

>0x7ffffff blocksize filesystem reporting

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2003-11/0014.html>

From: Kirk McKusick (mckusick_at_beastie.mckusick.com)

Date: 11/06/03

To: Peter Wemm <peter@FreeBSD.org>
Date: Wed, 05 Nov 2003 21:04:41 -0800

I have gone back and resurrected the changes for the updated statfs structure that were discussed on arch some months ago. Because they introduce replacement system calls for the statfs family of system calls, it is necessary to build and install a new kernel before doing a `make world'. Failure to do so will result in programs that use statfs (like df) failing with `bad system call' until a new kernel is booted. If the new kernel is booted first, it will handle both the new and old forms of the system call. If I have not received any objections to this change, I will check it into -current on the evening of Monday November 10th.

Kirk McKusick

=====
=====

Index: sys/sys/mount.h

RCS file: /usr/ncvs/src/sys/sys/mount.h,v

retrieving revision 1.148

diff -c -r1.148 mount.h

*** sys/sys/mount.h 1 Jul 2003 17:40:23 -0000 1.148

--- sys/sys/mount.h 5 Nov 2003 05:10:54 -0000

*** 63,75 ****

/*

* filesystem statistics

*/

! #define MFSNAMELEN 16 /* length of fs type name, including null */

! #define MNAMELEN (88 - 2 * sizeof(long)) /* size of on/from name bufs */

/* XXX getfsstat.2 is out of date with write and read counter changes here. */

/* XXX statfs.2 is out of date with read counter changes here. */

! struct statfs {

long f_spare2; /* placeholder */

long f_bsize; /* fundamental filesystem block size */

long f_iosize; /* optimal transfer block size */

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
--- 63,103 ----
/*
 * filesystem statistics
 */
+ #define MFSNAMELEN 16 /* length of type name including null */
+ #define MNAMELEN 80 /* size of on/from name bufs */
+ #define STATFS_VERSION 0x20030518 /* current version number */
+ struct statfs {
+ u_int32_t f_version; /* structure version number */
+ u_int32_t f_type; /* type of filesystem */
+ u_int64_t f_flags; /* copy of mount exported flags */
+ u_int64_t f_bsize; /* filesystem fragment size */
+ u_int64_t f_iosize; /* optimal transfer block size */
+ u_int64_t f_blocks; /* total data blocks in filesystem */
+ u_int64_t f_bfree; /* free blocks in filesystem */
+ int64_t f_bavail; /* free blocks avail to non-superuser */
+ u_int64_t f_files; /* total file nodes in filesystem */
+ int64_t f_ffree; /* free nodes avail to non-superuser */
+ u_int64_t f_syncwrites; /* count of sync writes since mount */
+ u_int64_t f_asyncwrites; /* count of async writes since mount */
+ u_int64_t f_syncreads; /* count of sync reads since mount */
+ u_int64_t f_asyncreads; /* count of async reads since mount */
+ u_int64_t f_spare[10]; /* unused spare */
+ u_int32_t f_namemax; /* maximum filename length */
+ uid_t f_owner; /* user that mounted the filesystem */
+ fsid_t f_fsid; /* filesystem id */
+ char f_charspare[76]; /* spare string space */
+ char f_fstypename[MFSNAMELEN]; /* filesystem type name */
+ char f_mntfromname[MNAMELEN]; /* mounted filesystem */
+ char f_mntonname[MNAMELEN]; /* directory on which mounted */
+ };

! #ifdef _KERNEL
! #define OMFSNAMELEN 16 /* length of fs type name, including null */
! #define OMNAMELEN (88 - 2 * sizeof(long)) /* size of on/from name bufs */

/* XXX getfsstat.2 is out of date with write and read counter changes here. */
/* XXX statfs.2 is out of date with read counter changes here. */
! struct ostatfs {
    long f_spare2; /* placeholder */
    long f_bsize; /* fundamental filesystem block size */
    long f_iosize; /* optimal transfer block size */
    *****
    *** 84,95 ***
    int f_flags; /* copy of mount exported flags */
    long f_syncwrites; /* count of sync writes since mount */
    long f_asyncwrites; /* count of async writes since mount */
! char f_fstypename[MFSNAMELEN]; /* fs type name */
! char f_mntonname[MNAMELEN]; /* directory on which mounted */
    long f_syncreads; /* count of sync reads since mount */
    long f_asyncreads; /* count of async reads since mount */

```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
short f_spare1; /* unused spare */
! char f_mntfromname[MNAMELEN]; /* mounted filesystem */
short f_spare2; /* unused spare */
/*
 * XXX on machines where longs are aligned to 8-byte boundaries, there
--- 112,123 ----
int f_flags; /* copy of mount exported flags */
long f_syncwrites; /* count of sync writes since mount */
long f_asyncwrites; /* count of async writes since mount */
! char ffstypename[OMFSNAMELEN]; /* fs type name */
! char f_mntonname[OMNAMELEN]; /* directory on which mounted */
long f_syncreads; /* count of sync reads since mount */
long f_asyncreads; /* count of async reads since mount */
short f_spare1; /* unused spare */
! char f_mntfromname[OMNAMELEN]; /* mounted filesystem */
short f_spare2; /* unused spare */
/*
 * XXX on machines where longs are aligned to 8-byte boundaries, there
*****
*** 99,105 ****
long f_spare[2]; /* unused spare */
};
```

```
- #ifdef _KERNEL
#define MMAXOPTIONLEN 65536 /* maximum length of a mount option */
```

```
TAILQ_HEAD(vnodelst, vnode);
--- 127,132 ----
```

