

# managing kallsyms\_addresses

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-01/msg13436.html>

---

- *From:* Robin Getz <[rgetz@xxxxxxxxxxxxxxxxxxxxxx](mailto:rgetz@xxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 31 Jan 2008 11:48:18 -0500
- 

When the kernel needs to find out what symbol is at a specific address, it uses `kallsyms_lookup()`. This seems to work pretty well – almost.

The problem is today, we don't to remove the symbols from the init section when the init section is freed. There is invalid data in `kallsyms_addresses`.

The problem I have been experiencing – If you have a module get loaded into a location which was init, then `kallsyms_lookup()` can return init labels, rather than the module labels. (since it looks up kernel labels before module labels).

What happens is if there is a OOPS in the module, the labels from the OOPS can point to init code (which doesn't exist), which confuses the heck out of users and developers...

There would be two solutions:

- when freeing the init section, remove all the init labels from the `kallsyms_addresses`, and resort/pack it.
- if the init section is unloaded, have `is_kernel_inittext` always return 0.

I assume that similar things need to be handled for module init too, but I have not run into that yet.

Thoughts?

--

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to [majordomo@xxxxxxxxxxxxxxxxxx](mailto:majordomo@xxxxxxxxxxxxxxxxxx)

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>