



Операционные системы

Лекция 2. Файловая система

Кафедра ВС, Бочкарев Борис Вячеславович

Осень-2019

Требования к файловой системе

1. Оно должно предоставлять возможность хранения огромного количества информации.
 2. Информация должна пережить прекращение работы использующего ее процесса.
 3. К информации должны иметь одновременный доступ несколько процессов.
-

Файл

Файл - именованная область данных на носителе информации, ограждает пользователя от подробностей о способе и месте хранения информации и деталей фактической работы дисковых устройств.

Типы файлов:

1. Обычные - содержат информацию пользователя.
2. Каталоги - системные файлы, предназначенные для поддержки структуры файловой системы.
3. Символьные специальные файлы - отображение последовательных устройств ввода-вывода (принтер, сеть, терминал).
4. Блочные специальные файлы - отображение дисков.

Операции с файлами

1. Create
 2. Delete
 3. Open
 4. Close
 5. Read
 6. Write
 7. Append
 8. Seek
 9. Rename
-

Доступ к файлам в ОС Linux

```
#include <fcntl.h>
```

```
int open(const char *path, int oflag, ...);
```

O_RDONLY : открыть только на чтение

O_WRONLY : открыть только на запись

O_RDWR : открыть на запись и на чтение

O_SYNC : синхронный ввод-вывод

O_APPEND : открыть файл на добавление

O_CREAT : создать файл, если его нет

O_TRUNC : открыть файл с очисткой

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

```
ssize_t read(int fildes, void *buf, size_t nbyte); // int[-1,0,...]
```

```
ssize_t write(int fildes, const void *buf, size_t nbyte); // int[-1, 0,...]
```

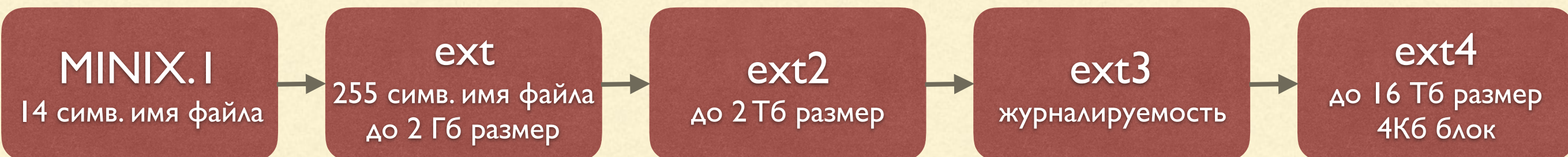
```
#include <io.h>
```

```
off_t lseek(int fildes, off_t offset, int whence); // SEEK_SET, SEEK_CUR, SEEK_END
```

```
stdin == 0
```

```
stdout == 1
```

Файловая система UNIX



bin	Двоичные (исполняемые) программы
dev	Специальные файлы для устройств ввода-вывода
etc	Разные системные файлы
lib	Библиотеки
usr	Каталоги пользователей

Абсолютный и
относительный путь?

Доступ к файлам в ОС Windows

CreateFile(szName, dwAccess, dwShareMode, lpSecurityAttributes,
dwCreationDisposition, dwFlags, hTemplateFile); // создание или открытие

szName - имя файла

dwAccess - доступ: GENERIC_READ, GENERIC_WRITE

dwShareMode - совместный доступ: FILE_SHARE_READ, FILE_SHARE_WRITE, FILE_SHARE_DELETE

lpSecurityAttributes - атрибут защиты // NULL

hTemplateFile - шаблон файла // NULL

dwCreationDisposition - как открыть: CREATE_NEW, CREATE_ALWAYS, OPEN_EXISTING, OPEN_ALWAYS

dwFlags - атрибуты у файла: FILE_ATTRIBUTE_NORMAL

ReadFile(hFile, lpBuff, dwBuffSize, &dwCount, NULL);

WriteFile(hFile, lpBuff, dwBuffSize, &dwCount, NULL);