Index: sys/kern/syscalls.master

=====
RCS file: /usr/ncvs/src/sys/kern/syscalls.master,v
retrieving revision 1.155
diff -c -r1.155 syscalls.master

```
*** sys/kern/syscalls.master 21 Oct 2003 07:03:27 -0000 1.155
--- sys/kern/syscalls.master 5 Nov 2003 05:10:54 -0000
```

```
*** 68,74 ****
15 STD POSIX { int chmod(char *path, int mode); }
16 STD POSIX { int chown(char *path, int uid, int gid); }
17 MSTD BSD { int obreak(char *nsize); } break obreak_args int
! 18 STD BSD { int getfsstat(struct statfs *buf, long bufsize, \
int flags); }
19 COMPAT POSIX { long lseek(int fd, long offset, int whence); }
20 MSTD POSIX { pid_t getpid(void); }
--- 68,74 ----
15 STD POSIX { int chmod(char *path, int mode); }
16 STD POSIX { int chown(char *path, int uid, int gid); }
17 MSTD BSD { int obreak(char *nsize); } break obreak_args int
! 18 COMPAT4 BSD { int getfsstat(struct ostatfs *buf, long bufsize, \
int flags); }
19 COMPAT POSIX { long lseek(int fd, long offset, int whence); }
```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
20 MSTD POSIX { pid_t getpid(void); }
*****
*** 252,259 ***
155 MNOIMPL BSD { int nfssvc(int flag, caddr_t argp); }
156 COMPAT BSD { int getdirentries(int fd, char *buf, u_int count, \
    long *basep); }
! 157 STD BSD { int statfs(char *path, struct statfs *buf); }
! 158 STD BSD { int fstatfs(int fd, struct statfs *buf); }
159 UNIMPL NOHIDE nosys
160 UNIMPL NOHIDE nosys
161 STD BSD { int getfh(char *fname, struct fhandle *fhp); }
---- 252,259 ----
155 MNOIMPL BSD { int nfssvc(int flag, caddr_t argp); }
156 COMPAT BSD { int getdirentries(int fd, char *buf, u_int count, \
    long *basep); }
! 157 COMPAT4 BSD { int statfs(char *path, struct ostatfs *buf); }
! 158 COMPAT4 BSD { int fstatfs(int fd, struct ostatfs *buf); }
159 UNIMPL NOHIDE nosys
160 UNIMPL NOHIDE nosys
161 STD BSD { int getfh(char *fname, struct fhandle *fhp); }
*****
*** 439,445 ***
295 UNIMPL NOHIDE nosys
296 UNIMPL NOHIDE nosys
; XXX 297 is 300 in NetBSD
! 297 STD BSD { int fhstatfs(const struct fhandle *u_fhp, struct statfs *buf); }
298 STD BSD { int fhopen(const struct fhandle *u_fhp, int flags); }
299 STD BSD { int fhstat(const struct fhandle *u_fhp, struct stat *sb); }
; syscall numbers for FreeBSD
---- 439,445 ----
295 UNIMPL NOHIDE nosys
296 UNIMPL NOHIDE nosys
; XXX 297 is 300 in NetBSD
! 297 COMPAT4 BSD { int fhstatfs(const struct fhandle *u_fhp, struct ostatfs *buf); }
298 STD BSD { int fhopen(const struct fhandle *u_fhp, int flags); }
299 STD BSD { int fhstat(const struct fhandle *u_fhp, struct stat *sb); }
; syscall numbers for FreeBSD
*****
*** 574,583 ***
    struct sf_hdr *hdr, off_t *sbytes, int flags); }
394 MSTD BSD { int mac_syscall(const char *policy, int call, \
    void *arg); }
! 395 UNIMPL NOHIDE nosys
! 396 UNIMPL NOHIDE nosys
! 397 UNIMPL NOHIDE nosys
! 398 UNIMPL NOHIDE nosys
399 UNIMPL NOHIDE nosys
400 MNOSTD BSD { int ksem_close(semid_t id); }
401 MNOSTD BSD { int ksem_post(semid_t id); }
---- 574,585 ----
    struct sf_hdr *hdr, off_t *sbytes, int flags); }
```

>0x7ffffff blocksize filesystem reporting

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
394 MSTD BSD { int mac_syscall(const char *policy, int call, \
        void *arg); }
! 395 STD BSD { int getfsstat(struct statfs *buf, long bufsize, \
! int flags); }
! 396 STD BSD { int statfs(char *path, struct statfs *buf); }
! 397 STD BSD { int fstatfs(int fd, struct statfs *buf); }
! 398 STD BSD { int fhstatfs(const struct fhandle *u_fhp, \
! struct statfs *buf); }
399 UNIMPL NOHIDE nosys
400 MNOSTD BSD { int ksem_close(semid_t id); }
401 MNOSTD BSD { int ksem_post(semid_t id); }
Index: sys/kern/vfs_syscalls.c
```

=====
RCS file: /usr/ncvs/src/sys/kern/vfs_syscalls.c,v

retrieving revision 1.332

diff -c -r1.332 vfs_syscalls.c

*** sys/kern/vfs_syscalls.c 19 Oct 2003 20:41:07 -0000 1.332

--- sys/kern/vfs_syscalls.c 5 Nov 2003 05:10:54 -0000

*** 226,236 ****

```
        struct statfs *buf;
    } /* *uap;
```

```
{
! register struct mount *mp;
! register struct statfs *sp;
    int error;
    struct nameidata nd;
- struct statfs sb;
```

```
    NDINIT(&nd, LOOKUP, FOLLOW, UIO_USERSPACE, uap->path, td);
    if ((error = namei(&nd)) != 0)
```

--- 226,235 ----

```
        struct statfs *buf;
    } /* *uap;
```

```
{
! struct mount *mp;
! struct statfs *sp, sb;
    int error;
    struct nameidata nd;
```

```
    NDINIT(&nd, LOOKUP, FOLLOW, UIO_USERSPACE, uap->path, td);
    if ((error = namei(&nd)) != 0)
```

*** 244,253 ****

```
        if (error)
            return (error);
#endif
        error = VFS_STATFS(mp, sp, td);
        if (error)
            return (error);
```

```
- sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
```

