# 《汇编语言》实验报告04

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 班级 | 2022秋 | 实验日期 | 2022.10.28 | 实验成绩 |  |
| 姓名 | 黄勖 | 学号 | 22920212204392 | | |
| 实验名称 | 汇编语言第四次实验 | | | | |
| 实  验  目  的  、  要  求 | 1. 熟练使用Debug（更加推荐采用程序形式），理解数据在内存中的存放，并理解并练习各种寻址方式。 2. 学习汇编语言的基本指令，学会阅读汇编代码并动手尝试编写一些算法 3. 了解传送指令、比较指令、条件转移指令、字符显示的具体编写方式 4. 学习简单程序结构 | | | | |
| 实  验  内  容  、  步  骤  及  结  果 | 1. **在长度为8的字节数组（无符号数）中，查找大于42H的无符号数的个数，存放在字节单元up中；等于42H的无符号数的个数，存放在字节单元equa中；小于42H的无符号数的个数，存放在字节单元down中。程序显示up equa down的值。**   **八个数：31H,21H,42H,52H,87H,23H,98H,01H**   * 1. **在数据段中存入八个数，为up,equa,down分配内存并初始化为0**      * 1. **初始化，把数组数据和42H进行比较，如果相等就转移到EQUAL\_NUMBER，如果小于42H就转移到DOWN\_NUMBER，如果大于42H就转移到UP\_NUMBER，并利用count变量判断是否已经计数完毕（如果count为0，说明比较完转移到RESULT，否则转移到COMPARE）**     **对每次的比较结果将每次的计数结果变量对应加一**       * 1. **输出结果，显示字符串upString，调用功能号02H，显示字符**      * 1. **退出程序**      * 1. **连接编译运行查看结果，得到up为3，equa为1，down为4，结果正确！** | | | | |
| 总  结 | 这一次实验我对汇编语言指令有了更深的理解，并且这一次实验的实践操作颇丰，在练习编码的过程中我对每一个指令的用途和用法有了更深的认识，通过一步步地解决问题，我的实践能力提高了，这让我受益匪浅；具体遇到问题的解决方案我在后文做了更详细的总结，在此就不多赘述；在未来我还要探索汇编语言的更多应用方面，寻找更多问题，并在发现问题的过程中继续提高我对汇编语言的掌握能力，这是一次颇有意义的实验！  **问题总结：**  **一、汇编语言中,为什么SI和DI不能同时使用汇编**  这两个寄存器的意思，SI源变址寄存器，DI目地变址寄存器，既然是变址寄存器，那么他们肯定是在某个地址的基础上进行偏移变化，由此我们就得出了需要基址寄存器。  要是把这两个寄存器同时使用，那地址变化的基址都没有，该怎么变化呢？在谁的基础上变化（也就是地址偏移）？  对于汇编中的规定，其实有时并不需要书上详细的介绍，我们都应该可以从中推导出这些规则，书上的那些介绍只是用来验证我们的推测的。或是对我们所掌握的知识的进行检测，用来说明我们所掌握的是对的！  **二、寄存器的总结**  **1:数据寄存器,一般称之为通用寄存器组**  8086 有8个8位数据寄存器，  这些8位寄存器可分别组成16位寄存器：  AH&AL＝AX：累加寄存器，常用于运算；  BH&BL＝BX：基址寄存器，常用于地址索引；  CH&CL＝CX：计数寄存器，常用于计数；  DH&DL＝DX：数据寄存器，常用于数据传递。  **2:地址寄存器/段地址寄存器**  为了运用所有的内存空间，8086设定了四个段寄存器，专门用来保存段地址：  CS（Code Segment）：代码段寄存器；  DS（Data Segment）：数据段寄存器；  SS（Stack Segment）：堆栈段寄存器；  ES（Extra Segment）：附加段寄存器。  **3：特殊功能的寄存器**  IP（Instruction Pointer）：指令指针寄存器，与CS配合使用，可跟踪程序的执行过程；  SP（Stack Pointer）：堆栈指针，与SS配合使用，可指向目前的堆栈位置。  BP（Base Pointer）：基址指针寄存器，可用作SS的一个相对基址位置；  SI（Source Index）：源变址寄存器可用来存放相对于DS段之源变址指针；  DI（Destination Index）：目的变址寄存器，可用来存放相对于 ES 段之目的变址指针。 | | | | |