

第10章 对文件的输入输出

10.1 C文件的有关基本知识

10.2 打开与关闭文件

10.3 顺序读写数据文件

10.4 随机读写数据文件

10.5 文件读写的出错检测

10.1 C文件的有关基本知识

10.1.1 什么是文件

10.1.2 文件名

10.1.3 文件的分类

10.1.4 文件缓冲区

10.1.5 文件类型指针

10.1.1 什么是文件

➤ 文件有不同的类型，在程序设计中，主要用到两种文件：

(1) 程序文件。包括源程序文件(后缀为.c)、目标文件(后缀为.obj)、可执行文件(后缀为.exe)等。这种文件的内容是程序代码。

(2) 数据文件。文件的内容不是程序，而是供程序运行时读写的数据，如在程序运行过程中输出到磁盘(或其他外部设备)的数据，或在程序运行过程中供读入的数据。如一批学生的成绩数据，或货物交易的数据等。

➤ 本章主要讨论对数据文件的操作

10.1.1 什么是文件

- 以前程序数据的输入和输出，从终端的键盘输入数据，运行结果输出到终端显示器上
- 常常需要将一些数据输出到磁盘上保存起来，以后使用
- 这就要用到磁盘文件
- 操作系统把各种设备都统一作为文件处理
- 从操作系统的角度看，每一个与主机相联的输入输出设备都看作是文件。例如，
 - ◆ 终端键盘是输入文件
 - ◆ 显示屏和打印机是输出文件

10.1.1 什么是文件

- “文件”指存储在外部介质上数据的集合
 - ◆一批数据是以文件的形式存放在外部介质上的
 - ◆操作系统是以文件为单位对数据进行管理
 - ◆想找存放在外部介质上的数据，先按文件名找到所指定的文件，然后再从该文件读数据
 - ◆要向外部介质上存储数据也必须先建立一个文件（以文件名作为标志），才能向它输出数据
- 输入输出是数据传送的过程，数据如流水一样从一处流向另一处，因此常将输入输出形象地称为流(stream)，即数据流。流表示了信息从源到目的端的流动。

10.1.1 什么是文件

- 输入：指数据从文件流向计算机内存
- 输出：指数据从计算机内存流向文件
- C语言把文件看作是一个字符（或字节）的序列。一个输入输出流就是一个字符流或字节（内容为二进制数据）流。
- 文件要有一个唯一的文件标识，以便用户识别和引用。
- 文件标识包括三部分：
 - (1)**文件路径
 - (2)**文件名主干
 - (3)**文件后缀

10.1.2 文件名

文件路径

文件名主干

文件扩展名

- 文件路径表示文件在外部存储设备中的位置。如：

D: \Code\C\example10-1.c

文件名

- ◆ 表示**example10-1.c**文件存放在**D**盘中的**Code**目录下的**C**子目录下面

- 根据数据的组织形式，数据文件可分为**ASCII文件**和**二进制文件**。

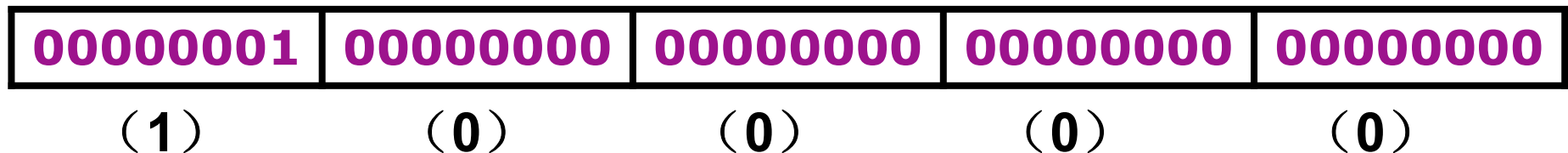
- ◆ **ASCII**文件又称文本文件，每一个字节放一个字符的**ASCII**代码

