Subject: [PATCH 2/2] move unneeded data to initdata section
Posted by den on Wed, 07 Nov 2007 11:57:38 GMT
View Forum Message <> Reply to Message

This patch reverts Eric's commit 2b008b0a8e96b726c603c5e1a5a7a509b5f61e35

It diets .text & .data section of the kernel if CONFIG_NET_NS is not set.
This is safe after list operations cleanup.

Signed-of-by: Denis V. Lunev <den@openvz.org>

```
--- ./drivers/net/loopback.c.reversed 2007-10-30 14:45:07.000000000 +0300
+++ ./drivers/net/loopback.c 2007-11-01 17:30:55.000000000 +0300
@@ -284,7 +284,7 @@ static __net_exit void loopback_net_exit
  unregister_netdev(dev);
 }

-static struct pernet_operations loopback_net_ops = {
+static struct pernet_operations __net_initdata loopback_net_ops = {
        .init = loopback_net_init,
        .exit = loopback_net_exit,
 };
--- ./fs/proc/proc_net.c.reversed 2007-10-30 14:45:07.000000000 +0300
+++ ./fs/proc/proc_net.c 2007-11-01 17:30:57.000000000 +0300
@@ -185,7 +185,7 @@ static __net_exit void proc_net_ns_exit(
  kfree(net->proc_net_root);
 }

-static struct pernet_operations proc_net_ns_ops = {
+static struct pernet_operations __net_initdata proc_net_ns_ops = {
  .init = proc_net_ns_init,
  .exit = proc_net_ns_exit,
 };
--- ./include/net/net_namespace.h.reversed 2007-10-30 14:45:07.000000000 +0300
+++ ./include/net/net_namespace.h 2007-11-01 17:30:58.000000000 +0300
@@ -102,9 +102,11 @@ static inline void release_net(struct ne
 #ifdef CONFIG_NET_NS
 #define __net_init
 #define __net_exit
+#define __net_initdata
 #else
 #define __net_init __init
 #define __net_exit __exit_refok
+#define __net_initdata __initdata
 #endif

 struct pernet_operations {
--- ./net/core/dev.c.reversed 2007-10-30 14:45:08.000000000 +0300
```

```
+++ ./net/core/dev.c 2007-11-01 17:30:58.000000000 +0300
@@ -2676,7 +2676,7 @@ static void __net_exit dev_proc_net_exit
  proc_net_remove(net, "dev");
 }

-static struct pernet_operations dev_proc_ops = {
+static struct pernet_operations __net_initdata dev_proc_ops = {
  .init = dev_proc_net_init,
  .exit = dev_proc_net_exit,
 };
@@ -4336,7 +4336,7 @@ static void __net_exit netdev_exit(struc
  kfree(net->dev_index_head);
 }

-static struct pernet_operations  netdev_net_ops = {
+static struct pernet_operations __net_initdata netdev_net_ops = {
  .init = netdev_init,
  .exit = netdev_exit,
 };
@@ -4367,7 +4367,7 @@ static void __net_exit default_device_ex
  rtnl_unlock();
 }

-static struct pernet_operations  default_device_ops = {
+static struct pernet_operations __net_initdata default_device_ops = {
  .exit = default_device_exit,
 };

--- ./net/core/dev_mcast.c.reversed 2007-10-30 14:45:08.000000000 +0300
+++ ./net/core/dev_mcast.c 2007-11-01 17:31:00.000000000 +0300
@@ -285,7 +285,7 @@ static void __net_exit dev_mc_net_exit(s
  proc_net_remove(net, "dev_mcast");
 }

-static struct pernet_operations dev_mc_net_ops = {
+static struct pernet_operations __net_initdata dev_mc_net_ops = {
  .init = dev_mc_net_init,
  .exit = dev_mc_net_exit,
 };
--- ./net/netlink/af_netlink.c.reversed 2007-10-30 14:45:08.000000000 +0300
+++ ./net/netlink/af_netlink.c 2007-11-01 17:31:01.000000000 +0300
@@ -1888,7 +1888,7 @@ static void __net_exit netlink_net_exit(
 #endif
 }

-static struct pernet_operations netlink_net_ops = {
+static struct pernet_operations __net_initdata netlink_net_ops = {
  .init = netlink_net_init,
```

```
  .exit = netlink_net_exit,
 };
```

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by davem on Tue, 13 Nov 2007 11:24:15 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Wed, 7 Nov 2007 15:01:00 +0300

> This patch reverts Eric's commit 2b008b0a8e96b726c603c5e1a5a7a509b5f61e35
>
> It diets .text & .data section of the kernel if CONFIG_NET_NS is not set.
> This is safe after list operations cleanup.
>
> Signed-of-by: Denis V. Lunev <den@openvz.org>

Applied, thanks Denis.

