

# [PATCH 8/9] x86-64 use r10 for current

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-11/9633.html>

---

**From:** Benjamin LaHaise (*bcr1\_at\_kvack.org*)

**Date:** 11/30/05

Date: Tue, 29 Nov 2005 23:22:12 -0500  
To: Andi Kleen <ak@suse.de>

Convert x86-64 to use r10 as current. This results in a significant code size savings for the kernel at the expense of reloading r10 on entry from interrupts or userland. No benchmarks that I am aware of show regressions with this change. Improvements are nothing exceptional, but the cachelines touched by the kernel are reduced.

```
text data bss dec hex filename
3970092 821904 317256 5109252 4df604 vmlinux
4013755 822016 317256 5153027 4ea103 vmlinux.save
```

```
---
arch/x86_64/Makefile | 1 +
arch/x86_64/ia32/ia32entry.S | 7 ++++++
arch/x86_64/kernel/entry.S | 9 ++++++--
arch/x86_64/kernel/process.c | 6 +++++-
arch/x86_64/kernel/setup64.c | 14 ++++++-----
include/asm-x86_64/current.h | 8 +-----
6 files changed, 30 insertions(+), 15 deletions(-)
applies-to: e08120f4e68ac8684c403dc1d28e4072d4088538
86f49f0ecfb1dfe935647e17e804ada7aadae7fb
diff --git a/arch/x86_64/Makefile b/arch/x86_64/Makefile
index a9cd42e..e547830 100644
--- a/arch/x86_64/Makefile
+++ b/arch/x86_64/Makefile
@@ -31,6 +31,7 @@ cflags-$(CONFIG_MK8) += $(call cc-option
cflags-$(CONFIG_MPSC) += $(call cc-option,-march=nocona)
CFLAGS += $(cflags-y)

+CFLAGS += -ffixed-r10
CFLAGS += -mno-red-zone
CFLAGS += -mcmmodel=kernel
CFLAGS += -pipe
diff --git a/arch/x86_64/ia32/ia32entry.S b/arch/x86_64/ia32/ia32entry.S
index e0eb0c7..c5b5918 100644
--- a/arch/x86_64/ia32/ia32entry.S
+++ b/arch/x86_64/ia32/ia32entry.S
@@ -99,6 +99,7 @@ sysenter_do_call:
    cmpl    $(IA32_NR_syscalls),%eax
    jae    ia32_badsys
    IA32_ARG_FIXUP 1
+    movq   %gs:pda_pcurrent,%r10
    call   *ia32_sys_call_table(,%rax,8)
    movq   %rax,RAX-ARGOFFSET(%rsp)
    GET_THREAD_INFO(%r10)
```

