

文件操作

概念知识

文件是操作系统管理数据的基本单位，文件一般包括三要素：文件路径、文件名、后缀。

D:\\C_WorkSpace\\Chapter_10\\file_1.txt

D:/C_WorkSpace/Chapter_10/file_1.txt

以上两个是文件的两种不同写法

根据文件中数据的组织形式的不同，可以把文件分为：文本文件和二进制文件（考试只考文本文件）

- 文本文件：把要存储的数据当成一系列字符组成，把每个字符的 ASCII 码值存入文件中。每个 ASCII 码值占一个字节，每个字节表示一个字符。故文本文件也称作字符文件或 ASCII 文件，是字符序列文件。
- 进制文件：把数据对应的二进制形式存储到文件中，是字节序列文件。

C语言文件操作

函数介绍

1. fopen()

fopen:文件打开操作

函数原型：FILE * fopen(char * pname, char * mode) mode: r read, w write (如果文件不存在，则建立一个新文件，如果文件存在，删除里面的所有东西) a append(追加)

正常返回：FILE * 一个指向文件在内存中的文件信息去的开头

异常返回：NULL，表示打开操作不成功

2. fclose()

fclose:文件关闭

函数原型：int fclose(FILE *fp);

正常返回：0

异常返回：EOF，表示文件在关闭时发生错误

3. fgetc()

fgetc:从fp中读取一个字符，作为返回值返回

函数原型: int fgetc(FILE *fp)

正常返回: 返回读取字符的代码

异常返回: 返回EOF

3. fputc()

fputc: 写一个字符到文件中

函数原型: int fputc(int ch, FILE*fp)

正常返回: 要写入的字符的代码

异常返回: 返回EOF

4. fgets()

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

fgets: 由fp指出的文件中读取n-1个字符，并把他们存放到有str指出的字符数组中，最后加上一个由字符串结束符"

5. fputs()

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

fputs: 把由str之处的字符串写入到fp所指的文件中

6. fprintf () 和 fscanf ()

```
1 fprintf(fp, "%s %d\t", str, &n); //制表符
2 printf("%s\t", str);
3 fscanf(fp, "%s\t", &str);
4 scanf("%s\t", &str);
```

7. feof(fp)

如果指针到了文件的末尾，返回false，否则返回true

file eof feof

```
1 if(!feof(fp)){ //如果指针没有到文件的末尾
2
3 }
```

字符串的操作

1. 整数转为字符串

```
1 char* itoa(int value,char*string,int radix);//value: 要转换的整数,  
   string: 转换后的字符串,radix: 转换进制数,如2,8,10,16 进制等。
```

这是库函数，我们可以写一个简单的类似函数

一个位的整数 0-9，怎么转为字符

char = int + '0';

int = char - '0';

56789154

%10 4 /10 5678915

5 5 0

```
1 void reverse(char* str,int k){           //字符数组逆转函数  
2     if(!str||k<=1)  
3         return;  
4     char temp;  
5     for(int i=0,j=k-1;i<j;){             //双指针操作  
6         temp = str[i];                   //swap  
7         str[i] = str[j];  
8         str[j] = temp;  
9         ++i;  
10        --j;  
11    }  
12 }  
13 int main(){  
14     temp = rand()%100;  
15     char str[10];  
16     int k = 0;  
17     printf("%d\t",temp);  
18     while(temp){                         //一个一个将数字转为字符  
19         str[k++] = temp%10 + '0';         //'1' = 1 + '1'  
20         temp /= 10;                       //减少个位数  
21     }  
22     str[k] = '\0';  
23     reverse(str,k);  
24 }
```

写随机数

1. 写随机种子

```
1 srand(time(0));           // #include <time.h>
```

2. 生成随机数

```
1 | temp = rand()%100 + 47;    //生成47到147的随机数
```