



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sigqueue.3p' command

\$ man sigqueue.3p

SIGQUEUE(3P) POSIX Programmer's Manual SIGQUEUE(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

sigqueue ? queue a signal to a process

SYNOPSIS

```
#include <signal.h>

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

DESCRIPTION

The sigqueue() function shall cause 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() function.

The sigqueue() function shall return immediately. If SA_SIGINFO is set for signo and if the resources were available to queue the signal, the signal shall be queued and sent to the receiving process. If SA_SIGINFO is not set for signo, then signo shall be sent at least once to the re?

ceiving process; it is unspecified whether value shall be sent to the receiving process as a result of this call.

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() function for signo, either signo or at least the pending, unblocked signal shall be delivered to the calling thread before the sigqueue() function returns. Should any multiple pending signals in the range SIGRTMIN to SIGRTMAX be selected for delivery, it shall be the lowest numbered one. The selection order between realtime and non-realtime signals, or between multiple pending non-realtime signals, is unspecified.

RETURN VALUE

Upon successful completion, the specified signal shall have been queued, and the sigqueue() function shall return a value of zero. Otherwise, the function shall return a value of -1 and set errno to indicate the error.

ERRORS

The sigqueue() function shall 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 appropriate privileges to send the signal to the receiving process.

ESRCH The process pid does not exist.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

The `sigqueue()` function allows an application to queue a realtime signal to itself or to another process, specifying the application-defined value. This is common practice in realtime applications on existing realtime systems. It was felt that specifying another function in the `sig...` name space already carved out for signals was preferable to extending the interface to `kill()`.

Such a function became necessary when the `put/get` event function of the message queues was removed. It should be noted that the `sigqueue()` function implies reduced performance in a security-conscious implementation as the access permissions between the sender and receiver have to be checked on each send when the `pid` is resolved into a target process. Such access checks were necessary only at message queue open in the previous interface.

The standard developers required that `sigqueue()` have the same semantics with respect to the null signal as `kill()`, and that the same permission checking be used. But because of the difficulty of implementing the "broadcast" semantic of `kill()` (for example, to process groups) and the interaction with resource allocation, this semantic was not adopted. The `sigqueue()` function queues a signal to a single process specified by the `pid` argument.

The `sigqueue()` function can fail if the system has insufficient resources to queue the signal. An explicit limit on the number of queued signals that a process could send was introduced. While the limit is "per-sender", this volume of POSIX.1-2017 does not specify that the resources be part of the state of the sender. This would require either that the sender be maintained after exit until all signals that it had sent to other processes were handled or that all such signals that had not yet been acted upon be removed from the queue(s) of the receivers. This volume of POSIX.1-2017 does not preclude this behavior, but an implementation that allocated queuing resources from a system-wide pool (with per-sender limits) and that leaves queued signals pending after the sender exits is also permitted.

None.

SEE ALSO

Section 2.8.1, Realtime Signals

The Base Definitions volume of POSIX.1?2017, <signal.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

SIGQUEUE(3P)