Full credit is given to the above companies including the OS that this PDF file was generated!

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

$ man waitid.3p

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

waitid ? wait for a child process to change state

SYNOPSIS

#include <sys/wait.h>

int waitid(idtype_t idtype, id_t id, siginfo_t *infop, int options);

DESCRIPTION

The waitid() function shall obtain status information (see Section 2.13, Status Information) pertaining to termination, stop, and/or con? tinue events in one of the caller's child processes.

The waitid() function shall cause the calling thread to become blocked until an error occurs or status information becomes available to the calling thread that satisfies all of the following properties (``match? ing status information''):

* The status information is from one of the child processes in the set of child processes specified by the idtype and id arguments.
* The state change in the status information matches one of the state change flags set in the options argument.

If matching status information is available prior to the call to waitid(), return shall be immediate. If matching status information is available for two or more child processes, the order in which their status is reported is unspecified.

As described in Section 2.13, Status Information, the waitid() function consumes the status information it obtains unless the WNOWAIT flag is set in the options argument.

The behavior when multiple threads are blocked in wait(), waitid(), or waitpid() is described in Section 2.13, Status Information.

The waitid() function shall record the obtained status information in the structure pointed to by infop. The fields of the structure pointed to by infop shall be filled in as described under ``Pointer to a Func? tion'' in Section 2.4.3, Signal Actions.

The idtype and id arguments are used to specify which children waitid() waits for.

If idtype is P_PID, waitid() shall wait for the child with a process ID equal to (pid_t)id.

If idtype is P_PGID, waitid() shall wait for any child with a process group ID equal to (pid_t)id.

If idtype is P_ALL, waitid() shall wait for any children and id is ig? nored.

The options argument is used to specify which state changes waitid() shall wait for. It is formed by OR'ing together the following flags:

WCONTINUED Status shall be returned for any continued child process whose status either has not been reported since it contin? ued from a job control stop or has been reported only by calls to waitid() with the WNOWAIT flag set.

WEXITED Wait for processes that have exited.

WNOHANG Do not hang if no status is available; return immediately.

WNOWAIT Keep the process whose status is returned in infop in a waitable state. This shall not affect the state of the process; the process may be waited for again after this call completes.

WSTOPPED Status shall be returned for any child that has stopped upon receipt of a signal, and whose status either has not been reported since it stopped or has been reported only by calls to waitid() with the WNOWAIT flag set.

Applications shall specify at least one of the flags WEXITED, WSTOPPED, or WCONTINUED to be OR'ed in with the options argument.

The application shall ensure that the infop argument points to a sig? info_t structure. If waitid() returns because a child process was found that satisfied the conditions indicated by the arguments idtype and op? tions, then the structure pointed to by infop shall be filled in by the system with the status of the process; the si_signo member shall be set equal to SIGCHLD. If waitid() returns because WNOHANG was specified and status is not available for any process specified by idtype and id, then the si_signo and si_pid members of the structure pointed to by in? fop shall be set to zero and the values of other members of the struc? ture are unspecified.

RETURN VALUE

If WNOHANG was specified and status is not available for any process specified by idtype and id, 0 shall be returned. If waitid() returns due to the change of state of one of its children, 0 shall be returned. Otherwise, -1 shall be returned and errno set to indicate the error.

ERRORS

The waitid() function shall fail if:

ECHILD The calling process has no existing unwaited-for child pro? cesses.

EINTR The waitid() function was interrupted by a signal.

EINVAL An invalid value was specified for options, or idtype and id specify an invalid set of processes.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Calls to waitid() with idtype equal to P_ALL will collect information

about any child process. This may result in interactions with other in? terfaces that may be waiting for their own children (such as by use of system()). For this reason it is recommended that portable applica? tions not use waitid() with idtype of P_ALL. See also APPLICATION USAGE for wait().

As specified in Consequences of Process Termination, if the calling process has SA_NOCLDWAIT set or has SIGCHLD set to SIG_IGN, then the termination of a child process will not cause status information to be? come available to a thread blocked in wait(), waitid(), or waitpid().

Thus, a thread blocked in one of the wait functions will remain blocked unless some other condition causes the thread to resume execution (such as an [ECHILD] failure due to no remaining children in the set of waited-for children).

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.4.3, Signal Actions, Section 2.13, Status Information, exec, exit(), wait()

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

files to man page format. To report such errors, see https://www.ker?

nel.org/doc/man-pages/reporting_bugs.html .