NAME

free_mntarg, **kernel_mount**, **mount_arg**, **mount_argb**, **mount_argf**, **mount_argsu -** functions provided as part of the kernel mount interface

SYNOPSIS

void
free_mntarg(*struct mntarg *ma*);

int
kernel_mount(*struct mntarg *ma, int flags*);

*struct mntarg **
mount_arg(*struct mntarg *ma, const char *name, const void *val, int len*);

*struct mntarg **
mount_argb(*struct mntarg *ma, int flag, const char *name*);

*struct mntarg **
mount_argf(*struct mntarg *ma, const char *name, const char *fmt, ...*);

*struct mntarg **
mount_argsu(*struct mntarg *ma, const char *name, const void *val, int len*);

DESCRIPTION

The **kernel_mount**() family of functions are provided as an API for building a list of mount arguments which will be used to mount file systems from inside the kernel. By accumulating a list of arguments, the API takes shape and provides the information necessary for the kernel to control the mount(8) utility. When an error occurs, the process will stop. This will not cause a panic(9).

The header of the structure is stored in *src/sys/kern/vfs_mount.c* which permits automatic structure creation to ease the mount process. Memory allocation must always be freed when the entire process is complete, it is an error otherwise.

The **free_mntarg**() function is used to free or clear the *mntarg* structure.

The **kernel_mount**() function pulls information from the structure to perform the mount request on a given file system. Additionally, the **kernel_mount**() function always calls the **free_mntarg**() function. If *ma* contains any error code generated during the construction, that code will be called and the file system mount will not be attempted.

The **mount_arg**() function takes a plain argument and crafts parts of the structure with regards to various mount options. If the length is a value less than 0, strlen(3) is used. This argument will be referenced until either **free_mntarg**() or **kernel_mount**() is called.

The **mount_argb**() function is used to add boolean arguments to the structure. The *flag* is the boolean value and *name* must start with "no", otherwise a panic will occur.

The **mount_argf**() function adds printf(9) style arguments to the current structure.

The **mount_argsu**() function will add arguments to the structure from a userland string.

EXAMPLES

An example of the ***_cmount**() function:

```
static int
msdosfs_cmount(struct mntarg *ma, void *data, int flags, struct thread *td)
{
        struct msdosfs_args args;
        int error;

        if (data == NULL)
                return (EINVAL);
        error = copyin(data, &args, sizeof(args));
        if (error)
                return (error);

        ma = mount_argsu(ma, "from", args.fspec, MAXPATHLEN);
        ma = mount_arg(ma, "export", &args.export, sizeof(args.export));
        ma = mount_argf(ma, "uid", "%d", args.uid);
        ma = mount_argf(ma, "gid", "%d", args.gid);
        ma = mount_argf(ma, "mask", "%d", args.mask);
        ma = mount_argf(ma, "dirmask", "%d", args.dirmask);

        ma = mount_argb(ma, args.flags & MSDOSFSMNT_SHORTNAME, "noshortname");
        ma = mount_argb(ma, args.flags & MSDOSFSMNT_LONGNAME, "nolongname");
        ma = mount_argb(ma, !(args.flags & MSDOSFSMNT_NOWIN95), "nowin95");
        ma = mount_argb(ma, args.flags & MSDOSFSMNT_KICONV, "nokiconv");

        ma = mount_argsu(ma, "cs_win", args.cs_win, MAXCSLEN);
        ma = mount_argsu(ma, "cs_dos", args.cs_dos, MAXCSLEN);
```

```
        ma = mount_argsu(ma, "cs_local", args.cs_local, MAXCSLEN);

        error = kernel_mount(ma, flags);

        return (error);
}
```

SEE ALSO

VFS(9), VFS_MOUNT(9)

HISTORY

The **kernel_mount**() family of functions and this manual page first appeared in FreeBSD 6.0.

AUTHORS

The **kernel_mount**() family of functions and API was developed by Poul-Henning Kamp <*phk@FreeBSD.org*>. This manual page was written by Tom Rhodes <*trhodes@FreeBSD.org*>.