>0x7ffffff blocksize filesystem reporting

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```

    if (suser(td)) {
        bcopy(sp, &sb, sizeof(sb));
        sb.f_fsid.val[0] = sb.f_fsid.val[1] = 0;
---- 243,257 ----
    if (error)
        return (error);
#endif
+ /*
+ * Set these in case the underlying filesystem fails to do so.
+ */
+ sp->f_version = STATFS_VERSION;
+ sp->f_namemax = NAME_MAX;
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
    error = VFS_STATFS(mp, sp, td);
    if (error)
        return (error);
    if (suser(td)) {
        bcopy(sp, &sb, sizeof(sb));
        sb.f_fsid.val[0] = sb.f_fsid.val[1] = 0;
*****
*** 276,284 ***
    {
        struct file *fp;
        struct mount *mp;
! register struct statfs *sp;
        int error;
- struct statfs sb;

        if ((error = getvnode(td->td_proc->p_fd, uap->fd, &fp)) != 0)
            return (error);
---- 280,287 ----
    {
        struct file *fp;
        struct mount *mp;
! struct statfs *sp, sb;
        int error;

        if ((error = getvnode(td->td_proc->p_fd, uap->fd, &fp)) != 0)
            return (error);
*****
*** 292,301 ***
        return (error);
#endif
    sp = &mp->mnt_stat;
    error = VFS_STATFS(mp, sp, td);
    if (error)
        return (error);
- sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
    if (suser(td)) {
        bcopy(sp, &sb, sizeof(sb));
        sb.f_fsid.val[0] = sb.f_fsid.val[1] = 0;

```

>0x7ffffff blocksize filesystem reporting

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```

---- 295,309 ----
        return (error);
    #endif
    sp = &mp->mnt_stat;
+ /*
+ * Set these in case the underlying filesystem fails to do so.
+ */
+ sp->f_version = STATFS_VERSION;
+ sp->f_namemax = NAME_MAX;
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
    error = VFS_STATFS(mp, sp, td);
    if (error)
        return (error);
    if (suser(td)) {
        bcopy(sp, &sb, sizeof(sb));
        sb.f_fsid.val[0] = sb.f_fsid.val[1] = 0;
*****
*** 323,330 ***
        int flags;
    } /* *uap;
    {
! register struct mount *mp, *nmp;
! register struct statfs *sp;
    caddr_t sfsp;
    long count, maxcount, error;

---- 331,338 ----
        int flags;
    } /* *uap;
    {
! struct mount *mp, *nmp;
! struct statfs *sp, sb;
    caddr_t sfsp;
    long count, maxcount, error;

*****
*** 346,351 ***
---- 354,366 ----
        if (sfsp && count < maxcount) {
            sp = &mp->mnt_stat;
        /*
+ * Set these in case the underlying filesystem
+ * fails to do so.
+ */
+ sp->f_version = STATFS_VERSION;
+ sp->f_namemax = NAME_MAX;
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
+ /*
            * If MNT_NOWAIT or MNT_LAZY is specified, do not
            * refresh the fsstat cache. MNT_NOWAIT or MNT_LAZY
            * overrides MNT_WAIT.

```

freebsd-arch: >0x7ffffff blocksizes filesystem reporting

```
*****
*** 358,364 ***
        vfs_unbusy(mp, td);
        continue;
    }
! sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
    error = copyout(sp, sfsp, sizeof(*sp));
    if (error) {
        vfs_unbusy(mp, td);
---- 373,383 ----
        vfs_unbusy(mp, td);
        continue;
    }
! if (suser(td)) {
! bcopy(sp, &sb, sizeof(sb));
! sb.f_fsid.val[0] = sb.f_fsid.val[1] = 0;
! sp = &sb;
! }
        error = copyout(sp, sfsp, sizeof(*sp));
        if (error) {
            vfs_unbusy(mp, td);
*****
*** 379,384 ***
---- 398,660 ----
        return (0);
    }

+ #ifdef COMPAT_FREEBSD4
+ /*
+ * Get old format filesystem statistics.
+ */
+ static void cvtstatfs(struct thread *, struct statfs *, struct ostatfs *);
+
+ #ifndef _SYS_SYSPROTO_H_
+ struct freebsd4_statfs_args {
+ char *path;
+ struct ostatfs *buf;
+ };
+ #endif
+ /* ARGSUSED */
+ int
+ freebsd4_statfs(td, uap)
+ struct thread *td;
+ struct freebsd4_statfs_args /* {
+ char *path;
+ struct ostatfs *buf;
+ } */ *uap;
+ {
+ struct mount *mp;
+ struct statfs *sp;
+ struct ostatfs osb;
```

>0x7ffffff blocksizes filesystem reporting

```

+ int error;
+ struct nameidata nd;
+
+ NDINIT(&nd, LOOKUP, FOLLOW, UIO_USERSPACE, uap->path, td);
+ if ((error = namei(&nd)) != 0)
+ return (error);
+ mp = nd.ni_vp->v_mount;
+ sp = &mp->mnt_stat;
+ NDFREE(&nd, NDF_ONLY_PNBUF);
+ vrele(nd.ni_vp);
+ #ifdef MAC
+ error = mac_check_mount_stat(td->td_ucred, mp);
+ if (error)
+ return (error);
+ #endif
+ error = VFS_STATFS(mp, sp, td);
+ if (error)
+ return (error);
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
+ cvtstatfs(td, sp, &osb);
+ return (copyout(&osb, uap->buf, sizeof(osb)));
+ }
+
+ /*
+  * Get filesystem statistics.
+  */
+ #ifdef _SYS_SYSPROTO_H_
+ struct freebsd4_fstatfs_args {
+ int fd;
+ struct ostatfs *buf;
+ };
+ #endif
+ /* ARGSUSED */
+ int
+ freebsd4_fstatfs(td, uap)
+ struct thread *td;
+ struct freebsd4_fstatfs_args /* {
+ int fd;
+ struct ostatfs *buf;
+ } */ *uap;
+ {
+ struct file *fp;
+ struct mount *mp;
+ struct statfs *sp;
+ struct ostatfs osb;
+ int error;
+
+ if ((error = getvnode(td->td_proc->p_fd, uap->fd, &fp)) != 0)
+ return (error);
+ mp = fp->f_vnode->v_mount;
+ fdrop(fp, td);

