第 1 章　汇编语言程序实验基础

1.1　汇编语言程序的开发过程

汇编语言程序的开发过程如图 1.1 所示。这个过程主要由编辑、汇编、连接几个步骤构成。

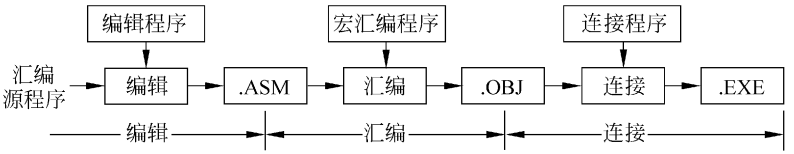

图 1.1　汇编语言程序的开发过程

1. 源程序的编辑

编辑过程就是调用编辑程序把源程序输入内存，生成一个扩展名为 ASM 的文本源文件并存入磁盘。如果是对原有的 ASM 文件进行修改，还会自动生成一个扩展名为 BAK 的备份文件，它保存的是修改前的 ASM 文件。用 DOS 提供的 EDIT. EXE 或其他全屏幕编辑软件都能完成编辑任务。

2. 源程序的汇编

汇编就是调用汇编程序（如 MASM 或 TASM）对源程序进行翻译，生成扩展名为 OBJ 的目标文件。在汇编过程中，若汇编器检查到源程序中有语法错误，则不生成目标文件，但给出错误提示信息。根据用户需要，汇编程序还可生成列表文件（LST 文件）和交叉引用文件（CRF 文件）。

3. 目标程序的连接

连接的过程是：调用连接程序（如 LINK 或 TLINK）将用户目标程序和库文件进行连接、定位，生成扩展名为 EXE 的可执行文件。连接时，如果在目标文件或库文件中找不到所需的连接信息，则连接程序给出错误提示信息，而不生成可执行文件。根据用户需要，连接程序还可以生成内存分配文件（MAP 文件）。

4. COM 文件的生成

按照 COM 文件的汇编格式设计的源程序，在生成 EXE 文件之后，还要转换成 COM 文件。如果连接时使用 TLINK. EXE 程序，在 TLINK 命令后加选项“/t”，可直接生成 COM 文件。但是宏汇编 MASK. EXE（5. 0 版本以下）不支持此项功能，还需要调用转换程序 EXE2BIN. EXE，才能将 EXE 文件转换成 COM 文件。假设用户的 EXE 文件、COM 文件

和转换程序都在同一目录下，则键入：

EXE2BIN "文件名".EXE "文件名".COM

如果不能转换就给出错误提示信息。

5. 调试可执行程序

可执行程序运行之后没有得到预想的结果，就需要重新审查源程序，找出算法或者逻辑错误，再重新编辑、汇编、连接和执行。在调试程序的时候，也可以借助调试软件。

1.2 汇编语言常用软件的使用方法

建议读者在C(或D)盘建立一个工作子目录，将汇编语言常用的软件装入工作子目录之下，在工作子目录下，完成编辑、汇编和连接等项操作，生成的用户文件也存放在工作子目录之下。本节在介绍常用软件使用方法时，不再说明文件的盘符和路径。

1.2.1 编辑程序(EDIT.EXE)

1. 启动EDIT进入编辑状态

(1) 在DOS环境下键入：EDIT ↙

这条命令执行EDIT.EXE程序，自动进入编辑状态，屏幕顶部显示主菜单，底部行显示提示信息，中间的22行为编辑窗口，用户使用编辑命令逐行输入源程序。由于该条命令没有给出待编辑的文件名，因此编辑后文件存盘时，要键入“Alt+F”，选择其中的Save功能，并通过会话给出文件名，才能将编辑后的文件存盘。

(2) 在DOS环境下键入：EDIT 待编辑的文件名.ASM ↙

例如，键入：EDIT ABC.ASM ↙

这条命令有3个作用：

① 执行EDIT.EXE程序，自动进入编辑状态，屏幕顶部显示主菜单。

② 如果工作目录中没有“ABC.ASM”文件，则编辑窗口没有信息，编辑后的文件存盘时，将自动存放在ABC.ASM文件中。

③ 如果工作目录中有“ABC.ASM”文件，系统将自动把ABC.ASM文件调入内存，并将开始的22行语句显示在编辑窗口供编辑。

2. 常用的编辑命令

Page Up	屏显上一页语句行
Page Down	屏显下一页语句行
↑、↓、←、→	光标上移、下移一行，左移、右移一位
Ctrl + ↑	屏幕信息向下移动一行
Ctrl + ↓	屏幕信息向上移动一行
Ctrl + ←	光标向左移动一个区段
Ctrl + →	光标向右移动一个区段

