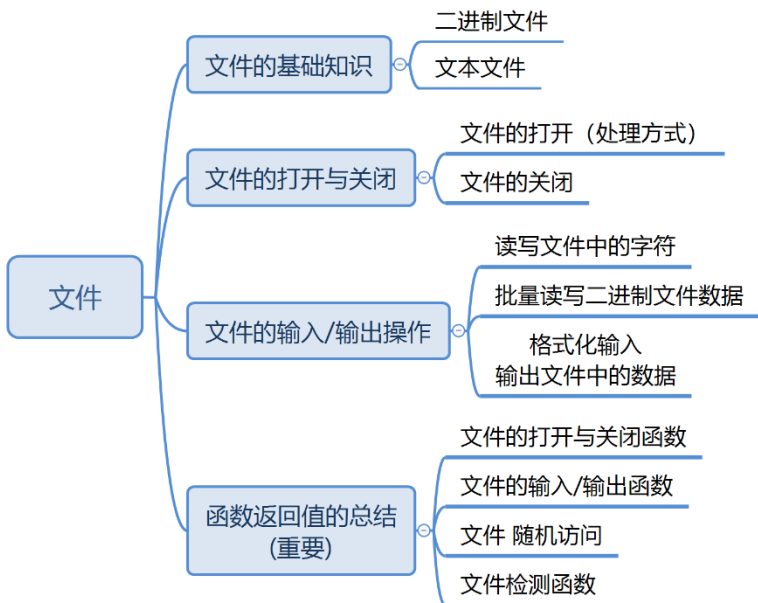




专题十 文件

【内容预览】



【知识清单】

10.1、文件的基础知识

10.1.1、二进制文件和文本文件

C 语言把文件看作一个字符（或字节）的序列，即由一个字符（字节）的数据顺序组成。根据组织形式，数据可分为两种类型，文本文件（也称 ASCII 文件）和二进制文件。

- 两者的区别:
- ①**文本文件中，数据以 ASCII 码形式存放，即每个 ASCII 码代表 1 个字符，输出与字符一一对应，一个字节代表一个字符**，因此便于对字符逐个进行处理。但占用存储空间大，且要花时间进行二进制与 ASCII 码的转换。
 - ②**在二进制文件中，一个字节不对应一个字符**，因此不能直接对字符逐个处理，但这种二进制文件可以节省外存空间和转换时间，存放数据的效率很高，所以当使用含有大量数值的文件时，可以考虑使用二进制文件。
 - ③**文本文件有单独的文本行，在写入文件时，会自动生成行结束标记；二进制文件不被分成单独的行，没有行结束标志。**

如：short int a=123;

短整型变量 **a** 在内存中占 2 个字节，在二进制文件中，变量 **a** 同样占 2 个字节。但在文本文件中，**a** 需要 3 个字节的存储空间。而如果 **a** 的值变为 1234，对二进制文件来说，变量 **a** 的值仍占 2 个字节，而在文本文件中则占 4 个字节。

10.2、文件的打开与关闭

10.2.1、文件结构体及文件类型指针

stdio.h 中定义了一个结构体类型 FILE，其中包含诸如缓冲区的位置，文件的状态,是否正在读取或写入，文件的当前位置及是否发生错误等有关所使用的文件的信息，这种结构体由系统而不是程序员管理。

对于每个要操作的文件，用户先要在程序中定义一个指向 FILE 类型的指针变量。如 FILE *fp;

10.2.2、文件的打开

在使用文件前必须打开文件。打开文件就是建立文件指针变量与为外部文件的联系，即建立流的通道。

格式：FILE *fp;

fp=fopen(filename,mode); //filename 中表明路径的斜线必须使用双斜线；如 “c:\\one\\file.c”

filename 表示文件名，即要使用的文件名；mode 指定了对文件的操作类型，即对文件的处理方式。若成功，则在内存中开辟一个存储空间存放 FILE 类型数据并返回其起始地址，否则返回 NULL。

