The Open Group Base Specifications Issue 7, 2018 edition
IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2008)

NAME

sys/stat.h - data returned by the stat() function

SYNOPSIS

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

DESCRIPTION

The *<sys/stat.h>* header shall define the structure of the data returned by the *fstat()*, *lstat()*, and *stat()* functions.

The *<sys/stat.h>* header shall define the **stat** structure, which shall include at least the following members:

dev_t st_dev	Device ID of device containing file.
ino_t st_ino	File serial number.
mode_t st_mode	Mode of file (see below).
nlink_t st_nlink	Number of hard links to the file.
uid_t st_uid	User ID of file.
gid_t st_gid	Group ID of file.
[XSI] ⌦	
dev_t st_rdev	Device ID (if file is character or block special).
⌫	
off_t st_size	For regular files, the file size in bytes. For symbolic links, the length in bytes of the pathname contained in the symbolic link.
[SHM] ⌦	
	For a shared memory object, the length in bytes.
⌫	
[TYM] ⌦	
	For a typed memory object, the length in bytes.
⌫	
	For other file types, the use of this field is unspecified.
struct timespec st_atim	Last data access timestamp.
struct timespec st_mtim	Last data modification timestamp.
struct timespec st_ctim	Last file status change timestamp.
[XSI] ⌦	
blksize_t st_blksize	A file system-specific preferred I/O block size for this object. In some file system types, this may vary from file to file.
blkcnt_t st_blocks	Number of blocks allocated for this object.
⌫	

The *st_ino* and *st_dev* fields taken together uniquely identify the file within the system.

The *<sys/stat.h>* header shall define the [XSI] ⌦ **blkcnt_t**, **blksize_t**, ⌫ **dev_t**, **ino_t**, **mode_t**, **nlink_t**, **uid_t**, **gid_t**, **off_t**, and **time_t** types as described in *<sys/types.h>*.

The *<sys/stat.h>* header shall define the **timespec** structure as described in *<time.h>*. Times shall be given in seconds since the Epoch.

Which structure members have meaningful values depends on the type of file. For further information, see the descriptions of *fstat()*, *lstat()*, and *stat()* in the System Interfaces volume of POSIX.1-2017.

For compatibility with earlier versions of this standard, the *st_atime* macro shall be defined with the value *st_atim.tv_sec*. Similarly, *st_ctime* and *st_mtime* shall be defined as macros with the values *st_ctim.tv_sec* and *st_mtim.tv_sec*, respectively.

The *<sys/stat.h>* header shall define the following symbolic constants for the file types encoded in type **mode_t**. The values shall be suitable for use in **#if** preprocessing directives:

S_IFMT
: [XSI] ⌦ Type of file.

S_IFBLK
: Block special.

S_IFCHR
: Character special.

S_IFIFO
: FIFO special.

S_IFREG
: Regular.

S_IFDIR
: Directory.

S_IFLNK
: Symbolic link.

S_IFSOCK
: Socket.

⌫

The *<sys/stat.h>* header shall define the following symbolic constants for the file mode bits encoded in type **mode_t**, with the indicated numeric values. These macros shall expand to an expression which has a type that allows them to be used, either singly or OR'ed together, as the third argument to *open()* without the need for a **mode_t** cast. The values shall be suitable for use in **#if** preprocessing directives.

Name	Numeric Value	Description
S_IRWXU	0700	Read, write, execute/search by owner.
S_IRUSR	0400	Read permission, owner.
S_IWUSR	0200	Write permission, owner.
S_IXUSR	0100	Execute/search permission, owner.
S_IRWXG	070	Read, write, execute/search by group.
S_IRGRP	040	Read permission, group.
S_IWGRP	020	Write permission, group.
S_IXGRP	010	Execute/search permission, group.
S_IRWXO	07	Read, write, execute/search by others.
S_IROTH	04	Read permission, others.
S_IWOTH	02	Write permission, others.
S_IXOTH	01	Execute/search permission, others.
S_ISUID	04000	Set-user-ID on execution.
S_ISGID	02000	Set-group-ID on execution.
[XSI] ⌦ S_ISVTX	01000	On directories, restricted deletion flag. ⌫

The following macros shall be provided to test whether a file is of the specified type. The value *m* supplied to the macros is the value of *st_mode* from a **stat** structure. The macro shall evaluate to a non-zero value if the test is true; 0 if the test is false.