Ctrl + Home	光标移到文件的开始行
Ctrl + End	光标移到文件的末尾
Home	光标移到当前行的开始
End	光标移到当前行的末尾
Enter	插入新的一行
Delete	删除光标所在位置上的一个字符
Backspace	删除光标左侧的一个字符
Ctrl+Y	删除光标所在行的全部信息

说明：

Ctrl+Y 是双键操作，即按下 Ctrl 不松手，再按 Y 键。其他的双键操作类同。

3. EDIT 的编辑功能

EDIT 采用两级下拉式菜单显示编辑功能。启动 EDIT 之后，屏幕顶部显示 6 个主菜单项，键入“Alt + 主菜单项首字母”即能显示相应的二级菜单；或者先按 Alt 键，然后按“←”、“→”键移动光条，再按回车键，也能显示二级菜单。显示二级菜单之后，按“↑”、“↓”键移动光条，再按回车键，即可选择相应的编辑功能；也可以通过热键操作选择编辑功能。本小节仅作简要介绍，对于比较复杂的编辑功能，下文再作专题描述。

(1) File　这项功能的二级菜单提供了与文件有关的操作。

New　建立一个新文件。

Open　从磁盘装入一个文件。

Save　当前编辑的文件存盘。

Save As　改用新的文件名存储当前文件。

Print　打印文件。

Exit　退出编辑状态，返回 DOS。

(2) Edit　对文件进行“块操作”。

(3) Search　完成字符串搜索与替换。

(4) View　进行多窗口编辑。

(5) Options　它有两个子功能。其一，设置 Tab 键控制的光标移动格数；其二，设置打印口。

(6) Help　显示帮助信息。

4. 块操作功能

若干连续的语句行称为一个“程序块”，一行中若干连续的字符称为一个“字符块”。主菜单 Edit 提供了“块定义”、“块复制”、“块转移”、“块删除”操作，尤其是块复制、块转移，为修改源程序提供了很大的方便。进行块复制、块转移、块删除之前，必须进行块定义。假设源程序的第 10～15 行为一个程序块，下面以此为例，介绍块操作的方法。

(1) 块定义操作步骤

① 按“↑”、“↓”键将光标移动到第 10 行。

② 按下 Shift 键(不松手)，再连续按“↓”键直到第 15 行，光标经过的行将变色，显示为

白底黑字，从而完成了“程序块”的定义。如果要定义字符块，应先将光标移动到字符块的首字符，然后按下 Shift 和“→”键，即可完成“字符块”的定义。

(2) 块复制操作步骤

假设把源程序的第 10～15 行复制一份到第 20～25 行。

① 首先完成第 10～15 行的块定义。

② 键入“Ctrl＋C”，将定义的程序块复制到 EDIT 指定的缓冲区。

③ 移动光标到第 20 行。

④ 键入“Ctrl＋ V”，将缓冲区中的程序块复制一份到第 20～25 行，就完成了“块复制”的操作。缓冲区中的程序块并不消失。

(3) 块转移操作步骤

假设把源程序中的第 10～15 行转移到第 20～25 行。

① 首先完成第 10～15 行的块定义。

② 键入“Ctrl＋ X”，将定义的程序块复制到 EDIT 指定的缓冲区，并且删除被定义的程序块。

③ 移动光标到第 20 行。

④ 键入“Ctrl＋ V”，将缓冲区中的程序块复制一份到第 20～25 行，就完成了“块转移”的操作。缓冲区中的程序块并不消失。

(4) 块删除操作步骤

首先进行块定义操作，然后按 Delete 键，被定义的程序块即被删除。

5. 多窗口编辑功能

当程序员正在编辑某个程序，同时又想参考另一个程序的时候，怎么办呢？主菜单 View 的多窗口编辑功能可以同时把两个程序调入编辑窗口，供程序员查阅。多窗口的编辑功能还可以实现文件之间的块复制、块转移。虽然如此，但读者应该慎重使用文件之间的块转移操作，其原因是不言而喻的。

