文件系统（第一部分）

薛泓彦 万炎广 程轶涵

预备：工具函数

将系统调用的第n个参数获取为int

```
int argint(int n,int* ip);
```

将系统调用的第n个参数获取为结构体指针并作相应检查

```
int argptr(int n,char** pp,int size);
```

将系统调用的第n个参数获取为字符串指针并作相应检查

```
int argstr(int n, char** pp);
```

syscall.c

预备：工具函数

将系统调用的第n个参数获取为文件描述符并获取描述符对应的文件

```
int argfd(int n,int* pfd,struct file** pf);
```

sysfile.c

open 和close 系统调用的作用是什么，系统调用的参数有哪些，各自代表什么含义?

作用是什么？

sys_open

传入文件名字符串和读写权限，打开对应文件并返回其文件描述符或者-1。

sys_close

传入文件描述符，关闭对应文件并返回0或-1。

系统调用的参数及其含义

sys_open

第1个参数,字符串,代表文件路径

```
argstr(0, &path)
```

第2个参数,整数,代表读写权限

```
argint(1, &omode)
```

sys_close

第1个参数,代表文件描述符

```
argfd(0, &fd, &f)
```

open 和close 中各种失败情况的原因都是什么？

Open失败　　4种情况

寻找返回-1的代码：

如果**路径字符串**和**权限变量**不合法(比如内存越界)

```
if(argstr(0, &path) < 0 || argint(1, &omode) < 0)
    return -1;
```

Open失败

```
if(omode & O_CREATE){
    //如果权限是[新建文件]
    //>>>>创建一个新inode如果创建失败
    ip = create(path, T_FILE, 0, 0);
    if(ip == 0){
        end_op();
        return -1;
    }
} else {
    //如果权限不是[新建文件]
    //>>>>寻找文件名对应的inode如果找不到
    if((ip = namei(path)) == 0){
        end_op();
        return -1;
}
```

Open失败

```
//>>>>开辟文件和文件描述符,如果其中某一个失败了
if((f = filealloc()) == 0 || (fd = fdalloc(f)) < 0){
if(f)
fileclose(f);
iunlockput(ip);
end_op();
return -1;
}
```

Close失败

- //>>>>如果文件描述符不合法
- if(argfd(0, &fd, &f) < 0)
  - return -1;

sys_open 函数中
f 变量是什么类型的变量，
里面的各个成员变量都代表什么，
在sys_open 函数中被设成了什么？
和fd 变量含义上有什么区别？

f 变量是什么类型的变量

```
struct file* f;//是一个文件结构体的指针


struct file {
      enum { FD_NONE, FD_PIPE, FD_INODE } type;
      //类型:普通,管道,inode
      int ref; //引用计数
      char readable; //可读?
      char writable; //可写?
      struct pipe *pipe; //管道
      struct inode *ip; //inode
      uint off;
};
```

里面的各个成员变量
都代表什么

```
f->type = FD_INODE;//类型:inode型
f->ip = ip;//inode指针:指向之前获得的
inode
f->off = 0;//
f->readable = !(omode & O_WRONLY);//可
读(除非只可写)
f->writable = (omode & O_WRONLY) ||
(omode & O_RDWR);//可写(仅可写或者读写)
```

f和fd的区别

- f指向一个文件结构体，保存了文件的相关信息
- fd是一个文件描述符号。打开文件表中的每个有效项是一个(struct file*)，文件描述符号就是打开文件表项的下标。

```
fd = fdalloc(f)
```

顺便回答Q6：fdalloc 函数的作用是什么？

为传入的文件开辟（打开）一个文件表项，并返回对应的文件描述符。

sys_close 函数中proc->ofile[fd] = 0 这一句代表了什么？

将打开文件表的这一项置为NULL(无效化)。

ofile 变量还有什么时候被赋值过?

```
void exit(void){
......
   for(fd = 0; fd < NOFILE; fd++){
   //遍历当前进程打开的所有文件
      if(curproc->ofile[fd]){//关闭文件
         fileclose(curproc->ofile[fd]);//清除描述符
         curproc->ofile[fd] = 0;
      }
   }
......
}
```

ofile 变量还有什么时候被赋值过?

```
static int fdalloc(struct file *f){
    int fd;
    struct proc *curproc = myproc();//遍历当前进程的打开文件表
    for(fd = 0; fd < NOFILE; fd++){//找到一个空项
        if(curproc->ofile[fd] == 0){
            curproc->ofile[fd] = f; //将这个项指向所给的文件
            return fd;
        }
    }
    return -1;
}
```

filealloc 函数的作用是什么，ftable 是一个什么数据结构，里面总共多少项？

函数作用

开辟一个文件结构体并返回指向这个结构体的指针。

ftable

是一个系统中所有打开文件的表。

定义

```
//file.c

struct {

        struct spinlock lock;//操作锁

        struct file file[NFILE];//文件结构数组

} ftable;
```

项数

NFILE=100

顺便回答Q6：fdalloc 函数的作用是什么？

为传入的文件开辟（打开）一个文件表项，并返回对应的文件描述符。