表 10-1:文件的处理方式

mode	处理方式	指定文件不存在	指定文件存在	含义
r	只读	出错	正常打开	以读方式打开一个文本文件
w	只写	建立新文件	文件原有内容丢失	以写方式打开一个文本文件
a	追加	建立新文件	在文件原有内容后追加	以追加方式打开一个文本文件
rb	只读	出错	正常打开	以读方式打开一个二进制文件
wb	只写	建立新文件	文件原有内容丢失	以写方式打开一个二进制文件
ab	追加	建立新文件	在文件原有内容后追加	以追加方式打开一个二进制文件
r+	读写	出错	正常打开	以读/写方式打开一个文本文件
w+	读写	建立新文件	文件原有内容丢失	以写/读方式打开一个文本文件
a+	读写	建立新文件	在文件原有内容后追加	以读/写方式打开一个文本文件
rb+	读写	出错	正常打开	以读/写方式打开一个二进制文件
wb+	读写	建立新文件	文件原有内容丢失	以写/读方式打开一个二进制文件
ab+	读写	建立新文件	在文件原有内容后追加	以读/写方式打开一个二进制文件

1. 在程序运行时，系统自动打开 3 个标准文件（标准流）：`stdin`、`stdout` 和 `stderr`。当程序需要从标准输入流（`stdin`）进行输入时，它从键盘获得输入；当程序向标准输出流（`stdout`）输出数据时，就把数据发送给终端屏幕；由库函数生成的出错信息送到标准出错流（`stderr`），即系统终端屏幕。
2. 当以读的方式打开的文件不存在时，会产生错误；也有可能文件存在，但没有使用权限，也会产生错误。

如果有任何错误，`fopen()`函数将返回 `NULL`。

在具体程序中，通常用下面的方式检查文件 `file` 是否正常打开，并显示相应的信息：

```
if((fp=fopen("file.c", "r"))==NULL){
    printf("文件打开失败，退出程序的执行\n");
    exit(0);
} //其中，exit()函数能关闭所有文件，终止程序。
```

10.2.3、文件的关闭

关闭文件就是断开文件指针变量与外部文件之间的联系，即关闭流。

格式：`fclose(fp)`;

1. 执行该函数时，首先释放文件缓冲区，同时保证文件的完整性。若成功地关闭了指定的文件，函数返回 0 值，否则函数返回 `EOF`（定义在 `stdio.h` 中）。
2. 由于文件是通过缓冲区进行读写的，所以当读写文件时，访问的是文件缓冲区，而不是直接访问文件。如果没有清除和保存缓冲区中的内容就终止程序，存放在缓冲区的数据就会丢失。而且大多数操作系统对于同时打开的文件数目是有限制的，因此，一旦不使用某文件，就一定要用 `fclose()`函数关闭它。

10.3、文件的输入/输出操作

10.3.1、读写文件中的字符

1. 字符读写函数

（1）函数 `fgetc()`用于从 1 个以只读或读写方式打开的文件读出字符。其函数原型为：

```
int fgetc(FILE *fp);
```

其中，`fp` 是由函数 `fopen()`返回的文件指针，函数功能是从 `fp` 所指文件中读 1 个字符，并将文件位置指针指向下 1 个字符。若读取成功，则返回该字符；若读到文件末尾，则返回 `EOF`。通常使用方法如下，

`ch=fgetc(fp)`;表示从 `fp` 所指文件中读 1 个字符，然后将其赋值给字符变量 `ch`。

（2）函数 `fputc()`用于将 1 个字符写到 `fp` 所指的文件中。其函数原型为：

```
int fputc(int c,FILE *fp);
```

函数的功能是将变量 `c` 写到 `fp` 所指文件中。若写入成功，则返回该字符；若写入错误，则返回 `EOF`。

使用方法：`fputc(ch,fp)`;

例：`while((ch=getchar())!='\n')`

```
    fputc(ch,fp);
```

2. 读写文件中的字符串

（1）从文件中读取字符串

函数原型：`char *fgets(char *str,int n,FILE *fp)`;

其中，`str` 是要读取的字符串，函数的功能是从 `fp` 所指的文件中读 `n-1` 个字符，存放在 `str` 所指的存放单元中。当读到回车换行符、文件末尾或读满 `n-1` 个字符，函数返回该字符串的首地址。函数返回值为指向所读串的指针，若出错或已到文件尾就返回 1 个空指针。如：`fgets(str,n,fp)`;`fgets()`函数执行结束时，文件指针会偏移至当前读完的字符之后的位置。

（2）向文件写入字符串

函数原型：`char *fputs(const char *str,FILE *fp)`;

函数功能是将 `str` 中的字符串写入 `fp` 所指文件中。若写入错误，则返回 `EOF`。如：`fputs(str,fp)`;

