

## Re: 2.4 vs 2.6 versions of include/linux/ioport.h

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-08/1166.html>

---

**From:** Gene Heskett ([gene.heskett\\_at\\_verizon.net](mailto:gene.heskett_at_verizon.net))

**Date:** 08/05/03

To: "Randy.Dunlap" <[rddunlap@osdl.org](mailto:rddunlap@osdl.org)>

Date: Tue, 5 Aug 2003 11:22:35 -0400

On Tuesday 05 August 2003 10:57, Randy.Dunlap wrote:

> On Tue, 5 Aug 2003 10:41:08 -0400 Gene Heskett

<[gene.heskett@verizon.net](mailto:gene.heskett@verizon.net)> wrote:

> | Greetings;

> |

> | In the 2.4 includes, find this in ioport.h

> | ----

> | /\* Compatibility cruft \*/

> | #define check\_region(start,n) \_\_check\_region(&ioport\_resource,

> | (start), (n))

> | [snip]

> | extern int \_\_check\_region(struct resource \*, unsigned long,

> | unsigned long);

> | ----

> | But in the 2.6 version, find this:

> | ----

> | /\* Compatibility cruft \*/

> | [snip]

> | extern int \_\_check\_region(struct resource \*, unsigned long,

> | unsigned long);

> | [snip]

> | static inline int \_\_deprecated check\_region(unsigned long s,

> | unsigned long n)

> | {

> | return \_\_check\_region(&ioport\_resource, s, n);

> | }

> | ----

> | First, the define itself is missing in the 2.6 version.

> |

> | Many drivers seem to use this call, and in that which I'm trying

> | to build, the nforce and advansys modules use it. And while the

> | modules seem to build, they do not run properly.

> |

> | I cannot run 2.6.x for extended tests because of the advansys

> | breakage this causes. I also haven't even tried to run X because

> | of the nforce error reported when its built, the same error as

> | attacks the advansys code.

Linux-Kernel: Re: 2.4 vs 2.6 versions of include/linux/ioport.h

```
>/
>/ Can I ask why this change was made, and is there a suitable
>/ replacement call available that these drivers could use instead of
>/ check_region(), as shown here in a snip from advansys.c?
>/ -----
>/ if (check_region(iop, ASC_IOADR_GAP) != 0) {
>/ ...
>/ if (check_region(iop_base, ASC_IOADR_GAP) != 0) {
>/ ...
>/
>/ Hopeing for some hints here.
>
> check_region() was racy. Use request_region() instead.
>
> if (!request_region(iop, ASC_IOADR_GAP, "advansys")) {
> ...
>
> if (!request_region(iop_base, ASC_IOADR, "advansys")) {
> ...
>
> Of course, if successful, this assigns the region to the driver,
> while check_region() didn't do this, so release_region() should be
> used as needed to return the resources.
```

Oops... I have a test compile going that changed those to check\_mem\_region. And while I didn't change the i2c stuff, which still reports the error, advansys.o built w/o error this time.

Ok, so I can change that to !request\_region, but I have NDI when to go about releasing it, if ever, for a kernel driver thats either there, and the hardware is used, or not because the hardware isn't present. It seems to me that if its allocated to this driver, and capable of being re-used at anytime, then the allocation should, once made, stand. Or is my view of the world skewed and it should be done at the bottom of whatever conditional is involved? Inquiring minds want to know. I guess it all depends on what happens if the call is repeated. Will it assign a new buffer each time?, thereby causing a memory leak, or will it find its been done once and return success anyway?

Thanks very much for the quick response.

```
>--
>~Randy
```

```
--
Cheers, Gene
AMD K6-III@500mhz 320M
Athlon1600XP@1400mhz 512M
99.27% setiathome rank, not too shabby for a WV hillbilly
Yahoo.com attornies please note, additions to this message
by Gene Heskett are:
```

## Linux-Kernel: Re: 2.4 vs 2.6 versions of include/linux/ioport.h

Copyright 2003 by Maurice Eugene Heskett, all rights reserved.

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to [majordomo@vger.kernel.org](mailto: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/>