**汇编语言程序设计课程实验报告**

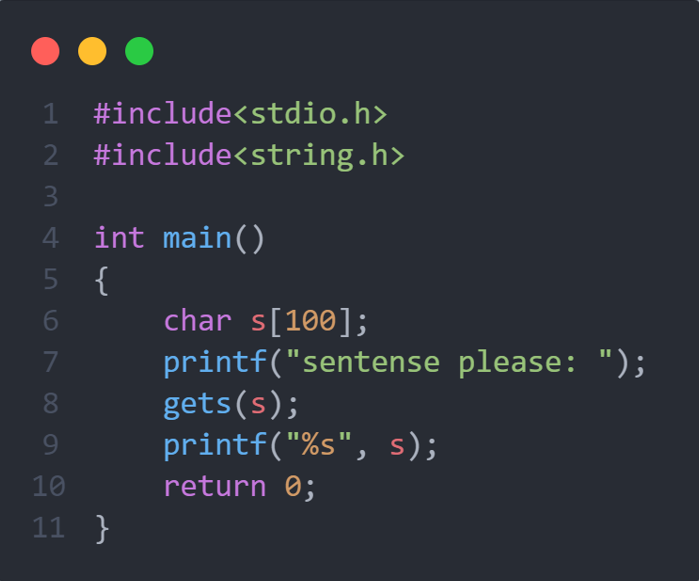
**实验1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 |  | 院系 | 计算机工程与科学学院 | 学号 |  |
| 实验目的： | | | | | |
| 1. 熟悉汇编语言编程环境； 2. 了解汇编语言程序代码框架； 3. 体会高级语言与汇编语言的优缺点； 4. 掌握汇编语言的调试方法。 | | | | | |
| 实验任务： | | | | | |
| 1. C语言实现例子程序； 2. 汇编语言实现例子程序； 3. 实践MASM环境的程序调试方法； 4. 对比上述两具实例的可执行代码长度、两种编程语言各自的优缺点。 | | | | | |

