Subject: Re: [PATCH 6/6] BC: kernel memory accounting (marks)
Posted by Geert Uytterhoeven on Thu, 24 Aug 2006 09:30:09 GMT
View Forum Message <> Reply to Message

On Wed, 23 Aug 2006, Dave Hansen wrote:
> I'm working on a patch to unify as many of the alloc_thread_info()
> functions as I can. That should at least give you one place to modify
> and track the thread_info allocations. I've only compiled for x86_64
> and i386, but I'm working on more. A preliminary version is attached.

> --- clean/include/asm-m68k/thread_info.h~unify-alloc-thread-info 2006-08-23
15:44:52.000000000 -0700
> +++ clean-dave/include/asm-m68k/thread_info.h 2006-08-23 15:45:32.000000000 -0700
> @@ -24,14 +24,7 @@ struct thread_info {
> }, \
> }
>
> -/* THREAD_SIZE should be 8k, so handle differently for 4k and 8k machines */
> -#if PAGE_SHIFT == 13 /* 8k machines */
> -#define alloc_thread_info(tsk) ((struct thread_info *)__get_free_pages(GFP_KERNEL,0))
> -#define free_thread_info(ti) free_pages((unsigned long)(ti),0)
> -#else /* otherwise assume 4k pages */
> -#define alloc_thread_info(tsk) ((struct thread_info *)__get_free_pages(GFP_KERNEL,1))
> -#define free_thread_info(ti) free_pages((unsigned long)(ti),1)
> -#endif /* PAGE_SHIFT == 13 */
> +#define THREAD_SHIFT 1
^
Shouldn't this be 13?

> --- /dev/null 2005-03-30 22:36:15.000000000 -0800
> +++ clean-dave/include/linux/thread_alloc.h 2006-08-23 16:00:41.000000000 -0700
> @@ -0,0 +1,42 @@
> +#ifndef _LINUX_THREAD_ALLOC
> +#define _LINUX_THREAD_ALLOC
> +
> +#ifndef THREAD_SHIFT
> +#define THREAD_SHIFT PAGE_SHIFT
> +#endif
> +#ifndef THREAD_ORDER
> +#define THREAD_ORDER (THREAD_SHIFT - PAGE_SHIFT)
> +#endif
> +
> +struct thread_info;
> +struct task;
> +
> +#if THREAD_SHIFT >= PAGE_SHIFT
> +static inline struct thread_info *alloc_thread_info(struct task_struct *tsk)
> +{

```
> + gfp_t flags = GFP_KERNEL;
> +#ifdef CONFIG_DEBUG_STACK_USAGE
> + flags |= __GFP_ZERO;
> +#endif
> + return (struct thread_info *)__get_free_pages(flags, THREAD_ORDER);
> +}
> +static inline void free_thread_info(struct thread_info *ti)
> +{
> + free_pages((unsigned long)ti, THREAD_ORDER);
> +}
> +#else /* THREAD_SHIFT < PAGE_SHIFT */
> +static inline struct thread_info *alloc_thread_info(struct task_struct *tsk)
> +{
> +#ifdef CONFIG_DEBUG_STACK_USAGE
> + return kzalloc(THREAD_SIZE, GFP_KERNEL);
> +#else
> + return kmalloc(THREAD_SIZE, GFP_KERNEL);
> +#endif
> +}
> +static inline void free_thread_info(struct thread_info *ti)
> +{
> + kfree(ti);
> +}
> +#endif /* THREAD_SHIFT < PAGE_SHIFT */
> +
> +#endif /* _LINUX_THREAD_ALLOC */
```

Gr{oetje,eeting}s,

Geert

--
Geert Uytterhoeven -- There's lots of Linux beyond ia32 -- geert@linux-m68k.org

In personal conversations with technical people, I call myself a hacker. But
when I'm talking to journalists I just say "programmer" or something like that.
-- Linus Torvalds