



石家莊鐵道大學
SHIJIAZHUANG TIEDAO UNIVERSITY

在线开放课程

C语言程序设计—文件操作

文件的读写函数

主讲：石玉晶

目录

- ◆ 1、fputc函数和fgetc函数
- ◆ 2、feof函数
- ◆ 3、fputs函数和fgets函数
- ◆ 4、fread函数和fwrite函数
- ◆ 5、fprintf函数和scanf函数

一、fputc函数和fgetc函数

■ fputc函数

用 fputc函数将一个字符输出到文件中。

调用形式:

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

- (1) ch 是要输出的字符，一个字符常量或字符变量；
- (2) fp是文件指针，它是从fopen函数得到的返回值。
- (3) fputc函数带回一个返回值，如果输出成功，则返回值就是输出的字符；如果输出失败，则返回EOF(-1)。
- (4) 被写入的文件可以用写、读写、追加方式打开。

一、fputc函数和fgetc函数

■ fgetc函数

用 fgetc函数是从文件读取一个字符。

```
ch = fgetc( fp )
```

文件必须是以读或读写方式打开的。

二、feof函数

■ feof函数

feof函数检查文件读写位置是否移到文件的末尾，即文件是否结束。

调用形式：

```
feof( fp )
```

文件结束，函数值为非 0(真)，否则为0(假)。

二、feof函数

■ 程序举例

【例 1】建立一个文件，将键入第一个“!”号前的若干个字符逐个写入该文件。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;    /* 步骤1: 定义文件指针 */
    char ch , filename[20];
    /* 输入写入数据的文件名 */
    printf( "Please input(file name):" );
    scanf( "%s" , filename );
    getchar();    /* 接收执行scanf函数时, 最后的回车 */
    if( ( fp = fopen( filename , "w" ) ) == NULL) /* 步骤2: 文件打开 */
    { printf( "Can not open file.\n" ); exit( 0 ); }
```

二、feof函数

```
printf( "Please input(end of !):" );  
ch = getchar( ); /* 接收输入的第一个字符 */  
while( ch != '!' )  
{  
    /* 步骤3: 文件写操作, 将字符ch输出到fp所指向的文件中 */  
    fputc( ch , fp );  
    ch = getchar( ); /* 逐个接收从键盘输入的字符 */  
}  
putchar( '\n' );  
fclose( fp ); /* 步骤4.文件关闭 */  
return 0 ;  
}
```

三、 fputs函数和fgets函数

■ fputs函数

fputs函数是将一个字符串输出到指定的文件。

调用形式：

```
fputs( str , fp )
```

str 是待输出的字符串，可以是字符串常量、存放字符串的字符数组名或指向字符串的指针。**fp**是文件指针。**str** 字符串末尾的'\0'不输出。

若函数调用成功，返回非负整数，否则返回**EOF(-1)**。

三、fputs函数和fgets函数

■ fgets函数

fgets函数是从指定的文件中读取一个字符串。

调用形式:

```
fgets( str , n , fp )
```

本函数的功能是从 **fp** 指针所指向的文件读取 **n-1** 个字符，并在最后加一个 **'\0'** 字符，然后把这 **n** 个字符存放到以 **str** 为起始地址的空间中。

如果在未读满 **n-1** 个字符时，遇到 **'\n'** 或 **EOF** (文件结束标志)，读入即结束，但将所遇到的 **'\n'** 也作为一个字符读入。

若调用 **fgets** 函数成功，则返回值为 **str**；如果调用失败，返回 **NULL**。

三、fputs函数和fgets函数

■ 程序举例

【例2】 编写程序使用fputs函数将键盘输入的字符串“hello, c program!”写入data.txt文件中，再使用fgets函数将刚写入文件的字符串读入内存并显示在屏幕上。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    FILE *fp ;
    char outstr[] = "hello , C program!" ;
    char instr[25] ;
    if( ( fp = fopen( "data.txt" ,"w" ) ) == NULL )
    {   printf( "Can not open data.txt!" ) ;   exit( 0 ) ;   }
```

三、fputs函数和fgets函数

```
fputs( outstr , fp );  
fclose( fp );  
if( ( fp = fopen( "data.txt" ,"r" ) ) == NULL )  
{  
    printf( "Can not open data.txt!" );  
    exit( 0 );  
}  
fgets( instr , strlen(outstr) + 1 , fp );  
puts( instr );  
fclose( fp );  
return 0 ;  
}
```