S_ISBLK(*m*)
Test for a block special file.

S_ISCHR(*m*)
Test for a character special file.

S_ISDIR(*m*)
Test for a directory.

S_ISFIFO(*m*)
Test for a pipe or FIFO special file.

S_ISREG(*m*)
Test for a regular file.

S_ISLNK(*m*)
Test for a symbolic link.

S_ISSOCK(*m*)
Test for a socket.

The implementation may implement message queues, semaphores, or shared memory objects as distinct file types. The following macros shall be provided to test whether a file is of the specified type. The value of the *buf* argument supplied to the macros is a pointer to a **stat** structure. The macro shall evaluate to a non-zero value if the specified object is implemented as a distinct file type and the specified file type is contained in the **stat** structure referenced by *buf*. Otherwise, the macro shall evaluate to zero.

S_TYPEISMQ(*buf*)
Test for a message queue.

S_TYPEISSEM(*buf*)
Test for a semaphore.

S_TYPEISSHM(*buf*)
Test for a shared memory object.

[TYM] ⌦ The implementation may implement typed memory objects as distinct file types, and the following macro shall test whether a file is of the specified type. The value of the *buf* argument supplied to the macros is a pointer to a **stat** structure. The macro shall evaluate to a non-zero value if the specified object is implemented as a distinct file type and the specified file type is contained in the **stat** structure referenced by *buf*. Otherwise, the macro shall evaluate to zero.

S_TYPEISTMO(*buf*)
Test macro for a typed memory object.

⌫

The *<sys/stat.h>* header shall define the following symbolic constants as distinct integer values outside of the range [0,999999999], for use with the *futimens()* and *utimensat()* functions: UTIME_NOW UTIME_OMIT

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
int    chmod(const char *, mode_t);
int    fchmod(int, mode_t);
int    fchmodat(int, const char *, mode_t, int);
int    fstat(int, struct stat *);
int    fstatat(int, const char *restrict, struct stat *restrict, int);
int    futimens(int, const struct timespec [2]);
int    lstat(const char *restrict, struct stat *restrict);
int    mkdir(const char *, mode_t);
int    mkdirat(int, const char *, mode_t);
int    mkfifo(const char *, mode_t);
int    mkfifoat(int, const char *, mode_t);
```

[XSI] ⌦

