《汇编语言程序设计实验》注意事项

请大家在网站上下载“2017实验报告样例”和“2017实验报告评分原则”，上机前做好准备，需要完成实验报告中除“实验记录与分析”和“总结与体会”以外的全部内容（实验报告电子版提交时间是当次实验课的课时结束后的次日，推迟提交将影响成绩）。上机时主要是验证所做结果是否正确，记录实验中看到的现象（如果截图，必须对截图中反映的信息给出文字说明），分析讨论各种现象的内在原理，找出错误的原因，以促进对课本知识的理解。

到机房上机要签到、以班级为单位安排座位、检查实验报告预习情况等；提前完成的需经过老师检查后方可离开机房。

请大家参考配套的《80X86汇编语言程序设计上机指南》（华中科技大学出版社2007年）相关内容。

上机使用的相关软件请从汇编教学网站上下载（比如MASM60TD的压缩包等）。自备U盘带相关软件和电子版实验报告，实验报告文件名格式：

CS1501\_ U201512345张三丰\_1.DOC

实验课程成绩=实验报告（70%）+现场实验操作表现（20%）+现场问答讨论（10%）

全部实验完成之后要提交实验报告的合订本纸质版，即5次实验报告的整合，作为终结成绩（每次实验后只需要提交电子版的实验报告，注意，为避免没有收到邮件，学生必须收到回复邮件后才算确认提交成功。5次实验完毕后，提交纸质报告，是5个报告的合订本，外加一个统一的封面）。实验报告评分中包括实验完成质量与撰写质量。

另外请注意，从实验报告评分原则可以看出，汇编语言程序设计实验更强调实验的过程与思考，而不是把某个功能的程序调通了就完事。一个如实记录了最终还没有调通程序的实验过程及思考应对过程的实验报告的成绩，也许会比一个程序没有问题但没有什么过程记录与思考的实验报告成绩高很多。

《汇编语言程序设计实验》

实验一 编程基础

本实验课内学时4+4，即占用两周课内实验课时，实验报告在第二次课内学时完成后的次日提交。

一．实验目的和要求

1. 掌握汇编源程序编辑工具、汇编程序、连接程序、调试工具TD的使用；
2. 理解数、符号、寻址方式等在计算机内的表现形式；
3. 理解指令执行与标志位改变之间的关系；
4. 熟悉常用的DOS功能调用；
5. 熟悉分支、循环程序的结构及控制方法，掌握分支、循环程序的调试方法；
6. 加深对转移指令及一些常用的汇编指令的理解。

二．实验内容

（上机实验环境说明：在机房计算机上建议大家使用VMWare Workstation的虚拟机环境，在自己的计算机上可以使用DOSBox虚拟机。源程序编辑工具可以使用记事本、EDIT、或C语言的编辑器；汇编程序使用MASM 6.0;连接程序使用LINK；调试工具使用TD；具体介绍可参见教材第7章及《80X86汇编语言程序设计上机指南》，相关软件可以到教学网站上下载）

