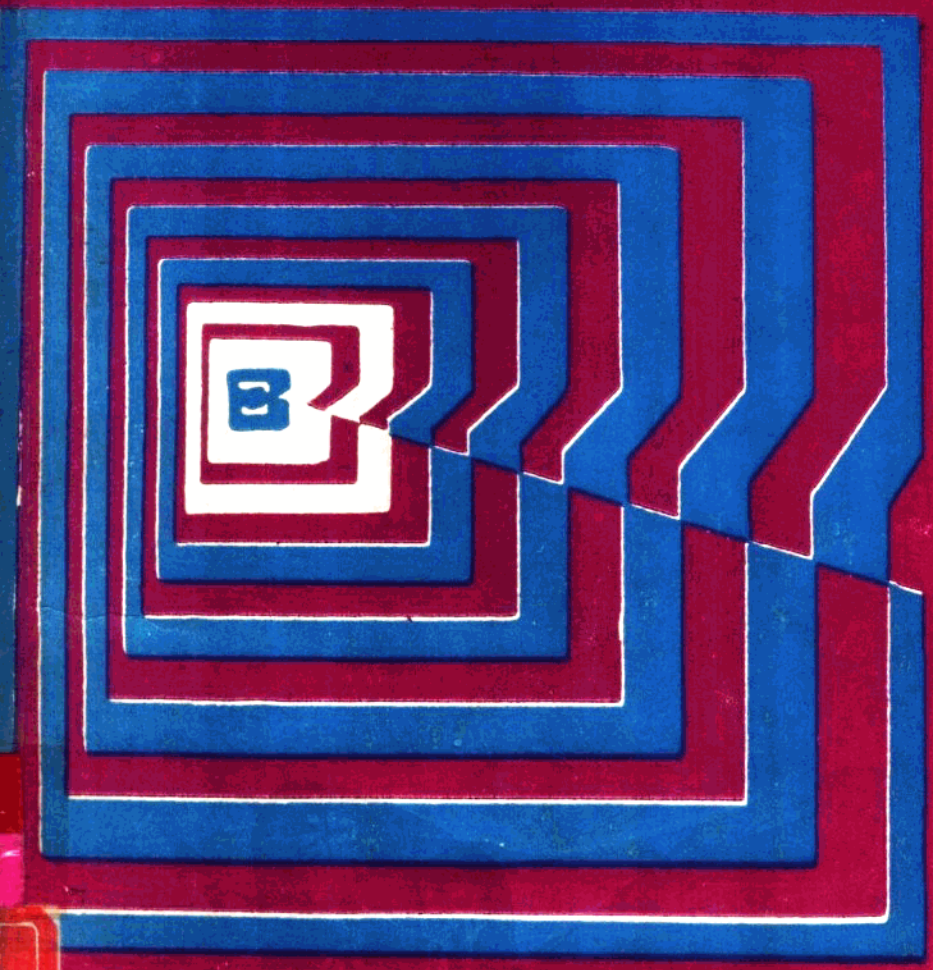
C语言编程技巧实用教程



雷自学编著

陕西电子编辑部

前　言

世界上，计算机语言数以千计，而最流行最受欢迎的不过只有少数几种，C 语言就是其中之一．这是因为 C 语言具有以下几方面优于其它计算机语言的特点：

.C 语言以小写英文字母为基础，表达方式灵活，程序紧凑，具有丰富的运算符和多种数据类型，类型转换方式十分灵活，便于编写程序；

.C 语言提供的与地址密切相关的指针以及位运算符，使它可以成功的取代汇编语言中大部分语句；

.C 语言提供了多种存储属性，其中寄存器变量的使用大大提高了程序的运行速度；

.C 语言是结构化语言，适用于大型程序的模块化设计．各模块分开编译的手段给开发大型程序带来方便；

.大部分 C 编译版本都提供了库函数，通过预处理进行宏调用，从而缩短了程序篇幅．

六十年代末七十年代初，美国人为了描述 UNIX 操作系统，在贝尔实验室构造了 C 语言．事实上，C 语言不仅适合于开发操作系统，而且由于它的高效率，所以也适用于科学计算，就是在应用程序设计方面，由于它提供了十分灵活的人-机交互方式，同样可以大显神威．目前，C 语言已经成为一种得到普遍应用的通用程序设计语言，各种大中小型机以及微型机上，几乎都配备了 C 语言．从七十年代初问世以来，C 语言就技压群芳，表现出极其强烈的生命力，成为软件人员的第一语言．

