School of Economics ang Management, Beihang University

Introduction to C Programming

Jichang Zhao

jichang@buaa.edu.cn

Data Files

Objectives

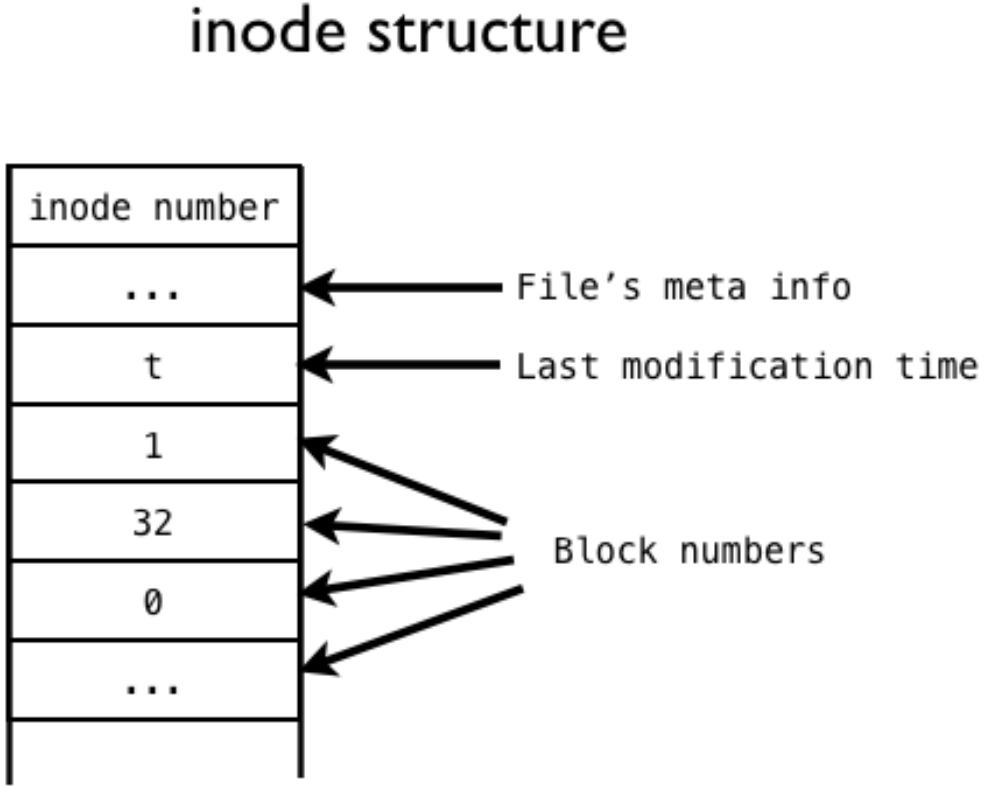
- Declaring, Opening, and Closing File Streams
- Reading from and Writing to Text Files
- Random File Access
- Passing and Returning Filenames
  - 参考教材 P396

补充：文件系统

- 文件系统是操作系统用于明确存储设备或分区上文件的方法和数据结构，也即在存储设备上组织文件的方法
  - 文件系统对文件存储设备的空间进行组织和分配，负责文件存储并对存入的文件进行保护和检索
  - 负责为用户建立文件，存入、读出、修改、转储文件，控制文件的存取，当用户不再使用时撤销文件等。

文件系统

- FAT32
  - U盘
- NTFS
  - Windows
- Ext
  - Linux
- HFS+
  - Mac OS



Declaring, Opening, and Closing File Streams

- To store and retrieve data outside a C program, you need two items:
  - A file
  - A file stream
  - 补充：流可以表示任意的输入或输出，文件只是流的一种
  - 补充：标准输入/输出流
  - 补充：重定向：> 输出重定向 <输入重定向
  - **DEMO: out.c, in.c**

