

# **УПРАВЛЕНИЕ ФАЙЛАМИ**

Системные вызовы и  
библиотеки Unix SVR4

# ЦЕЛИ РАЗДЕЛА

По окончании этого раздела вы сможете:

- определить доступность файла
- определить состояние файла
- изменить права доступа файла
- изменить владельца файла
- изменить время последнего доступа и изменения файла

# access(2)

## ИСПОЛЬЗОВАНИЕ

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

```
int access (const char *path, int  
    amode) ;
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

# Флаги access(2)

- R\_OK чтение
- W\_OK изменение
- X\_OK исполнение (поиск)
- F\_OK существование

# ulimit(2)

## ИСПОЛЬЗОВАНИЕ

```
#include <ulimit.h>
```

```
long ulimit (int cmd,.../* newlimit */);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - неотрицательное число

неуспех - -1 и errno установлена

# Команды ulimit(2)

**UL\_GETFSIZE** Возвращает текущее ограничение процесса на размер файла.

**UL\_SETFSIZE** Устанавливает ограничение на размер файла.

**UL\_GMEMLIM** Возвращает максимально допустимое значение границы выделяемой памяти.

**UL\_GDESLIM** Возвращает ограничение, устанавливаемое программно при конфигурации системы, на число файлов, которые процесс может открыть.

# umask(2)

## ИСПОЛЬЗОВАНИЕ

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

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

```
mode_t umask (mode_t cmask);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

предыдущее значение маски создания  
файла

# stat(2)

## ИСПОЛЬЗОВАНИЕ

```
#include <sys/types.h>
#include <sys/stat.h>
int stat (const char *path,
          struct stat *buf);
int lstat(const char *path,
          struct stat *buf);
int fstat (int fildes, struct stat *buf);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена



# struct stat

```
sys/stat.h:
struct stat {
    dev_t st_dev;
    ino_t st_ino;
    mode_t st_mode;
    nlink_t st_nlink;
    uid_t st_uid;
    gid_t st_gid;
    dev_t st_rdev;
    off_t st_size;
    time_t st_atim, st_mtim, st_ctim; };
#define st_atime st_atim.tv_sec;
#define st_mtime st_mtim.tv_sec;
```

# getpwent(3C)

## ИСПОЛЬЗОВАНИЕ

```
#include <pwd.h>

struct passwd *getpwent (void);
struct passwd *getpwuid (uid_t uid);
struct passwd *getpwnam (const char *name);
void setpwent (void);
void endpwent (void);
struct passwd *fgetpwent (FILE *f);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - указатель на структуру  
неуспех - NULL-указатель

# struct passwd

```
struct passwd {  
    char *pw_name;  
    char *pw_passwd;  
    uid_t pw_uid;  
    gid_t pw_gid;  
    char *pw_age;  
    char *pw_comment;  
    char *pw_gecos;  
    char *pw_dir;  
    char *pw_shell;  
};
```

# getgrent(3C)

## ИСПОЛЬЗОВАНИЕ

```
#include <grp.h>

struct group *getgrent (void);
struct group *getgrgid (gid_t gid);
struct group *getgrnam (char *name);
void setgrent (void);
void endgrent (void);
struct group *fgetgrent (FILE *f);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - указатель на структуру  
неуспех - NULL-указатель

# chmod(2)

## ИСПОЛЬЗОВАНИЕ

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

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

```
int chmod (const char *path,  
           mode_t mode);
```

```
int fchmod(int fildes, mode_t mode);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

# chown(2)

## ИСПОЛЬЗОВАНИЕ

```
#include <unistd.h>
#include <sys/stat.h>
int chown (const char *path, uid_t owner,
           gid_t group);
int lchown(const char *path, uid_t owner,
           gid_t group);
int fchown(int fildes, uid_t owner,
           gid_t group);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

# utime(2)

## ИСПОЛЬЗОВАНИЕ

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

```
#include <utime.h>
```

```
int utime (const char *path,  
           const struct utimbuf *times);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

# truncate(3C)

## ИСПОЛЬЗОВАНИЕ

```
#include <unistd.h>

int truncate(char *path,
             off_t length);

int ftruncate(int fildes,
             off_t length);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена



# pathfind(3G)

## ИСПОЛЬЗОВАНИЕ

```
gcc -lgen
```

```
#include <libgen.h>
```

```
char *pathfind(const char *path,  
               const char *name, const char *mode);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - указатель на путь

неуспех - NULL

# mktemp(3C)

## ИСПОЛЬЗОВАНИЕ

```
#include <stdlib.h>
```

```
char *mktemp(char * template);
```

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - указатель на имя временного  
файла

неуспех - NULL