

NAME

sigqueue - queue a signal to a process (REALTIME)

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <signal.h>
```

int

```
sigqueue(pid_t pid, int signo, const union sigval value);
```

DESCRIPTION

The **sigqueue()** system call causes the signal specified by *signo* to be sent with the value specified by *value* to the process specified by *pid*. If *signo* is zero (the null signal), error checking is performed but no signal is actually sent. The null signal can be used to check the validity of PID.

The conditions required for a process to have permission to queue a signal to another process are the same as for the **kill(2)** system call. The **sigqueue()** system call queues a signal to a single process specified by the *pid* argument.

The **sigqueue()** system call returns immediately. If the resources were available to queue the signal, the signal will be queued and sent to the receiving process.

If the value of *pid* causes *signo* to be generated for the sending process, and if *signo* is not blocked for the calling thread and if no other thread has *signo* unblocked or is waiting in a **sigwait()** system call for *signo*, either *signo* or at least the pending, unblocked signal will be delivered to the calling thread before **sigqueue()** returns. Should any multiple pending signals in the range SIGRTMIN to SIGRTMAX be selected for delivery, it is the lowest numbered one. The selection order between realtime and non-realtime signals, or between multiple pending non-realtime signals, is unspecified.

RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **sigqueue()** system call will fail if:

[EAGAIN] No resources are available to queue the signal. The process has already queued {SIGQUEUE_MAX} signals that are still pending at the receiver(s), or a system-

wide resource limit has been exceeded.

[EINVAL] The value of the *signo* argument is an invalid or unsupported signal number.

[EPERM] The process does not have the appropriate privilege to send the signal to the receiving process.

[ESRCH] The process *pid* does not exist.

SEE ALSO

kill(2), sigaction(2), sigpending(2), sigsuspend(2), sigtimedwait(2), sigwait(2), sigwaitinfo(2), pause(3), pthread_sigmask(3), siginfo(3)

STANDARDS

The **sigqueue()** system call conforms to IEEE Std 1003.1-2004 ("POSIX.1").

HISTORY

Support for POSIX realtime signal queue first appeared in FreeBSD 7.0.

CAVEATS

When using **sigqueue** to send signals to a process which might have a different ABI (for instance, one is 32-bit and the other 64-bit), the *sival_int* member of *value* can be delivered reliably, but the *sival_ptr* may be truncated in endian dependent ways and must not be relied on. Further, many pointer integrity schemes disallow sending pointers to other processes, and this technique should not be used in programs intended to be portable.