

简单文件操作

作者：20373559 杨恩源, 20373097王钧石

本教程适用于数据结构课程中对C语言文件操作尚不熟悉的同学，主要内容如下：

- 文件的打开和关闭
- 文件输入输出

文件的打开和关闭

C语言中，一个文件在使用前后，必须进行打开和关闭操作。

fopen() 函数

fopen 函数用于打开文件, 其调用格式为:

```
FILE *fopen(char *filename, *type);  
//FILE为结构指针，当前阶段可将其理解为为一个包含了文件管理有关信息的数据结构，即在打开文件时必须先定义一
```

*type	含义
"r"	打开文件(只读)
"w"	创建文件(只写)

一般来说，数据结构课程用这两种字符读写文件足够了。

*type 字符的含义类型不止此两种，感兴趣的同学可自行探索

假设成功的打开一个文件, fopen()函数返回文件指针, 否则返回空指针 (NULL)。由此可推断文件打开是否成功。

fclose() 函数

fclose() 函数用来关闭一个由 fopen() 函数打开的文件。

虽然在数据结构作业中忘记写 `fclose()` 并不会出现什么问题，但仍强烈建议遵循代码规范与逻辑，将**两者配套使用**！

同学们可在下面的几个示例代码中理解 `fopen()` 及 `fclose()` 的使用方法。

文件输入输出

在使用 `fopen()` 函数成功打开一个文件，并获得文件指针 `FILE *in` 之后，可使用 `fgetc()`、`fgets()` 等函数从中读取数据。

fgetc 函数

`fgetc` 函数原型为

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

`stream` 是文件指针，即上述的 `in`。该函数将从文件 `in` 中读取一个字符，并将读取的字符以 `int` 形式返回（ASCII码）。在读取失败的情况下（如达到文件末尾），返回 `-1`，即 `EOF`。

下面给出一个使用示例：

```
//改自BUAA数据结构课件DSPv2b_1_数据结构基础_p21
#include <stdio.h>
int main()
{
    int c;
    FILE *in, *out; // 为读写文件定义文件指针
    in = fopen("input.txt", "r");
    out = fopen("output.txt", "w");
    if(NULL == in) {
        printf ("Failed to open the file !\n");
        exit(0);
    }// 此条件句为演示FILE指针为空的含义
    // 打开文件input.txt和output.txt位于与该执行程序.exe文件同一目录下
    while ((c = fgetc(in)) != EOF) { // 从文件input.txt中依次读一个字符
        fputc(c, out); // 向文件output.txt中依次写一个字符
    }
    fclose(in);
    fclose(out);
    // 关闭两个打开的文件
    return 0;
}
```

fgets 函数

fgets 函数原型为

```
char *fgets(char *restrict str, int count, FILE *restrict stream);
```

其中参数的含义如下：

1. `str`：一个字符串的首地址，与 `gets()` 函数中传入的字符串一样。fgets 函数将读取的数据存入字符串 `str`。
2. `count`：一个整型变量，`count - 1` 表示最多读入的字符个数。gets() 函数正是因为没有最大读入字符数这一参数，在安全性上才出现了一些漏洞。该参数可以设置为 `str` 字符串的大小。
3. `stream`：待读取文件的文件指针，即上述的 `in` 指针。

在读取成功时，该函数返回传入的 `str`。读取失败（达到文件末尾等）时，返回 `null` 指针。因此可通过判断该函数返回值是否为 `null` 来判断是否读取结束。

该函数结合 `while` 循环可实现按行读取某文件的内容。下面给出一个使用的示例。

```
// 改自BUAA数据结构课件DSPv2b_1_数据结构基础_p26
#define MAXLINE 1024
char line[MAXLINE];

int main()
{
    // 打开文件
    FILE *fp;
    if((fp = fopen("file.txt", "r")) == NULL)
    {
        printf("Can't open file!\n");
        return 1;
    }
    // 从文件中按行读取，依次存入line中
    while(fgets(line, MAXLINE - 1, fp) != NULL)
    {
        if(index(line, '\n') != 0)
        {
            printf("%s", line);
        }
    }
    // ...

    // 关闭文件
    fclose(fp);

    // ...
}
```

注：换行符 '\n' 将被存入字符串末尾，可能需要更多的一些处理。

其他输入输出函数

一些常用的输入输出函数如下：

1. `fputs()` : 与 `puts()` 类似，向文件输出字符串
2. `fscanf()` : 与 `scanf()` 类似，按格式从文件读取字符串
3. `fprintf()` : 与 `printf()` 类似，按格式向文件输出字符串

这些函数均与同学们已经掌握的 `scanf()` 等函数用法类似，只是增加了一个文件指针的参数。同学们可以自行尝试，也可阅读本文末尾的参考资料。

重定向（`freopen` 函数）

上面已经介绍了文件的打开关闭、文件输入输出的方法，这里再简要说明文件重定向函数 `freopen()`，它也可以实现文件的输入输出。有时，这种方法更加简单。

在使用控制台输入输出的程序前，加入以下两行代码：

```
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
// 文件名input.txt, output.txt可更换
// "r"与"w"含义与fopen()中相同
```

之后便可直接使用 `scanf()`，`printf()` 等函数直接从文件输入、向文件输出。

在文件末尾关闭文件：

```
fclose(stdin);
fclose(stdout);
```

更多阅读资料

1. [C/C++语言参考](#)

可从中查看各种函数的用法，介绍较为全面，推荐大家进行查阅。

- `fopen`函数：<https://en.cppreference.com/w/c/io/fopen>

- fgetc函数 : <https://en.cppreference.com/w/c/io/fgetc>
- fgets函数 : <https://en.cppreference.com/w/c/io/fgets>
- freopen函数 : <https://en.cppreference.com/w/c/io/freopen>

2. B站入门视频讲解: <https://www.bilibili.com/video/BV17s411N78s?p=56>
P56-P61讲解文件的相关操作。