NAME

sigpending, rt_sigpending − examine pending signals

LIBRARY

Standard C library (*libc*, −*lc*)

SYNOPSIS

```
#include <signal.h>

int sigpending(sigset_t *set);
```

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

sigpending():
_POSIX_C_SOURCE

DESCRIPTION

sigpending() returns the set of signals that are pending for delivery to the calling thread (i.e., the signals which have been raised while blocked). The mask of pending signals is returned in*set*.

RETURN VALUE

sigpending() returns 0 on success. On failure, −1 is returned and *errno* is set to indicate the error.

ERRORS

EFAULT
set points to memory which is not a valid part of the process address space.

STANDARDS

POSIX.1-2001, POSIX.1-2008.

NOTES

See **sigsetops**(3) for details on manipulating signal sets.

If a signal is both blocked and has a disposition of "ignored", it is *not* added to the mask of pending signals when generated.

The set of signals that is pending for a thread is the union of the set of signals that is pending for that thread and the set of signals that is pending for the process as a whole; see **signal**(7).

A child created via **fork**(2) initially has an empty pending signal set; the pending signal set is preserved across an **execve**(2).

C library/kernel differences

The original Linux system call was named **sigpending**(). However, with the addition of real-time signals in Linux 2.2, the fixed-size, 32-bit *sigset_t* argument supported by that system call was no longer fit for purpose. Consequently, a new system call, **rt_sigpending**(), was added to support an enlarged *sigset_t* type. The new system call takes a second argument, *size_t sigsetsize*, which specifies the size in bytes of the signal set in *set*. The glibc**sigpending**() wrapper function hides these details from us, transparently calling **rt_sigpending**() when the kernel provides it.

BUGS

Up to and including glibc 2.2.1, there is a bug in the wrapper function for **sigpending**() which means that information about pending real-time signals is not correctly returned.

SEE ALSO

kill(2), **sigaction**(2), **signal**(2), **sigprocmask**(2), **sigsuspend**(2), **sigsetops**(3), **signal**(7)