**任务1. 《80X86汇编语言程序设计》教材中 P31的 1.14题。**

要求：(1) 直接在TD中输入指令，完成两个数的求和、求差的功能。求和/差后的结果放在(AH)中。

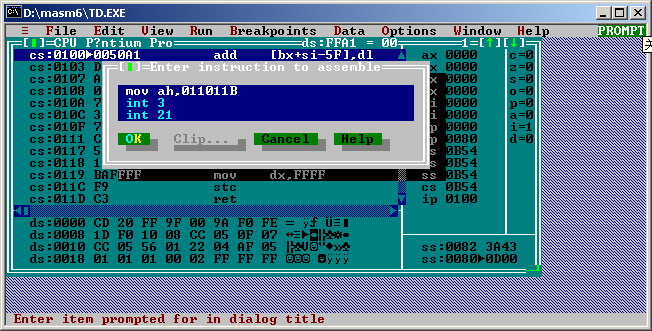
(2) 请事先指出执行指令后(AH)、标志位 SF、OF、CF、ZF的内容。

(3) 记录上机执行后的结果，与（2）中对应的内容比较。

(4)求差运算中，若将A、B视为有符号数，且A>B, 标志位有何特点？

若将A、B视为无符号数，且A>B, 标志位又有何特点？

**在TD中输入指令语句的操作提示：**将TD中的代码显示区置为当前区域，光标移到期望修改的行后，直接输入汇编指令；当输入了第一个字符时，TD自动弹出如下图所示的指令编辑窗口。每输入完一条指令，按回车键，这时输入的指令即可出现在光标处，同时光标自动下移一行，以便输入下一条指令。



**任务2. 《80X86汇编语言程序设计》教材中 P45的 2.3题。**

要求：（1）分别记录执行到“MOV CX，10”和“INT 21H”之前的(BX), (BP),(SI),(DI)各是多少。

（2）记录程序执行到退出之前数据段开始40个字节的内容，指出 程序运行结果是否与设想的一致。

(3)在标号LOPA前加上一段程序，实现新的功能：先显示提示信息“Press any key to begin!”, 然后，在按了一个键之后继续执行LOPA处的程序。

**操作提示：**使用TD.EXE调试程序时，应先单步执行各个语句，每执行一条语句，都应观察数据段中的内容以及相应寄存器的变化。首先注意观察对DS寄存器的赋值过程，并在TD的数据窗口定位待观察的数据区位置。其次，单步执行循环体两遍且正确理解了循环体语句的含义后，可在“MOV AH, 4CH”处设置断点，然后直接执行到断点处，回答(1)和(2)的问题。

**任务3. 《80X86汇编语言程序设计》教材中 P45的 2.4题的改写。**

要求：(1) 实现的功能不变，对数据段中变量访问时所用到的寻址方式中的寄存器改成32位寄存器。

(2) 内存单元中数据的访问采用变址寻址方式。

(3) 记录程序执行到退出之前数据段开始40个字节的内容，检查 程序运行结果是否与设想的一致。

(4)在TD代码窗口中观察并记录机器指令代码在内存中的存放形式，并与TD中提供的反汇编语句及自己编写的源程序语句进行对照，也与任务2做对比。（相似语句记录一条即可，重点理解机器码与汇编语句的对应关系，尤其注意操作数寻址方式的形式）。

（5）观察连续存放的二进制串在反汇编成汇编语言语句时，从不同字节位置开始反汇编，结果怎样？理解 IP/EIP指明指令起始位置的重要性。

**操作提示：**要让TD从任意指定地址开始反汇编，需要使用TD在代码显示区的Goto功能，即鼠标选中代码显示区，点击右键将显示带有Goto的菜单，选中Goto菜单项，输入CS:XXXX即可（XXXX是你希望录入的偏移地址）。

**任务4. 设计实现一个学生成绩查询的程序。**

1、实验背景

在以BUF为首址的字节数据存储区中，存放着n个学生的课程成绩表（百分制），每个学生的相关信息包括：姓名（占10个字节，结束符为数值0），语文成绩（1个字节），数学成绩（1个字节），英语成绩（1个字节），平均成绩（1个字节）。

例如：

N EQU 30

BUF DB ‘zhangsan’,0,0 ;学生姓名，不足10个字节的部分用0填充

DB 100, 85, 80,？ ; 平均成绩还未计算

DB ‘lisi’,6 DUP(0)

DB 80, 100, 70,?

DB N-3 DUP( ‘TempValue’,0,80,90,95,?) ;除了3个已经具体定义了学生信息的成绩表以外，其他学生的信息暂时假定为一样的。

DB ‘wangwu’，0，0，0，0 ；最后一个必须是自己名字的拼音

DB 85, 85, 100, ?

