

Системные вызовы для чтения атрибутов файла

Прототипы системных вызовов

```
#include <sys/stat.h>
#include <unistd.h>
int stat(char *filename, struct stat *buf);
int fstat(int fd, struct stat *buf);
int lstat(char *filename, struct stat *buf);
```

Описание системных вызовов

Настоящее описание не является полным описанием этих системных вызовов, а приспособлено для целей данного курса. Для получения полного описания обращайтесь в UNIX Manual.

Системные вызовы **stat**, **fstat** и **lstat** служат для получения информации об атрибутах файла.

Системный вызов **stat** читает информацию об атрибутах файла, на чье имя указывает параметр **filename**, и заполняет ими структуру, расположенную по адресу **buf**. Заметим, что имя файла должно быть полным, либо строиться относительно той директории, которая является текущей для процесса, совершившего вызов. Если имя файла относится к файлу типа "связь", то читается информация (рекурсивно!) об атрибутах файла, на который указывает символическая связь.

Системный вызов **lstat** идентичен системному вызову **stat** за одним исключением: если имя файла относится к файлу типа "связь", то читается информация о самом файле типа "связь".

Системный вызов **fstat** идентичен системному вызову **stat**, только файл задается не именем, а своим файловым дескриптором (естественно, файл к этому моменту должен быть открыт).

Для системных вызовов **stat** и **lstat** процессу не нужны никакие права доступа к указанному файлу, но могут понадобиться права для поиска во всех директориях, входящих в специфицированное имя файла.

Структура **stat** в различных версиях UNIX может быть описана по-разному. В Linux она содержит следующие поля:

```
struct stat {
```

```
    dev_t st_dev; /* устройство, на котором расположен файл */
    ino_t st_ino; /* номер индексного узла для файла */
    mode_t st_mode; /* тип файла и права доступа к нему */
    nlink_t st_nlink; /* счетчик числа жестких связей */
```

```

uid_t st_uid; /* идентификатор пользователя владельца */
gid_t st_gid; /* идентификатор группы владельца */
dev_t st_rdev; /* тип устройства для специальных файлов устройств */
off_t st_size; /* размер файла в байтах (если определен для данного типа
файлов) */
unsigned long st_blksize; /* размер блока для файловой системы */
unsigned long st_blocks; /* число выделенных блоков */
time_t st_atime; /* время последнего доступа к файлу */
time_t st_mtime; /* время последней модификации файла */
time_t st_ctime; /* время создания файла */

}

```

Для определения типа файла можно использовать следующие логические макросы, применяя их к значению поля **st_mode**:

```

S_ISLNK(m) - файл типа "связь"?
S_ISREG(m) - регулярный файл?
S_ISDIR(m) - директория?
S_ISCHR(m) - специальный файл символьного устройства?
S_ISBLK(m) - специальный файл блочного устройства?
S_ISFIFO(m) - файл типа FIFO?
S_ISSOCK(m) - файл типа socket?

```

Младшие 9 бит поля **st_mode** определяют права доступа к файлу подобно тому, как это делается в [маске создания файлов текущего процесса](#).

Системный вызов возвращает значение 0 при нормальном завершении и значение -1 при возникновении ошибки.