

FSH_SEGALLOC

Purpose

This function allocates a GDT or LDT selector. The selector will have read/write access. An FSD may call this function.

Calling Sequence

```
int far pascal FSH_SEGALLOC(flags, cbSeg, pSel)

unsigned short flags;
unsigned long cbSeg;
unsigned short far * pSel;
```

Where

flags indicate GDT/LDT, protection ring, swappable/non-swappable.

Bit 0x0001 off	indicates GDT selector returned.
Bit 0x0001 on	indicates LDT selector returned.
Bit 0x0002 off	indicates non-swappable memory.
Bit 0x0002 on	indicates swappable memory.
Bits 13 and 14	are the desired ring number.
All other bits	are reserved and must be zero.

cbSeg is the length of the segment.

pSel is the far address of the location where the allocated selector will be stored.

Returns

If no error is detected, a zero error code is returned. If an error is detected, one of the following error codes is returned:

- `ERROR_INTERRUPT` - the current thread received a signal.
- `ERROR_INVALID_PARAMETER` - the reserved bits in flags are set or requested size is too large.
- `ERROR_NOT_ENOUGH_MEMORY` - too much memory is allocated.

Remarks

It is strongly suggested that FSDs allocate all their data at protection level 0 for maximum protection from user programs.

GDT selectors are a scarce resource; the FSD must be prepared to expect an error for allocation of a GDT segment. The FSD should limit itself to a maximum of 10 GDT segments. It is suggested that a large segment be allocated for each type of data and divided into per-process records.

FSH_SEGALLOC may block.

Take care to avoid deadlocks between swappable segments and swapper requests.

Note: OS/2 does not validate input parameters. An FSD, therefore, should call *FSH_PROBEBUF* where appropriate.

From:

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

Permanent link:

<https://www.osfree.org/doku/doku.php?id=en:ibm:ifs:helpers:segalloc>

Last update: **2014/05/13 08:03**

