NAME

kexec_load - load a new kernel for later execution

SYNOPSIS

#include <linux/kexec.h>

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

Note: There is no glibc wrapper for this system call; see NOTES.

DESCRIPTION

The **kexec_load**() system call loads a new kernel that can be executed later by reboot(2).

The *flags* argument is a bit mask that controls the operation of the call. The following values can be specified in *flags*:

KEXEC_ON_CRASH (since Linux 2.6.13)
: Execute the new kernel automatically on a system crash.

KEXEC_PRESERVE_CONTEXT (since Linux 2.6.27)
: Preserve the system hardware and software states before executing the new kernel. This could be used for system suspend. This flag is available only if the kernel was configured with **CONFIG_KEXEC_JUMP**, and is effective only if *nr_segments* is greater than 0.

The high-order bits (corresponding to the mask 0xffff0000) of *flags* contain the architecture of the to-be-executed kernel. Specify (OR) the constant **KEXEC_ARCH_DEFAULT** to use the current architecture, or one of the following architecture constants **KEXEC_ARCH_386**, **KEXEC_ARCH_68K**, **KEXEC_ARCH_X86_64**, **KEXEC_ARCH_PPC**, **KEXEC_ARCH_PPC64**, **KEXEC_ARCH_IA_64**, **KEXEC_ARCH_ARM**, **KEXEC_ARCH_S390**, **KEXEC_ARCH_SH**, **KEXEC_ARCH_MIPS**, and **KEXEC_ARCH_MIPS_LE**. The architecture must be executable on the CPU of the system.

The *entry* argument is the physical entry address in the kernel image. The *nr_segments* argument is the number of segments pointed to by the *segments* pointer; the kernel imposes an (arbitrary) limit of 16 on the number of segments. The*se gments* argument is an array of *kexec_segment* structures which define the kernel layout:

```
struct kexec_segment {
void *buf; /* Buffer in user space */
size_t bufsz; /* Buffer length in user space */
void *mem; /* Physical address of kernel */
size_t memsz; /* Physical address length */
};
```

The kernel image defined by *segments* is copied from the calling process into previously reserved memory.

RETURN VALUE

On success, **kexec_load**() returns 0. On error, -1 is returned and *errno* is set to indicate the error.

ERRORS

EBUSY
: Another crash kernel is already being loaded or a crash kernel is already in use.

EINVAL
: *flags* is invalid; or *nr_segments* is too large

EPERM
: The caller does not have the **CAP_SYS_BOOT** capability.

VERSIONS

The **kexec_load**() system call first appeared in Linux 2.6.13.

CONFORMING TO

This system call is Linux-specific.

NOTES

Currently, there is no glibc support for **kexec_load**(). Call it using syscall(2).

The required constants are in the Linux kernel source file *linux/kexec.h*, which is not currently exported to glibc. Therefore, these constants must be defined manually.

This system call is available only if the kernel was configured with **CONFIG_KEXEC**.

SEE ALSO

reboot(2), syscall(2)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at http://www.kernel.org/doc/man-pages/.