- ◆ 数据在内存中是以二进制形式存储的

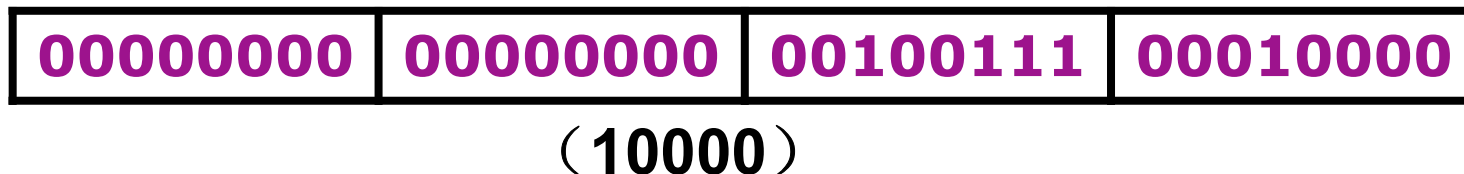
10.1.3 文件的分类

- 字符一律以**ASCII**形式存储
- 数值型数据既可以用**ASCII**形式存储，也可以用二进制形式存储
 - ◆ 如有整数**10000**，如果用**ASCII**码形式输出到磁盘，则在磁盘中占 5 个字节(每一个字符占一个字节)，而用二进制形式输出，则在磁盘上只占**4**个字节

ASCII形式



二进制形式

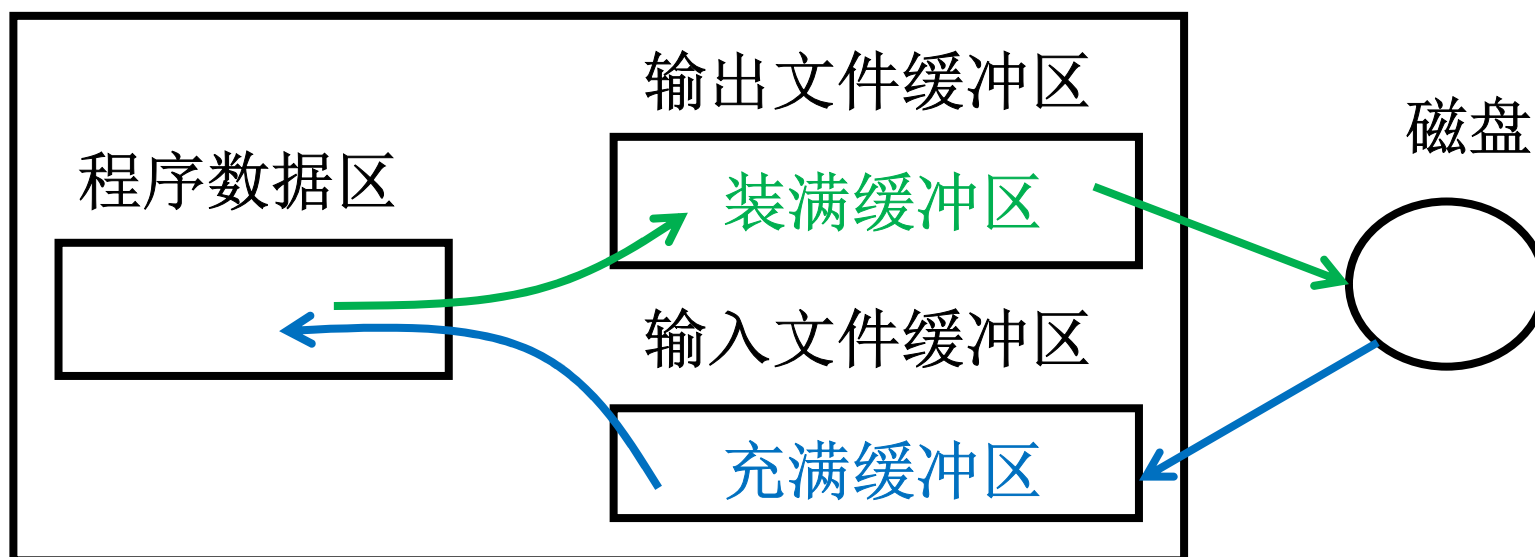


10.1.4 文件缓冲区

- C标准采用“缓冲文件系统”处理数据文件
- 所谓缓冲文件系统是指系统自动地在内存区为程序中每一个正在使用的文件开辟一个文件缓冲区
- 从内存向磁盘输出数据必须先送到内存中的缓冲区，装满缓冲区后才一起送到磁盘去
- 如果从磁盘向计算机读入数据，则一次从磁盘文件将一批数据输入到内存缓冲区（充满缓冲区），然后再从缓冲区逐个地将数据送到程序数据区（给程序变量）