hFile - файловый дескриптор

lpBuff - буфер для чтения/записи

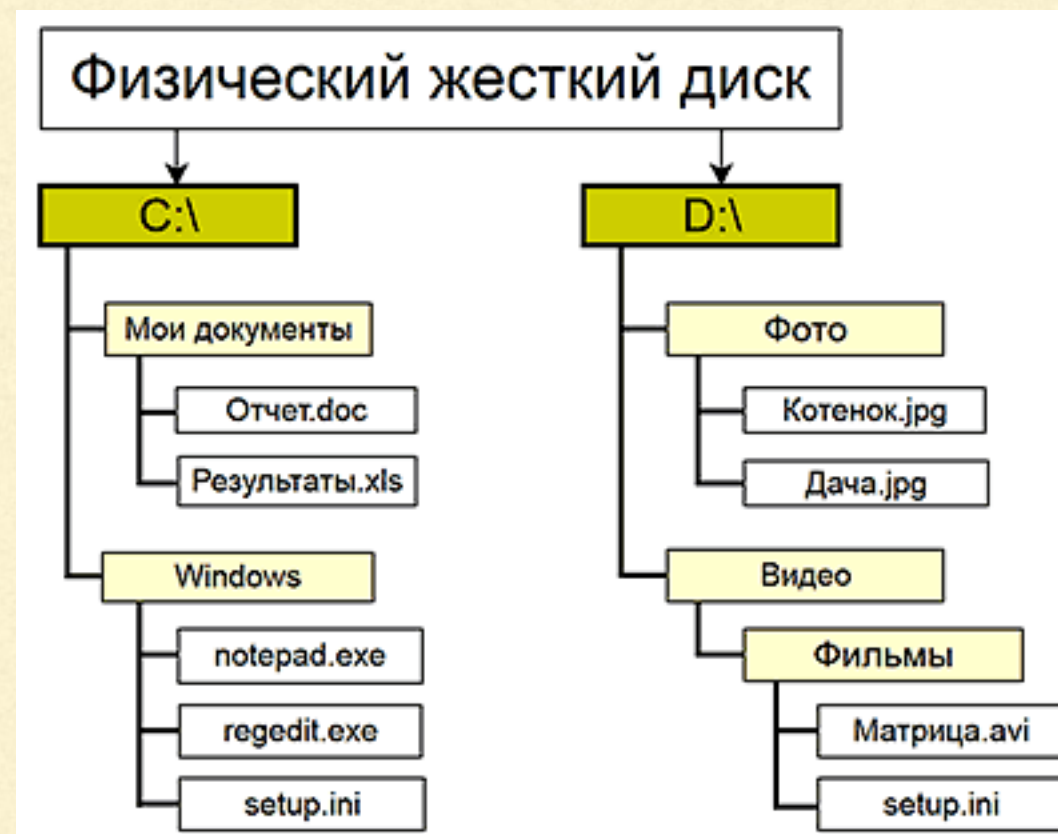
dwBuffSize - сколько необходимо считать/записать

