NAME

statvfs, fstatvfs - get filesystem statistics

SYNOPSIS

#include <sys/statvfs.h>

int statvfs(const char ****path*, struct statvfs ****buf***);**
int fstatvfs(int *fd***, struct statvfs ****buf***);**

DESCRIPTION

The function **statvfs**() returns information about a mounted filesystem. *p ath* is the pathname of any file within the mounted filesystem. *buf* is a poin ter to a *statvfs* structure defined approximately as follows:

```
struct statvfs {
unsigned long f_bsize; /* filesystem block size */
unsigned long f_frsize; /* fragment size */
fsblkcnt_t f_blocks; /* size of fs in f_frsize units */
fsblkcnt_t f_bfree; /* # free blocks */
fsblkcnt_t f_bavail; /* # free blocks for unprivileged users */
fsfilcnt_t f_files; /* # inodes */
fsfilcnt_t f_ffree; /* # free inodes */
fsfilcnt_t f_favail; /* # free inodes for unprivileged users */
unsigned long f_fsid; /* filesystem ID */
unsigned long f_flag; /* mount flags */
unsigned long f_namemax; /* maximum filename length */
};
```

Here the types *fsblkcnt_t* and *fsfilcnt_t* are defined in <*sys/types.h*>. Both used to be*unsigne d long*.

The field *f_flag* is a bit mask (of mount flags, see mount(8)). Bits defined by POSIX are

ST_RDONLY
Read-only filesystem.

ST_NOSUID
Set-user-ID/set-group-ID bits are ignored by exec(3).

It is unspecified whether all members of the returned struct have meaningful values on all filesystems.

fstatvfs() returns the same information about an open file referenced by descriptor *fd*.

RETURN VALUE

On success, zero is returned. On error, -1 is returned, and *errno* is set appropriately.

ERRORS

EACCES
(**statvfs**()) Search permission is denied for a component of the path prefix of *path*. (See also path_resolution(7).)

EBADF
(**fstatvfs**()) *fd* is not a valid open file descriptor.

EFAULT
Buf or *path* points to an invalid address.

EINTR
This call was interrupted by a signal.

EIO An I/O error occurred while reading from the filesystem.

ELOOP
(**statvfs**()) Too many symbolic links were encountered in translating *path*.

ENAMETOOLONG
(**statvfs**()) *path* is too long.

ENOENT
(**statvfs**()) The file referred to by *path* does not exist.

ENOMEM
Insufficient kernel memory was available.

ENOSYS
The filesystem does not support this call.

ENOTDIR
(**statvfs**()) A component of the path prefix of *path* is not a directory.

EOVERFLOW
Some values were too large to be represented in the returned struct.

ATTRIBUTES

Multithreading (see pthreads(7))

The **statvfs**() and **fstatvfs**() functions are thread-safe.

CONFORMING TO

POSIX.1-2001.

NOTES

The Linux kernel has system calls statfs(2) and fstatfs(2) to support this library call.

The current glibc implementations of

```
pathconf(path, _PC_REC_XFER_ALIGN);
pathconf(path, _PC_ALLOC_SIZE_MIN);
pathconf(path, _PC_REC_MIN_XFER_SIZE);
```

respectively use the *f_frsize*, *f_frsize*, and *f_bsize* fields of the return value of *statvfs(path,buf)*.

SEE ALSO

statfs(2)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at http://www.kernel.org/doc/man-pages/.