表22-1 标准流

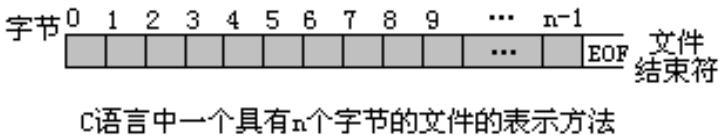
文件指针	流	默认的含义
stdin	标准输入	键盘
stdout	标准输出	屏幕
stderr	标准错误	屏幕

Files

- **File:** collection of data that is stored together under a common name, usually on a disk
- Each file has a unique filename, referred to as the file's **external name**
  - For example, `prices.dat` and `info.txt`
- There are two basic types of files
  - **Text files** (also known as **character-based files**)
    - store each individual character, such as a letter, digit, dollar sign, decimal point, and so on, using an individual character code
  - **Binary files**
    - use the same code as your computer processor uses internally for C's primitive data types
    - Advantage: speed and compactness

File Streams

- **File stream**
  - one-way transmission path used to connect a file stored on a physical device to a program



C语言中一个具有n个字节的文件的表示方法

- **Input file stream**
  - receives data from a file into a program
- **Output file stream**
  - sends data to a file
- P377编程注解

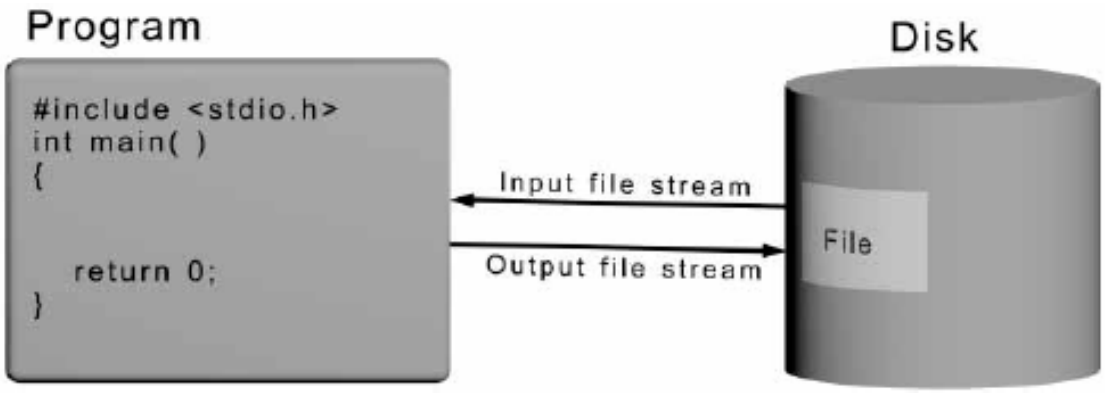


Figure 10.1 Input and output file streams

Declaring a File Stream

- For each file that your program uses, a file stream must be named (declared) and created (opened)
- Naming a file stream is accomplished by declaring a variable name to be of type `FILE`
  - `FILE` is a structure type
  - `FILE *inFile;` (文件指针)
    - * is necessary
    - Name is selected by programmer and internal to the program
    - The `FILE` data structure is declared in `stdio.h`

```
struct _iobuf {
        char *_ptr;
        int   _cnt;
        char *_base;
        int   _flag;
        int   _file;
        int   _charbuf;
        int   _bufsiz;
        char *_tmpfname;
        };
typedef struct _iobuf FILE;
```

Opening a File Stream

- Opening a file stream (or opening the file)
  - Establishes the physical communication link between the program and the data file
  - Equates a specific external filename to the name declared in the `FILE` declaration statement
- Use `fopen()` (declared in `stdio.h`)
  - `outFile = fopen("prices.bnd","w");`
  - `fileOut = fopen("prices.dat", "wb");`
  - `inFile = fopen("prices.bnd","r");`
    - If a file opened for reading does not exist, `fopen()` returns the NULL address value
    - 目录分隔符: / ,windows上如果用\，则需要写为\\进行转义

Opening a File Stream

表22-2 用于文本文件的模式字符串

字 符 串	含 义
"r"	打开文件用于读
"w"	打开文件用于写（文件不需要存在）
"a"	打开文件用于追加（文件不需要存在）
"r+"	打开文件用于读和写，从文件头开始
"w+"	打开文件用于读和写（如果文件存在就截去）
"a+"	打开文件用于读和写（如果文件存在就追加）

