



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 'sigsetjmp.3p' command

\$ man sigsetjmp.3p

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

sigsetjmp ? set jump point for a non-local goto

SYNOPSIS

```
#include <setjmp.h>

int sigsetjmp(sigjmp_buf env, int savemask);
```

DESCRIPTION

The sigsetjmp() function shall be equivalent to the setjmp() function, except as follows:

- * References to setjmp() are equivalent to sigsetjmp().
- * References to longjmp() are equivalent to siglongjmp().
- * If the value of the savemask argument is not 0, sigsetjmp() shall also save the current signal mask of the calling thread as part of the calling environment.

RETURN VALUE

If the return is from a successful direct invocation, sigsetjmp() shall return 0. If the return is from a call to siglongjmp(), sigsetjmp() shall return a non-zero value.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The distinction between `setjmp()/longjmp()` and `sigsetjmp()/siglongjmp()` is only significant for programs which use `sigaction()`, `sigprocmask()`, or `sigsuspend()`.

Note that since this function is defined in terms of `setjmp()`, if `save?` mask is zero, it is unspecified whether the signal mask is saved.

RATIONALE

The ISO C standard specifies various restrictions on the usage of the `setjmp()` macro in order to permit implementors to recognize the name in the compiler and not implement an actual function. These same restrictions apply to the `sigsetjmp()` macro.

There are processors that cannot easily support these calls, but this was not considered a sufficient reason to exclude them.

4.2 BSD, 4.3 BSD, and XSI-conformant systems provide functions named `_setjmp()` and `_longjmp()` that, together with `setjmp()` and `longjmp()`, provide the same functionality as `sigsetjmp()` and `siglongjmp()`. On those systems, `setjmp()` and `longjmp()` save and restore signal masks, while `_setjmp()` and `_longjmp()` do not. On System V Release 3 and in corresponding issues of the SVID, `setjmp()` and `longjmp()` are explicitly defined not to save and restore signal masks. In order to permit existing practice in both cases, the relation of `setjmp()` and `longjmp()` to signal masks is not specified, and a new set of functions is defined instead.

The `longjmp()` and `siglongjmp()` functions operate as in the previous issue provided the matching `setjmp()` or `sigsetjmp()` has been performed in the same thread. Non-local jumps into contexts saved by other threads would be at best a questionable practice and were not considered worthy of standardization.

FUTURE DIRECTIONS

None.

SEE ALSO

`pthread_sigmask()`, `siglongjmp()`, `signal()`, `sigsuspend()`

The Base Definitions volume of POSIX.1-2017, `<setjmp.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

SIGSETJMP(3P)