山东大学 计算机科学与技术 学院

汇编语言 课程实验报告

学号：202200130048	姓名：陈静雯	班级：6 班
实验题目：实验四：示例 2.6，2.7		
实验学时：2	实验日期：20241104	

实验目的：全面掌握汇编语言中的过程及其使用，进一步实践编程方法论。掌握通过全局变量、栈和寄存器传递过程参数与返回值的方法。掌握过程的模块化设计，及其嵌套与测试方法。掌握字节、字与双字数据的基本计算方法，以及部分系统调用。

实验环境：Windows10、DOSBox-0.74、Masm64

源程序清单：

1. rank.asm
2. scremp.asm

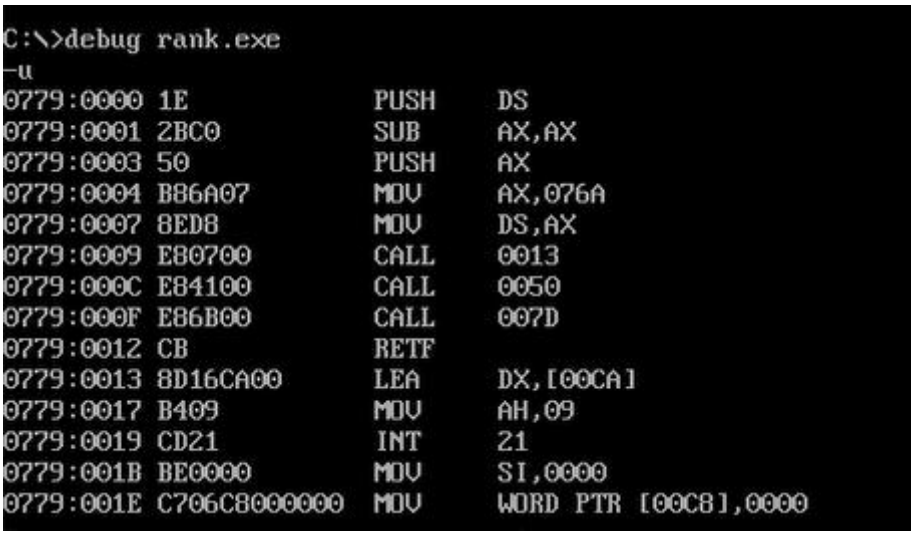
编译及运行结果：

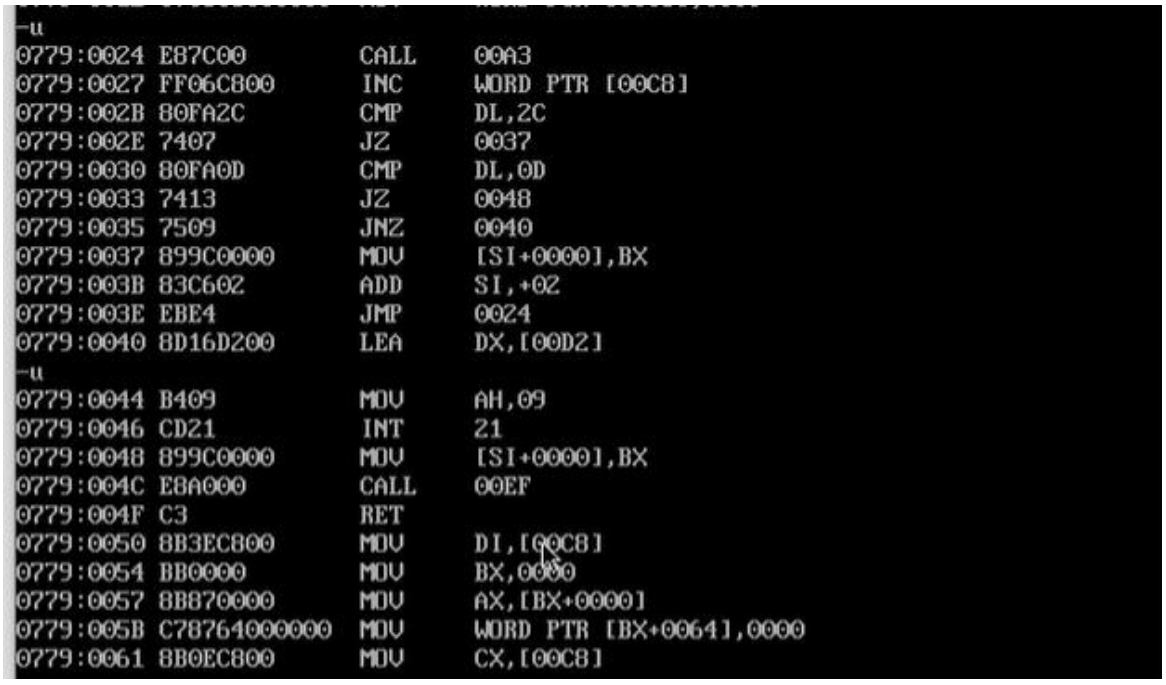
1.rank 运行结果

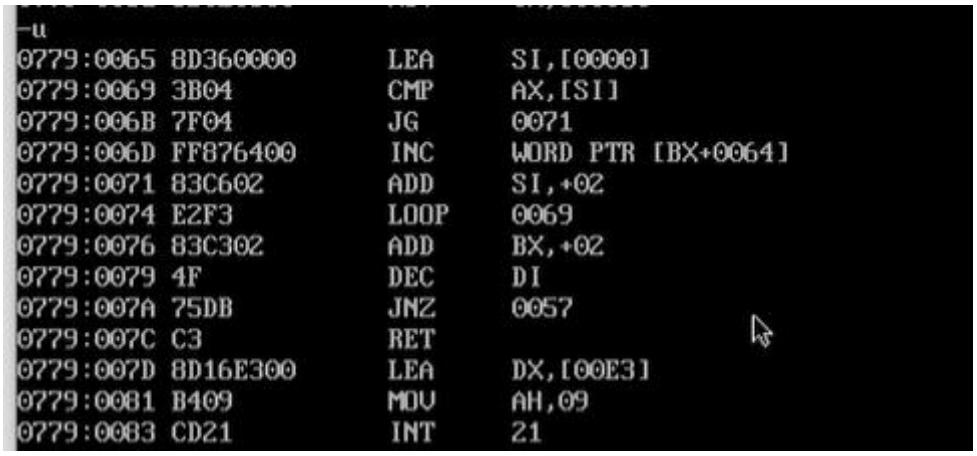
```