## Linux-Kernel: [PATCH 8/9] x86-64 use r10 for current

```

@@ -127,6 +128,7 @@ sysenter_tracesys:
    CLEAR_RREGS
    movq    $-ENOSYS,RAX(%rsp)      /* really needed? */
    movq    %rsp,%rdi              /* &pt_regs -> arg1 */
+   movq    %gs:pda_pcurrent,%r10
    call    syscall_trace_enter
    LOAD_ARGS ARGOFFSET /* reload args from stack in case ptrace changed it */
    RESTORE_REST
@@ -198,6 +200,7 @@ cstar_do_call:
    cmpl   $IA32_NR_syscalls,%eax
    jae   ia32_badsys
    IA32_ARG_FIXUP 1
+   movq    %gs:pda_pcurrent,%r10
    call   *ia32_sys_call_table(,%rax,8)
    movq   %rax,RAX-ARGOFFSET(%rsp)
    GET_THREAD_INFO(%r10)
@@ -220,6 +223,7 @@ cstar_tracesys:
    CLEAR_RREGS
    movq   $-ENOSYS,RAX(%rsp) /* really needed? */
    movq   %rsp,%rdi          /* &pt_regs -> arg1 */
+   movq   %gs:pda_pcurrent,%r10
    call   syscall_trace_enter
    LOAD_ARGS ARGOFFSET /* reload args from stack in case ptrace changed it */
    RESTORE_REST
@@ -282,6 +286,7 @@ ia32_do_syscall:
    cmpl   $(IA32_NR_syscalls),%eax
    jae   ia32_badsys
    IA32_ARG_FIXUP
+   movq   %gs:pda_pcurrent,%r10
    call   *ia32_sys_call_table(,%rax,8) # xxx: rip relative
ia32_sysret:
    movq   %rax,RAX-ARGOFFSET(%rsp)
@@ -291,6 +296,7 @@ ia32_tracesys:
    SAVE_REST
    movq   $-ENOSYS,RAX(%rsp) /* really needed? */
    movq   %rsp,%rdi          /* &pt_regs -> arg1 */
+   movq   %gs:pda_pcurrent,%r10
    call   syscall_trace_enter
    LOAD_ARGS ARGOFFSET /* reload args from stack in case ptrace changed it */
    RESTORE_REST
@@ -336,6 +342,7 @@ ENTRY(ia32_ptregs_common)
    CFI_ADJUST_CFA_OFFSET -8
    CFI_REGISTER rip, r11
    SAVE_REST
+   movq   %gs:pda_pcurrent,%r10
    call   *%rax
    RESTORE_REST
    jmp    ia32_sysret          /* misbalances the return cache */
diff --git a/arch/x86_64/kernel/entry.S b/arch/x86_64/kernel/entry.S
index 9ff4204..b2cec61 100644
--- a/arch/x86_64/kernel/entry.S
+++ b/arch/x86_64/kernel/entry.S
@@ -197,10 +197,11 @@ ENTRY(system_call)
    GET_THREAD_INFO(%rcx)
    testl  $( _TIF_SYSCALL_TRACE | _TIF_SYSCALL_AUDIT | _TIF_SECCOMP ), threadinfo_flags(%rcx)
    CFI_REMEMBER_STATE
+   movq   %r10,%rcx
+   movq   %gs:pda_pcurrent,%r10
    jnz   tracesys
    cmpq   $__NR_syscall_max,%rax
    ja    badsys
-   movq   %r10,%rcx

```

## Linux-Kernel: [PATCH 8/9] x86-64 use r10 for current

```

        call *sys_call_table(,%rax,8) # XXX:    rip relative
        movq %rax,RAX-ARGOFFSET(%rsp)
/*
@@ -263,6 +264,7 @@ badsys:
tracesys:
        CFI_RESTORE_STATE
        SAVE_REST
+       movq    %gs:pda_pcurrent,%r10
        movq $-ENOSYS,RAX(%rsp)
        FIXUP_TOP_OF_STACK %rdi
        movq %rsp,%rdi
@@ -272,6 +274,7 @@ tracesys:
        cmpq $__NR_syscall_max,%rax
        ja 1f
        movq %r10,%rcx /* fixup for C */
+       movq    %gs:pda_pcurrent,%r10
        call *sys_call_table(,%rax,8)
        movq %rax,RAX-ARGOFFSET(%rsp)
1:      SAVE_REST
@@ -495,6 +498,7 @@ ENTRY(stub_rt_sigreturn)
        swapgs
1:      incl    %gs:pda_irqcount          # RED-PEN should check preempt count
        movq %gs:pda_irqstackptr,%rax
+       movq    %gs:pda_pcurrent,%r10
        cmoveq %rax,%rsp /*todo This needs CFI annotation! */
        pushq %rdi          # save old stack
        CFI_ADJUST_CFA_OFFSET 8
@@ -684,6 +688,7 @@ ENTRY(spurious_interrupt)
        swapgs
        xorl %ebx,%ebx
1:      movq %rsp,%rdi
+       movq    %gs:pda_pcurrent,%r10
        movq ORIG_RAX(%rsp),%rsi
        movq $-1,ORIG_RAX(%rsp)
        call \sym
@@ -735,6 +740,7 @@ ENTRY(error_entry)
error_swapgs:
        swapgs
error_sti:
+       movq    %gs:pda_pcurrent,%r10
        movq %rdi,RDI(%rsp)
        movq %rsp,%rdi
        movq ORIG_RAX(%rsp),%rsi          /* get error code */
@@ -876,6 +882,7 @@ ENTRY(execve)
        CFI_STARTPROC
        FAKE_STACK_FRAME $0
        SAVE_ALL
+       movq    %gs:pda_pcurrent,%r10
        call sys_execve
        movq %rax, RAX(%rsp)
        RESTORE_REST
diff --git a/arch/x86_64/kernel/process.c b/arch/x86_64/kernel/process.c
index 28ebe45..5e4db7a 100644
--- a/arch/x86_64/kernel/process.c
+++ b/arch/x86_64/kernel/process.c
@@ -428,8 +428,10 @@ int copy_thread(int nr, unsigned long cl

        childregs->rax = 0;
        childregs->rsp = rsp;
-       if (rsp == ~0UL)
+       if (rsp == ~0UL) {
                childregs->rsp = (unsigned long)childregs;

