湖南科技大学计算机科学与工程学院

汇编语言 实验报告

**专业班级：** 22级计科六班

**姓 名：** 周俊哲

**学 号：** 2205010711

**指导教师：** 杨策

**时 间**： 2024.9.30—2024.10.9

**地 点**： 逸夫楼537

|  |
| --- |
| 指导教师评语：    **成绩： 等级：**  **签名：**  **年 月 日** |

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | 实验1 用Debug命令查看寄存器和内存中 的内容 | | |
| 实验性质  （必修、选修） |  | 实验类型（验证、设计、创新、综合） |  |
| 实验课时 |  | 实验日期 | 1、  2、  3、  4、  。。。 |
| 实验仪器设备以及实验软硬件要求 |  | | |
| 实验目的 | 要求掌握使用Debug命令查看寄存器和内存的方法。 | | |
| 实验内容（实验原理、运用的理论知识、算法、程序、步骤和方法）  （1）掌握安装使用汇编语言编译运行环境。  （2）使用Debug命令查看寄存器的值。  （3）使用Debug命令查看内存的内容。  Debug是DOS、Windows都提供的实模式（8086方式）程序的调试  工具。使用它，可以查看CPU各种寄存器中的内容、内存的情况和机  器码级跟踪程序的运行。  在本次实验中，需要用到的Debug命令及其相关的功能有：  （1）用Debug的R命令查看、改变CPU寄存器的内容；  （2）用Debug的D命令查看内存中的内容；  （3）用Debug的E命令改写内存中的内容。  【1】用R命令查看、改变CPU寄存器的内容。    用R命令修改寄存器中的值:    用R命令修改CS和IP中的内容:    用D查看内存10000H处的内容:    用E命令修改从1000:0开始的10个单元的内容: | | | |
| 实验结果与分析  学会了debug中R,D,E的命令的使用,了解了一些debug的基本使用方法 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | 实验2 上机过程及程序调试 | | |
| 实验性质  （必修、选修） |  | 实验类型（验证、设计、创新、综合） |  |
| 实验课时 |  | 实验日期 | 1、  2、  3、  4、  。。。 |
| 实验仪器设备以及实验软硬件要求 |  | | |
| 实验目的 | 1.通过第2章两个简单实例认识汇编语言程序，初步了解程序格  式；段定义；标号；DOS系统功能；操作数；指令；说明语句等作  用。  2.掌握汇编语言程序从设计到形成可执行程序文件的方法和步骤，  即编辑、汇编、连接。了解汇编语言的系统工作文件和工作环境。理  解汇编程序和连接程序的作用。理解列表文件内容。  3.初步了解程序的运行和调试方法。学会使用Debug主要命令和常  用的DOS命令。 | | |
| 实验内容（实验原理、运用的理论知识、算法、程序、步骤和方法）   1. 编辑第3章两个简单实例，通过汇编、连接。 2. 编写HELLO.ASM和test.asm, 在dosbox中用MASM和LINK命令对程序进行汇编和链接     【2】用Debug对HELLO程序进行调试。问题：  （1）该程序在内存中的起始地址是多少？结束地址是多少？共占  几个字节？    起始地址在0770:0005,结束地址在0700:000F;共占16字节.  （2）程序中定义的串‘HELLO,WORLD !$’在内存中的起始地址  是多少？请先用D命令显示该字符串，再用G命令运行对应程序段并显  示该串。  1.起始地址在076C:0000;  2.使用D命令显示:    3.使用G命令显示    （3）用P命令单步执行程序，查看相关寄存器的内存。  执行MOV AH,09后，AX由076C变为096C；  【3】  （1）用A命令从地址0770:000CH处输入汇编语句：  MOV AH,09  INT 21H  MOV AH,4C  INT 21H    使用G命令运行，可以看到程序输出了两遍Hello World，符合预期。  （2）如果用带有断点的G命令显示字符串‘HELLO,WORLD !'，  写出命令。    可以看到执行-g E后程序在调用INT 21结束程序前停下；再运行-g后程序结束。  （3）用T命令跟踪程序HELLO.EXE，写出每一步执行后，相关寄  存器的内容。    寄存器的相关内容如上图所示。  （4）用T命令跟踪程序，在跟踪执行INT 21时出现什么问题，如  何解决？进一步说明P命令和T命令的区别。   1. 使用T命令跟踪时，遇到INT 21H系统调用时会进入中断处理程序的内部，导致调试进入复杂的中断例程。 2. **解决方法**：在遇到中断指令时，使用P命令（过程跟踪）可以跳过中断处理代码，这样可以避免跟踪进入中断的细节。   **区别**：   * + T命令：逐条跟踪每一条指令，包括中断和过程调用。   + P命令：在遇到中断或过程调用时跳过，直接执行完中断或过程后返回。 | | | |
