alarm - Man Page

schedule an alarm signal

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.

Synopsis

```
#include <unistd.h>

unsigned alarm(unsigned seconds);
```

Description

The *alarm*() function shall cause the system to generate a SIGALRM signal for the process after the number of realtime seconds specified by *seconds* have elapsed. Processor scheduling delays may prevent the process from handling the signal as soon as it is generated.

If *seconds* is 0, a pending alarm request, if any, is canceled.

Alarm requests are not stacked; only one SIGALRM generation can be scheduled in this manner. If the SIGALRM signal has not yet been generated, the call shall result in rescheduling the time at which the SIGALRM signal is generated.

Interactions between *alarm*() and *setitimer*() are unspecified.

Return Value

If there is a previous *alarm*() request with time remaining, *alarm*() shall return a non-zero value that is the number of seconds until the previous request would have generated a SIGALRM signal. Otherwise, *alarm*() shall return 0.

Errors

The *alarm*() function is always successful, and no return value is reserved to indicate an error.

The following sections are informative.

Examples

alarm - Man Page

Application Usage

The *fork*() function clears pending alarms in the child process. A new process image created by one of the *exec* functions inherits the time left to an alarm signal in the image of the old process.

Application developers should note that the type of the argument *seconds* and the return value of *alarm*() is **unsigned**. That means that a Strictly Conforming POSIX System Interfaces Application cannot pass a value greater than the minimum guaranteed value for {UINT_MAX}, which the ISO C standard sets as 65535, and any application passing a larger value is restricting its portability. A different type was considered, but historical implementations, including those with a 16-bit type, consistently use either **unsigned** or **int**.

Application developers should be aware of possible interactions when the same process uses both the *alarm*() and *sleep*() functions.

Rationale

Many historical implementations (including Version 7 and System V) allow an alarm to occur up to a second early. Other implementations allow alarms up to half a second or one clock tick early or do not allow them to occur early at all. The latter is considered most appropriate, since it gives the most predictable behavior, especially since the signal can always be delayed for an indefinite amount of time due to scheduling. Applications can thus choose the *seconds* argument as the minimum amount of time they wish to have elapse before the signal.

The term “realtime” here and elsewhere (*sleep*(), *times*()) is intended to mean “wall clock” time as common English usage, and has nothing to do with “realtime operating systems”. It is in contrast to *virtual time*, which could be misinterpreted if just *time* were used.

In some implementations, including 4.3 BSD, very large values of the *seconds* argument are silently rounded down to an implementation-specific maximum value. This maximum is large enough (to the order of several months) that the effect is not noticeable.

There were two possible choices for alarm generation in multi-threaded applications: generation for the calling thread or generation for the process. The first option would not have been particularly useful since the alarm state is maintained on a per-process basis and the alarm that is established by the last invocation of *alarm*() is the only one that would be active.

Furthermore, allowing generation of an asynchronous signal for a thread would have introduced an exception to the overall signal model. This requires a compelling reason in order to be justified.

alarm - Man Page

None.

See Also

alarm(), exec, fork(), getitimer(), pause(), sigaction(), sleep()

The Base Definitions volume of POSIX.1-2017, <signal.h>, <unistd.h>

Copyright

Referenced By

exec(3p), fcntl(3p), fork(3p), getitimer(3p), lockf(3p), posix_spawn(3p), pselect(3p), signal.h(0p), sleep(1p), sleep(3p), times(3p), unistd.h(0p).

2017 IEEE/The Open Group POSIX Programmer's Manual

alarm - Man Page