```

## Linux-Kernel: [PATCH 8/9] x86-64 use r10 for current

```

+         childregs->r10 = (unsigned long)p;
+     }

    p->thread.rsp = (unsigned long) childregs;
    p->thread.rsp0 = (unsigned long) (childregs+1);
@@ -472,6 +474,7 @@ int copy_thread(int nr, unsigned long cl
    out:
        if (err && p->thread.io_bitmap_ptr) {
            kfree(p->thread.io_bitmap_ptr);
+           p->thread.io_bitmap_ptr = NULL;
            p->thread.io_bitmap_max = 0;
        }
        return err;
@@ -561,6 +564,7 @@ __switch_to(struct task_struct *prev_p,
    prev->user_rsp = read_pda(oldrsp);
    write_pda(oldrsp, next->user_rsp);
    write_pda(pcurrent, next_p);
+   current = next_p;
    write_pda(kernelstack,
        (unsigned long)next_p->thread_info + THREAD_SIZE - PDA_STACKOFFSET);

diff --git a/arch/x86_64/kernel/setup64.c b/arch/x86_64/kernel/setup64.c
index 3e81a04..3079869 100644
--- a/arch/x86_64/kernel/setup64.c
+++ b/arch/x86_64/kernel/setup64.c
@@ -123,6 +123,13 @@ void pda_init(int cpu)
    asm volatile("movl %0,%%fs ; movl %0,%%gs" :: "r" (0));
    wrmsrl(MSR_GS_BASE, cpu_pda + cpu);

+   if (cpu == 0) {
+       /* others are initialized in smpboot.c */
+       pda->pcurrent = &init_task;
+       pda->irqstackptr = boot_cpu_stack;
+   }

    current = pda->pcurrent;
    pda->cpunumber = cpu;
    pda->irqcount = -1;
    pda->kernelstack =
@@ -130,18 +137,13 @@ void pda_init(int cpu)
    pda->active_mm = &init_mm;
    pda->mmu_state = 0;

-   if (cpu == 0) {
-       /* others are initialized in smpboot.c */
-       pda->pcurrent = &init_task;
-       pda->irqstackptr = boot_cpu_stack;
-   } else {
+   if (cpu != 0) {
        pda->irqstackptr = (char *)
            __get_free_pages(GFP_ATOMIC, IRQSTACK_ORDER);
        if (!pda->irqstackptr)
            panic("cannot allocate irqstack for cpu %d", cpu);
    }

-
    pda->irqstackptr += IRQSTACKSIZE-64;
}

diff --git a/include/asm-x86_64/current.h b/include/asm-x86_64/current.h
index bc8adec..6675f2d 100644
--- a/include/asm-x86_64/current.h

```

## Linux-Kernel: [PATCH 8/9] x86-64 use r10 for current

```
+++ b/include/asm-x86_64/current.h
@@ -6,13 +6,7 @@ struct task_struct;

#include <asm/pda.h>

-static inline struct task_struct *get_current(void)
-{
-    struct task_struct *t = read_pda(pcurrent);
-    return t;
-}
-
-#define current get_current()
+register struct task_struct *current __asm__("%r10");

#else

---
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/
```