

用户都有读权限：

```
umask(DEF_UMASK);
fd = Open("foo.txt", O_CREAT|O_TRUNC|O_WRONLY, DEF_MODE);
```

掩码	描述
S_IRUSR S_IWUSR S_IXUSR	使用者（拥有者）能够读这个文件 使用者（拥有者）能够写这个文件 使用者（拥有者）能够执行这个文件
S_IRGRP S_IWGRP S_IXGRP	拥有者所在组的成员能够读这个文件 拥有者所在组的成员能够写这个文件 拥有者所在组的成员能够执行这个文件
S_IROTH S_IWOTH S_IXOTH	其他人（任何人）能够读这个文件 其他人（任何人）能够写这个文件 其他人（任何人）能够执行这个文件

图 10-2 访问权限位。在 sys/stat.h 中定义

最后，进程通过调用 close 函数关闭一个打开的文件。

```
#include <unistd.h>
int close(int fd);
```

返回：若成功则为 0，若出错则为 -1。

关闭一个已关闭的描述符会出错。



练习题 10.1 下面程序的输出是什么？

```
1  #include "csapp.h"
2
3  int main()
4  {
5      int fd1, fd2;
6
7      fd1 = Open("foo.txt", O_RDONLY, 0);
8      Close(fd1);
9      fd2 = Open("baz.txt", O_RDONLY, 0);
10     printf("fd2 = %d\n", fd2);
11     exit(0);
12 }
```

10.4 读和写文件

应用程序是通过分别调用 read 和 write 函数来执行输入和输出的。

```
#include <unistd.h>
```

```
ssize_t read(int fd, void *buf, size_t n);
```

返回：若成功则为读的字节数，若 EOF 则为 0，若出错为 -1。

```
ssize_t write(int fd, const void *buf, size_t n);
```

返回：若成功则为写的字节数，若出错则为 -1。