## NAME

**reboot** - reboot system or halt processor

## LIBRARY

Standard C Library (libc, -lc)

## SYNOPSIS

**#include <unistd.h>**
**#include <sys/reboot.h>**

*int*
**reboot**(*int howto*);

## DESCRIPTION

The **reboot**() system call reboots the system. Only the super-user may reboot a machine on demand. However, a reboot is invoked automatically in the event of unrecoverable system failures.

The *howto* argument is a mask of options; the system call interface allows the following options, defined in the include file <*sys/reboot.h*>, to be passed to the new kernel or the new bootstrap and init programs.

RB_AUTOBOOT The default, causing the system to reboot in its usual fashion.

RB_ASKNAME Normally the system only prompts the user if the loader specified root file system has an error. This flag forces it to always prompt the user for the root partition.

RB_DFLTROOT Use the compiled in root device. Normally, the system uses the device from which it was booted as the root device if possible. (The default behavior is dependent on the ability of the bootstrap program to determine the drive from which it was loaded, which is not possible on all systems.)

RB_DUMP Dump kernel memory before rebooting; see savecore(8) for more information.

RB_HALT The processor is simply halted; no reboot takes place. This option should be used with caution.

RB_POWERCYCLE After halting, the shutdown code will do what it can to turn off the power and then turn the power back on. This requires hardware support, usually an auxiliary microprocessor that can sequence the power supply. At present only the ipmi(4) driver implements this feature.

RB_POWEROFF After halting, the shutdown code will do what it can to turn off the power. This requires hardware support.

RB_KDB Load the symbol table and enable a built-in debugger in the system. This option will have no useful function if the kernel is not configured for debugging. Several other options have different meaning if combined with this option, although their use may not be possible via the **reboot**() system call. See ddb(4) for more information.

RB_NOSYNC Normally, the disks are sync'd (see sync(8)) before the processor is halted or rebooted. This option may be useful if file system changes have been made manually or if the processor is on fire.

RB_REROOT Instead of rebooting, unmount all filesystems except the one containing currently-running executable, and mount root filesystem using the same mechanism which is used during normal boot, based on vfs.root.mountfrom kenv(1) variable.

RB_RDONLY Initially mount the root file system read-only. This is currently the default, and this option has been deprecated.

RB_SINGLE Normally, the reboot procedure involves an automatic disk consistency check and then multi-user operations. RB_SINGLE prevents this, booting the system with a single-user shell on the console. RB_SINGLE is actually interpreted by the init(8) program in the newly booted system.

When no options are given (i.e., RB_AUTOBOOT is used), the system is rebooted from file "kernel" in the root file system of unit 0 of a disk chosen in a processor specific way. An automatic consistency check of the disks is normally performed (see fsck(8)).

## RETURN VALUES

If successful, this call never returns. Otherwise, a -1 is returned and an error is returned in the global variable *errno*.

## ERRORS

[EPERM] The caller is not the super-user.

## SEE ALSO

crash(8), halt(8), init(8), reboot(8), savecore(8), reboot(9)

## HISTORY

The **reboot**() system call appeared in 4.0BSD.