



## ***Linux Ubuntu 26.04 Manual Pages on command 'sgetmask.2'***

### ***\$ man sgetmask.2***

sgetmask(2)                      System Calls Manual                      sgetmask(2)

#### NAME

sgetmask, ssetmask - manipulation of signal mask (obsolete)

#### LIBRARY

Standard C library (libc, -lc)

#### SYNOPSIS

```
#include <sys/syscall.h>     /* Definition of SYS_* constants */  
  
#include <unistd.h>  
  
[[deprecated]] long syscall(SYS_sgetmask, void);  
  
[[deprecated]] long syscall(SYS_ssetmask, long newmask);
```

#### DESCRIPTION

These system calls are obsolete. Do not use them; use sigprocmask(2) instead.

sgetmask() returns the signal mask of the calling process.

ssetmask() sets the signal mask of the calling process to the value given in newmask. The previous signal mask is returned.

The signal masks dealt with by these two system calls are plain bit masks (unlike the sigset\_t used by sigprocmask(2)); use sigmask(3) to create and inspect these masks.

#### RETURN VALUE

sgetmask() always successfully returns the signal mask. ssetmask() always succeeds, and re? turns the previous signal mask.

#### ERRORS

These system calls always succeed.

#### STANDARDS

Linux.

## HISTORY

Since Linux 3.16, support for these system calls is optional, depending on whether the kernel was built with the `CONFIG_SGETMASK_SYSCALL` option.

## NOTES

These system calls are unaware of signal numbers greater than 31 (i.e., real-time signals).

These system calls do not exist on x86-64.

It is not possible to block `SIGSTOP` or `SIGKILL`.

## SEE ALSO

`sigprocmask(2)`, `signal(7)`

Linux man-pages 6.17

2026-02-08

`sgetmask(2)`