

**NAME**

abort - cause abnormal process termination

**SYNOPSIS**

```
#include <stdlib.h>
```

```
void abort(void);
```

**DESCRIPTION**

The **abort()** first unblocks the **SIGABRT** signal, and then raises that signal for the calling process. This results in the abnormal termination of the process unless the **SIGABRT** signal is caught and the signal handler does not return (see [longjmp\(3\)](#)).

If the **abort()** function causes process termination, all open streams are closed and flushed.

If the **SIGABRT** signal is ignored, or caught by a handler that returns, the **abort()** function will still terminate the process. It does this by restoring the default disposition for **SIGABRT** and then raising the signal for a second time.

**RETURN VALUE**

The **abort()** function never returns.

**ATTRIBUTES**

**Multithreading (see [pthread\(7\)](#))**

The **abort()** function is thread-safe.

**CONFORMING TO**

SVr4, POSIX.1-2001, 4.3BSD, C89, C99.

**SEE ALSO**

[gdb\(1\)](#), [sigaction\(2\)](#), [exit\(3\)](#), [longjmp\(3\)](#), [raise\(3\)](#)

**COLOPHON**

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.