C:\>rank
Grade? 67,76,34,100,86,97,54,99,75,98

Rank: 008,006,010,001,005,004,009,002,007,003,

C:\>rank
Grade? 78,99,65,89,74,98,84

Rank: 005,001,007,003,006,002,004,

C:\>rank
Grade? 87,99,100,76.
Input Error1

Rank: 003,002,001,004,
```

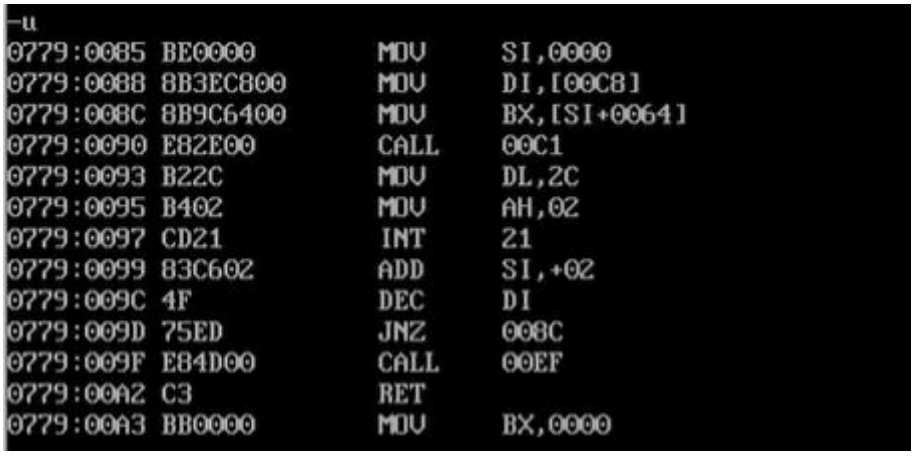
从键盘输入的一个班的学生成绩，并存放于 50 字的 grade 数组中，其中 grade 十 i 保存学号为 i+l 的学生的成绩。然后根据 grade 中的学生成绩，把学生名次填入 50 字的 rank 数组中，其中 rank 十 i 的内容是学号为 i+1 学生的名次。再按学号顺序把名次从终端上显示出来。