四、fread函数和fwrite函数

- fread函数和 fwrite函数分别用来读写二进制文件。

```
fread( buf , size , n , fp )
```

```
fwrite( buf , size , n , fp )
```

(1) **buf** 是一个内存地址(或指针)。对 **fread** 函数而言，它是从文件中读取数据存入此内存单元中。对 **fwrite** 函数而言，是需要输出数据的起始地址。

(2) **size** 是要读/写的一个数据块的字节数。

(3) **n** 是要进行读/写数据项的个数(每个数据项有 **size** 个字节)。

(4) **fp** 是文件指针，指向待读/写的文件。

(5) 如果 **fread** 函数或 **fwite** 函数调用成功，则返回读出或写入的数据块的个数(即 **n** 的值)。

四、fread函数和fwrite函数

【例3】使用块读/写函数建立一个有关员工数据类型的数据文件。

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 3
struct employee
{
    unsigned short int no ;
    char name[ 20 ] ;
    short int salary ;
}emp[6] , result[6] ;
```

四、fread函数和fwrite函数

```
void scanf_data( )
{
    int i ;
    printf( "Please input(%d-employee)\nName\tSalary\n" , SIZE ) ;
    for( i = 0 ; i < SIZE ; i++ )
    {
        emp[i].no = i + 1 ;
        scanf("%s%d" , emp[i].name , &emp[i].salary ) ;
    }
}
```

四、fread函数和fwrite函数

```
void save_data()  
{  
    FILE *fp ;  
    int i ;  
    if( ( fp = fopen( "employee.txt" , "wb" ) ) == NULL )  
    {  
        printf( "Can not open the file.\n" );  
        exit( 0 ) ;  
    }  
    for( i = 0 ; i < SIZE ; i++ )  
        if( fwrite( &emp[i] , sizeof( struct employee ) , 1 , fp ) != 1 )  
            printf( "Can not operate file!\n" ) ;  
    fclose( fp ) ;  
}
```

四、fread函数和fwrite函数

```
void write_data()
{
    FILE *fp;
    int i;
    if( ( fp = fopen( "employee.txt" , "rb" ) ) == NULL )
    {    printf( "Can not open the file.\n" ); exit( 0 );    }
    printf( "The data from disk:\n" );

    for( i = 0 ; i < SIZE ; i++ )
    {
        if( fread( &result[i] , sizeof( struct employee ) , 1 , fp ) != 1 )
            printf( "Can not operate file!" );
        printf( "%d , %s , %d \n" , result[i].no , result[i].name , result[i].salary ) ;
    }
    fclose( fp ) ;
}
```


四、fread函数和fwrite函数

```
int main()  
{  
    scanf_data();  
    save_data();  
    write_data();  
    return 0;  
}
```

```
Please input (3-employee)
```

```
Name      Salary
```

```
abc       1100✓
```

```
ef        1300✓
```

```
ghi       1200✓
```

```
The data from disk:
```

```
1 , abc , 1100
```

```
2 , ef , 1300
```

```
3 , ghi , 1200
```

五、fprintf函数和fscanf函数

- fprintf 函数、fscanf 函数与 printf 函数、scanf 函数作用相仿，都是格式化写读函数。只有一点不同：fprintf和fscanf函数的读/写对象是文件。

调用形式：

fprintf(文件指针 , 格式字符串 , 输出表列)
fscanf(文件指针 , 格式字符串 , 输入表列)

五、fprintf函数和fscanf函数

■ fprintf 函数

fprintf(文件指针 , 格式字符串 , 输出表列)

fprintf 函数按格式将内存中的数据转换成相应的字符, 以 ASCII 码形式输出到文本文件中。

例如:

```
fprintf( fp , "%c,%d" , ch , i );
```

将字符型变量 `ch` 和整型变量 `i` 的值按照 `%c` 和 `%d` 的格式输出到 `fp` 指向的文件中。如果 `ch = 'a'`, `i = 123`, 则输出到文件上的是以下字符串:

`a,123`

五、fprintf函数和fscanf函数

■ fscanf 函数

fscanf(文件指针 , 格式字符串 , 输入表列)

fscanf函数只能从文本文件中按格式输入。

例如，用以下fscanf函数从文件中读取 ASCII字符：

```
fscanf( fp , "%c,%d" , &ch , &i );
```

文件上如果有以下字符串：

a,123

则将文件中的数据'a'赋给字符型变量ch，123赋给整型变量i。

- ◆ 接下来将学习
- ◆ 10-4 习题解答

谢谢！