```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
+ if (mp == NULL)
+ return (EBADF);
+ #ifdef MAC
+ error = mac_check_mount_stat(td->td_ucred, mp);
+ if (error)
+ return (error);
+ #endif
+ sp = &mp->mnt_stat;
+ error = VFS_STATFS(mp, sp, td);
+ if (error)
+ return (error);
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
+ cvtstatfs(td, sp, &osb);
+ return (copyout(&osb, uap->buf, sizeof(osb)));
+ }
+
+ /*
+ * Get statistics on all filesystems.
+ */
+ #ifndef _SYS_SYSPROTO_H_
+ struct freebsd4_getfsstat_args {
+ struct ostatfs *buf;
+ long bufsize;
+ int flags;
+ };
+ #endif
+ int
+ freebsd4_getfsstat(td, uap)
+ struct thread *td;
+ register struct freebsd4_getfsstat_args /* {
+ struct ostatfs *buf;
+ long bufsize;
+ int flags;
+ } */ *uap;
+ {
+ struct mount *mp, *nmp;
+ struct statfs *sp;
+ struct ostatfs osb;
+ caddr_t sfsp;
+ long count, maxcount, error;
+
+ maxcount = uap->bufsize / sizeof(struct ostatfs);
+ sfsp = (caddr_t)uap->buf;
+ count = 0;
+ mtx_lock(&mountlist_mtx);
+ for (mp = TAILQ_FIRST(&mountlist); mp != NULL; mp = nmp) {
+ #ifdef MAC
+ if (mac_check_mount_stat(td->td_ucred, mp) != 0) {
+ nmp = TAILQ_NEXT(mp, mnt_list);
+ continue;
+ }

```

>0x7ffffff blocksize filesystem reporting

```

+ #endif
+ if (vfs_busy(mp, LK_NOWAIT, &mountlist_mtx, td)) {
+ nmp = TAILQ_NEXT(mp, mnt_list);
+ continue;
+ }
+ if (sfsp && count < maxcount) {
+ sp = &mp->mnt_stat;
+ /*
+ * If MNT_NOWAIT or MNT_LAZY is specified, do not
+ * refresh the fsstat cache. MNT_NOWAIT or MNT_LAZY
+ * overrides MNT_WAIT.
+ */
+ if (((uap->flags & (MNT_LAZY|MNT_NOWAIT)) == 0 ||
+ (uap->flags & MNT_WAIT)) &&
+ (error = VFS_STATFS(mp, sp, td))) {
+ mtx_lock(&mountlist_mtx);
+ nmp = TAILQ_NEXT(mp, mnt_list);
+ vfs_unbusy(mp, td);
+ continue;
+ }
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
+ cvtstatfs(td, sp, &osb);
+ error = copyout(&osb, sfsp, sizeof(osb));
+ if (error) {
+ vfs_unbusy(mp, td);
+ return (error);
+ }
+ sfsp += sizeof(osb);
+ }
+ count++;
+ mtx_lock(&mountlist_mtx);
+ nmp = TAILQ_NEXT(mp, mnt_list);
+ vfs_unbusy(mp, td);
+ }
+ mtx_unlock(&mountlist_mtx);
+ if (sfsp && count > maxcount)
+ td->td_retval[0] = maxcount;
+ else
+ td->td_retval[0] = count;
+ return (0);
+ }
+
+ /*
+ * Implement fstatfs() for (NFS) file handles.
+ */
+ #ifndef _SYS_SYSPROTO_H_
+ struct freebsd4_fhstatfs_args {
+ struct fhandle *u_fhp;
+ struct ostatfs *buf;
+ };
+ #endif

```

```

+ int
+ freebsd4_fhstatfs(td, uap)
+ struct thread *td;
+ struct freebsd4_fhstatfs_args /* {
+ struct fhandle *u_fhp;
+ struct ostatfs *buf;
+ } */ *uap;
+ {
+ struct statfs *sp;
+ struct mount *mp;
+ struct vnode *vp;
+ struct ostatfs osb;
+ fhandle_t fh;
+ int error;
+
+ /*
+ * Must be super user
+ */
+ error = suser(td);
+ if (error)
+ return (error);
+
+ if ((error = copyin(uap->u_fhp, &fh, sizeof(fhhandle_t))) != 0)
+ return (error);
+
+ if ((mp = vfs_getvfs(&fh.fh_fsid)) == NULL)
+ return (ESTALE);
+ if ((error = VFS_FHTOVP(mp, &fh.fh_fid, &vp)))
+ return (error);
+ mp = vp->v_mount;
+ sp = &mp->mnt_stat;
+ vput(vp);
+ #ifdef MAC
+ error = mac_check_mount_stat(td->td_ucred, mp);
+ if (error)
+ return (error);
+ #endif
+ if ((error = VFS_STATFS(mp, sp, td)) != 0)
+ return (error);
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
+ cvtstatfs(td, sp, &osb);
+ return (copyout(&osb, uap->buf, sizeof(osb)));
+ }
+
+ /*
+ * Convert a new format statfs structure to an old format statfs structure.
+ */
+ static void
+ cvtstatfs(td, nsp, osp)
+ struct thread *td;
+ struct statfs *nsp;

```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```

+ struct ostats *osp;
+ {
+
+ bzero(osp, sizeof(*osp));
+ osp->f_bsize = nsp->f_bsize;
+ osp->f_iosize = nsp->f_iosize;
+ osp->f_blocks = nsp->f_blocks;
+ osp->f_bfree = nsp->f_bfree;
+ osp->f_bavail = nsp->f_bavail;
+ osp->f_files = nsp->f_files;
+ osp->f_ffree = nsp->f_ffree;
+ osp->f_owner = nsp->f_owner;
+ osp->f_type = nsp->f_type;
+ osp->f_flags = nsp->f_flags;
+ osp->f_syncwrites = nsp->f_syncwrites;
+ osp->f_asyncwrites = nsp->f_asyncwrites;
+ osp->f_syncreads = nsp->f_syncreads;
+ osp->f_asyncreads = nsp->f_asyncreads;
+ bcopy(nsp->f_fstypename, osp->f_fstypename, MFSNAMELEN);
+ bcopy(nsp->f_mntonname, osp->f_mntonname, MNAMELEN);
+ bcopy(nsp->f_mntfromname, osp->f_mntfromname, MNAMELEN);
+ if (suser(td)) {
+ osp->f_fsid.val[0] = osp->f_fsid.val[1] = 0;
+ } else {
+ osp->f_fsid = nsp->f_fsid;
+ }
+ }
+ #endif /* COMPAT_FREEBSD4 */
+
+ /*
+  * Change current working directory to a given file descriptor.
+  */
+
+ *****
+ *** 3788,3797 ***
+     struct statfs *buf;
+     } /* *uap;
+ {
+ ! struct statfs *sp;
+     struct mount *mp;
+     struct vnode *vp;
+ - struct statfs sb;
+     fhandle_t fh;
+     int error;
+
+ ---- 4064,4072 ----
+     struct statfs *buf;
+     } /* *uap;
+ {
+ ! struct statfs *sp, sb;
+     struct mount *mp;
+     struct vnode *vp;