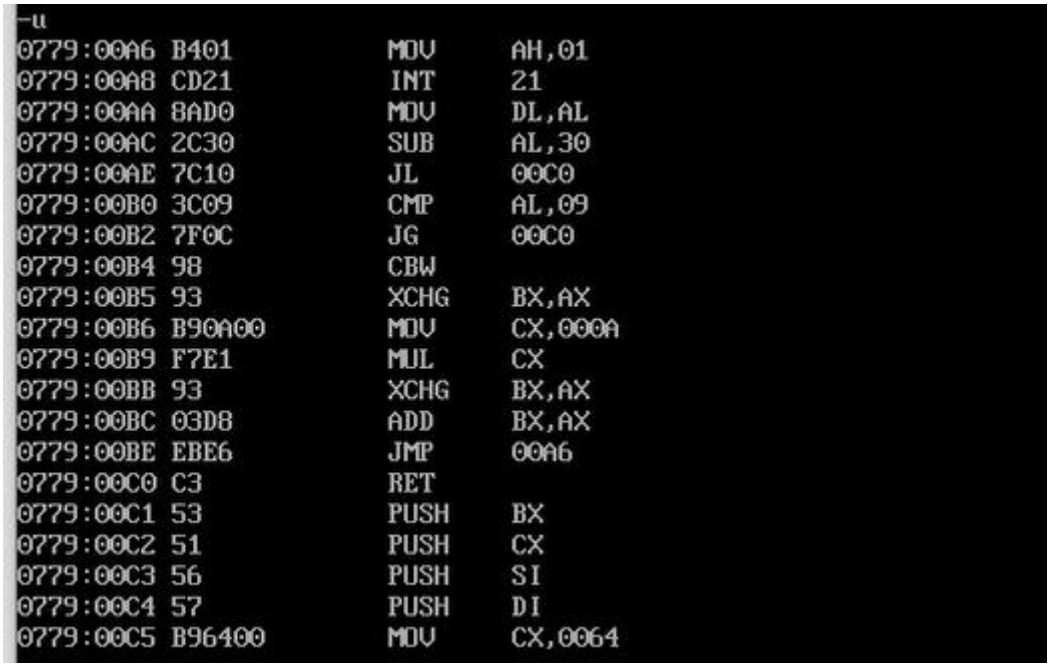
（1）Debug，显示反汇编语句



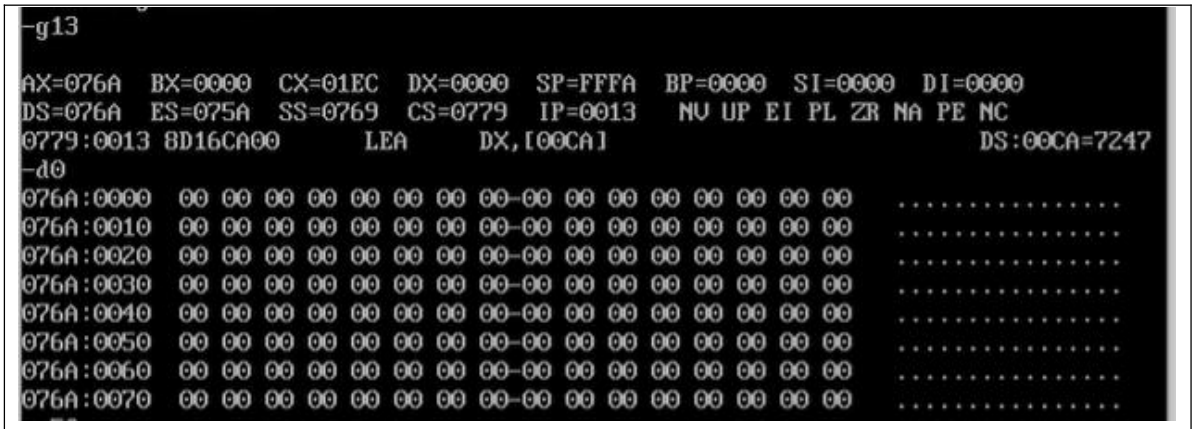




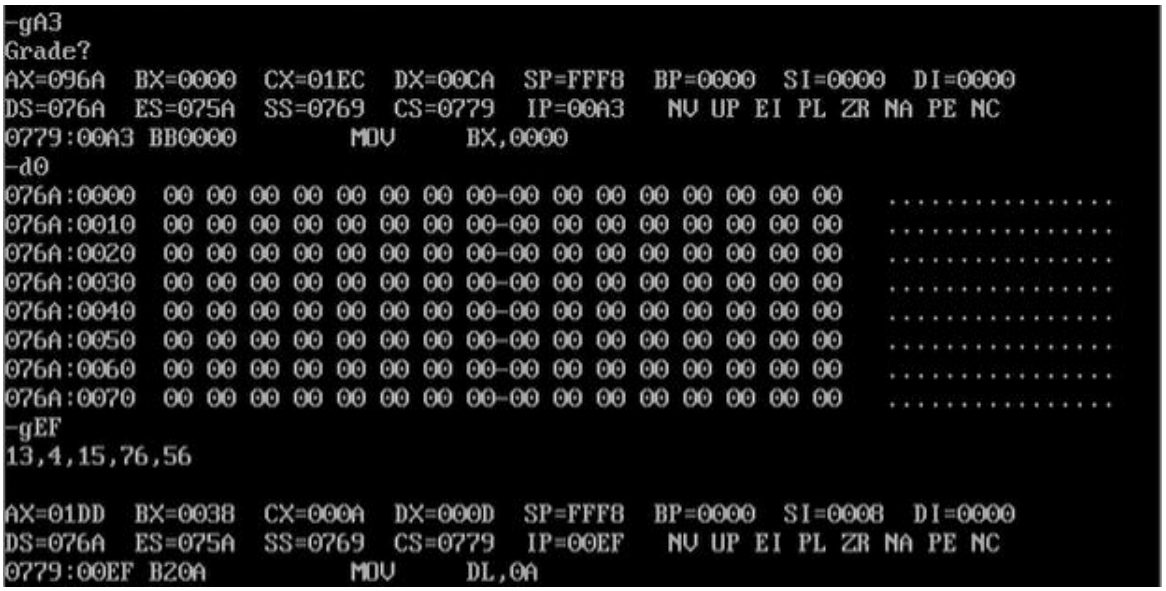




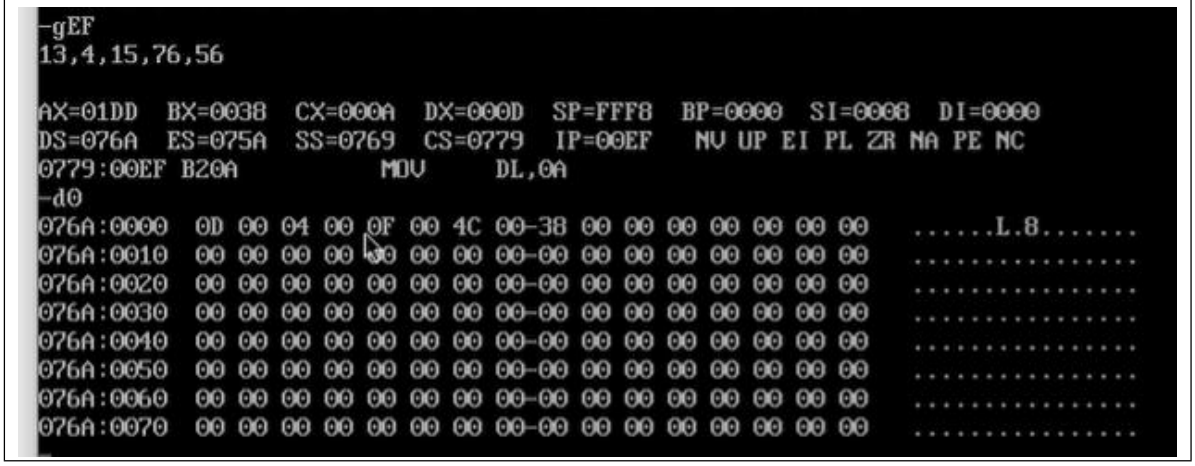
（2）input 模块从 0013 行开始



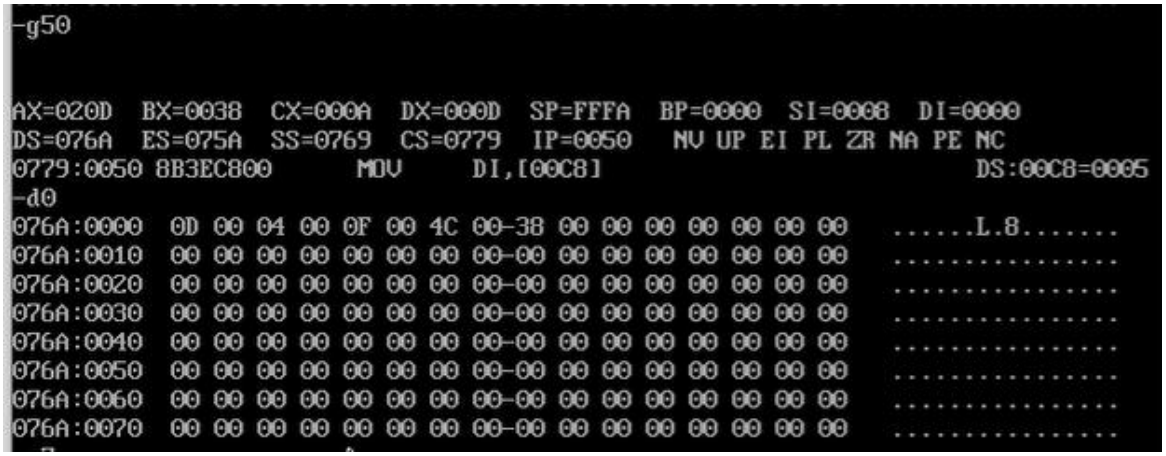
（3）decibin 模块从 A3 行开始，从键盘取一个十进制数转为二进制数存入寄存器



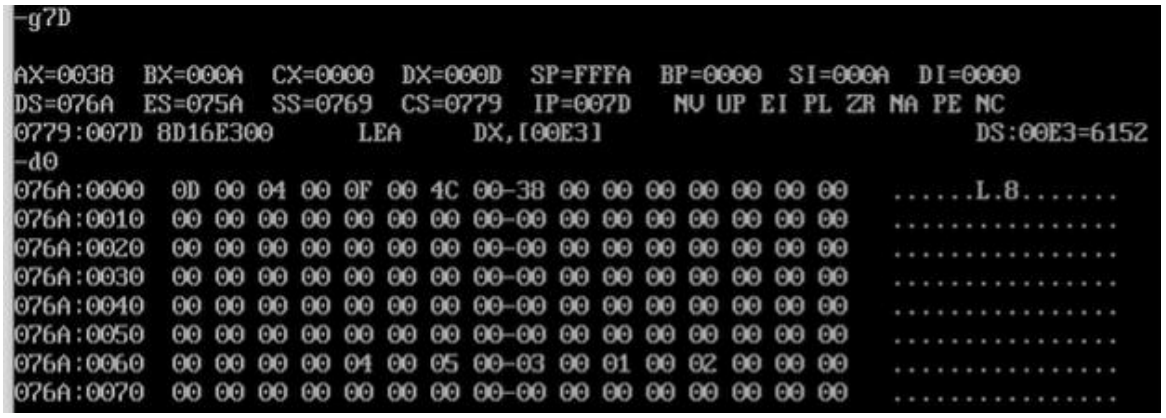
（4）crlf 从 EF 行开始，完成回车、换行操作，可以发现五个数字都被存入对应寄存器



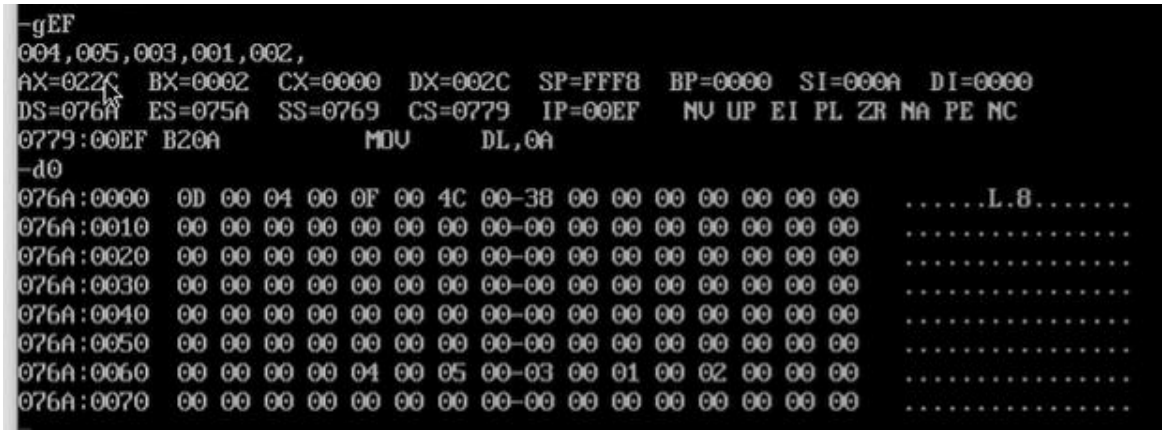
（5）rank 模块从 50 行开始，开始排序



（6）output 从 7D 行开始，把数据段 0064 开始的序号进行输出



（7）通过 EF 行开始的回车换行操作显示输出



2. scremp 运行结果

用户输入工作时间及工资率（即每小时的工资数），显示计算而得的工资数。

```
Hours worked?65535       Rate of pay?65535       Wage =  4294836225.00
                                                 Wage =  4294836225.00
Hours worked?0           Rate of pay?0           Wage =           0.00
                                                 Wage =           0.00
Hours worked?1           Rate of pay?1           Wage =           1.00
                                                 Wage =           1.00
```

