
实验&工具
               C:\VINDOVS\system32\cmd.exe
                                                                                                                               _ 🗆 ×
                  50770 + 450910 Bytes symbol space free
  DEV
                        0 Warning Errors
0 Severe Errors
Dev-C++
               C:∖masm>echo DEBUG_link
DEBUG_link
               C:\masm>link.exe 6-1.obj
  7
               Microsoft (R) Overlay Linker Version 3.65
Copyright (C) Microsoft Corp 1983—1988. All rights reserved.
  mean)
              Run File [6-1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment
               C:\masm>echo DEBUG_run
DEBUG_run
               C:\masm>6−1.exe
0543:0006 7865
              C:\masm>pause
Press any key to continue
```

3. 用单步方式运行程序

debug 6-1.exe

-t

4. 检查并记录各寄存器和存储单元内容的变化

```
_ 🗆 ×
        C:\WINDOWS\system32\cmd.exe - debug 6-1.exe
                                   CX=014C DX=0000
SS=0B31 CS=0B3E
                                                                                 BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
      AX=0B31
DS=0B21
                     BX = 0000
ES = 0B21
                                                                 SP=0000
IP=0003
                                                                                                                                               实验。
0B3E:0003 8ED8
                                                                                                                                                    BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
DS:0000=212F
 AX=0B31 BX=0000 CX=014C DX=0000 SP=0000 DS=0B31 ES=0B21 SS=0B31 CS=0B3E IP=0005 0B3E:0005 8D360000 LEA SI,[0000]
 Dev-
                                                                                                                                               → 转到
                                   CX=014C DX=0000 SP=0000
SS=0B31 CS=0B3E IP=0009
MOU CX,0064
       AX=0B31 BX=0000
DS=0B31 ES=0B21
0B3E:0009 B96400
                                                                                 BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
       AX=0B31 BX=0000
DS=0B31 ES=0B21
0B3E:000C E82200
                                    CX=0064 DX=0000 SP=0000
SS=0B31 CS=0B3E IP=000C
                                                                                BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
                                               CALL
       AX=0B31 BX=0000 CX=0064 DX=0000
DS=0B31 ES=0B21 SS=0B31 CS=0B3E
                                                                   SP=FFFE
IP=0031
                                                                                 BP=0000 SI=0000 DI=0000
                                                                                    NU UP EI PL NZ NA PO NC
       0B3E:0031 50
                                               PUSH
```

二. 加强型实验

1. 实验目的

进一步掌握主程序和子程序之间的调用关系及其调用方法.

2. 实验内容

用子程序结构编程计算: S=1!+2!+3!+4!+5!.

3. 实验仪器设备

- 硬件环境: AMD Ryzen 9.
- 软件环境: Windows XP Pro SP3 (in VMware Workstation 16 Pro).
- 编辑程序: Visual Studio Code 1.62.3.
- 调试程序: DEBUG.EXE.
- 编译程序: MASM.EXE.
- 连接程序: LINK.EXE.

4. 实验步骤

- 1. 检查并修改原程序
- 2. 编译和连接

```
C:\VINDOVS\system32\cad.exe

50928 + 450752 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\masm>cho DEBUG_link
DEBUG_link
C:\masm>link.exe 6-2.obj
Microsoft (R) Overlay Linker Version 3.65
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [6-2.EXE]:
List File INUL.MAP1:
Libraries [.LIB]:
LINK: warning L4021: no stack segment

C:\masm>cho DEBUG_run
DEBUG_run

C:\masm>6-2.exe
153
C:\masm>pause
Press any key to continue . . .
```

3. 用单步方式运行程序

debug 6-2.exe

-t

4. 检查并记录各寄存器和存储单元内容的变化

```
_ 🗆 ×
         C:\VINDOVS\system32\cmd.exe - debug 6-2.exe
                                                                                           BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
        AX=0B31 BX=0000
DS=0B21 ES=0B21
0B32:0003 8ED8
                                         CX=00AD DX=0000 SP=0000
SS=0B31 CS=0B32 IP=0003
MOU DS_AX
实验员
                                                                                                                                                                 AX=0B31 BX=0000
DS=0B31 ES=0B21
0B32:0005 B90500
                                                                          SP=0000
IP=0005
                                                                                           BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
                                                         DX=0000 SP=0
CS=0B32 IP=0
CX,0005
 Dev-
                                                                                                                                                               → 转到
        DS=0B31 ES=0B21 SS=0B31 CS=0B32 IP=0008
0B32:0008 E86700 CALL 0072
                                                                                             NU UP EI PL NZ NA PO NC
                                                                                          BP=0000 SI=0000 DI=0000
NU UP EI PL NZ NA PO NC
        AX=0B31 BX=0000 CX=0005 DX=0000 SP=FFFE
DS=0B31 ES=0B21 SS=0B31 CS=0B32 IP=0072
0B32:0072 81E1FF00 AND CX,00FF
                                                                                          BP=0000 SI=0000 DI=0000
NV UP EI PL NZ NA PE NC
        AX=0B31 BX=0000 CX=0005 DX=0000 SP=FFFE
DS=0B31 ES=0B21 SS=0B31 CS=0B32 IP=0076
0B32:0076 B80000 MOV AX,0000
```