```

```

fhandle_t fh;
int error;

*****
*** 3817,3825 ***
    if (error)
        return (error);
#endif
    if ((error = VFS_STATFS(mp, sp, td)) != 0)
        return (error);
- sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
    if (suser(td)) {
        bcopy(sp, &sb, sizeof(sb));
        sb.f_fsid.val[0] = sb.f_fsid.val[1] = 0;
---- 4092,4105 ----
    if (error)
        return (error);
#endif
+ /*
+ * Set these in case the underlying filesystem fails to do so.
+ */
+ sp->f_version = STATFS_VERSION;
+ sp->f_namemax = NAME_MAX;
+ sp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
    if ((error = VFS_STATFS(mp, sp, td)) != 0)
        return (error);
    if (suser(td)) {
        bcopy(sp, &sb, sizeof(sb));
        sb.f_fsid.val[0] = sb.f_fsid.val[1] = 0;
Index: sys/ufs/ffs/ffs_vfsops.c
=====
RCS file: /usr/src/sys/ufs/ffs/ffs_vfsops.c,v
retrieving revision 1.222
diff -c -r1.222 ffs_vfsops.c
*** sys/ufs/ffs/ffs_vfsops.c 2 Nov 2003 04:52:53 -0000 1.222
--- sys/ufs/ffs/ffs_vfsops.c 5 Nov 2003 05:10:54 -0000
*****
*** 1075,1080 ***
---- 1075,1081 ----
    fs = ump->um_fs;
    if (fs->fs_magic != FS_UFS1_MAGIC && fs->fs_magic != FS_UFS2_MAGIC)
        panic("ffs_statfs");
+ sbp->f_version = STATFS_VERSION;
    sbp->f_bsize = fs->fs_bsize;
    sbp->f_iosize = fs->fs_bsize;
    sbp->f_blocks = fs->fs_dsize;
*****
*** 1084,1091 ***
---- 1085,1102 ----
    dbtfsb(fs, fs->fs_pendingblocks);
    sbp->f_files = fs->fs_ncg * fs->fs_ipg - ROOTINO;

```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
sbp->f_ffree = fs->fs_cstotal.cs_nifree + fs->fs_pendinginodes;
+ sbp->f_namemax = NAME_MAX;
  if (sbp != &mp->mnt_stat) {
+ sbp->f_flags = mp->mnt_flag & MNT_VISFLAGMASK;
    sbp->f_type = mp->mnt_vfc->vfc_typenum;
+ sbp->f_syncwrites = mp->mnt_stat.f_syncwrites;
+ sbp->f_asyncwrites = mp->mnt_stat.f_asyncwrites;
+ sbp->f_syncreads = mp->mnt_stat.f_syncreads;
+ sbp->f_asyncreads = mp->mnt_stat.f_asyncreads;
+ sbp->f_owner = mp->mnt_stat.f_owner;
+ sbp->f_fsid = mp->mnt_stat.f_fsid;
+ bcopy((caddr_t)mp->mnt_stat.ffstypename,
+ (caddr_t)&sbp->f_fstypename[0], MFSNAMELEN);
    bcopy((caddr_t)mp->mnt_stat.f_mntonname,
          (caddr_t)&sbp->f_mntonname[0], MNAMELEN);
    bcopy((caddr_t)mp->mnt_stat.f_mntfromname,
```

Index: sys/kern/vfs_bio.c

=====
RCS file: /usr/ncvs/src/sys/kern/vfs_bio.c,v

retrieving revision 1.420

diff -c -r1.420 vfs_bio.c

*** sys/kern/vfs_bio.c 4 Nov 2003 06:30:00 -0000 1.420

--- sys/kern/vfs_bio.c 5 Nov 2003 05:10:54 -0000

*** 3239,3245 ****

```
                (int) m->pindex, (int)(foff >> 32),
                (int) foff & 0xffffffff, resid, i);
        if (!vn_isdisk(vp, NULL))
! printf(" iosize: %ld, lblkno: %jd, flags: 0x%x, npages: %d\n",
        bp->b_vp->v_mount->mnt_stat.f_iosize,
        (intmax_t) bp->b_lblkno,
        bp->b_flags, bp->b_npages);
```

--- 3239,3245 ----

```
                (int) m->pindex, (int)(foff >> 32),
                (int) foff & 0xffffffff, resid, i);
        if (!vn_isdisk(vp, NULL))
! printf(" iosize: %jd, lblkno: %jd, flags: 0x%x, npages: %d\n",
        bp->b_vp->v_mount->mnt_stat.f_iosize,
        (intmax_t) bp->b_lblkno,
        bp->b_flags, bp->b_npages);
```

Index: sys/kern/vfs_cluster.c

=====
RCS file: /usr/ncvs/src/sys/kern/vfs_cluster.c,v

retrieving revision 1.147

diff -c -r1.147 vfs_cluster.c

*** sys/kern/vfs_cluster.c 20 Oct 2003 18:24:38 -0000 1.147

--- sys/kern/vfs_cluster.c 5 Nov 2003 05:10:54 -0000

*** 327,333 ****

