Subject: [PATCH] Consolidate show_regs and show_registers for i386
Posted by Pavel Emelianov on Wed, 08 Aug 2007 14:16:39 GMT
View Forum Message <> Reply to Message

Both functions printk the same information, except for CRx and debug registers in the show_registers() one and a bit different manner. So move the common code into one place. This is already done for x86_64, so I think it's worth having the same on i386.

This saves 100 bytes of .rodata section :) ...
but only 8 from .text :(

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
 arch/i386/kernel/process.c |   56 +++++++++++++++++++++++++++++++++-------------
 arch/i386/kernel/traps.c   |   32 +++----------------------
 include/asm-i386/system.h  |    1
 3 files changed, 45 insertions(+), 44 deletions(-)

diff --git a/arch/i386/kernel/process.c b/arch/i386/kernel/process.c
index 238ac1b..ee40c19 100644
--- a/arch/i386/kernel/process.c
+++ b/arch/i386/kernel/process.c
@@ -299,34 +299,52 @@ static int __init idle_setup(char *str)
 }
 early_param("idle", idle_setup);

-void show_regs(struct pt_regs * regs)
+void __show_registers(struct pt_regs *regs, int all)
 {
  unsigned long cr0 = 0L, cr2 = 0L, cr3 = 0L, cr4 = 0L;
  unsigned long d0, d1, d2, d3, d6, d7;
+ unsigned long esp;
+ unsigned short ss, gs;
+
+ if (user_mode_vm(regs)) {
+  esp = regs->esp;
+  ss = regs->xss & 0xffff;
+  savesegment(gs, gs);
+ } else {
+  esp = (unsigned long) (&regs->esp);
+  savesegment(ss, ss);
+  savesegment(gs, gs);
+ }

  printk("\n");
```

```
- printk("Pid: %d, comm: %20s\n", current->pid, current->comm);
- printk("EIP: %04x:[<%08lx>] CPU: %d\n",0xffff & regs->xcs,regs->eip, smp_processor_id());
+ printk("Pid: %d, comm: %.*s %s (%s %.*s)\n",
+   current->pid, TASK_COMM_LEN, current->comm,
+   print_tainted(), init_utsname()->release,
+   (int)strcspn(init_utsname()->version, " "),
+   init_utsname()->version);
+
+ printk("EIP: %04x:[<%08lx>] EFLAGS: %08lx CPU: %d\n",
+   0xffff & regs->xcs, regs->eip, regs->eflags,
+   smp_processor_id());
  print_symbol("EIP is at %s\n", regs->eip);

- if (user_mode_vm(regs))
-  printk(" ESP: %04x:%08lx",0xffff & regs->xss,regs->esp);
- printk(" EFLAGS: %08lx    %s  (%s %.*s)\n",
-        regs->eflags, print_tainted(), init_utsname()->release,
-        (int)strcspn(init_utsname()->version, " "),
-        init_utsname()->version);
  printk("EAX: %08lx EBX: %08lx ECX: %08lx EDX: %08lx\n",
-  regs->eax,regs->ebx,regs->ecx,regs->edx);
- printk("ESI: %08lx EDI: %08lx EBP: %08lx",
-  regs->esi, regs->edi, regs->ebp);
- printk(" DS: %04x ES: %04x FS: %04x\n",
-        0xffff & regs->xds,0xffff & regs->xes, 0xffff & regs->xfs);
+  regs->eax, regs->ebx, regs->ecx, regs->edx);
+ printk("ESI: %08lx EDI: %08lx EBP: %08lx ESP: %08lx\n",
+  regs->esi, regs->edi, regs->ebp, esp);
+ printk(" DS: %04x ES: %04x FS: %04x GS: %04x SS: %04x\n",
+        regs->xds & 0xffff, regs->xes & 0xffff,
+        regs->xfs & 0xffff, gs, ss);
+
+ if (!all)
+  return;

  cr0 = read_cr0();
  cr2 = read_cr2();
  cr3 = read_cr3();
  cr4 = read_cr4_safe();
- printk("CR0: %08lx CR2: %08lx CR3: %08lx CR4: %08lx\n", cr0, cr2, cr3, cr4);
+ printk("CR0: %08lx CR2: %08lx CR3: %08lx CR4: %08lx\n",
+   cr0, cr2, cr3, cr4);

  get_debugreg(d0, 0);
  get_debugreg(d1, 1);
@@ -334,10 +352,16 @@ void show_regs(struct pt_regs * regs)
  get_debugreg(d3, 3);
  printk("DR0: %08lx DR1: %08lx DR2: %08lx DR3: %08lx\n",
```