(1) 进入多窗口编辑环境

假设正在编辑的程序是“1. ASM”，需要参考的程序是“2. ASM”。怎样进行多窗口编辑呢？步骤如下：

① 首先将“1. ASM”调入编辑窗口。

② 选择 View|Split Window，该项功能把编辑窗口一分为二，形成上、下两个窗口。

③ 选择 File|Open，将“2. ASM”程序装入内存。此时，上窗口显示的是“1. ASM”的语句行，下窗口显示“2. ASM”的语句行，按“↑”、“↓”键可以使下窗口程序滚动显示，从而达到查阅“2. ASM”程序的目的。

(2) 文件之间的块复制操作

在上述操作的基础上，怎样把“2. ASM”程序中的第 10～15 行语句复制到“1. ASM”程序的第 20～25 行呢？步骤如下：

① 按“↑”、“↓”键，将光标移动到下窗口“2. ASM”程序的第 10 行。

② 按下 Shift 键不松手，再按下“↓”键，完成第 10～15 行语句的“块定义”。

③ 键入“Ctrl＋C”，将定义的程序块复制到 EDIT 指定的缓冲区。

④ 选择 View|Close Window，这项功能将取消多窗口显示，使编辑窗口恢复，全屏幕显示“1. ASM” 程序的语句行。

⑤ 将光标移动到“1. ASM”程序的第 20 行。

⑥ 键入“Ctrl＋ V”，将缓冲区中的语句行复制到“1. ASM”程序的第 20～25 行，完成文件之间的“块复制”。

6. 字符串搜索与替换

变量、标号、寄存器名、立即数等都是字符串。在一个大型程序中，如果某字符串被多次引用，现在需要用另一个字符串去替换它，在这种情况下，由于语句行比较多，人工搜索与替换比较费事，而且容易遗漏，此时可借助“字符串搜索与替换”功能。主菜单 Search 提供的字符串搜索与替换功能，有两种搜索方式：一种是单步搜索，另外一种是连续搜索。操作步骤如下：

① 按“Ctrl＋Home”键，将光标移到文件的首部。

② 选择 Search|Replace，屏幕上立即出现一个对话框。在 Find what 一栏中键入要搜索的字符串，在 Replace with 一栏中键入用以替换的字符串。接下来选择搜索方式。

③ 用户将光标移动到 Replace 框，按下回车键之后，EDIT 自动进行单步搜索，每搜索到一个匹配的字符串，就停止搜索，屏幕显示另一个对话框，询问用户当前搜索到的字符串是否要替换它。用户把光标移到 Replace 框，按回车键，就实现替换功能；反之把光标移到 Skip 框按回车键，就是不替换。

④ 用户将光标移动到 Replace All 框，按下回车键之后，EDIT 将连续搜索并自动替换，不再征询用户意见。

1.2.2 宏汇编程序(MASM. EXE)

宏汇编程序以源程序(一定要有扩展名 ASM)为汇编对象。汇编过程中，宏汇编程序对源程序进行两次扫描，检查其语法错误，如果没有语法错误，则根据程序员的要求生成目标文件(OBJ)、列表文件(LST)和交叉引用文件(CRF)。宏汇编程序启动后，首先显示版本信息，然后逐条进行人机会话。

宏汇编程序可以生成文件名互不相同的 OBJ 文件、LST 文件和 CRF 文件，但这样做没有好处，一般都生成与源文件同名的目标文件(OBJ)。假设待汇编的源程序为“ABC. ASM”，并以此为例介绍人机会话过程。

1. 启动宏汇编程序的 3 种方法

(1) 在 DOS 环境下键入：MASM ↙

系统会给出下列机上提示

人机会话过程如下：

```
Source filename[.ASM]: ABC ↙          ; 键入源程序名
Object filename[ABC.OBJ]: ↙           ; 生成的目标文件为 ABC.OBJ
Source listing[NUL.LST]: ↙            ; 不生成 LST 文件
```

```
Cross_reference[NUL.CRF]: ↙                    ; 不生成 CRF 文件

ABC.ASM(10): error A2009: Symbol not defined: MESG
0  Warning Errors                              ; 0 个警告性错误
1  Severe Errors                               ; 1 个严重错误
```

