

FS_READ

Purpose

Read the specified number of bytes from a file to a buffer location.

Calling Sequence

```
int far pascal FS_READ(psffsi, psffsd, pData, pLen, IOflag)

struct sffsi far * psffsi;
struct sffsd far * psffsd;
char far * pData;
unsigned short far * pLen;
unsigned short IOflag;
```

Where

psffsi is a pointer to the file-system-independent portion of an open file instance.

sfi_position is the location within the file where the data is to be read from. The FSD should update the *sfi_position* field.

psffsd is a pointer to the file-system-dependent portion of an open file instance.

pData is the address of the application data area.

Addressing of this data area has not been validated by the kernel (see *FSH_PROBEBUF*).

pLen is a pointer to the length of the application data area.

On input, this is the number of bytes to be read. On output, this is the number of bytes successfully read. If the application data area is smaller than the length, no transfer is to take place. The FSD will not be called for zero length reads. The FSD does not need to verify this pointer.

IOflag indicates information about the operation on the handle.

<i>IOflag</i> == 0x0010	indicates write-through
<i>IOflag</i> == 0x0020	indicates no-cache

Remarks

If read is successful and is a file, the FSD should set *ST_SREAD* and *ST_PREAD* to make the kernel time stamp the last modification time in the SFT.

Of the information passed in *IOflag*, the write-through bit is a mandatory bit in that any data written to the block device must be put out on the medium before the device driver returns. The no-cache bit, on the other hand, is an advisory bit that says whether the data being transferred is worth caching or not.

From:

<http://www.osfree.org/doku/> - **osFree wiki**

Permanent link:

<http://www.osfree.org/doku/doku.php?id=en:ibm:ifs:routines:read>

Last update: **2014/05/13 02:02**

