

一. 从文件输入开始谈起

1. 文件的定义：**文件**是存储器中储存信息的区域，它是一种**抽象**。
2. 从编译器想到的：编译器要打开相应的纯文本文件，读取并处理文件内容，最后关闭文件。
由此可见，某些程序需要访问指定的文件。
3. 底层I/O和标准I/O：
 - ①**什么是底层I/O**？操作系统本身就有处理文件的方法——调用它自己的函数；如果C编译器也要处理文件，一个很自然的想法就是直接调用操作系统底层的函数。这种操作系统本身的函数就叫底层I/O。
 - ②**使用底层I/O有什么问题**？概括地讲就是不同系统间实现的不同。比如：**存储文件的方式**不同，文件有一些相关的信息——我们叫他文件的属性(大计基中叫它文件声明)，有些系统把文件的属性放在文件内；而还有些系统就把他们分开放；又比如：**文件中一行的结尾**也会有所不同，windows使用\r和\n(回车换行)，而Unix和Mac OS则使用\n,其他系统竟然还有使用\r的。再文章的后面还会提到另一种不同，就是**文件结尾**的不同。
 - ③**标准I/O的引入**：（原书并未涉及到标准I/O和底层I/O之间的关系）标准I/O从更高层面上处理文件，这样我们就不用担心不同系统之间的差异了。
4. 流：不同种类间的输入相差巨大，为了能够统一处理，C语言决定引入一种叫做“流”的中间量，然后把这种流附到键盘，屏幕或文件中。

二. 详谈文件结尾

1. 操作系统要用某种方式判断文件的开始和结束。这主要有两种方法：
 - ①**内嵌字符标记**：使用一个特殊的字符标记文件结尾，比如IBM-DOS,MS-DOS,CP/M使用Ctrl+Z（显示为^Z）标记文件结尾。
 - ②**利用文件的大小**：这个比较好理解。这种方法的优点就是可以包括Ctrl+D字符，新版的DOS和UNIX使用这种方法处理所有的文件。
2. 虽然不同的操作系统处理文件结尾的方式并不一样，好在C语言在读取文件结尾时一定会返回一个特殊的值，即EOF。scanf在检测文件结尾时也会返回EOF。这个值通常定义在stdio.h中。
3. EOF存在的意义：本章内容并不是文件相关，而是要读者**用键盘模拟文件结尾**。注意：
 - ①要把输入和输出设备视为存储设备上的普通的，自动打开的文件。
 - ②不同系统用键盘模拟文件结尾的方式不同。许多微型计算机都把**一行开始处**的Ctrl+Z识别为文件结尾信号，一些系统把**任意位置**的Ctrl+Z视为文件结尾信号。还有些系统：如大多数UNIX和Linux系统，把**一行开始处**的Ctrl+D视为文件结尾信号。

4. EOF的一些问题：EOF作为一种特殊的字符，必须要避开标准字符集，很多实现把EOF定义为一个负值（因为标准字符集的标号都是正的），那么一个无符号char是无法存贮这个EOF的值的，我建议使用signed char类型（虽然书上说使用int，但是这样占用的空间更大，而且让char失去了意义）。

5. EOF的历史来源：EOF的概念是在使用文本界面的命令行环境中产生的，在这种环境中，用户通过击键与程序交互，**由操作系统生成EOF信号**。

三. 缓冲输入

1. **缓冲与无缓冲**：回显用户输入的字符后立即重复打印该字符是无缓冲输入。不立即打印是缓冲输入。

2. **回显与无回显**：回显输入意味着用户输入的字符直接显示在屏幕上（如CMD），无回显输入意味着击键后对应的字符不显示（如Telnet）。

3. 缓冲的好处：

①将若干字符作为一个块进行传输比逐个发送这些字符更**节约时间**。

②如果用户打错字符，可以进行**修正**。

4. 行缓冲I/O和完全缓冲I/O：

①行缓冲：出现换行符时**刷新缓冲区**，如键盘输入。

②完全缓冲：当缓冲区被填满才**刷新缓冲区**，如文件输入。

四. 文件重定向

1. 一般而言，C程序使用**标准I/O包**查找标准输入作为输入源。现代计算机还允许程序从一个文件中查找输入，而不是从键盘。

2. 程序由两种方法使用文件：

①在C程序中使用函数来打开，写入，关闭文件。

②使用专门的用键盘和文件交互程序，重定向输入输出（本章着重讲这种）。

3. 重定向输入/输出：

①输入：XXX<XXX（注意“<”最好不要有空格）

②输出：XXX>XXX

4. 组合重定向：就是给可执行文件提供输入和输出的文件（由此可见重定向运算符不能连接多个程序）。格式：echo_eof<mywords>savewords（命令与重定向运算符无关），注意不能用重定向运算符读取多个文件的输入，也不能把输出定向至多个文件。

5. 其他符号：“>>”：将数据添加到现有文件的末尾。“|”：管道运算符，将一个文件的输入连接到另一个文件的输入。

6. 要点：

①重定向必须包含可执行文件（内含机器语言指令）

②重定向本质上是把stdin（或stdout）从输入设备（或输出设备，这里是显示器）

赋给文本文件③重定向后原文件的内容将被覆盖（当然也有不覆盖的方法，请见

5）；如果文件原来不存在，就创建一个新的文件，口诀总结：“**有则重写，无则创建**”。

④重定向是一个命令行的概念，在vs中，默认设置是把可执行文件放在项目文件夹的子文件夹中。

五. 知识拓展

1. 类型降级时，会给出警告（当然要激活较高等级的编译器）。

2. 苹果MAC OS X运行在Unix上，可以使用Terminal应用程序来使用UNIX命令行模式。

3. Linux中，一个相当于命令行的应用叫做ditto。

4. Winodws的命令行提示时模仿旧式DOS命令行环境的。（BUG:原书叫做window了。。）