File and Directory in Practice

Yajin Zhou (http://yajin.org)

Zhejiang University

Two Key Abstractions

- File
  - A linear array of bytes, with each you can read/write
  - Has a low-level name (user does not know) - **inode number**
    - Usually os does not know the exact type of the file
- Directory
  - Has a low level name
  - Its content is quite specific: contains a list of user-readable name to low-level name. Like (“foo”, inode number 10)
  - Each entry either points to file or other directory

File System Interface

- Create file

```
int fd = open("foo", O_CREAT | O_WRONLY | O_TRUNC);
```

- O_CREAT: create the file if not exists (not ***O_CREATE***)
- O_WRONLY: can only write to the file
- O_TRUNC: if the file exists, truncate it to zero bytes

The cat example

```
os@os:~/temp$ echo hello > foo
os@os:~/temp$ cat foo
hello
os@os:~/temp$
```

```
os@os:~/temp$ strace cat foo
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
open("foo", O_RDONLY)                   = 3
fstat(3, {st_mode=S_IFREG|0664, st_size=6, ...}) = 0
fadvise64(3, 0, 0, POSIX_FADV_SEQUENTIAL) = 0
mmap(NULL, 139264, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f703ea95000
read(3, "hello\n", 131072)              = 6
write(1, "hello\n", 6hello
)                       = 6
read(3, "", 131072)                     = 0
munmap(0x7f703ea95000, 139264)          = 0
close(3)                                = 0
close(1)                                = 0
close(2)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Seek

```
off_t lseek(int fildes, off_t offset, int whence);

If whence is SEEK_SET, the offset is set to offset bytes.
If whence is SEEK_CUR, the offset is set to its current
location plus offset bytes.
If whence is SEEK_END, the offset is set to the size of
the file plus offset bytes.
```

- Calling lseek does not perform a disk seek

fsync()

- write():
  - please write this data to persistent storage, at some point in the future. The file system, for performance reasons, will buffer such writes in memory for some time (say 5 seconds, or 30); at that later point in time, the write(s) will actually be issued to the storage device
- Fsync forces all dirty (not yet written) data to disk

```
int fd = open("foo", O_CREAT | O_WRONLY | O_TRUNC);
assert(fd > -1);
int rc = write(fd, buffer, size);
assert(rc == size);
rc = fsync(fd);
assert(rc == 0);
```

Getting Information About Files

```
struct stat {
    dev_t     st_dev;     /* ID of device containing file */
    ino_t     st_ino;     /* inode number */
    mode_t    st_mode;    /* protection */
    nlink_t   st_nlink;   /* number of hard links */
    uid_t     st_uid;     /* user ID of owner */
    gid_t     st_gid;     /* group ID of owner */
    dev_t     st_rdev;    /* device ID (if special file) */
    off_t     st_size;    /* total size, in bytes */
    blksize_t st_blksize; /* blocksize for filesystem I/O */
    blkcnt_t  st_blocks;  /* number of blocks allocated */
    time_t    st_atime;   /* time of last access */
    time_t    st_mtime;   /* time of last modification */
    time_t    st_ctime;   /* time of last status change */
};
```

```
os@os:~/temp$ stat foo
  File: 'foo'
  Size: 6               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 1328649     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-19 00:24:02.448286431 +0800
Modify: 2018-12-19 00:24:01.316296543 +0800
Change: 2018-12-19 00:24:01.316296543 +0800
 Birth: -
```

Information is kept in **inode**

Removing Files

```
os@os:~/temp$ strace rm foo
```

```
newfstatat(AT_FDCWD, "foo", {st_mode=S_IFREG|0664, st_size=6, ...}, AT_SYMLINK_NOFOLLOW) = 0
faccessat(AT_FDCWD, "foo", W_OK)        = 0
unlinkat(AT_FDCWD, "foo", 0)            = 0
lseek(0, 0, SEEK_CUR)                   = -1 ESPIPE (Illegal seek)
close(0)                                = 0
```

- Why we just “remove” or “delete” the file, but using “unlinkat”?

Making Directories

- strace mkdir foo

```
close(3)                                = 0
open("/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=2995, ...}) = 0
read(3, "# Locale name alias data base.\n#"..., 4096) = 2995
read(3, "", 4096)                       = 0
close(3)                                = 0
open("/usr/lib/locale/UTF-8/LC_CTYPE", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
mkdir("foo", 0777)                      = 0
close(1)                                = 0
close(2)                                = 0
```

```
os@os:~/temp/foo$ ls -al
total 8
drwxrwxr-x 2 os os 4096 Dec 20 23:56 .
drwxrwxr-x 3 os os 4096 Dec 20 23:56 ..
```


```
os@os:~/temp/foo$ ls -al
total 8
drwxrwxr-x 2 os os 4096 Dec 20 23:56 .
drwxrwxr-x 3 os os 4096 Dec 20 23:56 ..
```


- Even an empty directory has two entires

Reading Directories

