

[PATCH 4/9] x86-64 use task_thread_info()

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-11/9630.html>

From: Benjamin LaHaise (*bcr1_at_kvack.org*)

Date: 11/30/05

Date: Tue, 29 Nov 2005 23:21:43 -0500

To: Andi Kleen <ak@suse.de>

Prepare x86-64 for thread_info in task_struct by using the task_thread_info() macro instead of struct task_struct->thread_info directly.

```

---
 arch/x86_64/kernel/i387.c      |    2 +-
 arch/x86_64/kernel/process.c   |    2 +-
 arch/x86_64/kernel/smpboot.c  |    2 +-
 include/asm-x86_64/i387.h     |    8 +++++-
 4 files changed, 7 insertions(+), 7 deletions(-)
applies-to: f7870b25d1df24c7f544559ea8c7295a3a636dd5
dd7b8a32d0da262a26934e79da8c08fcb2c80e89
diff --git a/arch/x86_64/kernel/i387.c b/arch/x86_64/kernel/i387.c
index d9b22b6..a5d7e16 100644
--- a/arch/x86_64/kernel/i387.c
+++ b/arch/x86_64/kernel/i387.c
@@ -95,7 +95,7 @@ int save_i387(struct _fpstate __user *bu
     if (!used_math())
         return 0;
     clear_used_math(); /* trigger finit */
-    if (tsk->thread_info->status & TS_USED_FPU) {
+    if (task_thread_info(tsk)->status & TS_USED_FPU) {
         err = save_i387_checking((struct i387_fxsave_struct __user *)buf);
         if (err) return err;
         stts();
diff --git a/arch/x86_64/kernel/process.c b/arch/x86_64/kernel/process.c
index 7519fc5..28ebe45 100644
--- a/arch/x86_64/kernel/process.c
+++ b/arch/x86_64/kernel/process.c
@@ -435,7 +435,7 @@ int copy_thread(int nr, unsigned long cl
     p->thread.rsp0 = (unsigned long) (childregs+1);
     p->thread.userrsp = me->thread.userrsp;

-    set_ti_thread_flag(p->thread_info, TIF_FORK);
+    set_ti_thread_flag(task_thread_info(p), TIF_FORK);

     p->thread.fs = me->thread.fs;
     p->thread.gs = me->thread.gs;
diff --git a/arch/x86_64/kernel/smpboot.c b/arch/x86_64/kernel/smpboot.c
index 683c33f..50d018a 100644
--- a/arch/x86_64/kernel/smpboot.c
+++ b/arch/x86_64/kernel/smpboot.c
@@ -785,7 +785,7 @@ void do_rest:
     init_rsp = c_idle.idle->thread.rsp;
     per_cpu(init_tss,cpu).rsp0 = init_rsp;

```

Linux-Kernel: [PATCH 4/9] x86-64 use task_thread_info()

```
initial_code = start_secondary;
- clear_ti_thread_flag(c_idle.idle->thread_info, TIF_FORK);
+ clear_ti_thread_flag(task_thread_info(c_idle.idle), TIF_FORK);

printk(KERN_INFO "Booting processor %d/%d APIC 0x%x\n", cpu,
        cpus_weight(cpu_present_map),
diff --git a/include/asm-x86_64/i387.h b/include/asm-x86_64/i387.h
index aa39cfd..ae8b4c1 100644
--- a/include/asm-x86_64/i387.h
+++ b/include/asm-x86_64/i387.h
@@ -30,7 +30,7 @@ extern int save_i387(struct _fpstate __u
 */

#define unlazy_fpu(tsk) do { \
-     if ((tsk)->thread_info->status & TS_USEDFFPU) \
+     if (task_thread_info(tsk)->status & TS_USEDFFPU) \
        save_init_fpu(tsk); \
} while (0)

@@ -46,9 +46,9 @@ static inline void tolerant_fwait(void)
}

#define clear_fpu(tsk) do { \
-     if ((tsk)->thread_info->status & TS_USEDFFPU) { \
+     if (task_thread_info(tsk)->status & TS_USEDFFPU) { \
        tolerant_fwait(); \
        (tsk)->thread_info->status &= ~TS_USEDFFPU; \
+     task_thread_info(tsk)->status &= ~TS_USEDFFPU; \
        stts(); \
} \
} while (0)
@@ -135,7 +135,7 @@ static inline void save_init_fpu( struct
{
    asm volatile( "rex64 ; fxsave %0 ; fnclex"
                  : "=m" (tsk->thread.i387.fxsave));
-     tsk->thread_info->status &= ~TS_USEDFFPU;
+     task_thread_info(tsk)->status &= ~TS_USEDFFPU;
    stts();
}

---
0.99.9.GIT
-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to 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/
```