```
    GIANT_REQUIRED;
```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
KASSERT(size == vp->v_mount->mnt_stat.f_iosize,  
! ("cluster_rbuild: size %ld != filesize %ld\n",  
    size, vp->v_mount->mnt_stat.f_iosize));
```

```
/*  
--- 327,333 ----  
    GIANT_REQUIRED;
```

```
KASSERT(size == vp->v_mount->mnt_stat.f_iosize,  
! ("cluster_rbuild: size %ld != filesize %jd\n",  
    size, vp->v_mount->mnt_stat.f_iosize));
```

```
/*  
Index: sys/sys/syscall.h
```

```
=====  
RCS file: /usr/ncvs/src/sys/sys/syscall.h,v
```

```
retrieving revision 1.142
```

```
diff -c -r1.142 syscall.h
```

```
*** sys/sys/syscall.h 21 Oct 2003 07:03:27 -0000 1.142
```

```
--- sys/sys/syscall.h 5 Nov 2003 05:13:28 -0000
```

```
*****
```

```
*** 3,9 ****
```

```
*
```

```
* DO NOT EDIT-- this file is automatically generated.
```

```
* $FreeBSD$
```

```
! * created from FreeBSD: src/sys/kern/syscalls.master,v 1.154 2003/10/20 16:16:03 dwmalone Exp
```

```
*/
```

```
#define SYS_syscall 0
```

```
--- 3,9 ----
```

```
*
```

```
* DO NOT EDIT-- this file is automatically generated.
```

```
* $FreeBSD$
```

```
! * created from FreeBSD
```

```
*/
```

```
#define SYS_syscall 0
```

```
*****
```

```
*** 24,30 ****
```

```
#define SYS_chmod 15
```

```
#define SYS_chown 16
```

```
#define SYS_break 17
```

```
! #define SYS_getfsstat 18
```

```
/* 19 is old lseek */
```

```
#define SYS_getpid 20
```

```
#define SYS_mount 21
```

```
--- 24,30 ----
```

```
#define SYS_chmod 15
```

```
#define SYS_chown 16
```

```
#define SYS_break 17
```

```
! /* 18 is old getfsstat */
```

>0x7ffffff blocksize filesystem reporting

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
        /* 19 is old lseek */
#define SYS_getpid 20
#define SYS_mount 21
*****
*** 156,163 ***
        /* 150 is old getsockname */
#define SYS_nfssvc 155
        /* 156 is old getdirentries */
! #define SYS_statfs 157
! #define SYS_fstatfs 158
#define SYS_getfh 161
#define SYS_getdomainname 162
#define SYS_setdomainname 163
---- 156,163 ----
        /* 150 is old getsockname */
#define SYS_nfssvc 155
        /* 156 is old getdirentries */
! /* 157 is old statfs */
! /* 158 is old fstatfs */
#define SYS_getfh 161
#define SYS_getdomainname 162
#define SYS_setdomainname 163
*****
*** 221,227 ***
#define SYS_nstat 278
#define SYS_nfstat 279
#define SYS_nlstat 280
! #define SYS_fhstatfs 297
#define SYS_fhopen 298
#define SYS_fhstat 299
#define SYS_modnext 300
---- 221,227 ----
#define SYS_nstat 278
#define SYS_nfstat 279
#define SYS_nlstat 280
! /* 297 is old fhstatfs */
#define SYS_fhopen 298
#define SYS_fhstat 299
#define SYS_modnext 300
*****
*** 310,315 ***
---- 310,319 ----
#define SYS_uuidgen 392
#define SYS_sendfile 393
#define SYS_mac_syscall 394
+ #define SYS_getfsstat 395
+ #define SYS_statfs 396
+ #define SYS_fstatfs 397
+ #define SYS_fhstatfs 398
#define SYS_ksem_close 400
#define SYS_ksem_post 401
```

>0x7ffffff blocksize filesystem reporting

```
#define SYS_ksem_wait 402
Index: sys/sys/syscall.mk
```

```
=====
RCS file: /usr/ncvs/src/sys/sys/syscall.mk,v
```

```
retrieving revision 1.97
```

```
diff -c -r1.97 syscall.mk
```

```
*** sys/sys/syscall.mk 21 Oct 2003 07:03:27 -0000 1.97
```

```
--- sys/sys/syscall.mk 5 Nov 2003 05:13:28 -0000
```

```
*****
```

```
*** 1,7 ****
```

```
# FreeBSD system call names.
```

```
# DO NOT EDIT-- this file is automatically generated.
```

```
# $FreeBSD$
```

```
! # created from FreeBSD: src/sys/kern/syscalls.master,v 1.154 2003/10/20 16:16:03 dwmalone Exp
```

```
MIASM = \
    syscall.o \
    exit.o \
```

```
--- 1,7 ----
```

```
# FreeBSD system call names.
```

```
# DO NOT EDIT-- this file is automatically generated.
```

```
# $FreeBSD$
```

```
! # created from FreeBSD
```

```
MIASM = \
    syscall.o \
    exit.o \
```

```
*****
```

```
*** 19,25 ****
```

```
    chmod.o \
    chown.o \
    break.o \
```

```
- getfsstat.o \
```

```
    getpid.o \
    mount.o \
    unmount.o \
```

```
--- 19,24 ----
```

```
*****
```

```
*** 108,115 ****
```

```
    setsid.o \
    quotactl.o \
    nfssvc.o \
```

```
- statfs.o \
```

```
- fstatfs.o \
```

```
    getfh.o \
    getdomainname.o \
    setdomainname.o \
```

```
--- 107,112 ----
```

```
*****
```

```
*** 173,179 ****
```

```
    nstat.o \
    nfstat.o \
    nlstat.o \
```

```

- fhstatfs.o \
  fhopen.o \
  fhstat.o \
  modnext.o \
---- 170,175 ----
*****
*** 256,261 ***
---- 252,261 ----
  uuidgen.o \
  sendfile.o \
  mac_syscall.o \
+ getfsstat.o \
+ statfs.o \
+ fstatfs.o \
+ fhstatfs.o \
  ksem_close.o \
  ksem_post.o \
  ksem_wait.o \

```

Index: sys/sys/sysproto.h

=====
RCS file: /usr/ncvs/src/sys/sys/sysproto.h,v

retrieving revision 1.138

diff -c -r1.138 sysproto.h

*** sys/sys/sysproto.h 21 Oct 2003 07:03:27 -0000 1.138

--- sys/sys/sysproto.h 5 Nov 2003 05:13:29 -0000

*** 3,9 ****

*

* DO NOT EDIT-- this file is automatically generated.

* \$FreeBSD\$

! * created from FreeBSD: src/sys/kern/syscalls.master,v 1.154 2003/10/20 16:16:03 dwmalone Exp

*/

