

## Re: [PATCH][2.6] first/next\_cpu returns values > NR\_CPUS

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-08/0062.html>

---

**From:** Paul Jackson (*pj\_at\_sgi.com*)

**Date:** 08/01/04

Date: Sun, 1 Aug 2004 04:20:53 -0700  
To: Zwane Mwaikambo <zwane@linuxpower.ca>

Zwane wrote:

> *NR\_CPUS* was 3, the test case may as well be passing *first\_cpu* or *next\_cpu*  
> a value of 0 for the map.

So, if *NR\_CPUS* is 3, and you pass an empty map to *any\_online\_cpu()*  
on an i386, you get back not 3, as expected, but 32 ??

And this is because *find\_next\_bit(0, 3, 0)*, for example, returns 32,  
correct ??

Well ... no ... I must not be guessing your example right yet. Because  
in the above example, *first\_cpu(0)* will (should ?) return with *NR\_CPUS*,  
and the *for\_each\_cpu\_mask()* inside *any\_online\_cpu()* will end there.

Could you give me the rest of the numbers in a specific example?

Please ...

Hmmm ... perhaps you're saying you're passing a non-zero map to  
*any\_online\_cpu()*, but that the bits set in what you pass aren't  
online, which would end up calling *find\_next\_bit()*. Yeah - that  
must be it.

And indeed the i386 *find\_next\_bit()* code can't possibly be honoring a  
size < 32, because it doesn't even consider the size value until it has  
finished the first word without finding a set bit in the last 32-offset  
bits.

> *The "bug" in the i386 find\_next\_bit really*  
> *looks like a feature if you look at the code.*

What code, what feature, what bug ... Please be specific.

Are you referring to the apparent (if I am reading the code for  
*find\_next\_bit* in *arch/i386/lib/bitops.c* correctly) behaviour

of this find\_next\_bit() that it's really only coded for size some multiple of 32?

If so, then wouldn't whether this is a bug or a feature depend on what the other arch's do, and what (if there is anyway to know) was intended, and on what other code is expecting, and on what in the long term will be the least surprising behaviour, resulting in fewest bugs?

That is, are bitmaps only really supposed to work for integral multiples of unsigned longs, or are they supposed to honor fractional long sizes?

A quick look at some other arch's find\_next\_bit() leads me to suspect that they *do* handle fractional long sizes, unlike i386. And it was certainly my expectation that they should do so (returning, for example, 3, not 32, on an empty mask if called with size == 3). These routines *do* take a size that is a bit count, and I don't recall seeing any big hairy warnings that size better be a multiple of BITS\_PER\_LONG.

If all this is so, then i386 find\_next\_bit() is wrong. Possibly other some arch's too --- it's not code that I can read easily.

If not, then in addition to fixing cpumask.h, we'd better also consider whether we need to fix:

```
drivers/atm/lanai.c:
    vci = find_next_bit(lp, NUM_VCI, vci + 1);
include/linux/nodemask.h:
    return find_next_bit(srcp->bits, nbits, n+1);
kernel/sched.c:
    idx = find_next_bit(array->bitmap, MAX_PRIO, idx);
lib/idr.c:
    m = find_next_bit(&bm, IDR_SIZE, n);
mm/mempolicy.c:
    next = find_next_bit(policy->v.nodes, MAX_NUMNODES, 1+nid);
mm/mempolicy.c:
    nid = find_next_bit(pol->v.nodes, MAX_NUMNODES, nid+1);
```

Adding Matthew Dobson to this thread - since his new nodemask.h gets hit with this alot harder than cpumask.h, because it is more common to have a nodemask that isn't a multiple of a long in size.

--

```
I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.650.933.1373
```

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
Please read the FAQ at <http://www.tux.org/lkml/>