FS_MOUNT

Purpose

Examination of a volume by an FSD to see if it recognizes the file system format.

Calling Sequence

```
int far pascal FS_MOUNT(flag, pvpfsi, pvpfsd, hVPB, pBoot)

unsigned short flag;
struct vpfsi far * pvpfsi;
struct vpfsd far * pvpfsd;
unsigned short hVPB;
char far * pBoot;
```

Where

flag	**indicates operation requested.**
flag == 0	indicates that the FSD is requested to mount or accept a volume.
flag == 1	indicates that the FSD is being advised that the specified volume has been removed.
flag == 2	
flag == 3	indicates that the FSD is requested to accept the volume regardless of recognition in preparation for formatting for use with the FSD.
All other values are reserved.	

The value passed to the FSD will be valid.

pvpfsi is a pointer to the file-system-independent portion of VPB.

If the media contains an OS/2-recognizable boot sector, then the *vpi_vid* field contains the 32-bit identifier for that volume. If the media does not contain such a boot sector, the FSD must generate a unique label for the media and place it into the *vpi_vid* field.

pvpfsd is a pointer to the file-system-dependent portion of VPB.

The FSD may store information as necessary into this area.

hVPB is the handle to the volume

pBoot is a pointer to sector 0 read from the media.

This pointer is only valid when *flag* == 0.The buffer the pointer refers to must not be modified. The pointer is always valid and does not need to be verified when *flag* == 0. If a read error occurred, the buffer will contain zeroes.

Remarks

The FSD examines the volume presented and determine whether it recognizes the file system. If it does, it returns zero, after having filled in appropriate parts of the *vpfsi* and *vpfsd* data structures. The *vpi_vid* and *vpi_text* fields must be filled in by the FSD. If the FSD has an OS/2 format boot sector, it

must convert the label from the media into ASCIIZ form. The *vpi_hDev* field is filled in by OS/2. If the volume is unrecognized, the driver returns non-zero.

The *vpi_text* and *vpi_vid* must be updated by the FSD each time these values change.

The contents of the *vpfsd* data structure are as follows:

FLAG = 0 The FSD is expected to issue an *FSD_FINDDUPHVPB* to see if a duplicate VPB exists. If one does exist, the VPB fs dependent area of the new VPB is invalid and the new VPB will be unmounted after the FSD returns from the MOUNT. The FSD is expected to update the FS dependent area of the old duplicate VPB. If no duplicate VPB exists, the FSD should initialize the FS dependent area.

FLAG = 1 VPB FS dependent part is same as when FSD last modified it.

FLAG = 2 VPB FS dependent part is same as when FSD last modified it.

After media recognition time, the volume parameters may be examined using the *FSH_GETVOLPARM* call. The volume parameters should not be changed after media recognition time.

During a mount request, the FSD may examine other sectors on the media by using *FSH_DOVOLIO* to perform the I/O. If an uncertain-media return is detected, the FSD is expected to clean up and return an UNCERTAIN MEDIA error in order to allow the volume mount logic to restart on the newly-inserted media. The FSD must provide the buffer to use for additional I/O.

The OS/2 kernel manages the VPB through a reference count. All volume- specific objects are labeled with the appropriate volume handle and represent references to the VPB. When all kernel references to a volume disappear, *FS_MOUNT* is called with *flag* == 2, indicating a dismount request.

When the kernel detects that a volume has been removed from its driver, but there are still outstanding references to the volume, *FS_MOUNT* is called with *flag* == 1 to allow the FSD to drop clean (or other regenerable) data for the volume. Data which is dirty and cannot be regenerated should be kept so that it may be written to the volume when it is remounted in the drive.

When a volume is to be formatted for use with an FSD, the kernel calls the FSD's *FS_MOUNT* entry point with *flag* == 3 to allow the FSD to prepare for the format operation. The FSD should accept the volume even if it is not a volume of the type that FSD recognizes, since the point of format is to change the file system on the volume. The operation may fail if formatting does not make sense. (For example, an FSD which supports only CD-ROM.)

Since the hardware does not allow for kernel-mediated removal of media, it is certain that the unmount request is issued when the volume is not present in any drive.