Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'getitimer.3p' command

$ man getitimer.3p

GETITIMER(3P) POSIX Programmer's Manual GETITIMER(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

getitimer, setitimer ? get and set value of interval timer

SYNOPSIS

```
#include <sys/time.h>
int getitimer(int which, struct itimerval *value);
int setitimer(int which, const struct itimerval *restrict value,
    struct itimerval *restrict ovalue);
```

DESCRIPTION

The getitimer() function shall store the current value of the timer specified by which into the structure pointed to by value. The setitimer() function shall set the timer specified by which to the value specified in the structure pointed to by value, and if ovalue is not a null pointer, store the previous value of the timer in the struc? ture pointed to by ovalue.

A timer value is defined by the itimerval structure, specified in <sys/time.h>. If it_value is non-zero, it shall indicate the time to the next timer expiration. If it_interval is non-zero, it shall spec?

ify a value to be used in reloading it_value when the timer expires. Setting it_value to 0 shall disable a timer, regardless of the value of it_interval. Setting it_interval to 0 shall disable a timer after its next expiration (assuming it_value is non-zero).

Implementations may place limitations on the granularity of timer val? ues. For each interval timer, if the requested timer value requires a finer granularity than the implementation supports, the actual timer value shall be rounded up to the next supported value.

An XSI-conforming implementation provides each process with at least three interval timers, which are indicated by the which argument:

ITIMER_PROF Decrements both in process virtual time and when the sys? tem is running on behalf of the process. It is designed to be used by interpreters in statistically profiling the execution of interpreted programs. Each time the ITIMER_PROF timer expires, the SIGPROF signal is deliv? ered.

ITIMER_REAL Decrements in real time. A SIGALRM signal is delivered when this timer expires.

ITIMER_VIRTUAL
Decrements in process virtual time. It runs only when the process is executing. A SIGVTALRM signal is delivered when it expires.

The interaction between setitimer() and alarm() or sleep() is unspeci? fied.

RETURN VALUE

Upon successful completion, getitimer() or setitimer() shall return 0; otherwise, -1 shall be returned and errno set to indicate the error.

ERRORS

The setitimer() function shall fail if:

EINVAL The value argument is not in canonical form. (In canonical form, the number of microseconds is a non-negative integer less than 1000000 and the number of seconds is a non-negative integer.)

The getitimer() and setitimer() functions may fail if:

EINVAL The which argument is not recognized.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Applications should use the timer_gettime() and timer_settime() func?
tions instead of the obsolescent getitimer() and setitimer() functions,
respectively.

RATIONALE

None.

FUTURE DIRECTIONS

The getitimer() and setitimer() functions may be removed in a future
version.

SEE ALSO

alarm(), exec, sleep(), timer_getoverrun()

The Base Definitions volume of POSIX.1?2017, <signal.h>, <sys_time.h>

COPYRIGHT