NAME

kexec_load − loads a new kernel image to memory

SYNOPSIS

#include <syscall.h>
#include <kexec.h>

long kexec_load(unsigned long *entry***, unsigned long** *nr_segments***,**
struct kexec_segment ***segments*, unsigned long** *flags***);**

DESCRIPTION

kexec_load loads the new kernel from the current address space. This system call can only be used by root.

entry is a pointer to the entry point of newly loaded executable image. This is the memory location where kernel will jump to and start executing instructions of newly loaded image.

nr_segments denotes the number of segments which will be passed to **kexec_load**. The value must not be greater than *KEXEC_SEGMENT_MAX*.

segments denotes a pointer to the first element of an array of **kexec_segment** elements. A **kexec_segment** element contains the details of a segment to be loaded in memory.

flags Sixteen most significant bits of the flag are used to communicate the architecture information (**KEXEC_ARCH_***). The values for various architectures are same as defined by ELF specifications.

Lower sixteen bits have been reserved for miscellaneous information. Currently only one bit is being used and rest fifteen have been reserved for future use.

The least significant bit (**KEXEC_ON_CRASH**) can be set to inform the kernel that the memory memory image being loaded is to be executed upon a system crash and not regular boot. For regular boot, this bit is cleared.

RETURN VALUE

On success, zero is returned. On error, nonzero value is returned, and *errno* is set appropriately.

ERRORS

EPERM the calling process has not sufficient permissions (is not root).

EINVAL the *flags* argument contains an invalid combination of flags, or *nr_segments* is greater than **KEXEC_SEGMENT_MAX**.

ENOMEM there is not enough memory to store the kernel image.

EBUSY the memory location which should be written to is not available now.

AVAILABILITY

This syscall is implemented only since kernel 2.6.13