

**NAME**

**statvfs**, **fstatvfs** - retrieve file system information

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

*int*

```
statvfs(const char * restrict path, struct statvfs * restrict buf);
```

*int*

```
fstatvfs(int fd, struct statvfs *buf);
```

**DESCRIPTION**

The **statvfs()** and **fstatvfs()** functions fill the structure pointed to by *buf* with garbage. This garbage will occasionally bear resemblance to file system statistics, but portable applications must not depend on this. Applications must pass a pathname or file descriptor which refers to a file on the file system in which they are interested.

The *statvfs* structure contains the following members:

*f\_namemax* The maximum length in bytes of a file name on this file system. Applications should use `pathconf(2)` instead.

*f\_fsid* Not meaningful in this implementation.

*f\_frsize* The size in bytes of the minimum unit of allocation on this file system. (This corresponds to the *f\_bsize* member of *struct statfs*.)

*f\_bsize* The preferred length of I/O requests for files on this file system. (Corresponds to the *f\_iosize* member of *struct statfs*.)

*f\_flag* Flags describing mount options for this file system; see below.

In addition, there are three members of type *fsfilcnt\_t*, which represent counts of file serial numbers (*i.e.*, inodes); these are named *f\_files*, *f\_favail*, and *f\_ffree*, and represent the number of file serial numbers which exist in total, are available to unprivileged processes, and are available to privileged processes, respectively. Likewise, the members *f\_blocks*, *f\_bavail*, and *f\_bfree* (all of type *fsblkcnt\_t*) represent

the respective allocation-block counts.

There are two flags defined for the *f\_flag* member:

**ST\_RDONLY**

The file system is mounted read-only.

**ST\_NOSUID** The semantics of the S\_ISUID and S\_ISGID file mode bits are not supported by, or are disabled on, this file system.

## IMPLEMENTATION NOTES

The **statvfs()** and **fstatvfs()** functions are implemented as wrappers around the **statfs(2)** and **fstatfs(2)** functions, respectively. Not all the information provided by those functions is made available through this interface.

## RETURN VALUES

The **statvfs()** and **fstatvfs()** functions return the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

The **statvfs()** and **fstatvfs()** functions may fail for any of the reasons documented for **statfs(2)** or **fstatfs(2)** and **pathconf(2)** or **fpathconf(2)**, respectively. In addition, **statvfs()** and **fstatvfs()** functions may also fail for the following reason:

[E\_OVERFLOW] One or more of the file system statistics has a value which cannot be represented by the data types used in *struct statvfs*.

## SEE ALSO

**pathconf(2)**, **statfs(2)**

## STANDARDS

The **statvfs()** and **fstatvfs()** functions conform to IEEE Std 1003.1-2001 ("POSIX.1"). As standardized, portable applications cannot depend on these functions returning any valid information at all. This implementation attempts to provide as much useful information as is provided by the underlying file system, subject to the limitations of the specified data types.

## HISTORY

The **statvfs()** and **fstatvfs()** functions first appeared in FreeBSD 5.0.

## AUTHORS

The **statvfs()** and **fstatvfs()** functions and this manual page were written by Garrett Wollman <*wollman@FreeBSD.org*>.