


```

thread_start(void *arg)
{
    printf("Subthread starting infinite loop\n");
    for (;;)
        continue;
}

static void
pclock(char *msg, clockid_t cid)
{
    struct timespec ts;
    printf("%s", msg);
    if (clock_gettime(cid, &ts) == -1)
        err(EXIT_FAILURE, "clock_gettime");
    printf("%4jd.%03ld\n", (intmax_t) ts.tv_sec, ts.tv_nsec / 1000000);
}

int
main(void)
{
    pthread_t thread;
    clockid_t cid;
    int s;
    s = pthread_create(&thread, NULL, thread_start, NULL);
    if (s != 0)
        errc(EXIT_FAILURE, s, "pthread_create");
    printf("Main thread sleeping\n");
    sleep(1);
    printf("Main thread consuming some CPU time...\n");
    for (unsigned int j = 0; j < 2000000; j++)
        getppid();
    pclock("Process total CPU time: ", CLOCK_PROCESS_CPUTIME_ID);
    s = pthread_getcpuclockid(pthread_self(), &cid);
    if (s != 0)
        errc(EXIT_FAILURE, s, "pthread_getcpuclockid");
}

```

```

pclock("Main thread CPU time: ", cid);

/* The preceding 4 lines of code could have been replaced by:

    pclock("Main thread CPU time: ", CLOCK_THREAD_CPUTIME_ID); */

s = pthread_getcpuclockid(thread, &cid);

if (s != 0)

    errc(EXIT_FAILURE, s, "pthread_getcpuclockid");

pclock("Subthread CPU time: 1 ", cid);

exit(EXIT_SUCCESS);    /* Terminates both threads */

}

```

SEE ALSO

[clock_gettime\(2\)](#), [clock_settime\(2\)](#), [timer_create\(2\)](#), [clock_getcpuclockid\(3\)](#),
[pthread_self\(3\)](#), [pthreads\(7\)](#), [time\(7\)](#)

Linux man-pages 6.17

2025-08-20

[pthread_getcpuclockid\(3\)](#)