2、功能一：提示并输入待查询成绩的学生姓名

（1）使用9号DOS系统功能调用，提示用户输入学生姓名。

（2）使用10号DOS系统功能调用，输入学生姓名。输入的姓名字符串放在以in\_name为首址的存储区中。

（3）若只是输入了回车，则回到“(1)”处重新提示与输入；若仅仅输入字符q，则程序退出,否则，准备进入下一步处理。

3、功能二：以学生姓名查询有无该学生

（1）使用循环程序结构，在成绩表中查找该学生。

（2）若未找到，就提示用户该学生不存在，并回到“功能一（1）”的位置，提示并重新输入姓名。

（3）若找到，则将该学生课程成绩表的起始偏移地址保存到POIN字变量中。

**提示：**字符串比较时，当采用输入串的长度作为循环次数时，若因循环次数减为0而终止循环，则还要去判断成绩表中名字串的下一个字符是否是结束符0，若是，才能确定找到了（这样做是为了避免输入的名字仅仅是数据段中所定义名字的子集的误判情况）。

4、功能三：计算所有学生的平均成绩

使用算数运算相关指令计算并保存每一个学生的平均成绩。

平均成绩计算公式：(A\*2+B+C/2)/3.5，即将语文成绩A乘以权重2、英语成绩C除以权重2后，与数学成绩B一起求和，再计算该生的平均成绩。**要求避免溢出**。

**提示：**使用循环程序结构，注意寻址方式的灵活使用。把小数3.5转换成分数后再运算避免使用浮点数指令。

5、功能四：将功能二查到的学生的平均成绩进行等级判断，并显示判断结果。

（1）平均成绩等级显示方式：若平均成绩大于等于90分，显示“A”；大于等于80分，显示“B”；大于等于70分，显示“C”；大于等于60分，显示“D”；小于60分，显示“F”。

**提示：**使用分支程序结构，采用2号DOS系统功能调用显示结果。

（2）使用转移指令回到“功能一（1）”处（提示并输入姓名）。

进度提示：建议第一次上机学时完成任务1至3；第二次上机学时完成任务4。

变更提示：学生可以自行调整实验内容，但自选的内容要能完成实验目的与要求，并经老师同意。任务4的内容不宜自行更改太多，因后续实验还会用到。

**在上述任务中，需要大家思考、尝试、分析、总结的重要提示：**

**（请以合适的方式纳入到自己写的实验步骤、记录与分析以及总结中）**

本次实验中需要学生尝试体会的操作可分为三类：

1. TD的操作（直接在TD内输入指令，设置断点、单步执行程序、运行到断点、运行到指定的指令、查看寄存器的内容、查看标志寄存器的某个标志位的值、查看指定的存储单元的值、查看整个程序在内存中的存放方式）；

2. 将一个汇编源程序生成一个可执行文件的步骤，包括如何读懂汇编源程序在汇编过程中产生的错误信息；编辑工具的使用（使用Edit， notepad等需要注意的问题）；

