

**NAME**

clock - determine processor time

**SYNOPSIS**

```
#include <time.h>
```

```
clock_t clock(void);
```

**DESCRIPTION**

The **clock()** function returns an approximation of processor time used by the program.

**RETURN VALUE**

The value returned is the CPU time used so far as a *clock\_t*; to get the number of seconds used, divide by **CLOCKS\_PER\_SEC**. If the processor time used is not available or its value cannot be represented, the function returns the value *(clock\_t) -1*.

**CONFORMING TO**

C89, C99, POSIX.1-2001. POSIX requires that **CLOCKS\_PER\_SEC** equals 1000000 independent of the actual resolution.

**NOTES**

The C standard allows for arbitrary values at the start of the program; subtract the value returned from a call to **clock()** at the start of the program to get maximum portability.

Note that the time can wrap around. On a 32-bit system where **CLOCKS\_PER\_SEC** equals 1000000 this function will return the same value approximately every 72 minutes.

On several other implementations, the value returned by **clock()** also includes the times of any children whose status has been collected via [wait\(2\)](#) (or another wait-type call). Linux does not include the times of waited-for children in the value returned by **clock()**. The [times\(2\)](#) function, which explicitly returns (separate) information about the caller and its children, may be preferable.

In glibc 2.17 and earlier, **clock()** was implemented on top of [times\(2\)](#). For improved accuracy, since glibc 2.18, it is implemented on top of [clock\\_gettime\(2\)](#) (using the **CLOCK\_PROCESS\_CPUTIME\_ID** clock).

**SEE ALSO**

[clock\\_gettime\(2\)](#), [getrusage\(2\)](#), [times\(2\)](#)

**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/>.