

NAME

gsignal, ssignal - software signal facility

SYNOPSIS

```
#include <signal.h>
```

```
typedef void (*sig_handler_t)(int);
```

```
int gsignal(int signum);
```

```
sig_handler_t ssignal(int signum, sig_handler_t action);
```

Feature Test Macro Requirements for glibc (see [feature_test_macros\(7\)](#)):

```
gsignal(), ssignal(): _SVID_SOURCE
```

DESCRIPTION

Don't use these functions under Linux. Due to a historical mistake, under Linux these functions are aliases for [raise\(3\)](#) and [signal\(2\)](#), respectively.

Elsewhere, on System V-like systems, these functions implement software signaling, entirely independent of the classical [signal\(2\)](#) and [kill\(2\)](#) functions. The function **ssignal()** defines the action to take when the software signal with number *signum* is raised using the function **gsignal()**, and returns the previous such action or **SIG_DFL**. The function **gsignal()** does the following: if no action (or the action **SIG_DFL**) was specified for *signum*, then it does nothing and returns 0. If the action **SIG_IGN** was specified for *signum*, then it does nothing and returns 1. Otherwise, it resets the action to **SIG_DFL** and calls the action function with argument *signum*, and returns the value returned by that function. The range of possible values *signum* varies (often 1-15 or 1-17).

CONFORMING TO

These functions are available under AIX, DG/UX, HP-UX, SCO, Solaris, Tru64. They are called obsolete under most of these systems, and are broken under Linux libc and glibc. Some systems also have **gsignal_r()** and **ssignal_r()**.

SEE ALSO

[kill\(2\)](#), [signal\(2\)](#), [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/>.