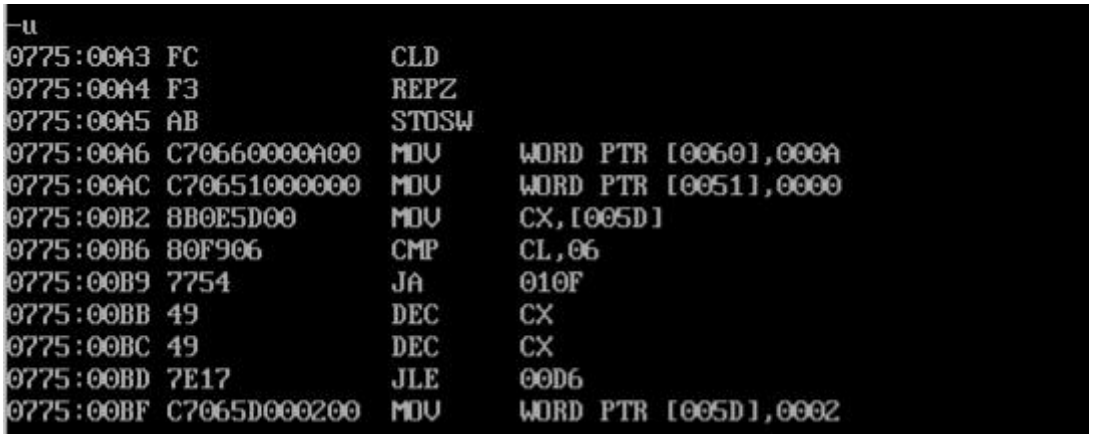
```
Hours worked?625         Rate of pay??00         Wage =      437500.00
                                                 Wage =      437500.00
Hours worked?65535       Rate of pay?999.9       Wage =    65528446.50
                                                 Wage =    65528446.50
Hours worked?65535       Rate of pay?99.99       Wage =     6552844.65
                                                 Wage =     6552844.65
Hours worked?65535       Rate of pay?9.999       Wage =      655284.47
                                                 Wage =      655284.47
Hours worked?65535       Rate of pay?.9999       Wage =       65528.45
                                                 Wage =       65528.45
Hours worked?999.9       Rate of pay?.9999       Wage =         999.80
                                                 Wage =         999.80
Hours worked?99.99       Rate of pay?.9999       Wage =          99.98
                                                 Wage =          99.98
Hours worked?9.999       Rate of pay?.9999       Wage =           0.00
                                                 Wage =           0.00
Hours worked?6553        Rate of pay?999.9       Wage =     6552344.70
                                                 Wage =     6552344.70
Hours worked?655.3       Rate of pay?999.9       Wage =      655234.47
                                                 Wage =      655234.47
```

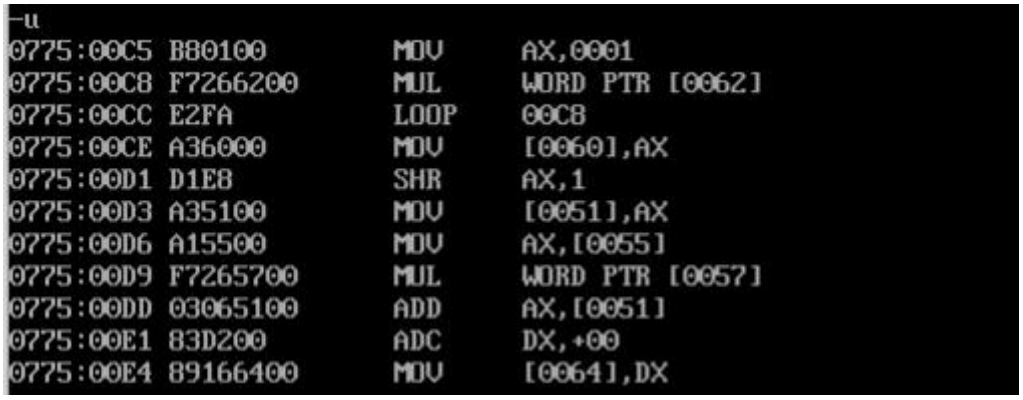
```
                                                 Wage =           1.00
Hours worked?65.53       Rate of pay?99.99       Wage =        6552.34
                                                 Wage =        6552.34
Hours worked?65536       Rate of pay?98          Wage =           0.00
                                                 Wage =           0.00
Hours worked?12          Rate of pay??0000       Wage =           0.00
                                                 Wage =           0.00
Hours worked?90000       Rate of pay?67          Wage =           0.00
                                                 Wage =           0.00
Hours worked?
```