| 实验结果与分析  1.通过第2章两个简单实例认识汇编语言程序，初步了解程序格  式；段定义；标号；DOS系统功能；操作数；指令；说明语句等作  用。  2.掌握汇编语言程序从设计到形成可执行程序文件的方法和步骤，  即编辑、汇编、连接。了解汇编语言的系统工作文件和工作环境。理  解汇编程序和连接程序的作用。理解列表文件内容。  3.初步了解程序的运行和调试方法。学会使用Debug主要命令和常  用的DOS命令。 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | 实验3 不同寻址方式的灵活运用 | | |
| 实验性质  （必修、选修） |  | 实验类型（验证、设计、创新、综合） |  |
| 实验课时 |  | 实验日期 | 1、  2、  3、  4、  。。。 |
| 实验仪器设备以及实验软硬件要求 |  | | |
| 实验目的 | 熟悉和掌握7种不同的操作数寻址方式的使用方法。 | | |
| 实验内容（实验原理、运用的理论知识、算法、程序、步骤和方法）  对指定内存单元的数据进行不同寻址方式的访问。   1. 立即寻址方式的使用      1. 寄存器寻址方式的使用      1. 直接寻址方式的使用      1. 寄存器间接寻址方式的使用      1. 寄存器相对寻址方式的使用      1. 基址变址寻址方式的使用 2. 相对基址变址寻址方式   思考题：参考以上寻址方式，完成下列实验内容。  1.两个操作数相减运算，结果放在数据段的偏移地址为0016H单元。  （1）AX,BX寄存器分别赋值为0038H和0010H。  （2）AX和BX的内容相减（SUB指令），结果在AX中（该指令为SUB AX,BX）。  （3）用直接寻址方式将相减的结果保存到0016H单元。  MOV AX, 0038H ; MOV BX, 0010H ;  SUB AX, BX  MOV [0016H], AX        2.两个操作数相加运算，结果放在附加段的0020H单元。  （1）AX的值为0034H。  （2）AX和65相加，结果在AX中（该指令为AND AX,65）。  （3）用寄存器间接寻址方式（段超越）保存运算结果。  MOV AX, 0034H  ADD AX, 0065H  MOV [ES:0020H], AX      3.将AX寄存器中的1234H写入数据段的0002H单元，读出0003H的  12H传送给BL寄存器（寻址方式自定）。  MOV AX, 1234H  MOV [0002H], AX  MOV BL, [0003H]= | | | |
| 实验结果与分析  熟悉和掌握7种不同的操作数寻址方式的使用方法。 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | 实验4 算术及位串处理程序 | | |
| 实验性质  （必修、选修） |  | 实验类型（验证、设计、创新、综合） |  |
| 实验课时 |  | 实验日期 | 1、  2、  3、  4、  。。。 |
| 实验仪器设备以及实验软硬件要求 |  | | |
| 实验目的 | 掌握多数位的算术运算、移位操作、字符串操作等程序的设计，  学习使用分支与循环等基本编程方法，熟练使用Debug。 | | |
| 【1】在数据段预先存放16个十六进制数的ASCII码，首地址为  ASC。从键盘输入一位十六进制数到BX，用ASC[BX]寻址方式找到对应  数位的ASCII码，并取出显示。  数据段：16 个十六进制数的 ASCII 码  DATA\_SEG SEGMENT  ASC DB '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'  DATA\_SEG ENDS  CODE\_SEG SEGMENT  ASSUME DS:DATA\_SEG  START:  MOV AX, DATA\_SEG  MOV DS, AX ; 初始化数据段    ; 从键盘输入一个十六进制字符  MOV AH, 01H ; 读入一个字符  INT 21H ; DOS 中断调用  SUB AL, 30H ; 将输入字符转为数值 ('0'-'9' 转换为 0-9)  CMP AL, 09H  JBE VALID\_INPUT  SUB AL, 07H ; 如果输入为 'A'-'F'，转换为 10-15  VALID\_INPUT:  MOV BX, AX ; BX 保存输入的数值    ; 寻址 ASC[BX]，显示相应的 ASCII 码  MOV DL, [ASC+BX]  MOV AH, 02H ; DOS 中断服务，显示字符  INT 21H    MOV AH, 4CH  INT 21H  CODE\_SEG ENDS  END START  【2】用16位指令编制程序，处理32位的加减乘除算术四则运算  题。  要求：  （1）所有变量均定义成字类型，其中应有负数。部分变量也可以  使用寄存器，在Debug下临时给出。程序必须在Debug下执行，以便验  证结果。  （2）跟踪程序，记录每条指令执行后的 ZF,SF,CF,OF标志。回答  每条指令执行后 ZF,SF,CF,OF标志设置的理由。  