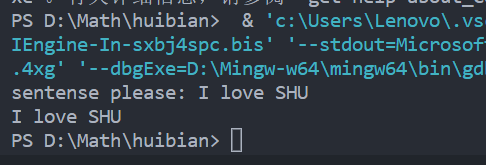
实验步骤：

1. C语言实验例子子程序

运行汇编子程序可知，这个程序的功能上是输入并显示一个字符串的功能，C语言实现如下：



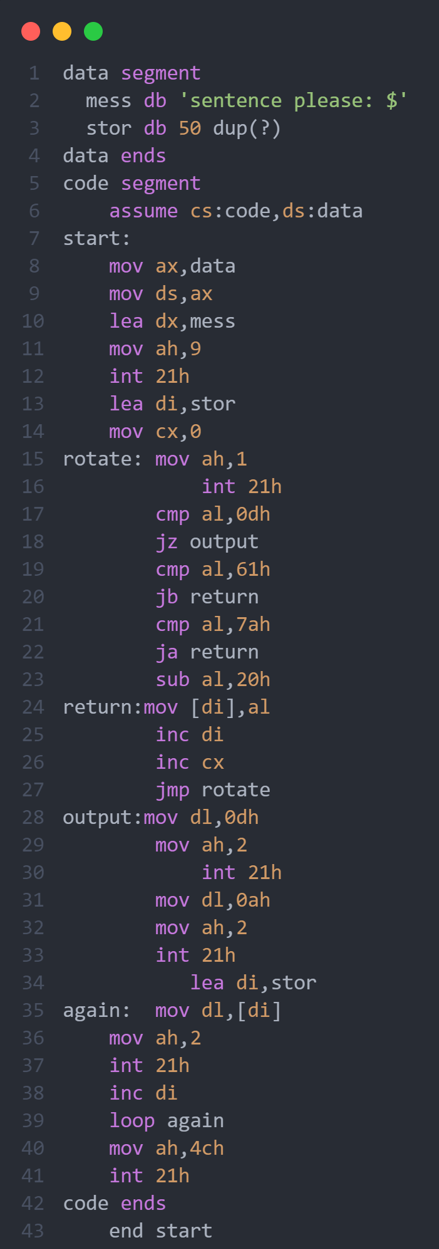
运行效果如下：



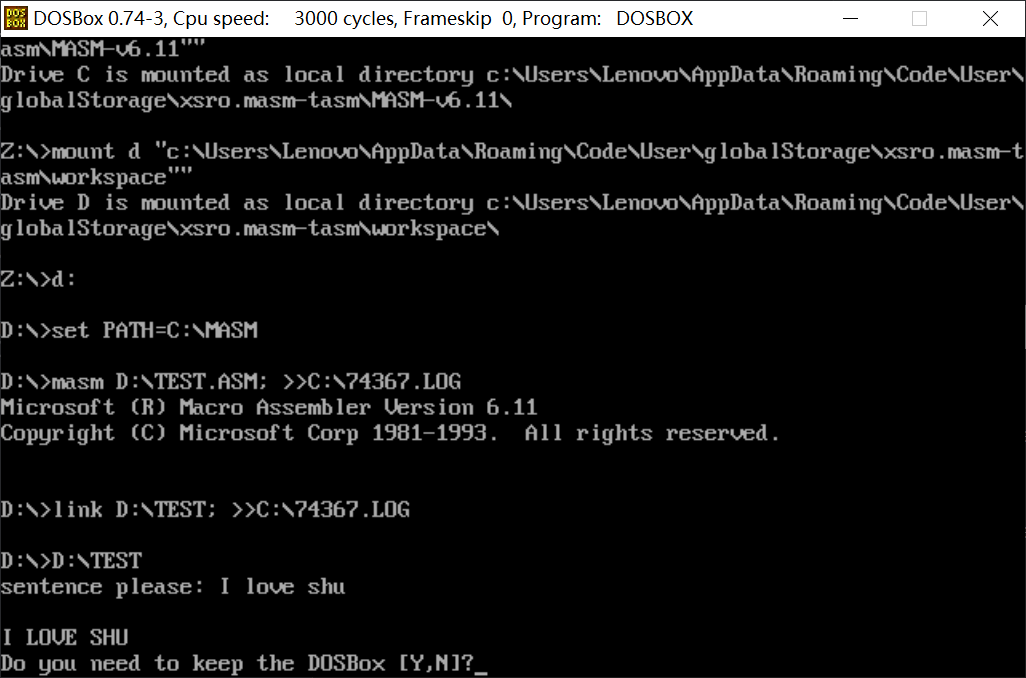
结果显示很好的使用C语言模拟汇编例子程序的效果，完成实验任务1。

1. 汇编语言实现例子程序；

在vs code中安装masm/tasm拓展，输入例子程序汇编代码，效果如下：



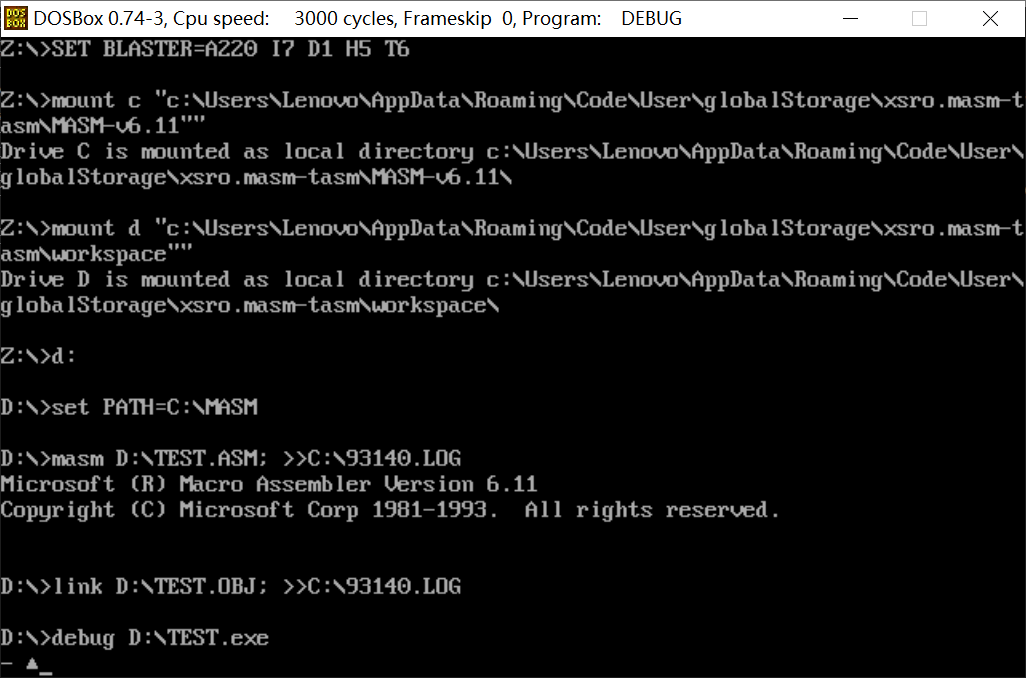
运行效果如下：



效果与使用C语言模拟的一致，完成实验任务2。