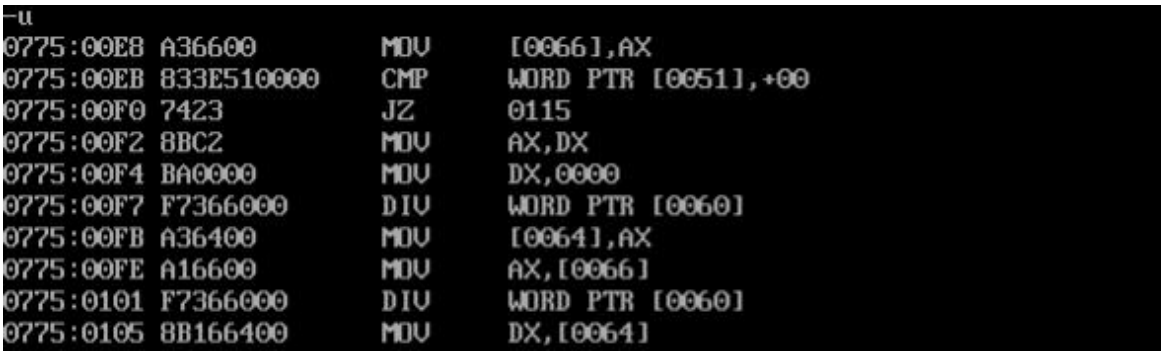
（1）debug 看反汇编语句

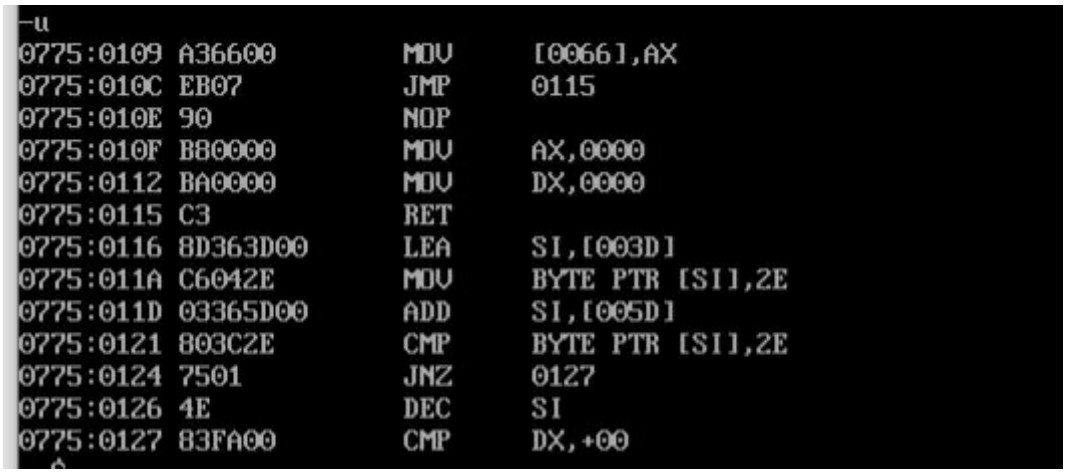
```
-u
0775:0000 1E            PUSH    DS
0775:0001 2BC0          SUB     AX,AX
0775:0003 50            PUSH    AX
0775:0004 B86E07        MOV     AX,076E
0775:0007 8ED8          MOV     DS,AX
0775:0009 8EC0          MOV     ES,AX
0775:000B B80006        MOV     AX,0600
0775:000E E8DF01        CALL    01F0
0775:0011 E8E601        CALL    01FA
0775:0014 E81F00        CALL    0036
0775:0017 803E010000    CMP     BYTE PTR [0001],00
0775:001C 7411          JZ      002F
0775:001E E84600        CALL    0067
```

```
-u
0775:0021 E85F00        CALL    0083
0775:0024 E87200        CALL    0099
0775:0027 E8EC00        CALL    0116
0775:002A E82A01        CALL    0157
0775:002D EBE5          JMP     0014
0775:002F B80006        MOV     AX,0600
0775:0032 E8BB01        CALL    01F0
0775:0035 CB            RETF
0775:0036 8D161000      LEA     DX,[0010]
0775:003A B409          MOV     AH,09
0775:003C CD21          INT     21
0775:003E 8D160000      LEA     DX,[0000]
-u
0775:0042 B40A          MOV     AH,0A
0775:0044 CD21          INT     21
0775:0046 803E010000    CMP     BYTE PTR [0001],00
0775:004B 7501          JNZ     004E
0775:004D C3            RET
0775:004E C606590019    MOV     BYTE PTR [0059],19
0775:0053 E8A401        CALL    01FA
0775:0056 8D161E00      LEA     DX,[001E]
0775:005A B409          MOV     AH,09
0775:005C CD21          INT     21
0775:005E 8D160800      LEA     DX,[0008]
-u
0775:0062 B40A          MOV     AH,0A
0775:0064 CD21          INT     21
0775:0066 C3            RET
0775:0067 C7065D000000  MOV     WORD PTR [005D],0000
0775:006D 8A0E0100      MOV     CL,[0001]
0775:0071 2AED          SUB     CH,CH
0775:0073 8D360100      LEA     SI,[0001]
0775:0077 03F1          ADD     SI,CX
0775:0079 E81A01        CALL    0196
0775:007C A15300        MOV     AX,[0053]
0775:007F A35500        MOV     [0055],AX
-u
0775:0082 C3            RET
0775:0083 8A0E0900      MOV     CL,[0009]
0775:0087 2AED          SUB     CH,CH
0775:0089 8D360900      LEA     SI,[0009]
0775:008D 03F1          ADD     SI,CX
0775:008F E80401        CALL    0196
0775:0092 A15300        MOV     AX,[0053]
0775:0095 A35700        MOV     [0057],AX
0775:0098 C3            RET
0775:0099 B90700        MOV     CX,0007
0775:009C 8D3E3200      LEA     DI,[0032]
0775:00A0 B83030        MOV     AX,3030
```


-u
0775:00A3 FC CLD
0775:00A4 F3 REPZ
0775:00A5 AB STOSW
0775:00A6 C70660000A00 MOV WORD PTR [0060],000A
0775:00AC C70651000000 MOV WORD PTR [0051],0000
0775:00B2 8B0E5D00 MOV CX,[005D]
0775:00B6 80F906 CMP CL,06
0775:00B9 7754 JA 010F
0775:00BB 49 DEC CX
0775:00BC 49 DEC CX
0775:00BD 7E17 JLE 00D6
0775:00BF C7065D000200 MOV WORD PTR [005D],0002


-u
0775:00C5 B80100 MOV AX,0001
0775:00C8 F7266200 MUL WORD PTR [0062]
0775:00CC E2FA LOOP 00C8
0775:00CE A36000 MOV [0060],AX
0775:00D1 D1E8 SHR AX,1
0775:00D3 A35100 MOV [0051],AX
0775:00D6 A15500 MOV AX,[0055]
0775:00D9 F7265700 MUL WORD PTR [0057]
0775:00DD 03065100 ADD AX,[0051]
0775:00E1 83D200 ADC DX,+00
0775:00E4 89166400 MOV [0064],DX


-u
0775:00E8 A36600 MOV [0066],AX
0775:00EB 833E510000 CMP WORD PTR [0051],+00
0775:00F0 7423 JZ 0115
0775:00F2 8BC2 MOV AX,DX
0775:00F4 BA0000 MOV DX,0000
0775:00F7 F7366000 DIV WORD PTR [0060]
0775:00FB A36400 MOV [0064],AX
0775:00FE A16600 MOV AX,[0066]
0775:0101 F7366000 DIV WORD PTR [0060]
0775:0105 8B166400 MOV DX,[0064]


