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：202120130276	姓名：王云强	班级：21.2 班
实验题目：实验四：例 2.6，例 2.7		
实验学时：2	实验日期：2023.11.10	
实验目的： 1.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。 2.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。 3.掌握过程的模块化设计，及其嵌套与测试方法。 4.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. Grade.ASM（示例 2.6 源程序） 2. Wage.ASM（示例 2.7 源程序）		
编译及运行结果： 示例 2.6 编译结果：		

```
C:\>masm Grade
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [Grade.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51564 + 448596 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link Grade

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [GRADE.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment
```

示例 2.6 运行结果：

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

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

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

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

C:\>Grade
GRADE? 87,99,100,76.
Input Error!
```

示例 2.7 编译结果：

```
C:\>masm Wage
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [Wage.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51680 + 448480 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link Wage

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [WAGE.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
```

示例 2.7 运行结果：

```
Hours worked?65535      Rate of pay?65535       Wage == 4294836225.00
Hours worked?0          Rate of pay?0           Wage ==          0.00
Hours worked?1          Rate of pay?1           Wage ==          1.00
Hours worked?625        Rate of pay??00         Wage ==     437500.00
Hours worked?65535      Rate of pay?999.9       Wage ==   65528446.50
Hours worked?12         Rate of pay??000        Wage ==        840.00
Hours worked?12         Rate of pay??0000       Wage ==        840.00
Hours worked?90000      Rate of pay?67          Wage ==          0.00
Hours worked?2.0        Rate of pay?2.0         Wage ==          4.00
Hours worked?2.1        Rate of pay?2.0         Wage ==          4.20
Hours worked?12.0       Rate of pay?2.0         Wage ==         25.56
Hours worked?12.0       Rate of pay?2.00        Wage ==         24.00
Hours worked? ▲_
```

问题及收获：

1. 示例 2.6 的代码解读：

数据段：GRADE 储存成绩数组，RANK 存放得到的结果的排名数组，COUNT 作为一个全局变量，三个 MESS 都是输入输出的提示。

DECIBIN 子程序：首先清空 BX 寄存器的值，方便存放中间使用和最终的结果。首先输入一个字符到 AL 中，先比较判断一下，输入的字符是不是 0-9 的字符，如果不是就退出。如果是，就先将 AX 和 BX 交换（此时 AX 为数字的第一个字符，BX 为空），让 AX 的值乘 10，再交换回来，将结果放入 BX（由于数字最大为 9，乘 10 不超过 10，所以虽然是 16 位之间的乘法，但是只会在 AX 中有结果，不会有高位产生），只会将 AX 结果再加到 BX 上（此时 AX 为 0），之后继续读下一个字符（数字）。

INPUT 子程序：先提示输入，之后从内存 GRADE 首地址开始，每读到一位数字判断其后是否是逗号（,），如果是，则将数字储存到 GRADE 对应数组指针所指向的地址，将指针移到下一位，并准备读入下一个字符；如果不是逗号，判断其是否是回车，如果回车，则调用输出回车的子程序（CRLF），返回主程序；如果都不是，则直接输出错误信息，再调用 CRLF。BINIDEC 子程序与 DEC_DIV 子程序共同进行作用：主要作用就是将 BX 的数字除以 100、10、1 来得到各位的值，并将商一位一位的打印在屏幕上。RANKP 子程序和之前的实验的 RANK 差不多：即分别把每一个成绩去与所有的成绩进行逐个比较,将其排名放在对应的 RANK 数组的位置中。

OUTPUT 子程序：主要作用就是从 RANK 数组中取出各个成绩的对应排名，

通过调用 BINIDEC 子程序，来将排名输出到屏幕上。

2.示例 2.7 的代码解读：

该程序比较长，此处只做简要分析（其他的部分都注释在代码中）。

主程序：先调用输入函数，将工作小时数和工资转化率先输入进来，并保存在内存中。之后调用 D10HOUR 和 E10RATE 函数将输入进来的工作小时数和工资转化率从 ASCII 码转换为真实数字。再调用 F10MULT 函数，计算最终结果（工资总额）。再调用 K10DISP 函数展示结果。最后再从头开始，进行下一个计算。

B10INPT 子程序：输入工作小时数和工资转化率，并分别储存在 HRSPAR 和 RATEPAR 中。

D10HOUR 和 E10RATE 子程序：两个程序基本一致，都是将输入的数字传入 M10ASBI 子程序中，该子程序的作用是将数字从末尾最后一位开始读入，最低位转换成个位数，然后记录一下小数的位数（NODEC）（如 12.25 转换为 1225，然后记录小数位数为 2）。

F10MULT 子程序：将工作小时数和工资转化率乘起来存入到内存中，然后算出移位因子（SHIFT=10^(NODEC-2)）和舍入（SHIFT/2）。

G10WAGE 子程序：主要是调整小数点位置，然后将真实数字转换回 ASCII 码，便于下一步输出。

3. 在示例 2.7 中乘法计算结果可能溢出 16 位（一个通用寄存器的最大长度），这个时候怎么处理？

答：正常调用 16 位数的乘法，然后分别储存高位和低位到内存中，在之

后的除法运算中，也依旧是分别除高位和低位，用两个 16 位的空间来储存答案。

4.在示例 2.7 中可能出现小数的问题，这个地方该怎么解决？

程序是这样解决的：首先记录两个数的小数位数之后，进行判断。如果超过六位数，则视为溢出（因为没法通过该算法计算）。如果小数位数在 3~6 之间，则引入移位因子（SHIFT = 10^(NODEC-2)）和舍入（SHIFT/2），然后将数字分别扩大对应的小数位数（相当于全部视为整数相乘），之后再将该结果加上舍入，得到的和除以移位因子视为最终结果（满足四舍五入，尽量地做到与原数据相近）。如果小数位数在 0~2 之间，则将数字分别扩大对应的小数位数（相当于全部视为整数相乘），之后在输出结果时，在对应位置输出小数点即可。