Subject: [PATCH 1/3] nitpick: make simple functions inline
Posted by Glauber Costa on Sun, 11 Dec 2011 14:45:36 GMT
View Forum Message <> Reply to Message

Those are quite simple bit-testing functions that are
only used within this file. Not reason for them not to
be inline.

Signed-off-by: Glauber Costa <glommer@parallels.com>
---
 kernel/cgroup.c | 4 ++--
 1 files changed, 2 insertions(+), 2 deletions(-)

diff --git a/kernel/cgroup.c b/kernel/cgroup.c
index d9d5648..e4b9d3c 100644
--- a/kernel/cgroup.c
+++ b/kernel/cgroup.c
@@ -241,12 +241,12 @@ static int cgroup_is_releasable(const struct cgroup *cgrp)
  return (cgrp->flags & bits) == bits;
 }

-static int notify_on_release(const struct cgroup *cgrp)
+static inline int notify_on_release(const struct cgroup *cgrp)
 {
  return test_bit(CGRP_NOTIFY_ON_RELEASE, &cgrp->flags);
 }

-static int clone_children(const struct cgroup *cgrp)
+static inline int clone_children(const struct cgroup *cgrp)
 {
  return test_bit(CGRP_CLONE_CHILDREN, &cgrp->flags);
 }
--
1.7.6.4

Subject: Re: [PATCH 1/3] nitpick: make simple functions inline
Posted by KOSAKI Motohiro on Sun, 11 Dec 2011 18:55:55 GMT
View Forum Message <> Reply to Message

> -static int notify_on_release(const struct cgroup *cgrp)
> +static inline int notify_on_release(const struct cgroup *cgrp)
> {
>   return test_bit(CGRP_NOTIFY_ON_RELEASE,&cgrp->flags);
> }
>
> -static int clone_children(const struct cgroup *cgrp)
> +static inline int clone_children(const struct cgroup *cgrp)

> {
>   return test_bit(CGRP_CLONE_CHILDREN,&cgrp->flags);
> }

Can you please tell us which compiler failed automatic inlining?
I suspect gcc is enough sane and we don't need this patch.

Subject: Re: [PATCH 1/3] nitpick: make simple functions inline
Posted by Glauber Costa on Sun, 11 Dec 2011 20:44:54 GMT
View Forum Message <> Reply to Message

On 12/11/2011 07:55 PM, KOSAKI Motohiro wrote:
>> -static int notify_on_release(const struct cgroup *cgrp)
>> +static inline int notify_on_release(const struct cgroup *cgrp)
>> {
>>   return test_bit(CGRP_NOTIFY_ON_RELEASE,&cgrp->flags);
>> }
>>
>> -static int clone_children(const struct cgroup *cgrp)
>> +static inline int clone_children(const struct cgroup *cgrp)
>> {
>>   return test_bit(CGRP_CLONE_CHILDREN,&cgrp->flags);
>> }
>
> Can you please tell us which compiler failed automatic inlining?
> I suspect gcc is enough sane and we don't need this patch.

Of course we don't need, that's the very definition of a "nitpick".
This patch is directed towards the reader, not the compiler. Maintainers
are free to take it or not, although I believe being explicit is better.

Subject: Re: [PATCH 1/3] nitpick: make simple functions inline
Posted by Tejun Heo on Mon, 12 Dec 2011 17:27:25 GMT
View Forum Message <> Reply to Message

Hello,

On Sun, Dec 11, 2011 at 09:44:54PM +0100, Glauber Costa wrote:
> On 12/11/2011 07:55 PM, KOSAKI Motohiro wrote:
> > Can you please tell us which compiler failed automatic inlining?
> > I suspect gcc is enough sane and we don't need this patch.
>
> Of course we don't need, that's the very definition of a "nitpick".
> This patch is directed towards the reader, not the compiler. Maintainers
> are free to take it or not, although I believe being explicit is better.

These days, I don't think adding inline buys us much (other than
explicit cases where always_inline or noinline is necessary). gcc
already does good enough job for inlining and 'inline' hint seems more
to hinder rather than help and I don't really see what it buys for
code readers either, so I won't be taking this one.

Thanks.

--
tejun

Subject: Re: [PATCH 1/3] nitpick: make simple functions inline
Posted by Li Zefan on Wed, 14 Dec 2011 01:09:25 GMT
View Forum Message <> Reply to Message

22:45, Glauber Costa wrote:
> Those are quite simple bit-testing functions that are
> only used within this file. Not reason for them not to
> be inline.
>

It's better to leave the optimization decision to gcc.

And I've confirmed they are inlined by gcc in my box.

(btw, please add "cgroup" prefix to the patch subject line)

> Signed-off-by: Glauber Costa <glommer@parallels.com>
> ---
> kernel/cgroup.c | 4 ++--
> 1 files changed, 2 insertions(+), 2 deletions(-)
>
> diff --git a/kernel/cgroup.c b/kernel/cgroup.c
> index d9d5648..e4b9d3c 100644
> --- a/kernel/cgroup.c
> +++ b/kernel/cgroup.c
> @@ -241,12 +241,12 @@ static int cgroup_is_releasable(const struct cgroup *cgrp)
> return (cgrp->flags & bits) == bits;
> }
>
> -static int notify_on_release(const struct cgroup *cgrp)
> +static inline int notify_on_release(const struct cgroup *cgrp)
> {
> return test_bit(CGRP_NOTIFY_ON_RELEASE, &cgrp->flags);
> }
>

> -static int clone_children(const struct cgroup *cgrp)
> +static inline int clone_children(const struct cgroup *cgrp)
> {
> return test_bit(CGRP_CLONE_CHILDREN, &cgrp->flags);
> }