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by ebiederm on Thu, 15 Nov 2007 14:32:40 GMT
View Forum Message <> Reply to Message

"Denis V. Lunev" <den@openvz.org> writes:

> This patch reverts Eric's commit 2b008b0a8e96b726c603c5e1a5a7a509b5f61e35
>
> It diets .text & .data section of the kernel if CONFIG_NET_NS is not set.
> This is safe after list operations cleanup.

Ok. This patch is technically safe because none of the touched
code can live in a module and so we never touch the exit code path.

However in the general case and as a code idiom this __net_initdata
on struct pernet_operations is fundamentally horribly broken.

Look at what happens if we use this idiom in module. There
is only one definition of __initdata ".init.data". The module
loader places all sections that begin with .init in a region of
memory that will be discarded after module initialization.

So in register_pernet_operations we pass in the a pointer to struct
pernet_operations and call the init method. Later when we remove the
module we again pass in the pointer to struct pernet_operations which
lived in an init section so it has been discarded. We dereference

that pointer to find the exit method and KABOOM!!!!

So I'm still opposed to __net_initdata on the grounds that at best
it is like putting our head under a guillotine and reaching up and
sawing at the row that holds the blade up with a pocket knife. It is
a think rope and a puny knife so you are safe for a while....

Eric

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by den on Thu, 15 Nov 2007 14:39:53 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> "Denis V. Lunev" <den@openvz.org> writes:
>
>> This patch reverts Eric's commit 2b008b0a8e96b726c603c5e1a5a7a509b5f61e35
>>
>> It diets .text & .data section of the kernel if CONFIG_NET_NS is not set.
>> This is safe after list operations cleanup.
>
> Ok. This patch is technically safe because none of the touched
> code can live in a module and so we never touch the exit code path.
>
> However in the general case and as a code idiom this __net_initdata
> on struct pernet_operations is fundamentally horribly broken.
>
> Look at what happens if we use this idiom in module. There
> is only one definition of __initdata ".init.data". The module
> loader places all sections that begin with .init in a region of
> memory that will be discarded after module initialization.

nothing is discarded after module load. Though, I can be wrong. Could
you point me to the exact place?

Regards,
 Den

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by Sam Ravnborg on Thu, 15 Nov 2007 15:14:29 GMT
View Forum Message <> Reply to Message

On Thu, Nov 15, 2007 at 05:42:04PM +0300, Denis V. Lunev wrote:
> Eric W. Biederman wrote:
> > "Denis V. Lunev" <den@openvz.org> writes:
> >
> >> This patch reverts Eric's commit 2b008b0a8e96b726c603c5e1a5a7a509b5f61e35
> >>
> >> It diets .text & .data section of the kernel if CONFIG_NET_NS is not set.
> >> This is safe after list operations cleanup.
> >
> > Ok. This patch is technically safe because none of the touched
> > code can live in a module and so we never touch the exit code path.
> >
> > However in the general case and as a code idiom this __net_initdata
> > on struct pernet_operations is fundamentally horribly broken.
> >
> > Look at what happens if we use this idiom in module. There
> > is only one definition of __initdata ".init.data". The module
> > loader places all sections that begin with .init in a region of
> > memory that will be discarded after module initialization.
>
> nothing is discarded after module load. Though, I can be wrong. Could
> you point me to the exact place?
If __initdata is not discarded after module load then we should do it.
There is no reason to waste __initdata RAM when the module is loaded.

 Sam

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by ebiederm on Thu, 15 Nov 2007 18:19:26 GMT
View Forum Message <> Reply to Message

Sam Ravnborg <sam@ravnborg.org> writes:

> On Thu, Nov 15, 2007 at 05:42:04PM +0300, Denis V. Lunev wrote:
>>
>> nothing is discarded after module load. Though, I can be wrong. Could
>> you point me to the exact place?
> If __initdata is not discarded after module load then we should do it.
> There is no reason to waste __initdata RAM when the module is loaded.

Down at the bottom of sys_init_module we have:

```
 /* Drop initial reference. */
 module_put(mod);
 unwind_remove_table(mod->unwind_info, 1);

 module_free(mod, mod->module_init);
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 mod->module_init = NULL;
 mod->init_size = 0;
 mod->init_text_size = 0;
 mutex_unlock(&module_mutex);

 return 0;
```

Which frees the memory for the .init sections.