10.1.4 文件缓冲区

➤从内存向磁盘输出数据



➤从磁盘文件向计算机内存读入数据

10.1.5 文件类型指针

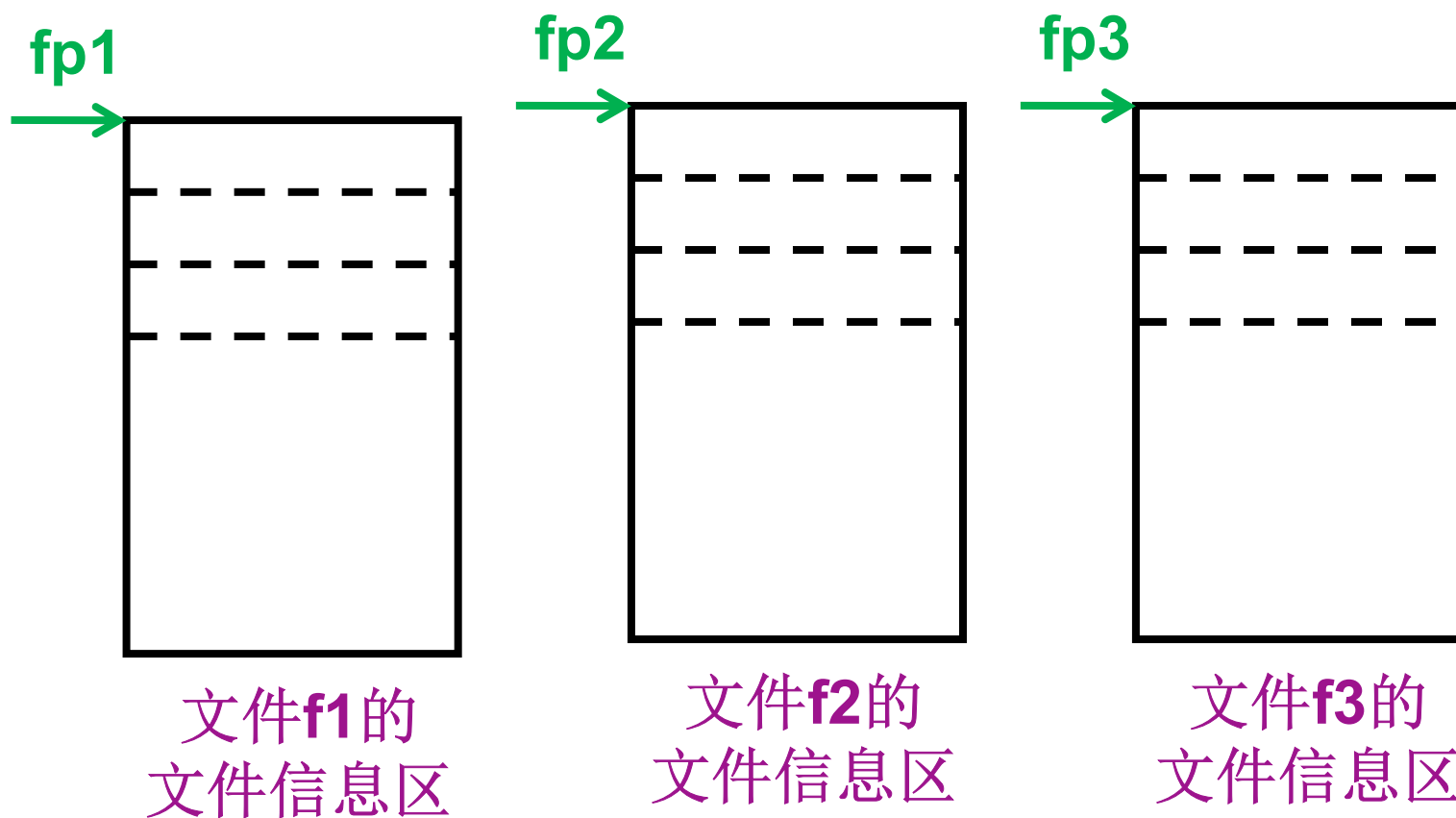
- 缓冲文件系统中，关键的概念是“文件类型指针”，简称“文件指针”
 - ◆ 每个被使用的文件都在内存中开辟一个相应的文件信息区，用来存放文件的有关信息（如文件的名字、文件状态及文件当前位置等）
 - ◆ 这些信息是保存在一个结构体变量中的。该结构体类型是由系统声明的，取名为**FILE**
- 声明**FILE**结构体类型的信息包含在头文件**<stdio.h>**中
- 一般设置一个指向**FILE**类型变量的指针变量，然后通过它来引用这些**FILE**类型变量

```
struct _iobuf {  
    char *_ptr;        //文件输入的下一个位置  
    int _cnt;          //当前缓冲区的相对位置  
    char *_base;      //文件基地址(起始位置)  
    int _flag;         //文件标识  
    int _file; //当前打开的文件序号 stdin 0 stdout 1 stderr 2  
    int _charbuf;     //检查缓冲区状况,如果无缓冲区则不读取  
    int _bufsiz;      //缓冲区大小  
    char *_tmpfname; //临时文件名  
};
```

```
typedef struct _iobuf FILE;
```

