NAME

clocks - various system timers

SYNOPSIS

#include <time.h>

DESCRIPTION

HZ is not part of the application interface in BSD.

There are many different real and virtual (timekeeping) clocks with different frequencies:

- The scheduling clock. This is a real clock with frequency that happens to be 100. It is not available to applications.

- The statistics clock. This is a real clock with frequency that happens to be 128. It is not directly available to applications.

- The clock reported by clock(3). This is a virtual clock with a frequency that happens to be 128. Its actual frequency is given by the macro CLOCKS_PER_SEC. Note that CLOCKS_PER_SEC may be floating point. Do not use clock(3) in new programs under FreeBSD. It is feeble compared with getrusage(2). It is provided for ANSI conformance. It is implemented by calling getrusage(2) and throwing away information and resolution.

- The clock reported by times(3). This is a virtual clock with a frequency that happens to be 128. Its actual frequency is given by the macro CLK_TCK (deprecated; do not use) and by **sysconf**(*_SC_CLK_TCK*) and by sysctl(3). Note that its frequency may be different from CLOCKS_PER_SEC. Do not use times(3) in new programs under FreeBSD. It is feeble compared with gettimeofday(2) together with getrusage(2). It is provided for POSIX conformance. It is implemented by calling gettimeofday(2) and getrusage(2) and throwing away information and resolution.

- The profiling clock. This is a real clock with frequency 1024. It is used mainly by moncontrol(3) and gprof(1). Applications should determine its actual frequency using sysctl(3) or by reading it from the header in the profiling data file.

- The mc146818a clock. This is a real clock with a nominal frequency of 32768. It is divided down to give the statistic clock and the profiling clock. It is not available to applications.

- The microseconds clock. This is a virtual clock with frequency 1000000. It is used for most timekeeping in BSD and is exported to applications in getrusage(2), gettimeofday(2), select(2), getitimer(2), etc. This is the clock that should normally be used by BSD applications.

- The i8254 clock. This is a real clock/timer with a nominal frequency of 1193182. It has three independent time counters to be used. It is divided down to give the scheduling clock. It is not available to applications.

- The TSC clock (64-bit register) on fifth-generation or later x86 systems. This is a real clock with a frequency that is equivalent to the number of cycles per second of the CPU(s). Its frequency can be found using the *machdep.tsc_freq* sysctl, if it is available. It is used to interpolate between values of the scheduling clock. It can be accessed using the PMIOTSTAMP request of perfmon(4).

- The ACPI clock. This is a real clock/timer with a nominal frequency of 3579545. It is accessed via a 24 or 32 bit register. Unlike the TSC clock, it maintains a constant tick rate even when the CPU sleeps or its clock rate changes. It is not available to applications.

Summary: if HZ is not 1000000 then the application is probably using the wrong clock.

SEE ALSO

gprof(1), clock_gettime(2), getitimer(2), getrusage(2), gettimeofday(2), select(2), clock(3), moncontrol(3), times(3)

AUTHORS

This manual page was written by Jörg Wunsch after a description posted by Bruce Evans.