NAME
timecounters - kernel time counters subsystem

SYNOPSIS
The kernel uses several types of time-related devices, such as: real time clocks, time counters and event timers. Real time clocks are responsible for tracking real world time, mostly when the system is down. Time counters are responsible for tracking purposes, when the system is running. Event timers are responsible for generating interrupts at a specified time or periodically, to run different time-based events. This page is about the second.

DESCRIPTION
Time counters are the lowest level of time tracking in the kernel. They provide monotonically increasing timestamps with known width and update frequency. They can overflow, drift, etc and so in raw form can be used only in very limited performance-critical places like the process scheduler.

More usable time is created by scaling the values read from the selected time counter and combining it with some offset, regularly updated by **tc_windup**() on **hardclock**() invocation.

Different platforms provide different kinds of timer hardware. The goal of the time counters subsystem is to provide a unified way to access that hardware.

Each driver implementing time counters registers them with the subsystem. It is possible to see the list of present time counters, via the *kern.timecounter* sysctl(8) variable:

kern.timecounter.choice: TSC-low(-100) HPET(950) i8254(0) ACPI-fast(900) dummy(-1000000)
kern.timecounter.tc.ACPI-fast.mask: 16777215
kern.timecounter.tc.ACPI-fast.counter: 13467909
kern.timecounter.tc.ACPI-fast.frequency: 3579545
kern.timecounter.tc.ACPI-fast.quality: 900
kern.timecounter.tc.i8254.mask: 65535
kern.timecounter.tc.i8254.counter: 62692
kern.timecounter.tc.i8254.frequency: 1193182
kern.timecounter.tc.i8254.quality: 0
kern.timecounter.tc.HPET.mask: 4294967295
kern.timecounter.tc.HPET.counter: 3013495652
kern.timecounter.tc.HPET.frequency: 14318180
kern.timecounter.tc.HPET.quality: 950
kern.timecounter.tc.TSC-low.mask: 4294967295
kern.timecounter.tc.TSC-low.counter: 4067509463
kern.timecounter.tc.TSC-low.frequency: 11458556

kern.timecounter.tc.TSC-low.quality: -100

The output nodes are defined as follows:

kern.timecounter.tc.X.mask is a bitmask, defining valid counter bits,

kern.timecounter.tc.X.counter is a present counter value,

kern.timecounter.tc.X.frequency is a counter update frequency,

kern.timecounter.tc.X.quality is an integral value, defining the quality of this time counter compared to others. A negative value means this time counter is broken and should not be used.

The time management code of the kernel automatically switches to a higher-quality time counter when it registers, unless the *kern.timecounter.hardware* sysctl has been used to choose a specific device.

There is no way to unregister a time counter once it has registered with the kernel. If a dynamically loaded module contains a time counter you will not be able to unload that module, even if the time counter it contains is not the one currently in use.

SEE ALSO
attimer(4), eventtimers(4), ffclock(4), hpet(4)