这种启动方式有 4 句会话。

第 1 句：请键入待汇编的源程序名。

第 2 句：询问生成的目标文件是否是 ABC.OBJ，按回车键表示认可。

第 3 句：询问是否要生成列表文件，按回车键表示不生成 LST 文件。如果要生成 LST 文件，请键入文件名。

第 4 句：询问是否要生成交叉引用文件，按回车键表示不生成 CRF 文件。

会话结束，宏汇编程序开始汇编，并给出汇编结果。上述源程序汇编后，有 1 个严重错误，即 ABC.ASM 程序的第 10 行，有 1 个符号名 MESG 没有定义过。程序员应当重新编辑，改正错误，然后再次汇编，直到没有语法错误为止。假设语法错误已经改正了。

(2) 在 DOS 环境下键入：MASM　待汇编的源文件名 ↙

例如：MASM ABC ↙

这种启动方式在命令行中给出待汇编的源文件名，因此人机会话只有 3 项内容，示范如下：

```
Object filename[ABC.OBJ]: ↙                   ; 生成的目标文件为 ABC.OBJ
Source listing[NUL.LST]: ↙                    ; 不生成 LST 文件
Cross_reference[NUL.CRF]: ↙                   ; 不生成 CRF 文件

0  Warning Errors                              ; 0 个警告性错误
0  Severe Errors                               ; 0 个严重错误
```

(3) 在 DOS 环境下键入：MASM　待汇编的源文件名；↙

例如：MASM ABC；↙

这种启动方式在命令行中给出待汇编的源文件名，并且以分号结束。按下回车键之后，没有人机会话过程，汇编结束后，同样给出汇编信息。如果源程序没有语法错误，则自动生成同名的 OBJ 文件，不生成 LST 文件和 CRF 文件。推荐使用这种方法。

2. 宏汇编参数

启动宏汇编程序的时候，可以携带参数。宏汇编参数是任选项，参数以“/”开头，紧跟在 MASM 之后，参数可以连用，在 DOS 环境下键入“MASM/H ↙”之后，屏幕上就能显示该版本宏汇编程序的参数表和各参数的功能。下面介绍几个比较有用的宏汇编参数：

/L　汇编后能生成 LST 文件。

/Z　如果源程序有错，汇编结束后，能显示有语法错误的语句行。

/Zi　产生用于 Code View 的符号信息。

1.2.3 连接程序(LINK.EXE)

经过汇编之后生成的目标文件(OBJ)不能在PC上运行,必须经过连接程序的连接和定位才能生成可执行文件(EXE)。

对于多模块程序(即源程序由1个主模块和若干子模块组成),汇编后生成的是各个模块的目标文件,连接程序要把各个模块的目标文件连接在一起,才能生成可执行文件。

如果设计源程序的时候,调用了库文件子程序(库文件由程序员自己创建,它是若干子程序的集合,文件扩展名为LIB),连接程序还要把被调用的库文件子程序和目标文件连接在一起才能生成可执行文件。

主教材和实验教程中的例题、实验均为单模块程序,而且没有库文件,因此在启动连接程序,回答问句"Libraries[.LIB]"的时候,请按下回车键,表示没有库文件。

启动连接程序也有3种方法,假设目标文件为ABC.OBJ,并以此为例介绍连接程序启动后的人机会话过程。

(1) 在DOS环境下键入:LINK ↙

人机会话过程如下:

```
Object Modules[.OBJ]: ABC ↙          ; 键入目标文件名
Run File[ABC.EXE]: ↙                 ; 生成的可执行文件为ABC.EXE
List File[NUL.MAP]: ↙                ; 不生成MAP文件
Libraries[.LIB]: ↙                   ; 没有库文件参与连接

LINK: Warning L4021: no stack segment
```

说明:

① 扩展名MAP是映像文件,它给出程序中每个逻辑段的位置和段长度,一般不需要生成MAP文件,因此在回答问句"List File[NUL.MAP]"时,按下回车键即可。

② 如果连接时发现错误,连接程序将会给出错误提示信息。

③ 连接程序要求生成EXE文件的源程序格式中,必须有堆栈段(堆栈段的标志是该逻辑段的段定义语句中有"STACK"段参数),否则连接之后,连接程序将给出"没有堆栈段"的警告信息。但是,这不代表源程序有语法错误。在程序设计中,如果没有大批数据进出堆栈,为了简化程序设计,程序员往往不单独设置堆栈段,DOS在装载没有堆栈段的EXE文件时,将会自动给程序分配至少128B的堆栈区。

