

Операционные системы

Файловый ввод-вывод

Олег Французов
2017

Файловая система

- Файл – набор данных, хранимый на ВЗУ и имеющий имя
- Файловая система – подсистема ОС, отвечающая за хранение данных на ВЗУ в виде именованных файлов
- А также структура данных на ВЗУ

Каталоги

- Каталог – особый тип файла, хранящий другие файлы
- Корневой каталог

Путь к файлу

- / – разделитель
- Локальное имя файла
- Абсолютный путь к файлу
- .. – родительский каталог

Относительный путь к файлу

```
$ touch /home/work/proj/task/prog.c
```

```
$ touch /home/fun/books/alice.txt
```

```
$ cd /home/work/proj
```

```
$ cat task/prog.c
```

```
...
```

```
$ cat ../../fun/books/alice.txt
```

```
...
```

Монтирование

- Единое дерево каталогов
- Нет букв дисков или аналогов
- На флэшке: `work/prog.c`
- Смонтирована на: `/mnt/flash`
- Полный путь: `/mnt/flash/work/prog.c`

Имена файлов и индексные дескрипторы

- В каталоге:
имя файла + номер индексного дескриптора
- ИД (i-node) содержит всю
информацию о файле
- Имя: любые символы, '\0' и '/'
- Но лучше воздержаться от:
*!?, служебных, пробела, не-латиницы

Жесткие ссылки

- 2+ имен ссылаются на один i-node
- Файл удаляется, когда нет ссылок
- Жесткие ссылки на каталоги запрещены

```
int link(const char *oldpath,  
         const char *newpath);  
int unlink(const char *path);
```

```
$ ln oldpath newpath
```

Символические ссылки

- Типы файлов: каталоги
- Симв. ссылка содержит имя другого файла
- Висячие ссылки

```
int symlink(const char *oldpath,  
            const char *newpath);
```

```
$ ln -s oldpath newpath
```

Права доступа к файлу

- Владелец, группа, остальные
- SetUid, SetGid, Sticky

```
hilt:~ oleg$ ls -l /bin/cat
-rwxr-xr-x  1 root  wheel  23520 Mar 12  2016 /bin/cat
^ 0755
```

Права доступа к файлу

```
$ chmod 644 myfile.c
```

```
$ chmod g+rw,o+r file.txt
```

```
# [ugoa] [+--=] [rwxSxtugo]
```

Открытие файла

```
int open(const char *name, int mode);  
int open(const char *name, int mode,  
         int perms);
```

mode:

```
O_RDONLY, O_WRONLY, O_RDWR  
O_APPEND, O_CREAT, O_TRUNC, O_EXCL
```

perms (-umask) = 0666 (типично)

Открытие файла

```
int open(const char *name, int mode);  
int open(const char *name, int mode,  
         int perms);
```

-> fd (файловый дескриптор) или -1

0 == stdin, 1 == stdout, 2 == stderr

Чтение из файла

```
int read(int fd, void *buf, int len);
```

→ сколько прочитано байт
или -1 в случае ошибки
или 0 при конце файла

блокируется, если нет данных в потоке

Запись в файл

```
int write(int fd, const void *buf,  
          int len);
```

→ сколько записано байт
или -1 в случае ошибки

Заккрытие файла

```
int close(int fd);
```

→ 0 в случае успеха
или -1 в случае ошибки

Все ФД закрываются автоматически
при завершении процесса

Позиционирование

```
int lseek(int fd, int offset,  
          int whence);
```

whence: SEEK_SET, SEEK_CUR, SEEK_END

→ новая позиция в файле
или -1 в случае ошибки

Устройства как файлы

```
$ cat /dev/cdrom > image.iso
```

```
$ cat myfile.ps > /dev/lp0
```

Типы устройств

- Поточковые (байт-ориентированные)
 - Терминал, принтер, мышь
 - `/dev/null`, `/dev/zero`, `/dev/random`
 - Не поддерживают `lseek()`
- Блочные (блок-ориентированные)
 - Диски

Дополнительные операции над устройствами

```
int ioctl(int fd, int request, ...);
```

```
// Пример:
```

```
int fd = open("/dev/cdrom",  
             O_RDONLY|O_NONBLOCK);
```

```
ioctl(fd, CDROMEJECT);
```

```
ioctl(fd, CDROMCLOSETRAY);
```

man 2 ioctl

```
IOCTL(2)                                BSD System Calls Manual                                IOCTL(2)

NAME
    ioctl -- control device

SYNOPSIS
    #include <sys/ioctl.h>

    int
    ioctl(int fildes, unsigned long request, ...);

DESCRIPTION
    The ioctl() function manipulates the underlying device parameters of special files. In particular, many operating characteristics of character special files (e.g. terminals) may be controlled with ioctl() requests. The argument fildes must be an open file descriptor.

    An ioctl request has encoded in it whether the argument is an ``in'' parameter or ``out'' parameter, and the size of the argument argp in bytes. Macros and defines used in specifying an ioctl request are located in the file <sys/ioctl.h>.

RETURN VALUES
    If an error has occurred, a value of -1 is returned and errno is set to
```

Q & A