```

#ifdef _SYS_SYSPROTO_H_

```

--- 3,9 ----

*

* DO NOT EDIT-- this file is automatically generated.

* \$FreeBSD\$

! * created from FreeBSD

*/

```

#ifdef _SYS_SYSPROTO_H_

```

*** 95,105 ****

```

struct obreak_args {
    char nsize_l_[PADL_(char *)]; char * nsize; char nsize_r_[PADR_(char *)];
};

```

```

- struct getfsstat_args {

```

```

- char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];

```

```

- char bufsize_l_[PADL_(long)]; long bufsize; char bufsize_r_[PADR_(long)];

```

```

- char flags_l_[PADL_(int)]; int flags; char flags_r_[PADR_(int)];

```

```

- };
  struct getpid_args {
    register_t dummy;
  };
---- 95,100 ----
*****
*** 491,504 ***
    char flag_l_[PADL_(int)]; int flag; char flag_r_[PADR_(int)];
    char argp_l_[PADL_(caddr_t)]; caddr_t argp; char argp_r_[PADR_(caddr_t)];
  };
- struct statfs_args {
- char path_l_[PADL_(char *)]; char * path; char path_r_[PADR_(char *)];
- char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];
- };
- struct fstatfs_args {
- char fd_l_[PADL_(int)]; int fd; char fd_r_[PADR_(int)];
- char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];
- };
  struct getfh_args {
    char fname_l_[PADL_(char *)]; char * fname; char fname_r_[PADR_(char *)];
    char fhp_l_[PADL_(struct fhandle *)]; struct fhandle * fhp; char fhp_r_[PADR_(struct fhandle *)];
---- 486,491 ----
*****
*** 778,787 ***
    char path_l_[PADL_(char *)]; char * path; char path_r_[PADR_(char *)];
    char ub_l_[PADL_(struct nstat *)]; struct nstat * ub; char ub_r_[PADR_(struct nstat *)];
  };
- struct fhstatfs_args {
- char u_fhp_l_[PADL_(const struct fhandle *)]; const struct fhandle * u_fhp; char u_fhp_r_[PADR_(const
struct fhandle *)];
- char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];
- };
  struct fhopen_args {
    char u_fhp_l_[PADL_(const struct fhandle *)]; const struct fhandle * u_fhp; char
u_fhp_r_[PADR_(const struct fhandle *)];
    char flags_l_[PADL_(int)]; int flags; char flags_r_[PADR_(int)];
---- 765,770 ----
*****
*** 1128,1133 ***
---- 1111,1133 ----
    char call_l_[PADL_(int)]; int call; char call_r_[PADR_(int)];
    char arg_l_[PADL_(void *)]; void * arg; char arg_r_[PADR_(void *)];
  };
+ struct getfsstat_args {
+ char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];
+ char bufsize_l_[PADL_(long)]; long bufsize; char bufsize_r_[PADR_(long)];
+ char flags_l_[PADL_(int)]; int flags; char flags_r_[PADR_(int)];
+ };
+ struct statfs_args {
+ char path_l_[PADL_(char *)]; char * path; char path_r_[PADR_(char *)];
+ char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];

```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
+ };
+ struct fstatfs_args {
+ char fd_l_[PADL_(int)]; int fd; char fd_r_[PADR_(int)];
+ char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];
+ };
+ struct fhstatfs_args {
+ char u_fhp_l_[PADL_(const struct fhandle *)]; const struct fhandle * u_fhp; char u_fhp_r_[PADR_(const
struct fhandle *)];
+ char buf_l_[PADL_(struct statfs *)]; struct statfs * buf; char buf_r_[PADR_(struct statfs *)];
+ };
  struct ksem_close_args {
    char id_l_[PADL_(semid_t)]; semid_t id; char id_r_[PADR_(semid_t)];
  };
*****
*** 1300,1306 ****
  int chmod(struct thread *, struct chmod_args *);
  int chown(struct thread *, struct chown_args *);
  int obreak(struct thread *, struct obreak_args *);
- int getfsstat(struct thread *, struct getfsstat_args *);
  int getpid(struct thread *, struct getpid_args *);
  int mount(struct thread *, struct mount_args *);
  int unmount(struct thread *, struct unmount_args *);
--- 1300,1305 ----
*****
*** 1389,1396 ****
  int setsid(struct thread *, struct setsid_args *);
  int quotactl(struct thread *, struct quotactl_args *);
  int nfssvc(struct thread *, struct nfssvc_args *);
- int statfs(struct thread *, struct statfs_args *);
- int fstatfs(struct thread *, struct fstatfs_args *);
  int getfh(struct thread *, struct getfh_args *);
  int getdomainname(struct thread *, struct getdomainname_args *);
  int setdomainname(struct thread *, struct setdomainname_args *);
--- 1388,1393 ----
*****
*** 1452,1458 ****
  int nstat(struct thread *, struct nstat_args *);
  int nfstat(struct thread *, struct nfstat_args *);
  int nlstat(struct thread *, struct nlstat_args *);
- int fhstatfs(struct thread *, struct fhstatfs_args *);
  int fhopen(struct thread *, struct fhopen_args *);
  int fhstat(struct thread *, struct fhstat_args *);
  int modnext(struct thread *, struct modnext_args *);
--- 1449,1454 ----
*****
*** 1536,1541 ****
--- 1532,1541 ----
  int uuidgen(struct thread *, struct uuidgen_args *);
  int sendfile(struct thread *, struct sendfile_args *);
  int mac_syscall(struct thread *, struct mac_syscall_args *);
+ int getfsstat(struct thread *, struct getfsstat_args *);
```

