NAME

siginfo – signal generation information

SYNOPSIS

#include <siginfo.h>

DESCRIPTION

If a process is catching a signal, it may request information that tells why the system generated that signal (see **sigaction**(2)). If a process is monitoring its children, it may receive information that tells why a child changed state (see **waitid**(2)). In either case, the system returns the information in a structure of type **siginfo_t**, which includes the following information:

```
int            si_signo   /* signal number */
int            si_errno   /* error number */
int            si_code    /* signal code */
union sigval   si_value   /* signal value */
```

si_signo contains the system-generated signal number. For the **waitid**(2) function, **si_signo** is always **SIGCHLD**.

If **si_errno** is non-zero, it contains an error number associated with this signal, as defined in **<errno.h>**.

si_code contains a code identifying the cause of the signal.

If the value of the **si_code** member is **SI_NOINFO**, only the **si_signo** member of **siginfo_t** is meaningful, and the value of all other members is unspecified.

User Signals

If the value of **si_code** is less than or equal to 0, then the signal was generated by a user process (see **kill**(2), **_lwp_kill**(2), **sigqueue**(3R), **sigsend**(2), **abort**(3C), and **raise**(3C)) and the **siginfo** structure contains the following additional information:

```
typedef long pid_t   si_pid   /* sending process ID */
typedef long uid_t   si_uid   /* sending user ID */
```

If the signal was generated by a user process, the following values are defined for **si_code**:

SI_USER the implementation sets **si_code** to **SI_USER** if the signal was sent by **kill**(2), **sigsend**(2), **raise**(3C) or **abort**(3C).

SI_LWP the signal was sent by **_lwp_kill**(2).

SI_QUEUE the signal was sent by **sigqueue**(3R).

SI_TIMER the signal was generated by the expiration of a timer created by **timer_settime**(3R).

SI_ASYNCIO the signal was generated by the completion of an asynchronous I/O request.

SI_MESGQ the signal was generated by the arrival of a message on an empty message queue. (see **mq_notify**(3R)).

si_value contains the application specified value, which is passed to the application's signal-catching function at the time of the signal delivery, if **si_code** is any of **SI_QUEUE**, **SI_TIMER**, **SI_ASYNCHIO**, or **SI_MESGQ**.

System Signals

Otherwise, **si_code** contains a positive value reflecting the reason why the system generated the signal:

Signal	Code	Reason
SIGILL	**ILL_ILLOPC**	illegal opcode
	ILL_ILLOPN	illegal operand
	ILL_ILLADR	illegal addressing mode
	ILL_ILLTRP	illegal trap
	ILL_PRVOPC	privileged opcode
	ILL_PRVREG	privileged register
	ILL_COPROC	co-processor error
	ILL_BADSTK	internal stack error
SIGFPE	**FPE_INTDIV**	integer divide by zero
	FPE_INTOVF	integer overflow
	FPE_FLTDIV	floating point divide by zero
	FPE_FLTOVF	floating point overflow
	FPE_FLTUND	floating point underflow
	FPE_FLTRES	floating point inexact result
	FPE_FLTINV	invalid floating point operation
	FPE_FLTSUB	subscript out of range
SIGSEGV	**SEGV_MAPERR**	address not mapped to object
	SEGV_ACCERR	invalid permissions for mapped object
SIGBUS	**BUS_ADRALN**	invalid address alignment
	BUS_ADRERR	non-existent physical address
	BUS_OBJERR	object specific hardware error
SIGTRAP	**TRAP_BRKPT**	process breakpoint
	TRAP_TRACE	process trace trap
SIGCHLD	**CLD_EXITED**	child has exited
	CLD_KILLED	child was killed
	CLD_DUMPED	child terminated abnormally
	CLD_TRAPPED	traced child has trapped
	CLD_STOPPED	child has stopped
	CLD_CONTINUED	stopped child had continued
SIGPOLL	**POLL_IN**	data input available
	POLL_OUT	output buffers available
	POLL_MSG	input message available
	POLL_ERR	I/O error
	POLL_PRI	high priority input available
	POLL_HUP	device disconnected

In addition, the following signal-dependent information is available for kernel-generated signals:

Signal	Field	Value
SIGILL **SIGFPE**	**caddr_t si_addr**	address of faulting instruction
SIGSEGV **SIGBUS**	**caddr_t si_addr**	address of faulting memory reference
SIGCHLD	**pid_t si_pid** **int si_status**	child process ID exit value or signal
SIGPOLL	**long si_band**	band event for **POLL_IN**, **POLL_OUT**, or **POLL_MSG**

SEE ALSO

_lwp_kill(2), **kill**(2), **sigaction**(2), **sigsend**(2), **waitid**(2), **abort**(3C), **raise**(3C), **aio_read**(3R), **mq_notify**(3R), **sigqueue**(3R), **timer_create**(3R), **timer_settime**(3R), **signal**(5)

NOTES

For **SIGCHLD** signals, if **si_code** is equal to **CLD_EXITED**, then **si_status** is equal to the exit value of the process; otherwise, it is equal to the signal that caused the process to change state. For some implementations, the exact value of **si_addr** may not be available; in that case, **si_addr** is guaranteed to be on the same page as the faulting instruction or memory reference.