NAME

hz, **tick**, **stathz**, **profhz** - system time model

SYNOPSIS

#include <sys/kernel.h>

extern int hz;
extern int tick;
extern int stathz;
extern int profhz; / deprecated */*

DESCRIPTION

FreeBSD utilizes periodic, one-shot, global or per-CPU timing hardware using eventtimers(9) to produce traditional clock behavior. These clocks regulate periodic events in the system.

The main clock is used to update the system's notion of time via timecounters(9) and to pace periodic system processing as documented in hardclock(9). That routine may be called once every 1 / *hz* seconds, though the call is omitted if no work is needed in the next tick and it has not been 0.5 seconds since the last call.

The stat clock running at *stathz* gathers statistics on the system and its processes. It computes values for getrusage(2) and statistics displayed by ps(1) and top(1).

Finally, a profiling clock may run at *profhz* to sample user program counter values for profiling purposes. This profiling mechanism has been replaced by the more functional hwpmc(4) and may be removed in a future version of FreeBSD.

tick is the length of time in microseconds of one system tick.

These system variables are also available as *struct clockinfo* from sysctl(3) and **kern.clockrate** from sysctl(8).

The current global and per-CPU CPU time usage is returned to the user in units of 1 / *stathz* ticks in the **kern.cp_time** and **kern.cp_times** sysctl MIBs.

The *hz* rate may be overridden by defining HZ in the kernel configuration file or setting **kern.hz** system tuneable via loader.conf(5).

The current default is 1000 Hz for a tick of 1 ms for real hardware. For virtual machine guests, the default is 100 Hz for a tick of 10 ms. Only override the default value if you really know what you are

doing. Due to the adaptive nature of timeouts, changing this value has less effect than it had in the past.

SEE ALSO

ps(1), top(1), setitimer(2), timer_settime(2), loader.conf(5), eventtimers(9), hardclock(9), microtime(9), time_second(9), timecounters(9)

IMPLEMENTATION NOTES

Historically, both the *stathz* and *profhz* clocks have run off the same physical timer running at the slower rate when no process is using the profile features, or at the higher rate when at least one process is using it. Although the interface is deprecated by IEEE Std 1003.1-2008 ("POSIX.1") in favor of timer_settime(2), several programs still use setitimer(2) and *ITIMER_PROF* for a higher-resolution periodic interrupt than has been traditionally available.

Historically, hardclock(9) has also been run off a separate interrupt, except on constrained platforms that lack enough periodic interrupt sources. FreeBSD uses eventtimers(9) to abstract these details away, though some old code may still harbor assumptions that are an imperfect fit to this abstraction.

timecounters(9) are limited to 32-bits and wrap after about a second, so we must update the time hands they maintain at least every half second to get the proper wrapping math. In addition, *kern.cp_times* needs to updated at least once a second so that the values displayed by top(1) update every second.