>0x7ffffff blocksize filesystem reporting

freebsd-arch: >0x7ffffff blocksize filesystem reporting

```
+ int statfs(struct thread *, struct statfs_args *);
+ int fstatfs(struct thread *, struct fstatfs_args *);
+ int fhstatfs(struct thread *, struct fhstatfs_args *);
  int ksem_close(struct thread *, struct ksem_close_args *);
  int ksem_post(struct thread *, struct ksem_post_args *);
  int ksem_wait(struct thread *, struct ksem_wait_args *);
*****
*** 1748,1753 ***
--- 1748,1770 ----

#ifdef COMPAT_FREEBSD4

+ struct freebsd4_getfsstat_args {
+ char buf_l_[PADL_(struct ostats *)]; struct ostats * buf; char buf_r_[PADR_(struct ostats *)];
+ char bufsize_l_[PADL_(long)]; long bufsize; char bufsize_r_[PADR_(long)];
+ char flags_l_[PADL_(int)]; int flags; char flags_r_[PADR_(int)];
+ };
+ struct freebsd4_statfs_args {
+ char path_l_[PADL_(char *)]; char * path; char path_r_[PADR_(char *)];
+ char buf_l_[PADL_(struct ostats *)]; struct ostats * buf; char buf_r_[PADR_(struct ostats *)];
+ };
+ struct freebsd4_fstatfs_args {
+ char fd_l_[PADL_(int)]; int fd; char fd_r_[PADR_(int)];
+ char buf_l_[PADL_(struct ostats *)]; struct ostats * buf; char buf_r_[PADR_(struct ostats *)];
+ };
+ struct freebsd4_fhstatfs_args {
+ char u_fhp_l_[PADL_(const struct fhandle *)]; const struct fhandle * u_fhp; char u_fhp_r_[PADR_(const
struct fhandle *)];
+ char buf_l_[PADL_(struct ostats *)]; struct ostats * buf; char buf_r_[PADR_(struct ostats *)];
+ };
  struct freebsd4_sendfile_args {
    char fd_l_[PADL_(int)]; int fd; char fd_r_[PADR_(int)];
    char s_l_[PADL_(int)]; int s; char s_r_[PADR_(int)];
*****
*** 1765,1770 ***
--- 1782,1791 ----
  struct freebsd4_sigreturn_args {
    char signtxp_l_[PADL_(const struct ucontext4 *)]; const struct ucontext4 * signtxp; char
signtxp_r_[PADR_(const struct ucontext4 *)];
  };
+ int freebsd4_getfsstat(struct thread *, struct freebsd4_getfsstat_args *);
+ int freebsd4_statfs(struct thread *, struct freebsd4_statfs_args *);
+ int freebsd4_fstatfs(struct thread *, struct freebsd4_fstatfs_args *);
+ int freebsd4_fhstatfs(struct thread *, struct freebsd4_fhstatfs_args *);
  int freebsd4_sendfile(struct thread *, struct freebsd4_sendfile_args *);
  int freebsd4_sigaction(struct thread *, struct freebsd4_sigaction_args *);
  int freebsd4_sigreturn(struct thread *, struct freebsd4_sigreturn_args *);
Index: sys/kern/init_sysent.c
```

```
=====
RCS file: /usr/ncvs/src/sys/kern/init_sysent.c,v
retrieving revision 1.158
```

freebsd-arch: >0x7ffffff blocksize filesystem reporting

diff -c -r1.158 init_sysent.c

*** sys/kern/init_sysent.c 21 Oct 2003 07:03:27 -0000 1.158

--- sys/kern/init_sysent.c 5 Nov 2003 05:13:29 -0000

*** 3,9 ****

*

* DO NOT EDIT--- this file is automatically generated.

* \$FreeBSD\$

! * created from FreeBSD: src/sys/kern/syscalls.master,v 1.154 2003/10/20 16:16:03 dwmalone Exp

*/

#include "opt_compat.h"

--- 3,9 ----

*

* DO NOT EDIT--- this file is automatically generated.

* \$FreeBSD\$

! * created from FreeBSD

*/

#include "opt_compat.h"

*** 46,52 ****

{ AS(chmod_args), (sy_call_t *)chmod }, /* 15 = chmod */

{ AS(chown_args), (sy_call_t *)chown }, /* 16 = chown */

{ SYF_MPSAFE | AS(obreak_args), (sy_call_t *)obreak }, /* 17 = break */

! { AS(getfsstat_args), (sy_call_t *)getfsstat }, /* 18 = getfsstat */

{ compat(AS(olseek_args),lseek) }, /* 19 = old lseek */

{ SYF_MPSAFE | 0, (sy_call_t *)getpid }, /* 20 = getpid */

{ AS(mount_args), (sy_call_t *)mount }, /* 21 = mount */

--- 46,52 ----

{ AS(chmod_args), (sy_call_t *)chmod }, /* 15 = chmod */

{ AS(chown_args), (sy_call_t *)chown }, /* 16 = chown */

{ SYF_MPSAFE | AS(obreak_args), (sy_call_t *)obreak }, /* 17 = break */

! { compat4(AS(freebsd4_getfsstat_args),getfsstat) }, /* 18 = old getfsstat */

{ compat(AS(olseek_args),lseek) }, /* 19 = old lseek */

{ SYF_MPSAFE | 0, (sy_call_t *)getpid }, /* 20 = getpid */

{ AS(mount_args), (sy_call_t *)mount }, /* 21 = mount */

*** 185,192 ****

{ 0, (sy_call_t *)nosys }, /* 154 = nosys */

{ SYF_MPSAFE | AS(nfssvc_args), (sy_call_t *)nosys }, /* 155 = nfssvc */

{ compat(AS(ogetdirentries_args),getdirentries) }, /* 156 = old getdirentries */

! { AS(statfs_args), (sy_call_t *)statfs }, /* 157 = statfs */

! { AS(fstatfs_args), (sy_call_t *)fstatfs }, /* 158 = fstatfs */

{ 0, (sy_call_t *)nosys }, /* 159 = nosys */

{ 0, (sy_call_t *)nosys }, /* 160 = nosys */

{ AS(getfh_args), (sy_call_t *)getfh }, /* 161 = getfh */

--- 185,192 ----

{ 0, (sy_call_t *)nosys }, /* 154 = nosys */

{ SYF_MPSAFE | AS(nfssvc_args), (sy_call_t *)nosys }, /* 155 = nfssvc */

{ compat(AS(ogetdirentries_args),getdirentries) }, /* 156 = old getdirentries */

>0x7ffffff blocksize filesystem reporting

freebsd-arch: >0x7ffffff blocksizes filesystem reporting

! { compat4(