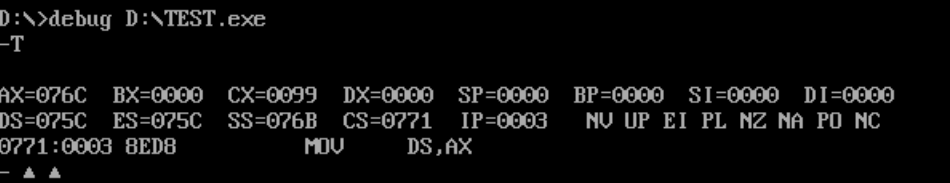
1. 实践MASM环境的程序调试方法。

在vs code中点击鼠标右键，点击调试当前运行程序，可以得到以下界面。



随后使用8086DEBUG中的部分命令进行操作。

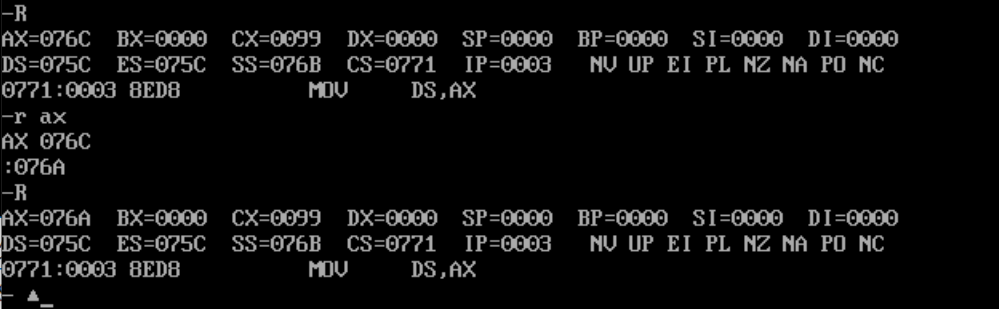
1. 追踪指令T，执行下一条机器指令，效果如下：



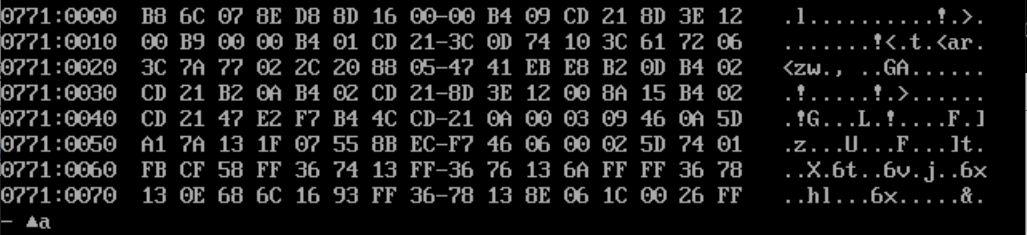
可以看到显示出执行的第一条指令MOV DS,AX，由于指令很多，所以需要多次使用T指令才能执行完程序。

1. R指令

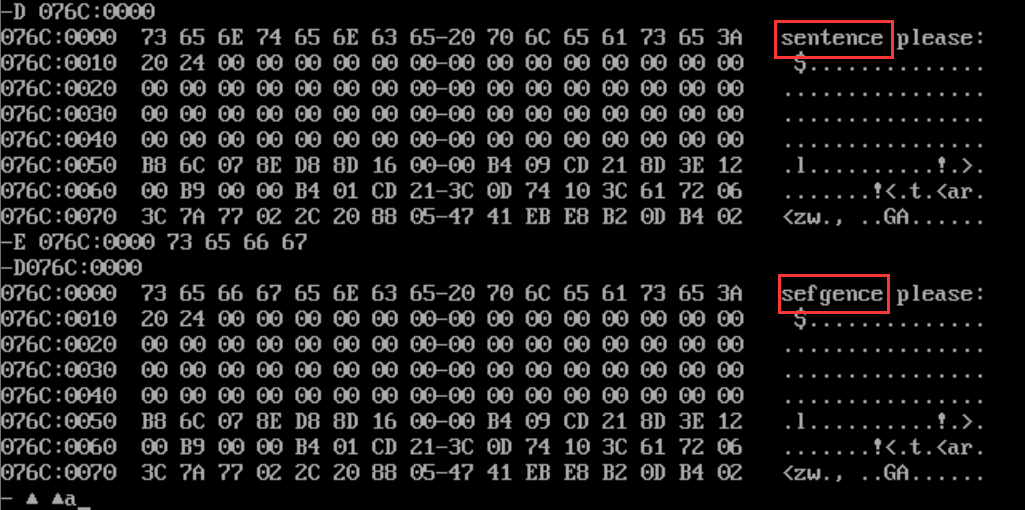
R指令，查看CPU寄存器中的内容，效果与T指令相同，在R指令后面加上寄存器的名称，可以修改对应寄存器的内容，效果如下：



