Лабораторная 7

Используемые функции:

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

int stat(const char *restrict pathname,
         struct stat *restrict statbuf);
```

Эти функции возвращают информацию о файле в буфер, на который указывает второй аргумент . Для самого файла не требуется никаких разрешений , только право выполнить поиск для всех каталогов в pathname, которые ведут к файлу. stat() на который указывает путь, переданный первым аргументом

```
struct stat {
    dev_t     st_dev;         /* ID of device containing file */
    ino_t     st_ino;         /* Inode number */
    mode_t    st_mode;        /* File type and mode */
    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;     /* Block size for filesystem I/O */
    blkcnt_t  st_blocks;      /* Number of 512B blocks allocated */

    /* Since Linux 2.6, the kernel supports nanosecond
       precision for the following timestamp fields.
       For the details before Linux 2.6, see NOTES. */

    struct timespec st_atim;  /* Time of last access */
    struct timespec st_mtim;  /* Time of last modification */
    struct timespec st_ctim;  /* Time of last status change */

#define st_atime st_atim.tv_sec      /* Backward compatibility */
#define st_mtime st_mtim.tv_sec
#define st_ctime st_ctim.tv_sec
};
```

st_mode - Это поле содержит тип файла и режим (mode).

Возвращаемое значение:

успех — ноль

При ошибке возвращается значение -1, а значение errno устанавливается для указания ошибки.

Возможные ошибки:

EACCES Отказано в разрешении на поиск для одного из каталогов в префиксе пути pathname.

ENAMETOOLONG слишком длинный путь.

ENOENT аргумент pathname не существует или является висячей символической ссылкой.

ENOMEM Недостаточно памяти

ENOTDIR
Компонент префикса пути pathname не является каталогом.

EOVERFLOW Путь относится к файлу, размер, номер inode или количество блоков которого не могут быть представлены соответственно в типах из структуры stat.

Можем на основе этой структуры, а в частности поля st_mode, опередить тип файла.

в POSIX определяются дополнительные макросы, позволяющие проверять тип файла в st_mode будет написано более кратко:

S_ISREG(m)
 это обычный файл?

S_ISDIR(m)
Каталог?

S_ISCHR(m)
Символьное устройство?

S_ISBLK (m)

блокирующее устройство?

S_ISFIFO(m)
FIFO (именованный канал)?

S_ISLNK (m)
Символическая ссылка? (Не в POSIX.1-1996.)

S_ISSOCK(m)
сокет? (Не в POSIX.1-1996.)

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

DIR *opendir(const char *name);
```

Функция opendir() открывает поток каталогов, соответствующий имени переданного каталога, и возвращает указатель на поток каталогов . Поток указывает в первую запись в каталоге.

Возвращаемое значение:

успех — указатель на поток каталогов.

При ошибке возвращается значение NULL, а значение errno устанавливается для указания ошибки.

Возможные ошибки:

EACCES недостаточно прав для открытия каталога

EMFILE Достигнут лимит на количество открытых файловых дескрипторов для каждого процесса.

ENFILE Достигнут общесистемный лимит на общее количество открытых файлов

ENOENT Каталог не существует, или имя является пустой строкой.

ENOMEM Недостаточно памяти для завершения операции.

ENOTDIR Имя переданное функции не является каталогом.

```
#include <dirent.h>

struct dirent *readdir(DIR *dirp);
```

Функция readdir() возвращает указатель на другую структуру представляющий следующую запись каталога в переданном потоке каталогов. Он возвращает значение NULL при достижении конца потока каталогов или при возникновении ошибки.

```
struct dirent {
    ino_t          d_ino;       /* Inode number */
    off_t          d_off;       /* Not an offset; see below */
    unsigned short d_reclen;    /* Length of this record */
    unsigned char  d_type;      /* Type of file; not supported
                                   by all filesystem types */
    char           d_name[256]; /* Null-terminated filename */
};
```

d_name This field contains the null terminated filename.

Возвращаемое значение:

успех — указатель на структуру dirent.

Если достигнут конец потока каталогов, возвращается значение NULL и errno не изменен. Если возникает ошибка, возвращается значение NULL и значение errno установлено для указания на ошибку.

Чтобы отличить конец потока от ошибки, установите значение errno равным нулю перед вызовом readdir() а затем проверьте значение errno, если возвращается значение NULL.

Возможные ошибки:

EBADF Недопустимый дескриптор потока каталогов dirp

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

int closedir(DIR *dirp);
```

Функция closedir() закрывает поток каталогов, связанный с переданным указаетелем. Успешный вызов closedir() также закрывает базовый файловый дескриптор. Переданный дескриптор потока каталогов недоступен после этого вызова.

Возращаемое значение:

Функция closedir() возвращает 0 при успешном выполнении. При ошибке значение -1 равно возвращается, и значение errno устанавливается для указания на ошибку.

Возможные ошибки:

EBADF Недопустимый дескриптор потока каталогов dirp.