近几年来，微型计算机在我国不断推广和普及，而且随着微机档次的不断增高，XENIX 操作系统（类似 UNIX 操作系统）和 PC-DOS 一样，成了微型机上的又一主力操作系统，由于 C 语言和 UNIX 的特殊关系，加上它本身的许多优点，越来越多的科技工作者都对 C 语言产生了浓厚的兴趣，他们迫切要求学习 C 语言．因为 C 语言以往主要用在大中小型机上，微机上的广大科技工作人员对 C 语言了解不多，仅仅凭道听途说，以为 C 语言如何如何难学．凭心而论，C 语言的基本成份也象其它高级语言一样是很容易了解和使用的．编者曾给来自各条战线上的不同年龄层次、不同文化程度和拥有不同计算机知识的人士，用这本书的原稿讲过课，效果较好．实践证明，C 语言能够成为广大科技人员手中的工具．

编程技巧历来都是十分诱人的话题．尤其是计算机应用的初期阶段，由于硬件方面的原因，如内存小、速度慢等等，为了一点小小的改进，程序员可以废寝忘食和挖空心思地追求程序的技巧性．随着硬件的发展，内存和速度已不再成为问题，过分追求技巧反而会弄巧成拙．好的程序首先应该是正确的，而且又是朴实、易读、易懂的．编者以为，熟悉您打算使用的计算机语言，全面掌握该语言的各种语句和函数的特性，使之按照软件工程论的方法用于一个应用程序中，这就是基本的编程技巧．书中含有近百个 C 语言的源程序，其中大部分程序编者都在 IBM-PC 及长城系列微型计算机上编译通过，有些程序有一定的实用价值，读者稍加修改或不用修改就可使用，在介绍这些例程的同时，也强调了一些程序设计经验、技术和风格．当然，提供这些程序的主要目的还是为了帮助读者学习

和理解 C 语言的基本成分，至于编程技巧，主要应当来自于经常的实践和不断的总结。编者相信，只要在学习这本书的同时，又能把书中的例程以及练习题在您的系统上一一编译通过，那么，您编程的技巧就会有一个大的提高。请记住：熟能生巧!

本书内容丰富，起点低，详略得当，实用性强。既可作为从事计算机工作的广大科技人员的自修读本或培训教材，也可作为大学非计算机专业学生的教材或参考书。

由于编者水平所限，书中有不妥及错误之处，诚请批评指正，编者不胜感激。

本书在编写过程中，得到了张忠智高级工程师以及编者同事们的大力帮助和支持，对此表示谢意。

编者　　1990.12.16.

目　录

第一章　程序结构及其运行

本章通过几个引例概括介绍 C 语言的基本语法规定，同时给出 C 语言源程序的基本结构，并说明如何在微机上编辑和运行 C 程序。内容包括：

- C 语言源程序的基本结构
- C 程序在 PC-DOS 操作系统下的实施过程
- C 程序在 XENIX 操作系统下的实施过程
- C 程序中使用汉字

第一节　程序的基本结构

学习一种新的语言，最好先读懂几个由这种语言写的程序，弄清该语言的基本规定，然后自己仿写几个程序，有条件再上机试一试您的程序（这一点很重要），这是学习计算机语言的最行之有效的途径。

先看一个最简单的 C 语言源程序。

例 1-1:

```
main( )
{
        printf ("This is a program\n");
}
```

这就是一个简单的但又是完整的 C 语言程序。这个程序的功能是在显示屏上输出：

```
This is a program
```

通过此例要弄清以下问题：

①一个 C 语言写的程序都由函数组成，一个程序中只能有一个以 main 命名的函数，称为主函数；

②main 后的括号 () 中可以有参数，也可以象该例那样没有参数，但无论有参或无参，括号均不能没有；

③稍后您可以知道，一个程序中可以有多个函数，但 main 函数总是第一个接受执行权，即 C 程序总是从 main 函数开始执行；

④接着 main() 后的一对花括号 { 和 }，是函数体开始和终了的标志符号，相当于某些高级语言程序中的保留字 begin 和 end，程序中的括号必须配对出现，花括号中的语句是函数的主体；

⑤printf 一句最后的分号"；"是语句终止号，不能缺省也不允许用其它符号代替，它是语句的必要组成部分，必需写；

⑥C 语言程序的保留字一律用小写英文字母书写（这一点务必请习惯用大小写不分的语言编程的读者注意！）；

⑦语句行不要行号，每行的起始位置无特殊规定，但一般把程序写成锯齿状以增加可读性.