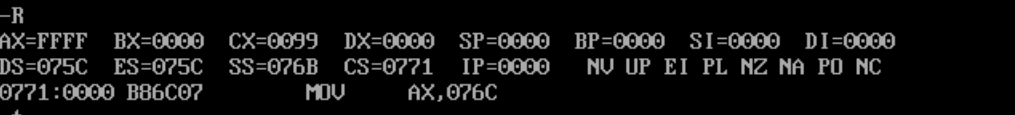
1. D指令，查看内存信息，效果如下：



1. E指令，修改内存单元内容，用法：-E 地址 内容表 将指定地址开始的内存单元中的内容替换成内容表中的内容。效果如下：



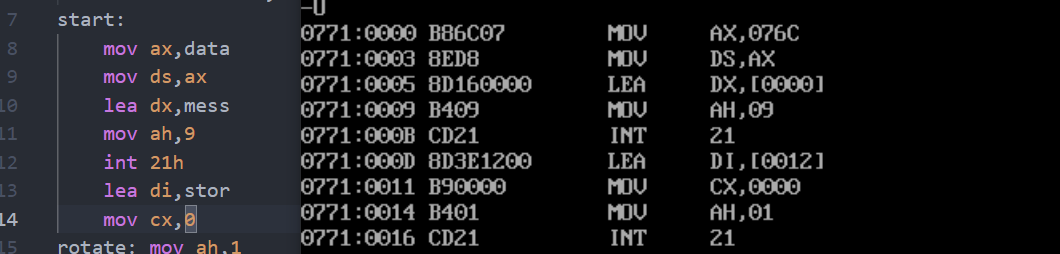
1. R指令，检查和修改寄存器内容，首先可以检查CPU内部所有寄存器内容和标志位信息。效果如下：



查阅资料可得各个标志位含义如下表：



1. U指令，由exe文件反汇编得到机械指令。格式：-U [地址]该命令从指定地址开始，反汇编 （指把机器代码翻译成汇编语句）32个字节，若地址省略，则从上一个U命令的最后一条指令的下一个单元开始显示32个字节。效果如下：



1. 退出命令Q

退出DEBUG模式，结束程序。

由以上内容，我初步实践了MASM环境的程序调试方法，完成实验任务3。

1. 通过对比汇编语言和C语言，可以观察到C语言的语句比汇编语言简短非常多，以下是两种语言的优缺点。
2. C语言

|  |  |
| --- | --- |
| 优点 | 缺点 |
| 易读和易学。 C语言的语法结构更接近自然语言，代码更易于阅读和维护 | 对于某些场景，特定的优化需要用汇编来实现 |
| 可以很方便的移植到不同平台 | 底层操作和细节控制汇编语言更为强大 |
| 有许多第三方库和工具支持 | 缺乏内建的安全机制，容易产生野指针 |

1. 汇编语言

|  |  |
| --- | --- |
| 优点 | 缺点 |
| 精确控制：汇编语言允许开发者直接控制底层硬件和处理器 | 汇编语言的语法和语义相对较为复杂，不容易理解和学习。 |
| 可以实现高度优化的代码，针对特定的硬件平台和应用需求进行性能调整 | 可读性和可维护性差，维护汇编代码更加困难和耗时 |
| 在逆向工程和安全领域，汇编语言是非常重要的工具，可以帮助分析和理解底层系统 | 可移植性较差，需要针对不同的硬件平台进行修改和适配。 |

1. 实验体会

汇编语言不同于高级程序设计语言，在计算机组成原理课中我初步接触到了汇编语言，当真正使用到它来编程，写一个简单的小程序时仍是觉得具有较大的挑战性，很多在高级语言上一条代码就能解决的问题再汇编语言却需要“绕很多弯路”，接触时间不长，一时间没有适应汇编语言的逻辑操作，我相信经过通过后面汇编语言的学习，我会进一步增加我对计算机的理解。