

Re: [PATCH] x86: generic versions of find\_first\_(zero\_)bit, convert i386

## Re: [PATCH] x86: generic versions of find\_first\_(zero\_)bit, convert i386

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg12126.html>

---

- *From:* Alexander van Heukelum <[heukelum@xxxxxxxxxxxxxx](mailto:heukelum@xxxxxxxxxxxxxx)>
  - *Date:* Mon, 31 Mar 2008 21:38:31 +0200
- 

On Mon, Mar 31, 2008 at 10:22:40AM -0700, Stephen Hemminger wrote:

On Mon, 31 Mar 2008 19:15:06 +0200

Alexander van Heukelum <[heukelum@xxxxxxxxxxxxxx](mailto:heukelum@xxxxxxxxxxxxxx)> wrote:

Generic versions of `__find_first_bit` and `__find_first_zero_bit` are introduced as simplified versions of `__find_next_bit` and `__find_next_zero_bit`. Their compilation and use are guarded by a new config variable `GENERIC_FIND_FIRST_BIT`.

The generic versions of `find_first_bit` and `find_first_zero_bit` are implemented in terms of the newly introduced `__find_first_bit` and `__find_first_zero_bit`.

This patch also converts i386 to the generic functions. The text size shrinks slightly due to unInlining of the `find_*_bit` functions.

```
text data bss dec hex filename
4764939 480324 622592 5867855 59894f vmlinux (i386 defconfig before)
4764645 480324 622592 5867561 598829 vmlinux (i386 defconfig after)
```

Signed-off-by: Alexander van Heukelum <[heukelum@xxxxxxxxxxxxxx](mailto:heukelum@xxxxxxxxxxxxxx)>

Size isn't everything, what is the performance difference?

Hi,

Performance should not change too much. UnInlining of the functions has some impact, of course, but this should be visible only for small bitmap sizes. Measuring the performance impact by doing artificial benchmarks is a bit problematic too, because it is hard to guess what patterns are important. Anyhow, I hacked together a program (in userspace) that searches for a bit in a bitmap. In pseudo code:

Re: [PATCH] x86: generic versions of find\_first\_(zero\_)bit, convert i386

```
bitmap <- [0...]  
for bitmapsizes=1 to 512  
for bitposition=0 to bitmapsizes-1  
find_first_bit in bitmap  
bitmap[bitposition] <- 1  
find_first_bit in bitmap  
bitmap[bitposition] <- 0
```

After each find\_first\_bit, the result is checked against the expected result. A similar test is done for searching zero bits. The two tests are performed 1000 times in a loop. On a 2.4GHz (P-IV-type) Xeon, I get the following results:

```
$ gcc -DNEW -fomit-frame-pointer -Os find_first_bit.c && time ./a.out  
real 0m15.006s  
$ nm -nStd  
0000000134513492 00000000000000065 T find_first_bit  
0000000134513557 00000000000000062 T find_first_zero_bit  
0000000134513619 00000000000000190 T testzerobit  
0000000134513809 00000000000000187 T testonebit  
0000000134513996 00000000000000045 T main
```

and

```
$ gcc -fomit-frame-pointer -Os find_first_bit.c && time ./a.out  
real 0m17.617s  
0000000134513492 00000000000000293 T testzerobit  
0000000134513785 00000000000000240 T testonebit  
0000000134514025 00000000000000045 T main
```

So in this particular case, on this particular machine, with this particular mix of searches, the new code is somewhat faster, even though it is out-of-line.

A similar, but more convincing, change was seen when the assembly versions for find\_next\_bit and find\_next\_zero\_bit were replaced by the generic ones.

Greetings,  
Alexander

—

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

Re: [PATCH] x86: generic versions of find\_first\_(zero\_)bit, convert i386