另外，还应了解

④printf 是格式化打印语句的保留字；

⑤printf 后括号里的符号"\n"表示以后输出的内容要打印在下一行上，即代表换行.

最后，通过引例 1 的学习，得出 C 语言程序的最基本结构是

```
main()
 {
      语句序列
 }
```

让我们再看一个比较完整的程序.

例 1-2:

```
/ * 源程序文件名: prog2.c * /
main( )
{
    int    a, b, c, sum;
    scanf ("%d",&a);
    b = 10;   c = 20;
    sum = a+b+c;
    printf("sum = %d\n", sum);
}
```

如果输入 30，则输出就是 60.

通过引例 2 要了解以下问题:

①关于注释

注释语句由符号"/ * "开始，" * / "结束，注释没有长度和语句量的限制，中间如有多行，每行首部可以只写一个" * "；注释不允许嵌套，如果写出如下"注释"语句:

```
/ * ..... / * ..... * / ..... * /
```

那么，第二个"/ * "将被认为是注释的一部分，而第一个"/ * "将结束注释，其后的"注

释"内容可能导致编译时的一系列错误的出现.

提倡在程序中使用注释语句，这可增强程序的可读性．除了最简单的和最常见的函数外，所有函数都应在顶部加上注释，注明函数功能是什么、怎样调用以及返回什么值等．

②程序中的

```
int  a, b, c, sum;
```

一句是变量说明语句，int 是整型数据类型的说明符，它把后面的 a、b、c 和 sum 说明为整型变量;

③scanf 是格式读语句的保留字，其中%d 是格式控制字符，它表示要从终端读一个整型数，&a 表示把读来的数输给变量 a;

④程序体共有六条语句，运行时，从上到下、从左到右（因为一行可以写多条语句，如 b＝10;　c＝20;　就写在同一行中）逐条执行.

⑤b＝10;　c＝20;　以及 sum＝a+b+c;

都是赋值语句，功能是把"＝"号右端的值赋给左端的变量，关于赋值语句后面还要详细介绍.

如果某段程序在一个程序中要多次重复出现，这种情况下，一些高级语言采用过程或子程序的技术，C 语言则使用函数调用技术．如下例

例 1-3: 函数调用示例

```
main( )
{
   pr( );
}
pr( )
{
printf("Hello from ");
printf("function printf.\n);
}
```

这是由两个函数 main( )和 pr( )组成的程序，在执行时，首先执行 main( )函数，当执行到 pr( );一句时，程序就去调用 pr( )函数并执行该函数，执行完此函数后，自动返回到 main( )函数，并接着执行 pr( )语句后面的语句．一个 C 程序可以包含多个象 pr( )这样的被调用函数，所以程序更一般的结构应该是下面的形式:

```
main( )
{
    语句序列
}
f1( )
{
    语句序列 1
```

```
}
f2( )
{
    语句序列 2
}
.....
fn( )
{
    语句序列 n
}
```

其中 f1、f2、.....fn 是被调用的函数，它们在程序中的位置以及次序都是无关紧要的.

上面三个引例，源程序较短，都放在一个文件中．如果程序很长，这将增加程序的编译时间，C 语言允许把程序分成几块，分别放在不同的文件中，并且分别编译．分别编译的优点是当您只改变一个文件中的代码时，您不必重新编译整个程序，而只要编译需要修改的那一个文件即可．另外，当您搞了一个时期的程序设计后，您可能积累了一些有用的函数，您就可以把它们分别放在不同的文件中，以便在其它程序中随时调用．

下面是一个分别编译的例子．

例 1-4：源程序分别被编辑在两个文件中．

名为 ABC.C 的第一个文件内容如下:

```
main( )
{
   pr( );
}
```

名为 XYZ.C 的第二个文件内容如下:

```
pr( )
{
   printf("Hello from ");
   printf("function printf.\n);
}
```

这实际是把引例 3 同一个程序中的两个函数分放在两个文件中，编译时应分别进行编译，然后再连接生成一个可执行文件．在执行时，第一个文件将调用第二个文件，第二个文件 XYZ 相当于是一个过程或子程序，而第一个文件 ABC 相当于主程序．