-u
0775:0109 A36600 MOV [0066],AX
0775:010C EB07 JMP 0115
0775:010E 90 NOP
0775:010F B80000 MOV AX,0000
0775:0112 BA0000 MOV DX,0000
0775:0115 C3 RET
0775:0116 8D363D00 LEA SI,[003D]
0775:011A C6042E MOV BYTE PTR [SI],2E
0775:011D 03365D00 ADD SI,[005D]
0775:0121 803C2E CMP BYTE PTR [SI],2E
0775:0124 7501 JNZ 0127
0775:0126 4E DEC SI
0775:0127 83FA00 CMP DX,+00

```
-u
0775:012A 7505          JNZ     0131
0775:012C 3D0A00        CMP     AX,000A
0775:012F 7222          JB      0153
0775:0131 8BC2          MOV     AX,DX
0775:0133 BA0000        MOV     DX,0000
0775:0136 F7366200      DIV     WORD PTR [0062]
0775:013A A36400        MOV     [0064],AX
0775:013D A16600        MOV     AX,[0066]
0775:0140 F7366200      DIV     WORD PTR [0062]
0775:0144 A36600        MOV     [0066],AX
0775:0147 80CA30        OR      DL,30
```

```
-u
0775:014A 8814          MOV     [SI],DL
0775:014C 4E            DEC     SI
0775:014D 8B166400      MOV     DX,[0064]
0775:0151 EBCE          JMP     0121
0775:0153 0C30          OR      AL,30
0775:0155 8804          MOV     [SI],AL
0775:0157 C606590032    MOV     BYTE PTR [0059],32
0775:015C E89B00        CALL    01FA
0775:015F B90A00        MOV     CX,000A
0775:0162 8D363200      LEA     SI,[0032]
0775:0166 803C30        CMP     BYTE PTR [SI],30
0775:0169 7506          JNZ     0171
```

```
-u
0775:016B C60420        MOV     BYTE PTR [SI],20
0775:016E 46            INC     SI
0775:016F E2F5          LOOP    0166
0775:0171 8D162B00      LEA     DX,[002B]
0775:0175 B409          MOV     AH,09
0775:0177 CD21          INT     21
0775:0179 803E5F0014    CMP     BYTE PTR [005F],14
0775:017E 7307          JNB     0187
0775:0180 FE065F00      INC     BYTE PTR [005F]
0775:0184 EB0F          JMP     0195
0775:0186 90            NOP
0775:0187 B80106        MOV     AX,0601
0775:018A E86300        CALL    01F0
-u
0775:018D C606590000    MOV     BYTE PTR [0059],00
0775:0192 E86500        CALL    01FA
0775:0195 C3            RET
0775:0196 C7065B000100  MOV     WORD PTR [005B],0001
0775:019C C70653000000  MOV     WORD PTR [0053],0000
0775:01A2 C6065A0000    MOV     BYTE PTR [005A],00
0775:01A7 2BDB          SUB     BX,BX
0775:01A9 8A04          MOV     AL,[SI]
0775:01AB 3C2E          CMP     AL,2E
```

（2）g36,0036 开始是 b10inpt 输入模块，接收 hour 和 rate

```
AX=0200  BX=0000  CX=0000  DX=0000  SP=003A  BP=0000  SI=0000  DI=0000
DS=076E  ES=076E  SS=076A  CS=0775  IP=0036   NV UP EI PL ZR NA PE NC
0775:0036 8D161000      LEA     DX,[0010]                       DS:0010=6F48
-d0
076E:0000  06 00 00 00 00 00 00 00-06 00 00 00 00 00 00 00   ................
076E:0010  48 6F 75 72 73 20 77 6F-72 6B 65 64 3F 24 52 61   Hours worked?$Ra
076E:0020  74 65 20 6F 66 20 70 61-79 3F 24 57 61 67 65 20   te of pay?$Wage
076E:0030  3D 20 30 30 30 30 30 30-30 30 30 30 30 30 30 30   = 0000000000000
076E:0040  0D 0A 24 0D 0A 4F 76 65-72 66 6C 6F 77 21 0D 0A   ..$..Overflow!..
076E:0050  24 00 00 00 00 00 00 00-00 00 00 01 00 00 00 00   $...............
076E:0060  00 00 0A 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076E:0070  1E 2B C0 50 B8 6E 07 8E-D8 8E C0 B8 00 06 E8 DF   .+.P.n..........
-g67
Hours worked?
```

（3）0067 开始是 d10hour 模块，把 hour 的 asc 码转为二进制数

```
0775:0067 C7065D000000  MOV     WORD PTR [005D],0000            DS:005D=0000
-d0
-76E:0000  06 00 00 00 00 00 00 00-06 00 00 00 00 00 00 00   ................
-76E:0010  48 6F 75 72 73 20 77 6F-72 6B 65 64 3F 24 52 61   Hours worked?$Ra
-76E:0020  74 65 20 6F 66 20 70 61-79 3F 24 57 61 67 65 20   te of pay?$Wage
-76E:0030  3D 20 30 30 30 30 30 30-30 30 30 30 30 30 30 30   = 0000000000000
-76E:0040  0D 0A 24 0D 0A 4F 76 65-72 66 6C 6F 77 21 0D 0A   ..$..Overflow!..
-76E:0050  24 00 00 00 00 00 00 00-00 00 00 01 00 00 00 00   $...............
-76E:0060  00 00 0A 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-76E:0070  1E 2B C0 50 B8 6E 07 8E-D8 8E C0 B8 00 06 E8 DF   .+.P.n..........
-g67
Hours worked?65535
AX=0A00  BX=0000  CX=0000  DX=0008  SP=003A  BP=0000  SI=0000  DI=0000
DS=076E  ES=076E  SS=076A  CS=0775  IP=0067   NV UP EI PL NZ NA PE NC
0775:0067 C7065D000000  MOV     WORD PTR [005D],0000            DS:005D=0000
-d0
076E:0000  06 05 36 35 35 33 35 0D-06 05 39 39 2E 39 39 0D   ..65535...99.99.
076E:0010  48 6F 75 72 73 20 77 6F-72 6B 65 64 3F 24 52 61   Hours worked?$Ra
076E:0020  74 65 20 6F 66 20 70 61-79 3F 24 57 61 67 65 20   te of pay?$Wage
076E:0030  3D 20 30 30 30 30 30 30-30 30 30 30 30 30 30 30   = 0000000000000
076E:0040  0D 0A 24 0D 0A 4F 76 65-72 66 6C 6F 77 21 0D 0A   ..$..Overflow!..
076E:0050  24 00 00 00 00 00 00 00-00 19 00 01 00 00 00 00   $...............
076E:0060  00 00 0A 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076E:0070  1E 2B C0 50 B8 6E 07 8E-D8 8E C0 B8 00 06 E8 DF   .+.P.n..........
-
```