10.3.2、批量读写二进制文件数据

1.fread()函数

函数原型: `unsigned fread(void *buffer,size-t size,size-t count,FILE *fp);`

其中, `buffer` 是待读入数据块的起始地址; `size` 是每个数据块的大小 (字节数); `count` 表示最多允许读取的数据块个数。

函数功能: 从 `fp` 所指的文件中读取数据块并存储到 `buffer` 指向的内存中。函数返回的是实际读到的数据块的个数。

2.fwrite()函数

函数原型: `unsigned fwrite(const void *buffer,size-t size,size-t count,FILE *fp)`

`buffer` 是待输出数据块的起始地址; `size` 表示每个数据块的大小 (待输出的每个数据块的字节数); `count` 表示最多允许写入的数据块个数 (每个数据块 `size` 个字节)。

函数功能: 将 `buffer` 指向的内存中的数据块写入 `fp` 所指的文件。

表 10-2: `fread` 和 `fwrite` 实现对文件批量数据的读写

创建文件	<pre>FILE *fp; fp=fopen("data", "wb"); fwrite(student,SIZE,k,fp); fclose(fp);</pre>
读取文件	<pre>FILE *fp; fp=fopen("data", "rb"); fread(student,SIZE,k,fp); fclose(fp);</pre>

10.3.3、格式化输入输出文件中的数据

1.fscanf()函数用于按指定格式从文件中读取数据

函数原型: `int fscanf(FILE *fp,格式控制字符串, 输出项列表);`

如: `fscanf(fp,"%d", &x);` 函数表示从 `fp` 所指文件中, 按 `int` 型格式读入数据, 存放在整型变量 `x` 中。

2.fprintf()函数用于按指定格式向文件中写数据

函数原型: `int fprintf(FILE *fp, 格式控制子串, 输出项列表);`

如: `fprintf(fp,"%d ",x);` 函数表示按 `int` 格式将变量 `x` 的值写入 `fp` 所指的文件中。

`fscanf()`函数和 `fprintf()`函数的执行很像 `scanf()`和 `printf()`, 不同之处在于读入和写出的数据都面向指定的文件, 而不像 `scanf()`和 `printf()`那样面向 `stdin` 和 `stdout`。

10.4、函数返回值的总结

10.4.1、文件的打开和关闭函数

`fopen()`函数: 若打开成功, 返回起始地址, 否则返回 `NULL` (定义在头文件 `stdio.h` 中)。

`fclose()`函数: 若成功关闭指定文件, 则函数返回 `0` 值, 否则返回 `EOF` (定义在头文件 `stdio.h`)。

10.4.2、文件的输入输出函数

`fgetc()`函数: 若读取成功, 则返回该字符; 若读到文件末尾, 则返回 `EOF`。

`fputc()`函数: 若写入成功, 则返回该字符; 若读到文件末尾, 则返回 `EOF`。

`fgets()`函数: 若读取成功, 则函数返回指向所读串指针, 即返回该字符串的首地址; 若出错或已到文件尾, 就返回一个空指针 `NULL`。

`fputs()`函数: 若写入错误, 则返回 `EOF`。

fread()函数: 函数返回的是实际读到的数据块个数。

fwrite()函数: 函数返回的是实际写入的数据块个数。

10.4.3、文件的随机访问

1.rewind()函数: 函数无返回值。

调用形式: rewind(fp);

功能: 将文件位置指针复位到文件的首部。

2.ftell()函数: 函数返回 fp 所指文件的文件位置指针当前所指位置(字节数), 若出错返回-1。

调用形式: long loc;

loc=ftell(fp);

功能: 求出当前文件的处理位置。

3.fseek()函数: 调用形式: fseek(fp,offset,whence);

功能: 将 fp 的文件指针从 whence 开始, 移动 offset 个字节, 指向下 1 个要读取数据的位置。

10.4.4、文件检测函数

feof(文件指针)函数: 若文件结束, 则返回值为 1 (真), 否则为 0 (假)。

功能: 检测是否到文件尾, 它既适用于文本文件, 又适用于二进制文件。

【解题技巧】

例 10.1 若要用 fopen 函数打开一个二进制文件, 目的是将数据添加进该文件, 则文件处理方式字符串应是 ()

A)"ab"

B)"wb"