至此，您对 C 程序应当有了一个概括地了解了．

第二节　PC-DOS 的基本知识

C 语言具有很好的移植性，所以，几乎所有的微型计算机都可以运行 C 语言。而要使用一台计算机，首先要掌握它的操作系统。常见的 IBM 系列和国产长城系列微型计算机上，大都使用 PC-DOS（即 Personal Computer Disk Operating System）操作系统。这一部分内容不属于 C 语言范畴，但为了编辑和运行 C 语言程序，又应当了解。如果您已经熟悉 PC-DOS，则可跳过这几节而直接去学习第二章的内容。如果您还不熟悉 PC-DOS，那么您至少应当掌握以下几方面的内容。

一、文件

文件是有关信息的集合。例如，一个 C 语言源程序、操作系统的某一个命令等都是作为一个一个文件存放在软盘或硬盘上的。存放在盘上的文件应该有自己的名字，文件的名字一般由文件名和扩展名两部分组成，其中文件名由 1-8 个字符组成，扩展名以园点"."符开始，后跟 1-3 个字符。扩展名可有可无，但有些文件名必须有扩展名。

关于文件名应注意：

①文件名和扩展名所能使用的字符并非任意的，一般 26 个英文字母和 0-9 数字字符都可用作文件名，详细规定请参考有关 DOS 手册。

②存放在同一盘上的不同内容的文件，都应当而且必须使用不同的文件名。

③给文件命名应尽量使用有一定含义的名字，以便容易识别，如一个具有求和功能的 C 语言程序，可以命名为 PROG1.C，但不如命名为 SUM.C 便于识别。

④通配符？和 *

通配符？和 * 也叫多义字符，如果文件名中含有一个？，那么该问号位置上可认为是任一字符，如

prog?.c

可以代表盘上的下列文件：

prog1.c
prog2.c
prog3.c
prog9.c
progx.c
proga.c

等等。

如果文件名中含有一个 *，那么从该星号位置起，后面可认为是任何合法的字符，如

p * .c

可以代表盘上以字母 P 打头且扩展名为.C 的所有文件。

操作系统对文件的存访都是通过文件的名字实施的。

二、 PC-DOS 的启动方法

要使用计算机，就要启动 DOS. DOS 通常被存放在软盘上，我们称它为 DOS 盘。有两种启动 DOS 的方法。

(一) 计算机未加电时，按以下方法启动:

①插 DOS 盘于 A 驱动器中，并关好门;

②再给外围设备如显示器、打印机（如果要用的话）等加电;

③最后给主机加电，稍等便可进入 DOS 状态。

(二) 计算机已经加电，可按以下方法启动:

①把 DOS 盘插入 A 驱动器中，并关好门;

②按下 CTRL+ALT+DEL 键并同时放开，稍等便可进入 DOS 状态。

进入 DOS 状态的标志是屏幕左边出现 DOS 提示符"A>"，这时就可使用 DOS 命令。

如果把 PC-DOS 装入硬盘，也可从硬盘启动，方法与上述过程类似，不同的是: 这时，A 驱动器的门必需是打开的。由硬盘启动时，系统的提示符变为"C>"。

三、 DOS 常用命令

(一) 列目录命令 DIR

格式一　A>DIR　　纵向显示 A 盘上的全部文件名目录

格式二　A>DIR/W　横向显示当前盘 A 盘上的全部文件名目录

格式三　A>DIR/P　按页显示 A 盘上的全部文件名目录

利用 DIR 命令还可显示某一盘上的某一个或某一类文件的目录（这要用到多义字符)。

(二) 拷贝命令 COPY

①同名拷贝:

A>COPY　SUM.C　B:　PROG1.C

功能是把 A 盘上的 SUM.C 文件拷贝到 B 盘上，文件名不变。

②改名拷贝，如

A>COPY SUM.C B:　PROG1.C

功能是把 A 盘上的 SUM.C 文件拷贝到 B 盘上，文件名改为 PROG1.C。

(三) 删除命令 DEL

A>DEL　SUM.C

功能是把当前盘上的无用的文件 SUM.C 从盘上删除。当您要删除盘上所有文件时，可用下面方法

A>DEL　*.*

当您敲了回车键后，DOS 会提示:

Are　you　suer? (Y/N)