（3）0083，e10rate 模块，把 rate 转成二进制

```
-g83

AX=FFFF  BX=0005  CX=0000  DX=0001  SP=003A  BP=0000  SI=0001  DI=0000
DS=076E  ES=076E  SS=076A  CS=0775  IP=0083   NV UP EI PL ZR NA PE NC
0775:0083 8A0E0900      MOV     CL,[0009]                       DS:0009=05
-d0
076E:0000  06 05 36 35 35 33 35 0D-06 05 39 39 2E 39 39 0D   ..65535...99.99.
076E:0010  48 6F 75 72 73 20 77 6F-72 6B 65 64 3F 24 52 61   Hours worked?$Ra
076E:0020  74 65 20 6F 66 20 70 61-79 3F 24 57 61 67 65 20   te of pay?$Wage
076E:0030  3D 20 30 30 30 30 30 30-30 30 30 30 30 30 30 30   = 000000000000000
076E:0040  0D 0A 24 0D 0A 4F 76 65-72 66 6C 6F 77 21 0D 0A   ..$..Overflow!..
076E:0050  24 00 00 FF FF FF FF 00-00 19 00 A0 86 00 00 00   $...............
076E:0060  00 00 0A 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076E:0070  1E 2B C0 50 B8 6E 07 8E-D8 8E C0 B8 00 06 E8 DF   .+.P.n..........
-
```

（4）0099，f10mult 模块，计算工资

```
-g99

AX=270F  BX=0002  CX=0000  DX=0000  SP=003A  BP=0000  SI=0009  DI=0000
DS=076E  ES=076E  SS=076A  CS=0775  IP=0099   NV UP EI PL NZ NA PO NC
0775:0099 B90700        MOV     CX,0007
-g116

AX=D8F1  BX=0002  CX=0000  DX=270E  SP=003A  BP=0000  SI=0009  DI=0040
DS=076E  ES=076E  SS=076A  CS=0775  IP=0116   NV UP EI PL ZR NA PE NC
0775:0116 8D363D00      LEA     SI,[003D]                       DS:003D=3030
-d0
076E:0000  06 05 36 35 35 33 35 0D-06 05 39 39 2E 39 39 0D   ..65535...99.99.
076E:0010  48 6F 75 72 73 20 77 6F-72 6B 65 64 3F 24 52 61   Hours worked?$Ra
076E:0020  74 65 20 6F 66 20 70 61-79 3F 24 57 61 67 65 20   te of pay?$Wage
076E:0030  3D 20 30 30 30 30 30 30-30 30 30 30 30 30 30 30   = 000000000000000
076E:0040  0D 0A 24 0D 0A 4F 76 65-72 66 6C 6F 77 21 0D 0A   ..$..Overflow!..
076E:0050  24 00 00 0F 27 FF FF 0F-27 19 01 10 27 02 00 00   $...'...'...'...
076E:0060  0A 00 0A 00 0E 27 F1 D8-00 00 00 00 00 00 00 00   .....'..........
076E:0070  1E 2B C0 50 B8 6E 07 8E-D8 8E C0 B8 00 06 E8 DF   .+.P.n..........
```

（5）0157 开始是 k10disp 模块，显示 wage

```
-g157

AX=0036  BX=0002  CX=0000  DX=0000  SP=003A  BP=0000  SI=0036  DI=0040
DS=076E  ES=076E  SS=076A  CS=0775  IP=0157   NV UP EI PL NZ NA PE NC
0775:0157 C606590032    MOV     BYTE PTR [0059],32              DS:0059=19
-d0
076E:0000  06 05 36 35 35 33 35 0D-06 05 39 39 2E 39 39 0D   ..65535...99.99.
076E:0010  48 6F 75 72 73 20 77 6F-72 6B 65 64 3F 24 52 61   Hours worked?$Ra
076E:0020  74 65 20 6F 66 20 70 61-79 3F 24 57 61 67 65 20   te of pay?$Wage
076E:0030  3D 20 30 30 30 30 36 35-35 32 38 34 34 2E 36 35   = 00006552844.65
076E:0040  0D 0A 24 0D 0A 4F 76 65-72 66 6C 6F 77 21 0D 0A   ..$..Overflow!..
076E:0050  24 00 00 0F 27 FF FF 0F-27 19 01 10 27 02 00 00   $...'...'...'...
076E:0060  0A 00 0A 00 00 00 06 00-00 00 00 00 00 00 00 00   ................
076E:0070  1E 2B C0 50 B8 6E 07 8E-D8 8E C0 B8 00 06 E8 DF   .+.P.n..........
```

（6）0195，k10disp 模块结束，将结果显示在终端

问题及收获：

1. 学会使用 DEBUG 或 TD 分别调试各个子程序，对其分模块进行单元测试，然后再将其组装成整个程序做整体测试。

2. 了解使用到的 DOS 系统调用，掌握示例程序中设计的 2 进制转 10 进制算法以及关于小数的乘除法处理。