dwCount - сколько реально байт прочитано/записано

Файловая система Windows



18 446 744 073 709 551 616 байт



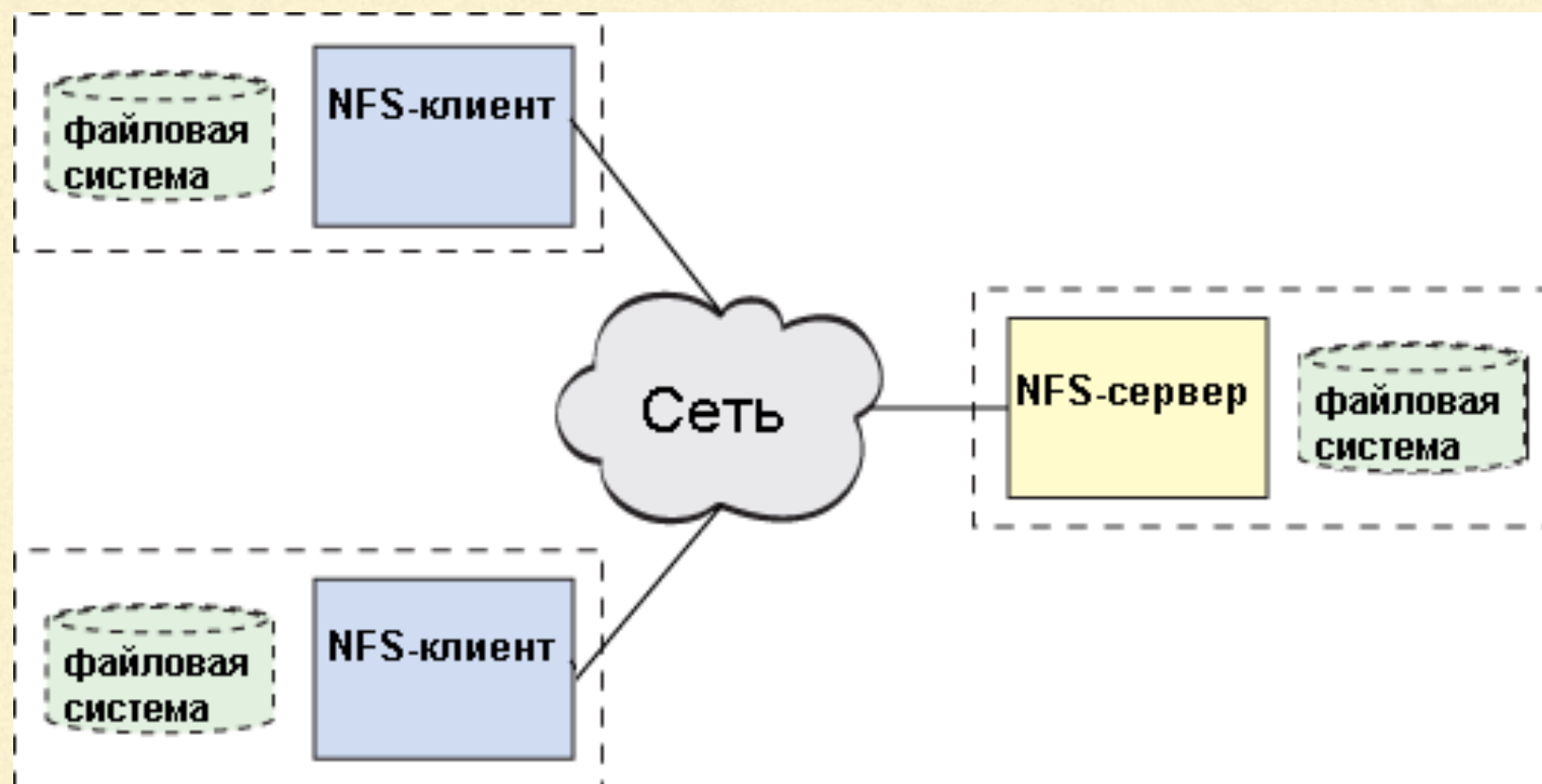
Дефрагментация диска

Процесс перераспределения фрагментов файлов на дисках для обеспечения непрерывной последовательности кластеров.

Дефрагментацию на flash и ssd проводить не рекомендуется!

NFS (Network File System)

Протокол сетевого доступа к файловым системам. Позволяет подключать (монтировать) удаленные файловые системы через сеть.



SYSFS

Виртуальная файловая система, экспортирующая в пользовательское пространство информацию от ядра Linux о присутствующих устройствах и драйверах.

Монтируется в точку */sys/*

/sys/devices/ - дерево устройств

/sys/bus/ - шины, зарегистрированные в ядре

/sys/drivers/ - каталог драйверов

/sys/block/ - каталог блочных устройств

/sys/class/ - группировка устройств в классы

proc

Виртуальная файловая система, представляющая собой интерфейс к нескольким структурам данных ядра. В /proc/ для каждого запускаемого процесса создается подкаталог, названный по pid процесса.

Служебные файлы:

loadavg - среднее число процессов, пытающихся запуститься в одно и то же время за последние 1 мин, 5 мин и 15 мин

meminfo - информацию о доступной оперативной памяти

uptime - время работы системы: общее и идеальное

proc. Отображение процессов

В каждой директории процесса существуют следующие файлы:

cmdline - командная строка процесса

exe - исполняемый файл процесса

fd - директория с файловыми дескрипторами

root - ссылка на корневой каталог процесса

stat - статусная информация о процесса

https://www.opennet.ru/docs/RUS/kernel2prog/linux_khg_03.html

Работа с директориями в ОС Linux

```
#include <sys/types.h>
#include <dirent.h>
DIR *opendir(const char *name);
```

Пример

```
struct dirent *pDirent;
DIR *pDir = opendir (argv[1]);
while ((pDirent = readdir(pDir)) != NULL) {
    printf ("%s\n", pDirent->d_name);
}
closedir (pDir);
```

```
struct dirent {
    ino_t      d_ino;        /* inode */
    off_t      d_off;        /* not used */
    unsigned short d_reclen; /* размер записи */
    unsigned char d_type;    /* тип записи */
    char        d_name[256]; /* имя записи */
};
```

d_type in dirent

DT_BLK : блочное устройство
DT_CHR : символьное устройство
DT_DIR : директория
DT_FIFO : именованный канал
DT_LNK : ссылка
DT_REG : регулярный файл
DT SOCK : UNIX сокет
DT_UNKNOWN : ???

Информация о файле в ОС Linux

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
```

```
int stat(const char *file_name, struct stat *buf);
```

```
struct stat {
```

```
    dev_t      st_dev;      /* устройство */
```

```
    ino_t      st_ino;      /* inode */
```

```
    mode_t     st_mode;     /* режим доступа */
```

```
    nlink_t    st_nlink;    /* количество жестких ссылок */
```

```
    uid_t      st_uid;      /* идентификатор пользователя-владельца */
```

```
    gid_t      st_gid;      /* идентификатор группы-владельца */
```

```
    dev_t      st_rdev;     /* тип устройства */
```

```
    off_t      st_size;     /* общий размер в байтах */
```

```
    blksize_t  st_blksize;  /* размер блока ввода-вывода в файловой системе */
```

```
    blkcnt_t   st_blocks;   /* количество выделенных блоков */
```

```
    time_t     st_atime;    /* время последнего доступа */
```

```
    time_t     st_mtime;    /* время последней модификации */
```

```
    time_t     st_ctime;    /* время последнего изменения */
```

```
};
```