3. 操作数寻址方式的改变，对目标码产生的影响； DOS系统功能调用中应注意的问题。

各实验任务中的上述三类操作的具体提示如下表：

|  |  |  |
| --- | --- | --- |
| 任务编号 | 序号 | 学生尝试的操作 |
| 任务1 | 1 | 打开TD之后，如何在代码区输入一条指令，并且执行这条指令？ |
| 2 | 如何在代码区输入若干条指令后，再从输入的第一条指令开始执行？ |
| 3 | 在输入一条指令中的数据时，若以16进制输入，需要注意什么问题？ |
| 4 | 输入指令MOV AH, -128并执行 查看AH寄存器的内容 |
| 5 | 任务1中的(1)~(4) |
| 6 | 执行一条指令后，如何查看寄存器的值（含32位寄存器）？如何修改寄存器的值？ |
| 7 | 执行一条指令后，如何查看标志寄存器的值？ |
| 8 | 经过6，7后，总结加减法对标志寄存器的影响？ |
| 任务2和任务3 | 1 | 如何将一可执行程序调入TD，并查看代码区？理解机器码与汇编指令之间的对应关系？ |
| 2 | 如何设置断点并运行到断点？ |
| 3 | 如何使程序运行到光标的当前点？（假设活动光标在代码区，指向某一条指令） |
| 4 | 如何单步执行一条指令？（多种方法） |
| 5 | 在数据区找到数据段的方法？思考：是否可以用这一方法查看代码段甚至整个程序?（至少有三种方法：(1)goto，DS：偏移地址，(2)goto, 直接输入：段寄存器的值：偏移地址；(3)直接在数据区用光标移动查找） |
| 6 | 修改某个指定的存储单元的值，如任务2中的BUF2 |
| 7 | 如何查看堆栈？ |
| 8 | 如何汇编一个汇编源程序并链接产生可执行代码？ |
| 9 | 如何读懂源程序在汇编过程中产生的错误？ |
| 10 | 使用DOS系统功能调用前应注意保存AX寄存器的值 |
| 11 | 查看BUF2等变量在TD中的表示形式并总结 |
| 12 | 查看寄存器间接寻址、变址寻址的汇编源程序经汇编、链接后在TD中的表示形式？总结源程序的指令和TD中的指令差异。 |
| 13 | 单步执行循环体2遍，查看数据段的变化 |

**对于任务4，可以思考与选做功能包括：**

（1）对输入的姓名进行合法性检查，比如：是否存在大小写字母之外的字符，等等；若存在非法的输入情形，则提示之后重新输入。

（2）对BUF中预先定义的课程成绩表中的所有成绩项进行合法性检查，判断是否符合实际（如超过100分等），不符合则提示并退出程序。预先定义的学生姓名中，如果有重名如何处理？

（3）除了显示平均成绩的等级外，还将平均成绩同时显示。

（4）由于有些课程体系中，成绩的满分可能会达到300分，此时，成绩项的定义必须采用字类型，那么，如何改造程序呢？采用数据结构来定义BUF存储区是否会更好？

**任务4中，重要操作提示：**

|  |  |  |
| --- | --- | --- |
| 功能编号 | 序号 | 学生尝试的操作 |
| 功能一、二 | 1 | 如何让9号功能调用显示的信息放在自己希望的位置？ |
| 2 | 如果在9号功能调用时，带显示字符串的结尾没有“$”结束符会怎样？ |
| 3 | 如果在9号功能调用前，未对DS赋值，或者未对DX给予正确的值，结果会如何？ |
| 4 | 10号功能调用时，输入的字符数超过定义的数量时，它是如何处理的？ |
| 5 | 查找学生姓名时，如何提高查找速度？ |
| 6 | 循环或转移时，是否有多种指令的组合方式实现？ |
| 7 | 注意观察转移指令机器码的编码方法，观察对应标号的偏移地址与该编码之间的关系。 |
| 8 | 检查循环程序，是否可能出现死循环的情况。 |
| 功能三、四 | 1 | 把除以3.5，改成乘2除以7。简化一下计算公式。 |
| 2 | 单步观察计算过程，结果是否正确？ |
| 3 | 用TD直接修改BUF区的成绩值，使得程序对应的每个分支都能得到验证。 |
| 4 | 选取特殊的值，观察计算结果溢出的情况。 |
|  |  |

**最后，对实验一的实验报告给出如下要求与提示：**

（1）**第一个4学时上机前，实验报告应完成任务1至3的关键步骤描述（也即你准备具体操作和观察哪几个事情），给出任务3的源程序清单（初稿，允许有错误）。**进入实验后，按照步骤完成实验操作并记录相关结果（可以截图），对结果进行分析说明。写出对任务1至3的总体结论与体会。

（2）**第二个4学时上机前，实验报告应完成任务4的步骤描述，程序设计思想、流程图、源程序初稿等。**进入实验后，按照步骤完成程序调试、实验操作并记录相关结果（可以截图），对结果进行分析说明。在任务1至3的总结基础上，写出对任务4的总体结论与体会。

（3）最后提交的实验报告应包含对实验目的和要求以及实验内容的描述，请复制上述页面中的相关内容，但需要把所有提示都删除。