```
int    mknod(const char *, mode_t, dev_t);
int    mknodat(int, const char *, mode_t, dev_t);
⌫
int    stat(const char *restrict, struct stat *restrict);
mode_t umask(mode_t);
int    utimensat(int, const char *, const struct timespec [2], int);
```

Inclusion of the *<sys/stat.h>* header may make visible all symbols from the *<time.h>* header.

The following sections are informative.

APPLICATION USAGE

Use of the macros is recommended for determining the type of a file.

RATIONALE

A conforming C-language application must include *<sys/stat.h>* for functions that have arguments or return values of type **mode_t**, so that symbolic values for that type can be used. An alternative would be to require that these constants are also defined by including *<sys/types.h>*.

The S_ISUID and S_ISGID bits may be cleared on any write, not just on *open()*, as some historical implementations do.

System calls that update the time entry fields in the **stat** structure must be documented by the implementors. POSIX-conforming systems should not update the time entry fields for functions listed in the System Interfaces volume of POSIX.1-2017 unless the standard requires that they do, except in the case of documented extensions to the standard.

Upon assignment, file timestamps are immediately converted to the resolution of the file system by truncation (i.e., the recorded time can be older than the actual time). For example, if the file system resolution is 1 microsecond, then a conforming *stat()* must always return an *st_mtim.tv_nsec* that is a multiple of 1000. Some older implementations returned higher-resolution timestamps while the *inode* information was cached, and then spontaneously truncated the *tv_nsec* fields when they were stored to and retrieved from disk, but this behavior does not conform.

Note that *st_dev* must be unique within a Local Area Network (LAN) in a ``system'' made up of multiple computers' file systems connected by a LAN.

Networked implementations of a POSIX-conforming system must guarantee that all files visible within the file tree (including parts of the tree that may be remotely mounted from other machines on the network) on each individual processor are uniquely identified by the combination of the *st_ino* and *st_dev* fields.

The unit for the *st_blocks* member of the **stat** structure is not defined within POSIX.1-2017. In some implementations it is 512 bytes. It may differ on a file system basis. There is no correlation between values of the *st_blocks* and *st_blksize*, and the *f_bsize* (from *<sys/statvfs.h>*) structure members.

Traditionally, some implementations defined the multiplier for *st_blocks* in *<sys/param.h>* as the symbol DEV_BSIZE.

Some earlier versions of this standard did not specify values for the file mode bit macros. The expectation was that some implementors might choose to use a different encoding for these bits than the traditional one, and that new applications would use symbolic file modes instead of numeric. This version of the standard specifies the traditional encoding, in recognition that nearly 20 years after the first publication of this standard numeric file modes are still in widespread use by application developers, and that all conforming implementations still use the traditional encoding.

FUTURE DIRECTIONS

No new S_IFMT symbolic names for the file type values of **mode_t** will be defined by POSIX.1-2017; if new file types are required, they will only be testable through *S_ISxx*() or *S_TYPEISxxx*() macros instead.

SEE ALSO

<sys/statvfs.h>, *<sys/types.h>*, *<time.h>*

XSH *chmod*, *fchmod*, *fstat*, *fstatat*, *futimens*, *mkdir*, *mkfifo*, *mknod*, *umask*

CHANGE HISTORY

First released in Issue 1. Derived from Issue 1 of the SVID.

Issue 5

The DESCRIPTION is updated for alignment with the POSIX Realtime Extension.

The type of *st_blksize* is changed from **long** to **blksize_t**; the type of *st_blocks* is changed from **long** to **blkcnt_t**.

Issue 6

The S_TYPEISMQ(), S_TYPEISSEM(), and S_TYPEISSHM() macros are unconditionally mandated.

The Open Group Corrigendum U035/4 is applied. In the DESCRIPTION, the types **blksize_t** and **blkcnt_t** have been described.

The following new requirements on POSIX implementations derive from alignment with the Single UNIX Specification:

- The **dev_t**, **ino_t**, **mode_t**, **nlink_t**, **uid_t**, **gid_t**, **off_t**, and **time_t** types are mandated.

S_IFSOCK and S_ISSOCK are added for sockets.

The description of **stat** structure members is changed to reflect contents when file type is a symbolic link.

The test macro S_TYPEISTMO is added for alignment with IEEE Std 1003.1j-2000.

The **restrict** keyword is added to the prototypes for *lstat()* and *stat()*.

The *lstat()* function is made mandatory.

IEEE Std 1003.1-2001/Cor 1-2002, item XBD/TC1/D6/17 is applied, adding text regarding the *st_blocks* member of the **stat** structure to the RATIONALE.

IEEE Std 1003.1-2001/Cor 2-2004, item XBD/TC2/D6/25 is applied, adding to the DESCRIPTION that the **timespec** structure may be defined as described in the *<time.h>* header.

Issue 7

SD5-XSH-ERN-161 is applied, updating the DESCRIPTION to clarify that the descriptions of the interfaces should be consulted in order to determine which structure members have meaningful values.

The *fchmodat()*, *fstatat()*, *mkdirat()*, *mkfifoat()*, *mknodat()*, and *utimensat()* functions are added from The Open Group Technical Standard, 2006, Extended API Set Part 2.

The *futimens()* function is added.

This reference page is clarified with respect to macros and symbolic constants.

Changes are made related to support for finegrained timestamps and the UTIME_NOW and UTIME_OMIT symbolic constants are added.

POSIX.1-2008, Technical Corrigendum 1, XBD/TC1-2008/0068 [207] is applied.

POSIX.1-2008, Technical Corrigendum 2, XBD/TC2-2008/0078 [531] is applied.

End of informative text.

return to top of page

[Main Index | XBD | XSH | XCU | XRAT]

<<< Previous Home Next >>>