10.1.5 文件类型指针

```
FILE *fp1,*fp2,*fp3;
```



10.2 打开与关闭文件

10.2.1 用**fopen**函数打开数据文件

10.2.2 用**fclose**函数关闭数据文件

10.2.1 用fopen函数打开数据文件

- 对文件读写之前应该“打开”该文件，在使用结束之后应“关闭”该文件。
- 所谓“**打开**”是指为文件建立相应的信息区(用来存放有关文件的信息)和文件缓冲区(用来暂时存放输入输出的数据)。
- 在编写程序时，在打开文件的同时，一般都指定一个指针变量指向该文件，也就是建立起指针变量与文件之间的联系，这样就可以通过该指针变量对文件进行读写
- 所谓“**关闭**”是指撤销文件信息区和文件缓冲区

10.2.1 用fopen函数打开数据文件

- **fopen**函数的调用方式为：
fopen(文件名,使用文件方式);
- 例如：
fopen("a1","r");
 - ◆表示要打开名为“**a1**”的文件，使用文件方式为“读入”
 - ◆**fopen**函数的返回值是指向**a1**文件的指针
- 需要将**fopen**函数的返回值赋给一个指向文件的指针变量，此后才能通过此指针变量对文件进行操作。如：
FILE *fp;
fp=fopen("a1","r");
 - ◆**fp**和文件**a1**关联，**fp**指向了**a1**文件

mode	meaning	附加说明
r	以读的方式打开文本文件	文件不存在打开失败
w	以写的方式打开文本文件	文件不存在创建 文件已存在截为空
wx	以写的方式创建文本文件	文件不存在创建 文件已存在打开失败
a	以向文件末尾追加内容的方式打开文本文件，不可读	文件不存在创建
rb	以读的方式打开二进制文件	文件不存在打开失败
wb	以写的方式打开二进制文件	文件不存在创建 文件已存在截为空
wbx	以写的方式创建二进制文件	文件不存在创建 文件已存在打开失败
ab	以向文件末尾追加内容的方式打开二进制文件	文件不存在创建
r+	以更新(包括读和写)的方式打开文本文件	文件不存在打开失败
w+	以更新的方式打开文本文件	文件不存在创建 文件已存在截为空
w+x	以更新的方式创建文本文件	文件不存在创建 文件已存在打开失败
a+	以向文件末尾追加内容的方式打开文本文件，更新	文件不存在创建
r+b or rb+	以读的方式打开二进制文件	文件不存在打开失败
w+b or wb+	以写的方式打开二进制文件	文件不存在创建 文件已存在截为空
w+bx or wb+x	以写的方式创建二进制文件	文件不存在创建 文件已存在打开失败
a+b or ab+	以向文件末尾追加内容的方式打开二进制文件	文件不存在创建

➤ 说明：

程序中可以使用的**3**个标准的流文件：标准输入流、标准输出流、标准出错输出流。

◆ 系统已对这**3**个文件指定了与终端的对应关系

◆ 标准输入流是从终端的输入

◆ 标准输出流是向终端的输出

◆ 标准出错输出流是当程序出错时将出错信息发送到终端

➤ 程序开始运行时系统自动打开这**3**个标准流文件。因此，程序编写者不需要在程序中用**fopen**函数打开它们。所以以前我们用到的从终端输入或输出到终端都不需要打开终端文件。

10.2.2 用fclose函数关闭数据文件

- 关闭文件用**fclose**函数。**fclose**函数调用的一般形式为

fclose(文件指针);

例如：

fclose (fp);

- 如果不关闭文件将会丢失数据。