```
    d0, d1, d2, d3);
+
  get_debugreg(d6, 6);
  get_debugreg(d7, 7);
- printk("DR6: %08lx DR7: %08lx\n", d6, d7);
+ printk("DR6: %08lx DR7: %08lx\n",
+   d6, d7);
+}

+void show_regs(struct pt_regs * regs)
+{
+ __show_registers(regs, 1);
  show_trace(NULL, regs, &regs->esp);
 }

diff --git a/arch/i386/kernel/traps.c b/arch/i386/kernel/traps.c
index 65eac87..a20c7ad 100644
--- a/arch/i386/kernel/traps.c
+++ b/arch/i386/kernel/traps.c
@@ -282,33 +282,9 @@ EXPORT_SYMBOL(dump_stack);
 void show_registers(struct pt_regs *regs)
 {
  int i;
- int in_kernel = 1;
- unsigned long esp;
- unsigned short ss, gs;
-
- esp = (unsigned long) (&regs->esp);
- savesegment(ss, ss);
- savesegment(gs, gs);
- if (user_mode_vm(regs)) {
-  in_kernel = 0;
-  esp = regs->esp;
-  ss = regs->xss & 0xffff;
- }
+
  print_modules();
- printk(KERN_EMERG "CPU:    %d\n"
-  KERN_EMERG "EIP:    %04x:[<%08lx>]    %s VLI\n"
-  KERN_EMERG "EFLAGS: %08lx   (%s %.*s)\n",
-  smp_processor_id(), 0xffff & regs->xcs, regs->eip,
-  print_tainted(), regs->eflags, init_utsname()->release,
-  (int)strcspn(init_utsname()->version, " "),
-  init_utsname()->version);
- print_symbol(KERN_EMERG "EIP is at %s\n", regs->eip);
- printk(KERN_EMERG "eax: %08lx   ebx: %08lx   ecx: %08lx   edx: %08lx\n",
-  regs->eax, regs->ebx, regs->ecx, regs->edx);
- printk(KERN_EMERG "esi: %08lx   edi: %08lx   ebp: %08lx   esp: %08lx\n",
```

```
-  regs->esi, regs->edi, regs->ebp, esp);
- printk(KERN_EMERG "ds: %04x   es: %04x   fs: %04x  gs: %04x  ss: %04x\n",
-        regs->xds & 0xffff, regs->xes & 0xffff, regs->xfs & 0xffff, gs, ss);
+ __show_registers(regs, 0);
  printk(KERN_EMERG "Process %.*s (pid: %d, ti=%p task=%p task.ti=%p)",
   TASK_COMM_LEN, current->comm, current->pid,
   current_thread_info(), current, task_thread_info(current));
@@ -316,14 +292,14 @@ void show_registers(struct pt_regs *regs
   * When in-kernel, we also print out the stack and code at the
   * time of the fault..
   */
- if (in_kernel) {
+ if (!user_mode_vm(regs)) {
   u8 *eip;
   unsigned int code_prologue = code_bytes * 43 / 64;
   unsigned int code_len = code_bytes;
   unsigned char c;

   printk("\n" KERN_EMERG "Stack: ");
-  show_stack_log_lvl(NULL, regs, (unsigned long *)esp, KERN_EMERG);
+  show_stack_log_lvl(NULL, regs, &regs->esp, KERN_EMERG);

   printk(KERN_EMERG "Code: ");

diff --git a/include/asm-i386/system.h b/include/asm-i386/system.h
index 609756c..c339b30 100644
--- a/include/asm-i386/system.h
+++ b/include/asm-i386/system.h
@@ -314,5 +314,6 @@ extern unsigned long arch_align_stack(un
 extern void free_init_pages(char *what, unsigned long begin, unsigned long end);

 void default_idle(void);
+void __show_registers(struct pt_regs *, int all);

 #endif
```

Subject: Re: [PATCH] Consolidate show_regs and show_registers for i386
Posted by Sam Ravnborg on Wed, 08 Aug 2007 18:15:30 GMT
View Forum Message <> Reply to Message

On Wed, Aug 08, 2007 at 06:16:39PM +0400, Pavel Emelyanov wrote:
> Both functions printk the same information, except for CRx and
> debug registers in the show_registers() one and a bit different
> manner. So move the common code into one place. This is already
> done for x86_64, so I think it's worth having the same on i386.

Hi Pavel.

As a general note that also applies to your mails...
Please do not include linux-arch for for issues that does only concern or are only relevant for a single architecture.

Sam