询问您是否真要删除，如果您改变主意，不想删除时，可回答 N 则撤销原来的命令。

(四) 显示打印命令 TYPE

A>TYPE　B:　PROG1.C

功能是在屏幕上显示 B 盘上的 PROG1.C 文件的内容。如果上述命令敲完后，在按回车键之前，先按一下 CTRL+P 键，再按回车键，该文件的内容将在打印机上打印出来，当然你的打印机事先已经接通。注意，该命令只对正文文件即 ASCII 码文件起作用，否则显示出不可识别的字符。

（五）建立子目录命令 MKDIR（简称 MD）

MD 命令的一般格式是

md　子目录名

子目录名就是您要建立的子目录名，包括盘符。如

A>MD　C:＼TURBOC

功能是在 C 盘上建立一个名为 TURBOC 的子目录。

（六）进入子目录命令 CHDIR（简称 CD）

A>CD　C:＼TURBOC

功能是进入 C 盘上的子目录 TURBOC，即把子目录 TURBOC 改变为当前目录。

（七）删除子目录命令 RMDIR（简称 RD）

RD 命令的一般格式是

rd　子目录名

子目录名就是您要删除的子目录名，包括盘符。如

A>RD C:＼TURBOC

功能是删除 C 盘上的子目录 TURBOC。但要注意，当您要删除一个子目录时，该子目录必须是空目录，即如果该目录里还存有文件或子子目录，您必须事先把它们都删除掉，否则您就不能删除该目录。

第三节　在 PC-DOS 下开发 C 语言程序的过程

写在纸上的程序，必须经过在计算机上编辑、编译、连接以后才能运行。在 PC-DOS 下，开发一个 C 语言程序，要经过以下步骤：

一、 编辑

所谓编辑，就是使用系统提供的编辑程序将写在纸上的程序以文件的形式存入磁盘（软盘或硬盘）。PC-DOS 提供的编辑程序叫行编辑程序，存放在 DOS 盘上，文件名是 EDLIN.COM。下面以例 2 求和程序为例，说明如何用行编辑程序把纸上的程序存到软盘上。

第一步　启动 DOS，屏幕出现 DOS 提示符"A>"

第二步　进入行编辑状态，应按下面的格式：

A>EDLIN　B: SUM.C

说明：①DOS 盘上必须有 EDLIN.COM 程序；

②准备把纸上的程序以 SUM.C 为名存放到 B 盘上，这里文件名是必须写的，如果

您忘了写而敲了回车键，行编辑程序会提示您写上文件名，注意 C 源程序的扩展名必须是".C";

③文件名敲完后按一下回车键，屏幕显示"New file"即新文件的字样，并出现行编辑的提示符"*"，说明系统已经处于行编辑的命令状态。可开始输入行编辑的命令。

第三步键入 i 并回车则进入行编辑的插入状态，可开始编辑源程序:

```
*i
    * 1: / * 源程序文件名: prog2.c * /
    * 2:main( )
    * 3:{
    * 4:    int    a, b, c, sum;
    * 5:    scanf ("%d",&a);
    * 6:    b=10;   c=20;
    * 7:    sum=a+b+c;
    * 8:    printf("sum=%d\n", sum);
    * 9:}
    * 10:c
*E
```

第 9 行程序结束，当第 10 行出现以后按 F6 或 CTRL+C 退出插入状态，回到"*"提示符下，键入存盘命令 E 并回车，退回 DOS 状态。这样该程序就以 SUM.C 为名，存入 B 盘上了。

如果程序在输入过程中有错，可在行编辑状态修改，如何修改，请参阅有关资料。

二、 编译

将 C 语言源程序存入软盘后，还要进行编译。编译的目的是把源程序转换成能直接在计算机上运行的目标代码。编译由编译程序来完成，不同版本的 C 语言编译程序的编译过程可能有所不同，有只要一次编译的，也有需要几次编译的。要进行两次编译的，编译程序名为 C1.EXE 和 C2.EXE。仍以前面的例子为例，说明编译的过程:

第一步　把 C 语言编译盘插入 A 驱动器，并关好门，把存有源程序 SUM.C 的盘插入 B 驱动器，关好门，然后敲入如下命令:

A>C1

并回车，屏幕首先显示 C 语言编译程序的版本号，接着显示四项提问，你必须一一给出正确的回答，如果回答有错，在未按回车键之前，可用退格键修改，如果已经按了回车键，则必须按 CTRL+Break 退回 DOS 提示符下，重新开始。四项提示的回答如下:

Source filename (.pas): b:sum	键入要编译的文件名
Object filename (sum.obj): b:	键入目标程序文件名，用默认名，存入B盘。下面两项提问均可按这样回答:
Source listing (NUL.LST): b:	询问是否要建立可打印文件
Object listing (NUL.COD): b:	询问是否要建立前后对照文件

源程序如果没有错误，第一次编译成功，出现如下提示:

Pass One No Errors Detected.

