

```

1: #ifndef _FS_STATVFS_H /* wrapper symbol for kernel use */
2: #define _FS_STATVFS_H /* subject to change without notice */
3:
4: #ident "@(#)kern:fs/statvfs.h 1.5"
5: #ident "$Header: $"
6:
7: #if defined(__cplusplus)
8: extern "C" {
9: #endif
10:
11: /*
12:  * Structure returned by statvfs(2).
13:  */
14:
15: #ifdef _KERNEL_HEADERS
16:
17: #include <util/types.h> /* REQUIRED */
18:
19: #elif defined(_KERNEL) || defined(_KMEMUSER)
20:
21: #include <sys/types.h> /* REQUIRED */
22:
23: #endif /* _KERNEL_HEADERS */
24:
25: #define FSTYPESZ 16
26:
27: typedef struct statvfs {
28:     u_long f_bsize; /* fundamental file system block size */
29:     u_long f_frsize; /* fragment size */
30:     u_long f_blocks; /* total # of blocks of f_frsize on fs */
31:     u_long f_bfree; /* total # of free blocks of f_frsize */
32:     u_long f_bavail; /* # of free blocks avail to non-superuser */
33:     u_long f_files; /* total # of file nodes (inodes) */
34:     u_long f_ffree; /* total # of free file nodes */
35:     u_long f_favail; /* # of free nodes avail to non-superuser */
36:     u_long f_fsid; /* file system id (dev for now) */
37:     char f_basetype[FSTYPESZ]; /* target fs type name, null-terminated */
38:     u_long f_flag; /* bit-mask of flags */
39:     u_long f_namemax; /* maximum file name length */
40:     char f_fstr[32]; /* filesystem-specific string */
41:     u_long f_filler[16]; /* reserved for future expansion */
42: } statvfs_t;
43:
44: /*
45:  * Flag definitions.
46:  */
47:
48: #define ST_RDONLY 0x01 /* read-only file system */
49: #define ST_NOSUID 0x02 /* does not support setuid/setgid semantics */
50: #define ST_NOTRUNC 0x04 /* does not truncate long file names */
51:
52: #if defined(__STDC__) && !defined(_KERNEL)
53: int statvfs(const char *, struct statvfs *);
54: int fstatvfs(int, struct statvfs *);
55: #endif
56:
57: #if defined(__cplusplus)
58: }
59: #endif
60:
61: #endif /* _FS_STATVFS_H */

```

```

1: /* Definition of `struct statvfs', information about a filesystem.
2: Copyright (C) 1998 Free Software Foundation, Inc.
3: This file is part of the GNU C Library.
4:
...
33:
34: struct statvfs
35: {
36:     unsigned long int f_bsize;
37:     unsigned long int f_frsize;
38: #ifndef __USE_FILE_OFFSET64
39:     __fsblkcnt_t f_blocks;
40:     __fsblkcnt_t f_bfree;
41:     __fsblkcnt_t f_bavail;
42:     __fsfilcnt_t f_files;
43:     __fsfilcnt_t f_ffree;
44:     __fsfilcnt_t f_favail;
45: #else
46:     __fsblkcnt64_t f_blocks;
47:     __fsblkcnt64_t f_bfree;
48:     __fsblkcnt64_t f_bavail;
49:     __fsfilcnt64_t f_files;
50:     __fsfilcnt64_t f_ffree;
51:     __fsfilcnt64_t f_favail;
52: #endif
53:     __fsid_t f_fsid;
54:     unsigned long int f_flag;
55:     unsigned long int f_namemax;
56:     unsigned int f_spare[6];
57: };
58:
59: #ifdef __USE_LARGEFILE64
60: struct statvfs64
61: {
62:     unsigned long int f_bsize;
63:     unsigned long int f_frsize;
64:     __fsblkcnt64_t f_blocks;
65:     __fsblkcnt64_t f_bfree;
66:     __fsblkcnt64_t f_bavail;
67:     __fsfilcnt64_t f_files;
68:     __fsfilcnt64_t f_ffree;
69:     __fsfilcnt64_t f_favail;
70:     __fsid_t f_fsid;
71:     unsigned long int f_flag;
72:     unsigned long int f_namemax;
73:     unsigned int f_spare[6];
74: };
75: #endif
76:
77: /* Definitions for the flag in `f_flag'. */
78: enum
79: {
80:     ST_RDONLY = 1,
81: #define ST_RDONLY ST_RDONLY
82:     ST_NOSUID = 2
83: #define ST_NOSUID ST_NOSUID
84: };

```