DATA\_SEG SEGMENT  WORD1 DW 1234H, 5678H ; 32 位操作数 1，低位和高位  WORD2 DW 2345H, 6789H ; 32 位操作数 2，低位和高位  RESULT\_ADD DW 2 DUP(?) ; 存放加法结果  RESULT\_SUB DW 2 DUP(?) ; 存放减法结果  DATA\_SEG ENDS  CODE\_SEG SEGMENT  ASSUME DS:DATA\_SEG  START:  MOV AX, DATA\_SEG  MOV DS, AX ; 初始化数据段    ; 32 位加法：WORD1 + WORD2  MOV AX, WORD1 ; 低 16 位相加  ADD AX, WORD2  MOV RESULT\_ADD, AX ; 存储加法低 16 位    MOV AX, WORD1+2 ; 高 16 位相加  ADC AX, WORD2+2 ; 处理进位  MOV RESULT\_ADD+2, AX ; 存储加法高 16 位  ; 32 位减法：WORD1 - WORD2  MOV AX, WORD1 ; 低 16 位相减  SUB AX, WORD2  MOV RESULT\_SUB, AX ; 存储减法低 16 位    MOV AX, WORD1+2 ; 高 16 位相减  SBB AX, WORD2+2 ; 处理借位  MOV RESULT\_SUB+2, AX ; 存储减法高 16 位    MOV AH, 4CH  INT 21H  CODE\_SEG ENDS  END START  【3】用字符串处理指令编制程序，处理字符串的比较和查找，显  示结果。  要求：  （1）字符串的比较程序中，一个字符串在数据段定义，另一个字  符串在程序执行时从键盘输入，必须定义键盘缓冲区，并指出不相等  的位置。  （2）在字符串中查找某个字符，字符串在数据段定义，要查找的  字符在程序执行时从键盘输入，并指出找到的位置。  DATA\_SEG SEGMENT  STRING1 DB 'HELLO', 0 ; 以 NULL 结尾的字符串  BUFFER DB 10 ; 缓冲区，用于存储键盘输入的字符串  MSG\_EQUAL DB 'Equal', 0  MSG\_NOT\_EQUAL DB 'Not Equal', 0  TARGET\_CHAR DB '?', 0 ; 用于存储待查找的字符  DATA\_SEG ENDS  CODE\_SEG SEGMENT  ASSUME DS:DATA\_SEG  START:  MOV AX, DATA\_SEG  MOV DS, AX ; 初始化数据段    ; 读取键盘输入字符串  MOV AH, 0AH ; 读字符串到缓冲区  LEA DX, BUFFER  INT 21H  ; 比较 STRING1 与输入的字符串  CLD ; 清除方向位  LEA SI, STRING1 ; 设置源地址  LEA DI, BUFFER+1 ; 设置目的地址  MOV CX, 5 ; 比较 5 个字符  REPE CMPSB ; 若字符相同则继续比较  JNE NOT\_EQUAL  EQUAL:  LEA DX, MSG\_EQUAL  JMP DISPLAY\_MSG  NOT\_EQUAL:  LEA DX, MSG\_NOT\_EQUAL  DISPLAY\_MSG:  MOV AH, 09H  INT 21H  ; 查找字符串中的字符  MOV AH, 01H ; 读取待查找的字符  INT 21H  MOV TARGET\_CHAR, AL  LEA DI, STRING1 ; 设置查找字符串的地址  MOV CX, 5 ; 查找字符串长度  MOV AL, TARGET\_CHAR ; 设置待查找的字符  REPE SCASB ; 查找  JNE NOT\_FOUND  FOUND:  ; 显示字符找到的位置  MOV AH, 09H  LEA DX, MSG\_EQUAL ; 假设 'Equal' 用作找到提示  INT 21H  JMP EXIT  NOT\_FOUND:  MOV AH, 09H  LEA DX, MSG\_NOT\_EQUAL ; 假设 'Not Equal' 用作未找到提示  INT 21H  EXIT:  MOV AH, 4CH  INT 21H  CODE\_SEG ENDS  END START  【4】AL字节内容反向排序。  CODE\_SEG SEGMENT  ASSUME CS:CODE\_SEG  START:  MOV AL, 5AH ; 示例数据：AL = 01011010B  MOV BL, 00H ; 清除 BL  MOV CX, 08H ; 准备循环处理 8 位  REVERSE\_BITS:  ROR AL, 1 ; 右移 AL，最低位到 CF  ROL BL, 1 ; 左移 BL，CF 到 BL 的最高位  LOOP REVERSE\_BITS ; 循环处理  MOV AL, BL ; 反转后的值存回 AL  ; 退出程序  MOV AH, 4CH  INT 21H  CODE\_SEG ENDS  END START | | | |
| 实验结果与分析  成功掌握多数位的算术运算、移位操作、字符串操作等程序的设计，  学习使用分支与循环等基本编程方法，熟练使用Debug。 | | | |