

**NAME**

**ffclock\_getcounter**, **ffclock\_getestimate**, **ffclock\_setestimate** - Retrieve feed-forward counter, get and set feed-forward clock estimates

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <sys/timeffc.h>
```

*int*

```
ffclock_getcounter(ffcounter *ffcount);
```

*int*

```
ffclock_getestimate(struct ffclock_estimate *cest);
```

*int*

```
ffclock_setestimate(struct ffclock_estimate *cest);
```

**DESCRIPTION**

The **ffclock** is an alternative method to synchronise the system clock. The **ffclock** implements a feed-forward paradigm and decouples the timestamping and timekeeping kernel functions. This ensures that past clock errors do not affect current timekeeping, an approach radically different from the feedback alternative implemented by the **ntpd** daemon when adjusting the system clock. The feed-forward approach has demonstrated better performance and higher robustness than a feedback approach when synchronising over the network.

In the feed-forward context, a *timestamp* is a cumulative value of the ticks of the timecounter, which can be converted into seconds by using the feed-forward *clock estimates*.

The **ffclock\_getcounter**() system call allows the calling process to retrieve the current value of the feed-forward counter maintained by the kernel.

The **ffclock\_getestimate**() and **ffclock\_setestimate**() system calls allow the caller to get and set the kernel's feed-forward clock parameter estimates respectively. The **ffclock\_setestimate**() system call should be invoked by a single instance of a feed-forward synchronisation daemon. The **ffclock\_getestimate**() system call can be called by any process to retrieve the feed-forward clock estimates.

The feed-forward approach does not require that the clock estimates be retrieved every time a timestamp

is to be converted into seconds. The number of system calls can therefore be greatly reduced if the calling process retrieves the clock estimates from the clock synchronisation daemon instead. The **ffclock\_getestimate()** must be used when the feed-forward synchronisation daemon is not running (see *USAGE* below).

The clock parameter estimates structure pointed to by *cest* is defined in *<sys/timeeffc.h>* as:

```
struct ffclock_estimate {
    struct bintime update_time; /* Time of last estimates update. */
    ffcounter    update_ffcount; /* Counter value at last update. */
    ffcounter    leapsec_next; /* Counter value of next leap second. */
    uint64_t     period; /* Estimate of counter period. */
    uint32_t     errb_abs; /* Bound on absolute clock error [ns]. */
    uint32_t     errb_rate; /* Bound on counter rate error [ps/s]. */
    uint32_t     status; /* Clock status. */
    int16_t     leapsec_total; /* All leap seconds seen so far. */
    int8_t      leapsec; /* Next leap second (in {-1,0,1}). */
};
```

Only the super-user may set the feed-forward clock estimates.

## RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

The following error codes may be set in *errno*:

[EFAULT]	The <i>ffcount</i> or <i>cest</i> pointer referenced invalid memory.
[EPERM]	A user other than the super-user attempted to set the feed-forward clock parameter estimates.

## USAGE

The feed-forward paradigm enables the definition of specialised clock functions.

In its simplest form, **ffclock\_getcounter()** can be used to establish strict order between events or to measure small time intervals very accurately with a minimum performance cost.

Different methods exist to access absolute time (or "wall-clock time") tracked by the ffclock. The

simplest method uses the `ffclock` sysctl interface `kern.ffclock` to make the system clock return the `ffclock` time. The `clock_gettime(2)` system call can then be used to retrieve the current time seen by the feed-forward clock. Note that this setting affects the entire system and that a feed-forward synchronisation daemon should be running.

A less automated method consists of retrieving the feed-forward counter timestamp from the kernel and using the feed-forward clock parameter estimates to convert the timestamp into seconds. The feed-forward clock parameter estimates can be retrieved from the kernel or from the synchronisation daemon directly (preferred). This method allows converting timestamps using different clock models as needed by the application, while collecting meaningful upper bounds on current clock error.

### SEE ALSO

`date(1)`, `adjtime(2)`, `clock_gettime(2)`, `ctime(3)`

### HISTORY

Feed-forward clock support first appeared in FreeBSD 10.0.

### AUTHORS

The feed-forward clock support was written by Julien Ridoux <[jridoux@unimelb.edu.au](mailto:jridoux@unimelb.edu.au)> in collaboration with Darryl Veitch <[dveitch@unimelb.edu.au](mailto:dveitch@unimelb.edu.au)> at the University of Melbourne under sponsorship from the FreeBSD Foundation.