

UNIX File System Calls

Basic File Operations - open

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

```
int open(const char *pathname, int flags);
int open(const char *pathname, int flags, mode_t mode);
```

```
int creat(const char *pathname, mode_t mode);
```

- Common Flags:

- 0_RDONLY
 - 0_CREAT
 - 0_WRONLY
 - 0_RDWR
 - 0_APPEND

- Mode: Like directory entry
- Returns: File Descriptor, or if negative, an error

Basic Operations - read, write

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

```
ssize_t read(int fd, void *buf, size_t count);
```

```
ssize_t write(int fd, const void *buf, size_t count);
```

- Returns actual number of bytes read/written

Basic Operations - lseek

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

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

- Whence choices:

SEEK_SET - absolute offset (bytes)
SEEK_CUR - current location plus offset
SEEK_END - last location plus offset

- Returns resulting offset location, or error if negative

Basic Operative - close

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

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

Basic Operation - unlink

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

```
int unlink(const char *pathname);
```

- Disassociate directory entry from i-node.
- Actual file is deleted after last reference to it is unlinked.

File Operations - stat, fstat, lstat

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

```
int stat(const char *pathname, struct stat *statbuf);  
int fstat(int fd, struct stat *statbuf);  
int lstat(const char *pathname, struct stat *statbuf);
```

- statbuf is a struct defined in sys/stat.h

File operations - statbuf

```
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
};
```