FreeBSD の msdosfs の実装より

FreeBSD 5.1-RELEASE のソースコードより

■ VNODE に関する操作？

/sys/fs/msdosfs/msdosfs_vnops.c

```
/* Global vfs data structures for msdosfs */
vop_t **msdosfs_vnodeop_p;
static struct vnodeopv_entry_desc msdosfs_vnodeop_entries[] = {
    { &vop_default_desc,        (vop_t *) vop_defaultop },
    { &vop_access_desc,     (vop_t *) msdosfs_access },
    { &vop_bmap_desc,       (vop_t *) msdosfs_bmap },
    { &vop_cachedlookup_desc,   (vop_t *) msdosfs_lookup },
    { &vop_close_desc,      (vop_t *) msdosfs_close },
    { &vop_create_desc,     (vop_t *) msdosfs_create },
    { &vop_fsync_desc,      (vop_t *) msdosfs_fsync },
    { &vop_getattr_desc,        (vop_t *) msdosfs_getattr },
    { &vop_inactive_desc,       (vop_t *) msdosfs_inactive },
    { &vop_link_desc,       (vop_t *) msdosfs_link },
    { &vop_lookup_desc,     (vop_t *) vfs_cache_lookup },
    { &vop_mkdir_desc,      (vop_t *) msdosfs_mkdir },
    { &vop_mknod_desc,      (vop_t *) msdosfs_mknod },
    { &vop_pathconf_desc,       (vop_t *) msdosfs_pathconf },
    { &vop_print_desc,      (vop_t *) msdosfs_print },
    { &vop_read_desc,       (vop_t *) msdosfs_read },
    { &vop_readdir_desc,        (vop_t *) msdosfs_readdir },
    { &vop_reclaim_desc,        (vop_t *) msdosfs_reclaim },
    { &vop_remove_desc,     (vop_t *) msdosfs_remove },
    { &vop_rename_desc,     (vop_t *) msdosfs_rename },
    { &vop_rmdir_desc,      (vop_t *) msdosfs_rmdir },
    { &vop_setattr_desc,        (vop_t *) msdosfs_setattr },
    { &vop_strategy_desc,       (vop_t *) msdosfs_strategy },
    { &vop_symlink_desc,        (vop_t *) msdosfs_symlink },
    { &vop_write_desc,      (vop_t *) msdosfs_write },
    { NULL, NULL }
};
static struct vnodeopv_desc msdosfs_vnodeop_opv_desc =
    { &msdosfs_vnodeop_p, msdosfs_vnodeop_entries };
```

■ VFS としてのインターフェース？

/sys/fs/msdosfs/msdosfs_vfsops.c

```
static struct vfsops msdosfs_vfsops = {
    .vfs_fhtovp =       msdosfs_fhtovp,
    .vfs_init =     msdosfs_init,
    .vfs_mount =        msdosfs_mount,
    .vfs_root =     msdosfs_root,
    .vfs_statfs =       msdosfs_statfs,
    .vfs_sync =     msdosfs_sync,
    .vfs_uninit =       msdosfs_uninit,
    .vfs_unmount =      msdosfs_unmount,
    .vfs_vptofh =       msdosfs_vptofh,
};
```

■ init 関数？

/sys/fs/msdosfs/msdosfs_denode.c

```
/*ARGSUSED*/
```

```
int
msdosfs_init(vfsp)
    struct vfsconf *vfsp;
{
    /*
     * The following lines prevent us from initializing the mutex
     * init multiple times.  I'm not sure why we get called multiple
     * times, but the following prevents the panic when we initalize
     * the mutext the second time.  XXX BAD XXX
     */
    if (dehash_init) {
        printf("Warning: msdosfs_init called more than once!?n");
        return (0);
    }
    dehash_init++;
    dehashtbl = hashinit(desiredvnodes/2, M_MSDOSFSMNT, &dehash);
    mtx_init(&dehash_mtx, "msdosfs dehash", NULL, MTX_DEF);
    return (0);
}
```

■ struct vfsconf

/usr/include/sys/mount.h

```
/*
 * Filesystem configuration information. One of these exists for each
 * type of filesystem supported by the kernel. These are searched at
 * mount time to identify the requested filesystem.
 */
struct vfsconf {
    struct  vfsops *vfc_vfsops; /* filesystem operations vector */
    char    vfc_name[MFSNAMELEN];   /* filesystem type name */
    int vfc_typenum;        /* historic filesystem type number */
    int vfc_refcount;       /* number mounted of this type */
    int vfc_flags;      /* permanent flags */
    struct  vfsoptdecl *vfc_opts;   /* mount options */
    struct  vfsconf *vfc_next;  /* next in list */
};
```

■ユーザーランド？気になる

```
/* Userland version of the struct vfsconf. */
struct xvfsconf{...}
```