这时就可立即进行第二次编译，命令和屏幕显示如下：

A>C2

Code Area Size=#00ab (171)

Cons Area Size=#0030 (48)

Data Area Size=#000A (10)

Pass Two No Errors Detected.

最后一行提示说明第二次编译成功，这样就可开始连接工作。

三、 连接

编译成功的标志是在 B 盘上生成一个扩展名为.OBJ 的目标文件，这可用 DIR 命令查看．这个目标文件还不能直接运行，必须经过连接装配，再生成一个可执行文件，才能运行．事实上连接程序起两个作用，一是把交给它的所有文件（如多个目标代码文件以及库函数等）连成一个程序；二是解决外部引用问题．所谓外部引用是指一个文件中的代码引用另一个文件中的代码。连接需要用连接程序 LINK.COM．启动连接程序可用下面两种方法中的任一种。

方法一是控制台回答方式：在 DOS 提示符下敲入连接程序的名字：

A>LINK

假定连接程序就存放在当前盘上，按回车后，首先显示连接程序的版本号，接着有四项提问，你必须依次给出正确的回答，回答的方法与编译时类似，我们以 SUM 文件为例具体说明如下：

GW Personal Computer Linker	
Version 2.10	这是 LINK 的版本号
Object Modules [.OBJ]: b:sum	询问目标模块名，回答 B:SUM
Run File [A:SUM.EXE]: B:	询问运行文件名，默认为 A:SUM.EXE 回答 B:是要把 SUM.EXE 存在 B 盘
List File [NUL.MAP]:	询问列表文件名，一般不建立，可直接回车
Libraries [.LIB]:	询问编译程序包中的库文件在哪个盘上， 在默认盘上，故直接回车

如果连接出错，则要重新调用 EDLIN 来修改，然后重新编译，再重新连接，直至无错．连接成功的标志是无错误信息而直接出现 DOS 操作系统的提示符"A>"。

方法二是命令行方式，格式如下：

A>LINK objlist, runfile, mapfile, liblist;

其中：

objlist: 是目标模块表，当有多个目标模块时，各模块之间用空格或加号"+"隔开

runfile: 是您给运行文件起的名字

mapfile: 是您给连接图象起的名字，

liblist: 是要用的库文件表，用空格或加号"+"隔开

后面两项可以不写，这样，连接程序就取默认值，但分隔符逗号必须写上．如

A > LINK B:SUM.OBJ,B:SUM.EXE,,;

同样可以得到一个可执行文件 SUM.EXE.

四、 运行

连接的结果是在 B 盘上生成一个可执行文件：SUM.EXE，运行很简单，只要在

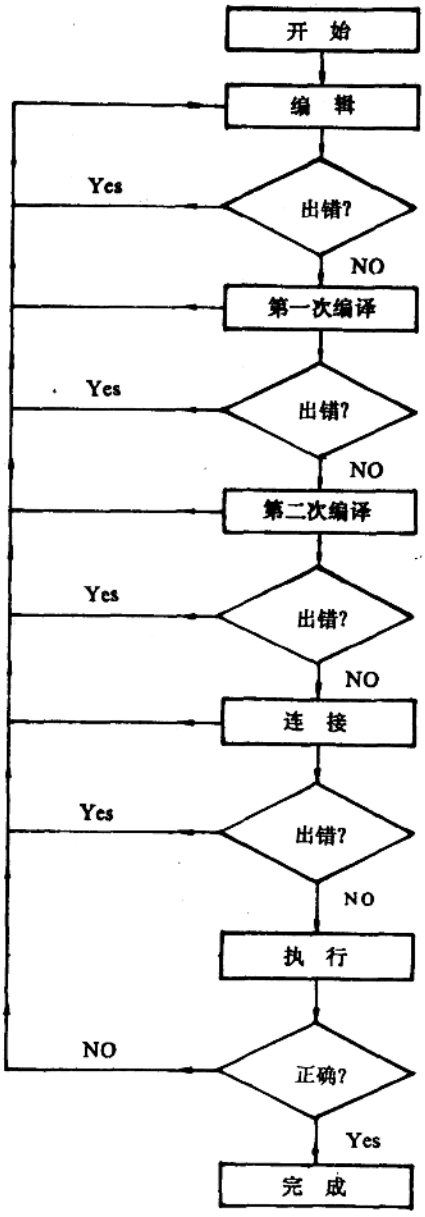


图 1

DOS 提示符下键入文件名 SUM 并回车即可．前面的操作过程都把 A 盘确定为默认盘，为了方便，现在把默认盘改为 B 盘:

