

**NAME**

alarm – set an alarm clock for delivery of a signal

**LIBRARY**

Standard C library (*libc*, *-lc*)

**SYNOPSIS**

**#include <unistd.h>**

**unsigned int alarm(unsigned int *seconds*);**

**DESCRIPTION**

**alarm()** arranges for a **SIGALRM** signal to be delivered to the calling process in *seconds* seconds.

If *seconds* is zero, any pending alarm is canceled.

In any event any previously set **alarm()** is canceled.

**RETURN VALUE**

**alarm()** returns the number of seconds remaining until any previously scheduled alarm was due to be delivered, or zero if there was no previously scheduled alarm.

**STANDARDS**

POSIX.1-2001, POSIX.1-2008, SVr4, 4.3BSD.

**NOTES**

**alarm()** and **setitimer(2)** share the same timer; calls to one will interfere with use of the other.

Alarms created by **alarm()** are preserved across **execve(2)** and are not inherited by children created via **fork(2)**.

**sleep(3)** may be implemented using **SIGALRM**; mixing calls to **alarm()** and **sleep(3)** is a bad idea.

Scheduling delays can, as ever, cause the execution of the process to be delayed by an arbitrary amount of time.

**SEE ALSO**

**gettimeofday(2)**, **pause(2)**, **select(2)**, **setitimer(2)**, **sigaction(2)**, **signal(2)**, **timer\_create(2)**, **timerfd\_create(2)**, **sleep(3)**, **time(7)**