Eric

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by Sam Ravnborg on Thu, 15 Nov 2007 18:43:34 GMT
View Forum Message <> Reply to Message

On Thu, Nov 15, 2007 at 11:19:26AM -0700, Eric W. Biederman wrote:
> Sam Ravnborg <sam@ravnborg.org> writes:
>
> > On Thu, Nov 15, 2007 at 05:42:04PM +0300, Denis V. Lunev wrote:
> >>
> >> nothing is discarded after module load. Though, I can be wrong. Could
> >> you point me to the exact place?
> > > If __initdata is not discarded after module load then we should do it.
> > > There is no reason to waste __initdata RAM when the module is loaded.
>
> Down at the bottom of sys_init_module we have:
>
> /* Drop initial reference. */
> module_put(mod);
> unwind_remove_table(mod->unwind_info, 1);
>
> module_free(mod, mod->module_init);
> ^^
> mod->module_init = NULL;

> mod->init_size = 0;
> mod->init_text_size = 0;
> mutex_unlock(&module_mutex);
>
> return 0;
>
> Which frees the memory for the .init sections.

Thanks for clarifying this Eric - should have looked myself..

 Sam

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by den on Thu, 15 Nov 2007 19:17:14 GMT
View Forum Message <> Reply to Message

Sam Ravnborg wrote:
> On Thu, Nov 15, 2007 at 11:19:26AM -0700, Eric W. Biederman wrote:
>> Sam Ravnborg <sam@ravnborg.org> writes:
>>
>>> On Thu, Nov 15, 2007 at 05:42:04PM +0300, Denis V. Lunev wrote:
>>>> nothing is discarded after module load. Though, I can be wrong. Could
>>>> you point me to the exact place?
>>> If __initdata is not discarded after module load then we should do it.
>>> There is no reason to waste __initdata RAM when the module is loaded.
>> Down at the bottom of sys_init_module we have:
>>
>> /* Drop initial reference. */
>> module_put(mod);
>> unwind_remove_table(mod->unwind_info, 1);
>>
>> module_free(mod, mod->module_init);
>> ^^
>> mod->module_init = NULL;
>> mod->init_size = 0;
>> mod->init_text_size = 0;
>> mutex_unlock(&module_mutex);
>>
>> return 0;
>>
>> Which frees the memory for the .init sections.
>
> Thanks for clarifying this Eric - should have looked myself..

clear :) I was wrong... Thank you for pointing this out.

will you mind against this?

```
diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index 5dd6d90..d136707 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -119,10 +119,14 @@ static inline struct net *maybe_get_net(struct net *net)
 #ifdef CONFIG_NET_NS
 #define __net_init
 #define __net_exit
-#define __net_initdata
 #else
 #define __net_init __init
 #define __net_exit __exit_refok
+#endif
+
+#if defined(CONFIG_NET_NS) || defined(MODULE)
+#define __net_initdata
+#else
 #define __net_initdata __initdata
 #endif
```

Subject: Re: [PATCH 2/2] move unneeded data to initdata section
Posted by Sam Ravnborg on Thu, 15 Nov 2007 19:34:35 GMT
View Forum Message <> Reply to Message

On Thu, Nov 15, 2007 at 10:17:14PM +0300, Denis V. Lunev wrote:
>
> will you mind against this?

> diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
> index 5dd6d90..d136707 100644
> --- a/include/net/net_namespace.h
> +++ b/include/net/net_namespace.h
> @@ -119,10 +119,14 @@ static inline struct net *maybe_get_net(struct net *net)
> #ifdef CONFIG_NET_NS
> #define __net_init
> #define __net_exit

> -#define __net_initdata
> #else
> #define __net_init __init
> #define __net_exit __exit_refok
> +#endif
> +
> +#if defined(CONFIG_NET_NS) || defined(MODULE)
> +#define __net_initdata
> +#else
> #define __net_initdata __initdata
> #endif

n principle I am against this approach.
__initdata is far too overloaded with different stuff.

A much more preferred approach should be to create new sections
named for example .init.data.net and .init.data.net.module

And then in include/asm-generic/vmlinux.lds.h decide the
location of these sections.

On top of this we would have to teach modpost about these new sections.
But the advantage of this approach is that the section mismatch
checks are *independent* of the module being a MODULE or build-in.
The check will still happen.

In this way we avoid the situation where a warning only pops up
in certain configurations.

To do so will obviously require a bit more linker script
consolidation but if you or some else could step in a do this
it would be great!

 Sam

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers