SIGNAL

#include <signal.h>

typedef void (*sighandler_t)(int);

sighandler_t signal(int signum, sighandler_t handler);

DESCRIPTION

If the signal signum is delivered to the process, then one of the following happens:

* If the disposition is set to SIG_IGN, then the signal is ignored.

* If the disposition is set to SIG_DFL, then the default action associated with the signal occurs.

* If the disposition is set to a function, then first either the disposition is reset to SIG_DFL, or the signal is blocked, and then handler is called with argument signum. If invocation of the handler caused the signal to be blocked, then the signal is unblocked upon return from the handler.

The signals SIGKILL and SIGSTOP cannot be caught or ignored.

RETURN VALUE

signal() returns the previous value of the signal handler, or SIG_ERR on error.

KILL

kill - send a signal to a process

#include <sys/types.h>

#include <signal.h>

int kill(pid_t pid, int sig);

If pid is positive, then signal sig is sent to the process with the ID specified by pid.

If pid equals 0, then sig is sent to every process in the process group of the calling process.

If pid equals -1, then sig is sent to every process for which the calling process has permission to send signals, except for process 1 (init), but see below.

If pid is less than -1, then sig is sent to every process in the process group whose ID is -pid.

If sig is 0, then no signal is sent, but existence and permission checks are still performed; this can be used to check for the existence of a process ID or process group ID that the caller is permitted to signal.

RETURN VALUE

On success (at least one signal was sent), zero is returned. On error, -1 is returned

ERRORS

EINVAL An invalid signal was specified.

EPERM The process does not have permission to send the signal to any of the target processes.

ESRCH The process or process group does not exist. Note that an existing process might be a zombie, a process that has terminated execution, but has not yet been waited for.