```c
int main(int argc, char *argv[]) {
    DIR *dp = opendir(".");
    assert(dp != NULL);
    struct dirent *d;
    while ((d = readdir(dp)) != NULL) {
        printf("%d %s\n", (int) d->d_ino, d->d_name);
    }
    closedir(dp);
    return 0;
}
```

```
os@os:~/temp/foo$ ls -l
total 16
-rwxrwxr-x 1 os os 8760 Dec 21 00:00 main
-rw-rw-r-- 1 os os  232 Dec 21 00:00 main.c
```

```
os@os:~/temp/foo$ ./main
1328648 main
1328642 ..
1328651 main.c
1328649 .
os@os:~/temp/foo$ _
```

```c
struct dirent {
    char           d_name[256]; /* filename */
    ino_t          d_ino;       /* inode number */
    off_t          d_off;       /* offset to the next dirent */
    unsigned short d_reclen;    /* length of this record */
    unsigned char  d_type;      /* type of file */
};
```

Hard Links

- link() system call: system call takes two arguments, an old pathname and a new one
  - another way to refer to the same file

```
os@os:~/temp/foo$ echo hello > file
os@os:~/temp/foo$ ln file file2
os@os:~/temp/foo$ cat file2
hello
os@os:~/temp/foo$ ls -i file file2
1328650 file  1328650 file2
os@os:~/temp/foo$
```

  - same inode number

Why removing a file calls unlink

- Create a file
  - making a structure (the inode) that will track virtually all relevant information about the file
  - **linking** a human-readable name to that file (or inode), and putting that link into a directory
- To remove a file, we **unlink** it

Reference count

```
os@os:~/temp/foo$ rm file
os@os:~/temp/foo$ cat file2
hello
os@os:~/temp/foo$
```

- rm: unlink the file, and check the reference count of the **inode**

```
os@os:~/temp/foo$ echo hello >file
os@os:~/temp/foo$ stat file
  File: 'file'
  Size: 6               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 1328650     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-21 00:09:14.763058468 +0800
Modify: 2018-12-21 00:09:14.763058468 +0800
Change: 2018-12-21 00:09:14.763058468 +0800
 Birth: -
os@os:~/temp/foo$ ln file file2
os@os:~/temp/foo$ stat file2
  File: 'file2'
  Size: 6               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 1328650     Links: 2
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-21 00:09:14.763058468 +0800
Modify: 2018-12-21 00:09:14.763058468 +0800
Change: 2018-12-21 00:09:24.247463616 +0800
 Birth: -
os@os:~/temp/foo$ ln file2 file3
os@os:~/temp/foo$ stat file
  File: 'file'
  Size: 6               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 1328650     Links: 3
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-21 00:09:14.763058468 +0800
Modify: 2018-12-21 00:09:14.763058468 +0800
Change: 2018-12-21 00:09:33.975880487 +0800
 Birth: -
```

```
os@os:~/temp/foo$ rm file
os@os:~/temp/foo$ stat file2
  File: 'file2'
  Size: 6               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 1328650     Links: 2
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-21 00:09:14.763058468 +0800
Modify: 2018-12-21 00:09:14.763058468 +0800
Change: 2018-12-21 00:09:40.776172622 +0800
 Birth: -
os@os:~/temp/foo$ rm file2
os@os:~/temp/foo$ stat file3
  File: 'file3'
  Size: 6               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 1328650     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-21 00:09:14.763058468 +0800
Modify: 2018-12-21 00:09:14.763058468 +0800
Change: 2018-12-21 00:09:47.092444491 +0800
 Birth: -
```

Symbolic links

```
os@os:~/temp/foo$ echo hello > file
os@os:~/temp/foo$ ln -s file file2
```

```
os@os:~/temp/foo$ ls -l
total 4
-rw-rw-r-- 1 os os 6 Dec 21 00:11 file
lrwxrwxrwx 1 os os 4 Dec 21 00:12 file2 -> file
os@os:~/temp/foo$ stat file
  File: 'file'
  Size: 6               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 1328650     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-21 00:11:54.636605978 +0800
Modify: 2018-12-21 00:11:54.636605978 +0800
Change: 2018-12-21 00:11:54.636605978 +0800
 Birth: -
os@os:~/temp/foo$ stat file2
  File: 'file2' -> 'file'
  Size: 4               Blocks: 0          IO Block: 4096   symbolic link
Device: 801h/2049d      Inode: 1328653     Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/      os)   Gid: ( 1000/      os)
Access: 2018-12-21 00:12:02.300152148 +0800
Modify: 2018-12-21 00:12:00.960229971 +0800
Change: 2018-12-21 00:12:00.960229971 +0800
 Birth: -
```

```
os@os:~/temp/foo$ rm file
os@os:~/temp/foo$ cat file2
cat: file2: No such file or directory
os@os:~/temp/foo$ ls -l
total 0
lrwxrwxrwx 1 os os 4 Dec 21 00:12 file2 -> file
os@os:~/temp/foo$
```

Mounting a file system

```
os@os:~/temp/foo$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=475364k,nr_inodes=118841,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=100384k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
```