

1. `chmod` : 更改文件 9 个属性 (读, 写, 执行)

9 个属性是 owner/group/others 各自的 read/write/execute 权限

`chmod [-R] xyz` 文件或目录

way 1

xyz 是权限属性, 是 rwx 属性值相加 r:4 w:2 x:1

通过符号类型改变文件权限

u: 用户 group: 组 others: 组

`chmod u=rwx, g=rX, o=r` 文件名

way 2

2. `File * stream`

`File` 是一个结构体, `stream` 则是一个指向 `File` 的结构体指针  
传入 `stream` 这个参数, 操作的是 `File`

3. 空格不是空字符, 它俩不是一个概念。

4. `fgets` 和 `gets()`

`char* fgets (char*s, int count, file*stream)`

`char* gets (char*s)` 返回值应当理解为  
返回字符串 `s` 吧。

`fgetc` 从 `stream` 指定的流中读 `count-1` 到字符串 `S`。  
并在末尾添加空字符

↓  
`S` 指定的字符串。

为什么 `S` 是字符串???

应当是从地址 `S` 开始读 `count-1` 个字符吧

`gets` 从标准输入流 `stream` 中读入完整字符  
并删除换行符, 加上空字符结尾。

---

## 5. `fgetc`、`getc`、`getchar`

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

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

```
int getchar (void);
```

都是读一个字符然后将它强制转换为 `int` 返回。

`getchar` 从标准输入流 `stdin` 中读

相当于 `getc (stdin)`

`fgetc` 和 `getc` 相同

---

6.

```
int fputc (int c, FILE *stream)
```

```
int putc (int c, FILE *stream)
```

```
int putchar (int c)
```

都  $c$  转化为 无符号  $\text{char}$  型, 写入  $\text{stream}$  指定流中  
并返回  $c$ .

$\text{putchar}(\text{int } c)$  等价于  $\text{putchar}(\text{stdout})$

---

7.

```
int fputs (const char *s, FILE *stream)
```

```
int puts (const char *s)
```

$\text{fputs}(s)$  输出以  $\backslash 0$  结尾的字符串  $s$  到  $\text{stream}$  指定流中  
但结尾的空字符不写入, 也不添加  $\backslash n$   
但是  $\text{puts}$  会加入一个  $\backslash n$ .