(2) 在DOS环境下键入:LINK　目标文件名 ↙

例如:LINK ABC ↙

这种启动方式在命令行中直接给出待连接的目标文件名,按下回车键之后,只进行后3项人机会话。

(3) 在DOS环境下键入:LINK　待连接的目标文件名; ↙

例如:LINK ABC; ↙

这种启动方式在命令行中给出待连接的目标文件名,并以分号结束,按下回车键之后,没有人机会话过程。如果程序没有错误,连接后自动生成同名的EXE文件,不生成MAP文件。如果没有库文件参与连接,这种方法是比较快捷的。

生成可执行文件(EXE)之后,在 DOS 环境下,键入其文件名即可执行了。必须指出,没有汇编错误,没有连接错误,只能说明源程序没有语法错误;如果源程序在算法或者其他方面有错,那么执行结果也是不正确的,此时应认真分析,找出原因,再重新编辑、汇编和连接。

1.2.4 调试程序(CV.EXE)

CV 是 Code View 的缩写,CV.EXE 是 MASM 5.1 软件包中的一个调试程序,它能够显示待调试的源程序及其目标代码。在 Code View 控制下,可以慢速地或者单步运行程序,监视程序的转向,可以实时地观察相关寄存器内容的变化,也可以检查和修改内存单元的数据,本小节仅介绍 Code View 用于调试汇编语言程序的常用命令。假设CV.EXE和待调试的程序(假设为 ABC.EXE)都在同一目录下。

(1) 启动 Code View 的命令

在 DOS 环境下键入:CV　待调试的文件名 ↙

例如:CV ABC ↙

Code View 启动后,屏幕显示如图 1.2 所示。

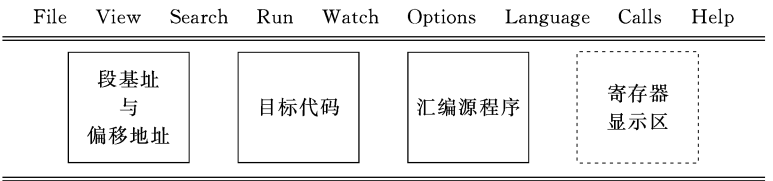

图 1.2　Code View 启动后的屏幕显示

Code View 采用两级下拉式菜单,顶部为 9 个主菜单项。键入“Alt ＋ 主菜单项首字母”可以显示属下的二级菜单,或者先按下 Alt,再按“←”、“→”键,移动蓝色小光条使之覆盖某一菜单项,然后再按回车键,也能显示其属下的二级菜单。二级菜单项显示后,按“↑”、“↓”键移动蓝色小光条,再按回车键即可选择相应的菜单项功能。按下 Esc 键可以关闭二级菜单或者当前的显示框。

