

Re: select & gettimeofday

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.system/2004-02/0271.html>

From: Floyd Davidson (floyd_at_barrow.com)

Date: 02/15/04

Date: Sun, 15 Feb 2004 04:09:47 -0900

Moritz Franosch <mail@Franosch.org> wrote:

> *Floyd Davidson <floyd@barrow.com> writes:*

>

>> *Moritz Franosch <mail@Franosch.org> wrote:*

>> >

>> > *As TIMEOUT specifies the _maximum_ time to wait, select should*

>> > *_always_ return _before_ the timeout, i.e. before 10 ms or before 3*

>> > *ms, respectively. It does not in the case TIMEOUT=3 ms (in this case*

>> > *it returns after about 10 ms). Cases where select returns after*

>> > *TIMEOUT are a kernel and/or glibc bug.*

>>

>> *There is no bug, and in fact you cannot tell, in a user process,*

>> *exactly when select() returns!*

>

> *In this case the documentation should say so, e.g. "Select can return*

> *before timeout, in any case. Else, timeout is the approximate amount*

> *of time elapsed before select returns." and not "timeout is an upper*

> *bound on the amount of time elapsed before select returns."*

>

> *The function select does not work as described. If something does not*

> *work as described there is a bug, either in the documentation or*

> *elsewhere.*

>

>> *The traditional UNIX scheduler has a 10ms granularity, which*

>> *made sense with slow processors. The 2.6.x kernel changed that,*

>> *and the program will demonstrate finer granularity with the*

>> *program you showed.*

>

> *No, surprisingly it does not. Why?!*

>

> *I've set tm.tv_usec=3000 (3 ms) in the example run below.*

>

> *jfranosch@koma:~/tmp/test> cat /proc/version*

> *Linux version 2.6.1 (jfranosch@cell2) (gcc version 3.2.2) #1 SMP Tue Feb 3 20:04:16 CET 2004*

> *jfranosch@koma:~/tmp/test> g++ ./select.cpp*

> *jfranosch@koma:~/tmp/test> ./a.out*

> *2.97*

> *10.238*

>9.799
>10.727
>10.578
>9.835
>9.966

I'm not at all certain what you are doing. However, I've got a test program that runs 1000 iterations each with usleep(), nanaosleep(), select(), and with no delay, and the results from select() on a 2.6.0 kernel are, for select(),

select (Microseconds, rounded to the nearest 100.)

Most often seen values Lowest values

1000 (count: 383) 700 (count: 5)
900 (count: 355) 800 (count: 23)
1100 (count: 78) 900 (count: 355)
1800 (count: 43) 1000 (count: 383)
1200 (count: 40) 1100 (count: 78)

As noted, the times are in microseconds rounded to the nearest 100. The measured delay was 1.00 milliseconds 383 times, 0.9 ms 355 times, 1.1 78 times, and so on.

Five times the delay was only 0.7 ms, which was the lowest delay registered.

The high counts are not of interest because that depends entirely on system load. That run was on a basically idle SMP system. The counts shown add up to 927, hence there were 73 counts that are other than in the 5 lowest or 5 most often seen.

Run on a 2.4.15 kernel (not an SMP box though), the results look like this,

select (Microseconds, rounded to the nearest 100.)

Most often seen values Lowest values

9900 (count: 564) 9900 (count: 564)
10000 (count: 435) 10000 (count: 435)
10100 (count: 1) 10100 (count: 1)

The difference is exactly what would be expected with a scheduler dealing with an HZ value that is 100 instead of 1000.

--

Floyd L. Davidson
Ukpeagvik (Barrow, Alaska)

<http://web.newsquy.com/floyd_davidson>
floyd@barrow.com