

汇编语言程序设计课程设计 第一次实验

DEBUG基本命令与数据传输指令

本次实验目的

- 熟练掌握DEBUG的基本调试命令，能够使用DEBUG编写、调试汇编语言程序片段。
- 在理解数据传输指令的基础上按照实验内容中指定的程序片段对程序进行调试和记录；

实验报告的主要内容

- 1) 在实验报告中说明所学DEBUG基本命令的使用方法。
- 2) 在实验报告中记载每条指令执行后，寄存器、标志位的变化情况，解释指令的执行原理。

本次实验指定的源代码

- MOV AL, 01H
- MOV SI, 0002H
- LEA SI, [SI]
- MOV BYTE PTR [SI], 80H
- LAHF
- XCHG AL, AH
- SAHF
- XCHG AH, [SI]
- SAHF

DEBUG调试工具

- 调试8086/8088指令集程序的基本工具，只要安装了WINDOWS系统，系统在“命令提示符”方式下都能运行这个程序。
- 目前WINDOWS系统下运行的DEBUG是建立在“8086虚拟机”的环境下，并不意味着真实CPU芯片就是8086。

DEBUG中数据表示形式与源程序的区别

- DEBUG中能够输入的数据（包括逻辑地址）与DEBUG在屏幕上回显的数据均为十六进制，其它进位记数制在DEBUG中均不能使用。
- 通常的文本源程序中（不是在DEBUG中输入的程序片段），可以混合使用各种进位记数制。

DEBUG的运行与退出

- 在命令行直接输入“DEBUG”并回车，则运行DEBUG程序，其提示符为一个短横线。
- 如果要退出DEBUG，则使用q命令：
- -q
- 退出DEBUG后，窗口退回到DOS命令行提示符。

使用r命令查看寄存器内容

- 1.寄存器查看/编辑命令 r
- 功能：查看或手工修改寄存器内容以及标志位状态。
- 使用：
- -r
- -r寄存器名称（不需要空格）
- （演示）

DEBUG基本调试命令

- 2. 汇编命令 **a**
- 功能：从键盘入汇编指令，由DEBUG转换为机器指令并存放在内存单元中。
- 使用方法：
 - **-a**
 - **-a XXXX:XXXX**
 - （演示）

使用**a**命令修改程序片段

- 如果程序片段中有某条指令在输入时出现语法错误，则DEBUG会自动提示出错的位置，且不会将错误指令生成为机器指令。
- 如果程序片段中某条指令在输入时出现逻辑错误（语法上正确），只能退出**a**命令状态，重新使用**a**命令并给出出错指令的逻辑地址进行修改。

使用a命令修改程序片段

- 如果新输入指令的机器指令字节数比以前的指令短，则补以NOP指令（单字节的空操作指令）。
- 如果新输入指令的机器指令字节数比以前的指令长，则从该指令起的后续所有指令序列都必须重新录入。
- 因此，a命令只适合输入较短的程序片段。
- （演示）

使用U命令查看代码段中指令

- 3. 反汇编命令U
- 功能：将指定逻辑地址起始的机器指令序列反汇编为汇编指令序列，并在屏幕上回显。
- 使用：
- -U
- -U XXXX: XXXX
- （演示）

U命令注意事项

- 给出代码起始地址时一定要给准确，否则可能造成整段代码牛头不对马嘴。

使用T命令单步调试程序片段

- 4. 单步调试命令T
- 功能：单步执行下一条指令，执行完毕后自动暂停，在屏幕上回显寄存器内容（同R命令），然后等待用户输入下一个DEBUG命令。
- 使用：
- -t
- （演示）