NAME

modify_ldt − get or set ldt

SYNOPSIS

#include <sys/types.h>

int modify_ldt(int *func***, void ****ptr***, unsigned long** *bytecount***);**

DESCRIPTION

modify_ldt() reads or writes the local descriptor table (ldt) for a process. The ldt is a per-process memory management table used by the i386 processor. For more information on this table, see an Intel 386 processor handbook.

When *func* is 0, **modify_ldt**() reads the ldt into the memory pointed to by *ptr*. The number of bytes read is the smaller of *bytecount* and the actual size of the ldt.

When *func* is 1, **modify_ldt**() modifies one ldt entry. *ptr* points to a *user_desc* structure and *bytecount* must equal the size of this structure.

The *user_desc* structure is defined in *<asm/ldt.h>* as:

```
struct user_desc {
    unsigned int  entry_number;
    unsigned long base_addr;
    unsigned int  limit;
    unsigned int  seg_32bit:1;
    unsigned int  contents:2;
    unsigned int  read_exec_only:1;
    unsigned int  limit_in_pages:1;
    unsigned int  seg_not_present:1;
    unsigned int  useable:1;
};
```

In Linux 2.4 and earlier, this structure was named *modify_ldt_ldt_s*.

RETURN VALUE

On success, **modify_ldt**() returns either the actual number of bytes read (for reading) or 0 (for writing). On failure, **modify_ldt**() returns −1 and sets *errno* to indicate the error.

ERRORS

EFAULT
ptr points outside the address space.

EINVAL
ptr is 0, or *func* is 1 and *bytecount* is not equal to the size of the structure *modify_ldt_ldt_s*, or *func* is 1 and the new ldt entry has invalid values.

ENOSYS
func is neither 0 nor 1.

CONFORMING TO

This call is Linux-specific and should not be used in programs intended to be portable.

NOTES

Glibc does not provide a wrapper for this system call; call it using **syscall**(2).

SEE ALSO

vm86(2)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.