

[PATCH 7/9] x86-64 move thread_info into task_struct

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-11/9628.html>

From: Benjamin LaHaise (*bcr1_at_kvack.org*)

Date: 11/30/05

Date: Tue, 29 Nov 2005 23:22:05 -0500

To: Andi Kleen <ak@suse.de>

On x86-64, move thread_info from the stack into task_struct. This has benefits for the use of registers in entry.S when current is moved into a register. Take the easy approach of making GET_THREAD_INFO() return a pointer to current and make the asm-offset.c aware of this new usage.

```
arch/i386/oprofile/nmi_int.c      |    1 +
arch/x86_64/kernel/asm-offsets.c  |    2 +-
arch/x86_64/kernel/genapic_cluster.c |    1 +
arch/x86_64/kernel/genapic_flat.c |    1 +
arch/x86_64/kernel/setup64.c     |    2 +-
include/asm-x86_64/desc.h         |    1 +
include/asm-x86_64/processor.h    |   10 +++-----
include/asm-x86_64/system.h       |    6 +++---
include/asm-x86_64/thread_info.h  |   31 ++++++++-----
```

9 files changed, 25 insertions(+), 30 deletions(-)

applies-to: 747a43be5747e1c8e25f5769bdb9e4a1b8029138

bb489ebel733165426cbbaba24a2ed51d6952d88

diff --git a/arch/i386/oprofile/nmi_int.c b/arch/i386/oprofile/nmi_int.c

index 0493e8b..1e91d22 100644

--- a/arch/i386/oprofile/nmi_int.c

+++ b/arch/i386/oprofile/nmi_int.c

@@ -13,6 +13,7 @@

#include <linux/oprofile.h>

#include <linux/sysdev.h>

#include <linux/slab.h>

+#include <linux/sched.h>

#include <asm/nmi.h>

#include <asm/msr.h>

#include <asm/apic.h>

diff --git a/arch/x86_64/kernel/asm-offsets.c b/arch/x86_64/kernel/asm-offsets.c

index aaa6d38..66ebe60 100644

--- a/arch/x86_64/kernel/asm-offsets.c

+++ b/arch/x86_64/kernel/asm-offsets.c

@@ -29,7 +29,7 @@ int main(void)

ENTRY(pid);

BLANK();

#undef ENTRY

+#define ENTRY(entry) DEFINE(threadinfo_ ## entry, offsetof(struct thread_info, entry))

+#define ENTRY(entry) DEFINE(threadinfo_ ## entry, offsetof(struct task_struct, thread.info.entry

ENTRY(flags);

ENTRY(addr_limit);

ENTRY(preempt_count);

Linux-Kernel: [PATCH 7/9] x86-64 move thread_info into task_struct

```
diff --git a/arch/x86_64/kernel/genapic_cluster.c b/arch/x86_64/kernel/genapic_cluster.c
index a472d62..42531a9 100644
--- a/arch/x86_64/kernel/genapic_cluster.c
+++ b/arch/x86_64/kernel/genapic_cluster.c
@@ -16,6 +16,7 @@
#include <linux/kernel.h>
#include <linux/ctype.h>
#include <linux/init.h>
+#include <linux/sched.h>
#include <asm/smp.h>
#include <asm/ipi.h>

diff --git a/arch/x86_64/kernel/genapic_flat.c b/arch/x86_64/kernel/genapic_flat.c
index 9da3edb..28b775f 100644
--- a/arch/x86_64/kernel/genapic_flat.c
+++ b/arch/x86_64/kernel/genapic_flat.c
@@ -15,6 +15,7 @@
#include <linux/kernel.h>
#include <linux/ctype.h>
#include <linux/init.h>
+#include <linux/sched.h>
#include <asm/smp.h>
#include <asm/ipi.h>

diff --git a/arch/x86_64/kernel/setup64.c b/arch/x86_64/kernel/setup64.c
index 06dc354..3e81a04 100644
--- a/arch/x86_64/kernel/setup64.c
+++ b/arch/x86_64/kernel/setup64.c
@@ -126,7 +126,7 @@ void pda_init(int cpu)
    pda->cpunumber = cpu;
    pda->irqcount = -1;
    pda->kernelstack =
-        (unsigned long)stack_thread_info() - PDA_STACKOFFSET + THREAD_SIZE;
+        (unsigned long)current->thread_info - PDA_STACKOFFSET + THREAD_SIZE;
    pda->active_mm = &init_mm;
    pda->mmu_state = 0;

diff --git a/include/asm-x86_64/desc.h b/include/asm-x86_64/desc.h
index 3376486..ece54d4 100644
--- a/include/asm-x86_64/desc.h
+++ b/include/asm-x86_64/desc.h
@@ -9,6 +9,7 @@
#include <linux/string.h>
#include <linux/smp.h>
+#include <linux/sched.h>

#include <asm/segment.h>
#include <asm/mmu.h>

diff --git a/include/asm-x86_64/processor.h b/include/asm-x86_64/processor.h
index 4861246..7f24beb 100644
--- a/include/asm-x86_64/processor.h
+++ b/include/asm-x86_64/processor.h
@@ -20,6 +20,7 @@
#include <asm/mmsegment.h>
#include <asm/percpu.h>
#include <linux/personality.h>
+#include <linux/thread_info.h>

#define TF_MASK                0x00000100
#define IF_MASK                0x00000200
@@ -230,6 +231,7 @@ DECLARE_PER_CPU(struct tss_struct, init_t
```