C)"rb"

D)"r"

正解: A

分析: "wb": 以写方式打开一个二进制文件; "rb": 以读方式打开一个二进制文件; "r": 以读方式打开一个文本文件。

例 10.2 程序中调用标准库函数 feof 来判断被处理文件是否结束, 如果内部记录指针未到尾, 其返回值是 ()

A)EOF

B)1

C)0

D)文件的最后一个字符

正解: C

分析: feof 是 C 语言标准库函数, 其原型在 stdio.h 中, 其功能是检测文件是否结束, 如果文件结束, 则返回非 0 值, 否则返回 0。

例 10.3

写出下面程序执行后文件 data.txt 中的内容。

```
#include <stdio.h>

void putchartofile(char c)
{
    FILE *f;
    f=fopen("data.txt","a");
    fputc(c,f);
    fclose(f);
}

void main()
{
    char ch;
```

```
for(ch='A';ch<='Z';ch++)
    putchartofile(ch);
}
```

正解：ABCDEFGHIJKLMNOPQRSTUVWXYZ

分析：这段程序的功能是将 A~Z 依次写入文件中。

【精选习题】

答案 P128

基础篇

1. 如果需要打开一个已经存在的非空文件"Demo"进行修改, 正确的方法是()
A)fp=fopen("Demo","r") B)fp=fopen("Demo","ab+")
C)fp=fopen("Demo","r+") D)fp=fopen("Demo","w+")
2. C 语言的文件操作过程中, 调用_____函数可以使得文件的读写位置指针回到文件的开始处。
3. 设已用文件指针 fp 正确打开了文件, 以下语句段的作用是()

```
int x=0;
fseek(fp,0,SEEK_END);
x=ftell(fp);
```


A)将记录指针移到文件尾 B)重置记录指针
C)计算文件的长度 D)判断是否到文件结束处
4. _____文件是把内存中的数据, 按其在内存中的存储形式原样存放在计算机外部设备的指定文件中。
5. 下列叙述中正确的是 ()
A) 当对文件的读(写)操作完成后, 必须将它关闭, 否则可能导致数据丢失
B) 打开一个已存在的文件并进行写操作后, 原有文件中的全部数据必定被覆盖
C) 在一个程序中对文件进行写操作后, 必须先关闭该文件然后再打开才能读到第一个数据
D) C 语言中的文件是流式文件, 因此只能顺序存取文件
6. 下列关于 C 语言文件的叙述中正确的是 ()
A) 文件由一系列数据依次排列组成, 只能构成二进制文件
B) 文件由结构序列组成, 可以构成二进制文件或文本文件
C) 文件由数据序列组成, 可以构成二进制文件或文本文件
D) 文件由字符序列组成, 其类型只能是文本文件

提高篇

1. 下面程序的功能是: 读出文本文件 infor.txt 中的所有数据并显示在屏幕上, 请填空完成程序。

```
#include<stdio.h>
void main()
{ FILE *fp;
  char ch;
```

```

if((fp=fopen("infor.txt", "r"))==NULL)
{
    printf("Can't open file!\n ");
    return;
}
ch=fgetc(fp);
while(_____)
{
    putchar(ch);
    ch=fgetc(fp);
}
_____
}

```

2. 设有以下语句：struct st{long num;char name[10];}stu[30], *p=stu;给该数组赋值后，将其所有元素值写入 fp 指向的文件。正确的语句是（ ）

A) fwrite(30*sizeof(struct st),1,fp);
 B) fwrite(p,sizeof(struct st)*30,30,fp);
 C) fwrite(stu[0],12,30,fp);
 D) fwrite(p,sizeof(struct st),30,fp);

3. 有以下程序；

```

#include<stdio.h>
main()
{
    FILE *fp;int a[10]={1,2,3},i,n;
    fp = fopen("d1.dat","w");
    for(i=0;i<3;i++)
        fprintf(fp,"%d",a[i]);
    fprintf(fp,"\n");
    fclose(fp);
    fp = fopen("d1.dat","r");
    fscanf(fp,"%d",&n);
    fclose(fp);
    printf("%d\n",n);
}

```

该程序的输出是（ ）

A)321 B)12300 C)1 D)123

4. 编程实现功能：将文本文件 a.txt 中的所有大写英文字母挑选出来转化为对应的小写字母后写入到文本文件 b.txt 中。