表22-3 用于二进制文件的模式字符串

字 符 串	含 义
"rb"	打开文件用于读
"wb"	打开文件用于写（文件不需要存在）
"ab"	打开文件用于追加（文件不需要存在）
"r+b"或者"rb+"	打开文件用于读和写，从文件头开始
"w+b"或者"wb+"	打开文件用于读和写（如果文件存在就截去）
"a+b"或者"ab+"	打开文件用于读和写（如果文件存在就追加）

Closing a File Stream

- A file stream is closed using `fclose()`
  - `fclose()` breaks the link between the file's external and internal names, releasing the internal file pointer name, which can then be used for another file
  - `fclose(inFile);`
- Because all computers have a limit on the maximum number of files that can be open at one time, **closing files that are no longer needed makes good sense**
- Open files existing at the end of normal program execution are closed by the operating system
- DEMO foc.c

补充：exit

- 如果打开文件失败，需要直接终止程序执行的话
  - exit(-1)
  - exit()就是退出，传入的参数是程序退出时的状态码，0表示正常退出，其他表示非正常退出，一般用-1，标准C里有EXIT_SUCCESS和EXIT_FAILURE两个宏，用exit(EXIT_SUCCESS);可读性更好
  - 使用exit()时，程序直接终止
  - P380, 程序10.1
  - P382编程注解

Reading from and Writing to Text Files

- Prototypes in `stdio.h`
- Examples
  - `fputc('a',outFile);`
  - `fputs("Hello world!",outFile);`
  - `fprintf(outFile,"%s %n",descrip,price);`
  - **DEMO fw.c**

Reading from a Text File

- Prototypes in `stdio.h`
- Examples
  - `fgetc(inFile);`
  - `fgets(message,10,inFile);`
  - `fscanf(inFile,"%lf",&price);`
- `fgetc()` and `fscanf()` return EOF when the end-of-file marker is detected
- `fgets()` returns a NULL instead
- DEMO fr.c

Function	Description
`fgetc(`*`filename`*`)`	Read a character from the file.
`fgets(`*`stringname,n,filename,`*`)`	Read *n* –1 characters from the file and store the characters in the given string name.
`fscanf(`*`filename,"format",&args`*`)`	Read values for the listed arguments from the file, according to the format.

Random File Access

- `rewind()` resets the current position to the start of the file
  - `rewind(inFile)`
- `fseek()` allows the programmer to move to any position in the file
  - `fseek(fileName, offset, origin)`
  - Origin: `SEEK_SET(起始)`, `SEEK_CUR(当前)`, and `SEEK_END（末尾）`
- `ftell()` returns the offset value of the next character that will be read or written
  - `ftell(inFile);`

Random File Access (continued)

- Examples of `fseek()` are
  - `fseek(inFile,4L,SEEK_SET);`
  - `fseek(inFile,4L,SEEK_CUR);`
  - `fseek(inFile,-4L,SEEK_CUR);`
  - `fseek(inFile,0L,SEEK_SET);`
  - `fseek(inFile,0L,SEEK_END);`
  - `fseek(inFile,-10L,SEEK_END);`
  - **DEMO fs.c**

补充：块的输入和输出

- fread函数
  - size_t fread(a, sizeof(a[0]),sizeof(a)/sizeof(a[0]),fp);
- fwrite函数
  - size_t fwrite(&s, sizeof(s), 1, fp);
- DEMO: br.c(作业第8题）

Homework

- 1. P391, 5 (建议逐个字符读）
- 2. P392, 6
- 3. P395, 3
- 4. P407, 4
- 5. 实现程序将文本文件倒序输出到新的文件。
- 6. 《现代方法第2版》P416, 3
- 7.《现代方法第2版》P418, 17
- 8.《现代方法第2版》 P416, 6
- 9.《现代方法第2版》P416, 7（选做，可以用第6题的程序查看是否能够正确地压缩与解压）