Linux-Kernel: [PATCH 7/9] x86-64 move thread_info into task_struct

```

#define ARCH_MIN_TASKALIGN      16

struct thread_struct {
+   struct thread_info info;
   unsigned long  rsp0;
   unsigned long  rsp;
   unsigned long  userrsp;          /* Copy from PDA */
@@ -257,6 +259,7 @@ struct thread_struct {
} __attribute__((aligned(16)));

#define INIT_THREAD { \
+   .info = INIT_THREAD_INFO(init_task), \
   .rsp0 = (unsigned long)&init_stack + sizeof(init_stack) \
}

@@ -467,13 +470,6 @@ static inline void __mwait(unsigned long
: : "a" (eax), "c" (ecx));
}

-#define stack_current() \
-({
-   struct thread_info *ti;
-   asm("andq %%rsp,%0; : "=r" (ti) : "0" (CURRENT_MASK)); \
-   ti->task;
-})

#define cache_line_size() (boot_cpu_data.x86_cache_alignment)

extern unsigned long boot_option_idle_override;
diff --git a/include/asm-x86_64/system.h b/include/asm-x86_64/system.h
index 85348e0..d2cbbc3 100644
--- a/include/asm-x86_64/system.h
+++ b/include/asm-x86_64/system.h
@@ -34,17 +34,15 @@
        ".globl thread_return\n"
        "thread_return:\n\t"
        "movq %%gs:%P[pda_pcurrent],%rsi\n\t"
-       "movq %P[thread_info](%rsi),%r8\n\t"
-       LOCK "btr  %[tif_fork],%P[ti_flags](%r8)\n\t"
+       LOCK "btr  %[tif_fork],%P[ti_flags](%rsi)\n\t"
        "movq %%rax,%rdi\n\t"
        "jc   ret_from_fork\n\t"
        RESTORE_CONTEXT
        : "=a" (last)
        : [next] "S" (next), [prev] "D" (prev),
          [threadrsp] "i" (offsetof(struct task_struct, thread.rsp)), \
-         [ti_flags] "i" (offsetof(struct thread_info, flags)), \
+         [ti_flags] "i" (offsetof(struct task_struct, thread.info.flags)), \
          [tif_fork] "i" (TIF_FORK),
-         [thread_info] "i" (offsetof(struct task_struct, thread_info)), \
          [pda_pcurrent] "i" (offsetof(struct x8664_pda, pcurrent)) \
        : "memory", "cc" __EXTRA_CLOBBER)

diff --git a/include/asm-x86_64/thread_info.h b/include/asm-x86_64/thread_info.h
index 08eb6e4..0c90a18 100644
--- a/include/asm-x86_64/thread_info.h
+++ b/include/asm-x86_64/thread_info.h
@@ -57,20 +57,16 @@ struct thread_info {
#define init_thread_info      (init_thread_union.thread_info)
#define init_stack            (init_thread_union.stack)

-static inline struct thread_info *current_thread_info(void)

```

Linux-Kernel: [PATCH 7/9] x86-64 move thread_info into task_struct

```
-{
-   struct thread_info *ti;
-   ti = (void *) (read_pda(kernelstack) + PDA_STACKOFFSET - THREAD_SIZE);
-   return ti;
-}
-
-/* do not use in interrupt context */
-static inline struct thread_info *stack_thread_info(void)
-{
-   struct thread_info *ti;
-   __asm__("andq %%rsp,%0; : "=r" (ti) : "0" (~(THREAD_SIZE - 1)));
-   return ti;
-}
+#define task_thread_info(t) (&(t)->thread.info)
+#define current_thread_info() task_thread_info(current)
+
+#define setup_thread_stack(p, org) ({
+   task_thread_info(p)->task = (p);
+})
+
+#define end_of_stack(p) ((unsigned long *) (p)->thread.info + 1)
+
+#define __HAVE_THREAD_FUNCTIONS
+
+/* thread information allocation */
+#define alloc_thread_info(tsk) \
@@ -81,10 +77,11 @@ static inline struct thread_info *stack_
+
+/* else */
+
+/* how to get the thread information struct from ASM */
+/* How to get the thread information struct from ASM. We use a pointer to
+ * current and make the asm offsets point to * ->thread.info.<field>
+ */
+#define GET_THREAD_INFO(reg) \
-   movq %gs:pda_kernelstack,reg ; \
-   subq $(THREAD_SIZE-PDA_STACKOFFSET),reg
+   movq %gs:pda_pcurrent,reg
+
+#endif
+
+---
+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/
```