A > B:

即只要在 A > 后键入驱动器号 B 再加一个冒号": "并回车即可．这时提示符改为"B > "．

下面就来运行 SUM 程序:

B > SUM

按回车后，光标移到下一行，等待您输入数据，比如您输入下面两数

65　−20

回车后，运行结果便显示在屏幕上:

65　　−20　　45

即 65 加−20 的和为 45，程序运行结果与予期的结果相同，说明程序是正确的．如果运行结果与予期的不同，说明程序有错，就需要重新修改程序，重新编辑，重新编译，重新连接，重新运行，直至正确无误为止。

对多个源文件程序如 ABC.C 和 XYZ.C 需先分别编译，其编译方法与前面相同，然后再把编译生成的目标文件如 ABC.OBJ 和 XYZ.OBJ 进行连接，用命令行方式象下面这样连接:

A > LINK　B: ABC. OBJ+B: XYZ. OBJ, B: PROG. EXE,,;

回车后,就可在 B 盘上生成一个名为 PROG.EXE 的可执行文件．

综上所述，可以看出，在 PC−DOS 下，开发一个 C 语言程序，其基本过程如图 1 所示:

第四节　在 XENIX 下开发 C 语言程序的过程

在 XENIX 下开发一个 C 语言程序，要比在 PC−DOS 下容易实现，其步骤如下．

一、 进入 XENIX 操作系统．

1) 注册

XENIX 是多用户操作系统，任何时候都有可能是几个人同时在使用系统，因此，只有合法的用户才能有效地使用 XENIX 系统．所谓合法用户是指经过正确注册的用户，即当您打开一台与系统相连的终端，准备进入和使用系统，屏幕显示

l o g i n:

时，如果您能正确回答用户名字（这个名字是由系统管理员事先为您建立的一个独立的用户目录)，那么您就有可能进入和使用系统了．说有可能是注册过程还没有结束，系统还要询问口令

p a s s w o r d:

这个口令是系统管理员为您建立用户目录时您自己告诉系统的，只有回答正确才可以进入和使用系统。

2）提示符

正确回答用户名和口令后，系统显示一个美元符和光标如下：

$ _

这时，您就可以使用 XENIX 系统了。

3）注销

当您使用完系统后，一定要记住从系统退出。退出系统的方法很简单，只要敲一下＜Ctrl+d＞键就可以了。退出系统的标志是屏幕重新显示：

login:

字样，这时您就可以关机离开。

二、 shell 命令

进入系统之后，终端即处于终端命令解释程序 shell 的控制之下。以后您输入的命令都是经过 shell 解释执行的。shell 命令的一般形式为：

$ 命令名　任选项.....　参数.....

其中 $ 是 shell 的提示符，命令名是您要 shell 执行的命令或可执行文件的名字（可带有路径名），任选项用来对命令的功能作进一步的描述，一个命令可以不带或带有多个任选项，参数一般与命令的处理对象有关，其个数随具体命令而异。

1）改变当前目录命令 cd＼

$ cd＼　　表示回到根目录

$ cd＼目录名　表示转到目录名指定的目录

$ cd..＼　　表示向前退回一级目录

2）显示当前目录 pwd

$ pwd

回车后显示出您当前所处于的目录及路径名。

3）列文件目录内容命令 ls

$ ls　任选项　目录名

实际上，目录名也是任选的，不选时列出当前目录。常用任选项是-1，如

$ ls　-1

回车后显示器屏幕上每一行列出一个文件或一个目录，并有其它一些参数指明该文件或目录的一些性质。

4）删除文件命令 rm

$ rm　文件名

回车后删除了文件名指定的那个不想再保留的文件。

5）显示文件内容的命令 cat

$ cat　文件名

回车后在屏幕上显示出文件名指定的文件的内容。该命令还有联接的作用，如

$ cat　文件一　文件二

回车后，屏幕上先显示文件一的内容，接着显示文件二的内容。

6）文件复制命令 cp