

NAME

vfscnf - vfs configuration information

SYNOPSIS

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

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

```
int
```

```
vfscnf_register(struct vfscnf *vfc);
```

```
int
```

```
vfscnf_unregister(struct vfscnf *vfc);
```

```
int
```

```
vfscnf_modevent(module_t mod, int type, void *data);
```

DESCRIPTION

Each file system type known to the kernel has a *vfscnf* structure that contains the information required to create a new mount of that file systems type.

```
struct vfscnf {
    struct    vfsops *vfc_vfsops;          /* file system operations vector */
    char      vfc_name[MFSNAMELEN]; /* file system type name */
    int       vfc_tynenum;                /* historic file system type number */
    int       vfc_refcount;                /* number mounted of this type */
    int       vfc_flags;                   /* permanent flags */
    struct    vfscnf *vfc_next; /* next in list */
};
```

When a new file system is mounted, *mount(2)* does a lookup of the *vfscnf* structure by its name, and if it is not already registered, attempts to load a kernel module for it. The file system operations for the new mount point are taken from *vfc_vfsops*, and *mnt_vfc* in the *mount* structure is made to point directly at the *vfscnf* structure for the file system type. The file system type number is taken from *vfc_tynenum* which was assigned in **vfscnf_register()**, and the mount flags are taken from a mask of *vfc_flags*. Each time a file system of a given type is mounted, *vfc_refcount* is incremented.

vfscnf_register() takes a new *vfscnf* structure and adds it to the list of existing file systems. If the type has not already been registered, it is initialized by calling the **vfscnf_init()** function in the file system operations vector. **vfscnf_register()** also updates the oid's of any sysctl nodes for this file system type to be the same as the newly assigned type number.

vfs_unregister() unlinks *vfc* from the list of registered file system types if there are currently no mounted instances. If the **vfs_uninit()** function in the file systems initialization vector is defined, it is called.

vfs_modevent() is registered by **VFS_SET()** to handle the loading and unloading of file system kernel modules. In the case of **MOD_LOAD**, **vfs_register()** is called. In the case of **MOD_UNLOAD**, **vfs_unregister()** is called.

RETURN VALUES

vfs_register() returns 0 if successful; otherwise, **EEXIST** is returned indicating that the file system type has already been registered.

vfs_unregister() returns 0 if successful. If no *vfscnf* entry can be found matching the name in *vfc*, **EINVAL** is returned. If the reference count of mounted instances of the file system type is not zero, **EBUSY** is returned. If **vfs_uninit()** is called, any errors it returns will be returned by **vfs_unregister()**.

vfs_modevent() returns the result of the call to **vfs_register()** or **vfs_unregister()**, whatever the case.

SEE ALSO

mount(2), *vfs_rootmountalloc*(9), **VFS_SET**(9)

AUTHORS

This manual page was written by Chad David <*davidc@acns.ab.ca*>.