主菜单以下,两条双线框之内是调试显示窗口,显示待调试程序的前 18 行语句和相应的目标代码,一条长长的蓝色光条覆盖程序的启动指令。

双线框以下为命令窗口,显示 Code View 的版本和版权说明。“>”号是 Code View 的提示符,“—”是闪烁的光标。

(2) 扩大/ 缩小显示窗口的命令:Ctrl＋T/Ctrl＋G

键入“Ctrl＋T”可以扩大显示窗口,使其最多能显示 21 行指令;键入“Ctrl＋G”,则减小显示窗口的面积。

(3) 光标移动命令:F6

按 F6 键可以使光标在上下窗口之间移动,当光标移到上窗口时,再按“↑”、“↓”键可以使光标移动到某一程序行,按 PgUp 或 PgDn 键可以换页显示。

(4) 打开/关闭寄存器显示窗口的命令:F2

按 F2 键可以打开或关闭寄存器显示窗口。

(5) 单步执行命令：F8

每按一次 F8 键，就执行一条指令，随后蓝色光条移动到下一次要执行的程序行。如果寄存器窗口是打开的话，可以实时地观察到相关寄存器内容的变化。

(6) 单步执行命令：F10

F10 键也是单步执行命令。它和 F8 命令不同的是：按 F10 单步执行的时候，不跟踪子程序的执行过程。

(7) 重新设置启动行的命令：Run|Restart

选择主菜单 Run 属下的 Restart，可以使蓝色光条重新覆盖启动指令。

(8) 断点设置命令：F9

在调试过程中，如果想使程序连续运行到某一指令行自动停止，以便观察前一段程序的执行结果，怎么办呢？应当先将该指令行设置为“断点”，设置断点的步骤如下：

先按 F6 键，使光标上移到显示窗口，再按“↑”、“↓”键使光标移动到指定的程序行，之后按 F9 则断点设置成功(再次按 F9 可取消该断点)。如法炮制，可以设置多个断点。

(9) 取消全部断点的命令：Run|Clear Breakpoints

选择主菜单 Run 属下的 Clear Breakpoints，可以一次性地消除已设置的全部断点。

(10) 连续运行命令：F5 或者 Run|Start

当蓝色光条覆盖启动指令的时候，在全部断点清除的条件下，按 F5 键，或者选择主菜单 Run 属下的 Start，可以使程序连续运行直到程序结束；如果有断点，则连续运行到断点处自动停止。

(11) 连续地慢速运行命令：Run|Execute

当蓝色光条覆盖启动指令的时候，在全部断点清除的条件下，选择主菜单 Run 属下的 Execute 可以使程序慢速运行(1s 执行 3～4 条指令)，直到程序结束；如果有断点，则连续运行到断点处自动停止。

(12) 屏幕切换命令：F4

如果程序运行后要在屏幕上显示运行结果，怎样观察呢？按 F4 键，则屏幕显示切换到第 0 页启动 Code View 之前的状态，即可观察程序的运行结果，敲击任意键之后，屏幕又恢复显示 Code View 窗口。

(13) 寄存器显示与修改命令：R＋寄存器名称

只要打开寄存器显示窗口，就可以显示寄存器的内容，若要修改寄存器内容，先将光标移到命令窗口(即下窗口)，再键入“R”命令。Code View 将以不带后缀 H 的十六进制数显示寄存器的内容等待修改。修改时，若键入的是十进制数，应先键入“N10 ↙”，然后再键入十进制数。若键入的是十六进制数，应先键入“N16↙”，然后再键入十六进制数。

例如，将 AX 的内容从 1000(十六进制数为 3E8H)改为 2000(十六进制数为 7D0H)，操作步骤如下：

```
>N10 ↙   ; 或者 n10 ↙,表示要键入的数为十进制数
>RAX ↙   ; 键入“R”命令,显示 AX 的内容
AX  03E8  ; Code View 显示 AX 的内容为 3E8H,换行显示“:”,等待修改
:   2000  ; 程序员键入十进制数 2000,Code View 自动转换成二进制数并写入 AX
```

例如，将 AX 的内容从 7D0H 改为 3E8H，操作步骤如下：

```
>N16 ↙   ；或者 n16 ↙，表示要键入的数为十六进制数
>RAX ↙   ；键入“R”命令，显示 AX 的内容
AX  07D0  ；Code View 显示 AX 的内容为 7D0H，换行显示“：”，等待修改
:   3E8   ；程序员键入十六进制数 3E8，Code View 自动转换成二进制数并写入 AX
```

(14) 显示存储单元内容的命令：D＋段基址：偏移地址

将光标移到下窗口，先键入“N16 ↙”，然后再键入“D”命令，即可显示从指定单元开始的存储单元的内容。例如：

```
>N16 ↙                ；或者 n16↙
>D3F16：0000 ↙        ；键入“D”命令，后跟存储单元地址就能显示其内容
```

(15) 修改存储单元内容的命令：E＋段基址：偏移地址

将光标移到下窗口，先键入“N16 ↙”，然后再键入“E”命令，即可显示该单元的内容，等待修改。例如：

```
>N16 ↙                ；或者 n16 ↙
>E3F17：0010 ↙        ；键入“E”命令，后跟存储单元地址
3F17：0010  B8._      ；Code View 显示该单元的内容为 B8H，在闪烁的光标处键入两位十六进制数，然后键入回车，修改完毕。
```

(16) 退出 Code View 的命令：Q

按 Q 键，再按回车，即可退出 Code View 返回 DOS，或者选择主菜单项 File 属下的 Exit，也可以返回 DOS。