

[2.6 patch] i386: always use 4k stacks

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-11/9710.html>

From: Adrian Bunk (*bunk_at_stusta.de*)

Date: 11/30/05

Date: Wed, 30 Nov 2005 14:16:24 +0100

To: Andrew Morton <akpm@osdl.org>

It seems most problems with 4k stacks are already resolved at least in -mm.

I'd like to see this patch to always use 4k stacks in -mm now for finding any remaining problems before submitting this patch for 2.6.16.

Signed-off-by: Adrian Bunk <bunk@stusta.de>

Acked-by: Arjan van de Ven <arjan@infradead.org>

This patch was already sent on:

- 23 Nov 2005

- 14 Nov 2005

```
arch/i386/Kconfig.debug      | 10 -----
arch/i386/kernel/irq.c       | 10 -----
include/asm-i386/irq.h       | 11 +++-----
include/asm-i386/module.h    |  8 +-----
include/asm-i386/thread_info.h |  6 +-----
```

5 files changed, 5 insertions(+), 40 deletions(-)

--- linux-2.6.14-mm2-full/arch/i386/Kconfig.debug.old 2005-11-14 01:30:54.000000000 +0100

+++ linux-2.6.14-mm2-full/arch/i386/Kconfig.debug 2005-11-14 01:31:06.000000000 +0100

@@ -52,16 +52,6 @@

portion of the kernel code won't be covered by a 2MB TLB anymore.
If in doubt, say "N".

-config 4KSTACKS

- bool "Use 4Kb for kernel stacks instead of 8Kb"

- depends on DEBUG_KERNEL

- help

- If you say Y here the kernel will use a 4Kb stacksize for the
- kernel stack attached to each process/thread. This facilitates
- running more threads on a system and also reduces the pressure
- on the VM subsystem for higher order allocations. This option
- will also use IRQ stacks to compensate for the reduced stackspace.

- config X86_FIND_SMP_CONFIG

bool

depends on X86_LOCAL_APIC || X86_VOYAGER

--- linux-2.6.14-mm2-full/include/asm-i386/irq.h.old 2005-11-14 01:31:18.000000000 +0100

+++ linux-2.6.14-mm2-full/include/asm-i386/irq.h 2005-11-14 01:31:29.000000000 +0100

@@ -27,14 +27,9 @@

```
# define ARCH_HAS_NMI_WATCHDOG /* See include/linux/nmi.h */
#endif
```

Linux-Kernel: [2.6 patch] i386: always use 4k stacks

```

-#ifdef CONFIG_4KSTACKS
- extern void irq_ctx_init(int cpu);
- extern void irq_ctx_exit(int cpu);
-# define __ARCH_HAS_DO_SOFTIRQ
-#else
-# define irq_ctx_init(cpu) do { } while (0)
-# define irq_ctx_exit(cpu) do { } while (0)
-#endif
+extern void irq_ctx_init(int cpu);
+extern void irq_ctx_exit(int cpu);
+#define __ARCH_HAS_DO_SOFTIRQ

#ifdef CONFIG_IRQBALANCE
extern int irqbalance_disable(char *str);
--- linux-2.6.14-mm2-full/include/asm-i386/thread_info.h.old      2005-11-14 01:31:45.000000000 +0100
+++ linux-2.6.14-mm2-full/include/asm-i386/thread_info.h          2005-11-14 01:32:11.000000000 +0100
@@ -53,11 +53,7 @@
#endif

#define PREEMPT_ACTIVE          0x10000000
-#ifdef CONFIG_4KSTACKS
-#define THREAD_SIZE            (4096)
-#else
-#define THREAD_SIZE            (8192)
-#endif
+#define THREAD_SIZE            (4096)

#define STACK_WARN              (THREAD_SIZE/8)
/*
--- linux-2.6.14-mm2-full/include/asm-i386/module.h.old          2005-11-14 01:32:18.000000000 +0100
+++ linux-2.6.14-mm2-full/include/asm-i386/module.h              2005-11-14 01:32:42.000000000 +0100
@@ -64,12 +64,6 @@
#define MODULE_REGPARM " "
#endif

-#ifdef CONFIG_4KSTACKS
-#define MODULE_STACKSIZE "4KSTACKS "
-#else
-#define MODULE_STACKSIZE " "
-#endif
-
-#define MODULE_ARCH_VERMAGIC MODULE_PROC_FAMILY MODULE_REGPARM MODULE_STACKSIZE
+#define MODULE_ARCH_VERMAGIC MODULE_PROC_FAMILY MODULE_REGPARM

#endif /* _ASM_I386_MODULE_H */
--- linux-2.6.14-mm2-full/arch/i386/kernel/irq.c.old              2005-11-14 01:32:56.000000000 +0100
+++ linux-2.6.14-mm2-full/arch/i386/kernel/irq.c                  2005-11-14 01:33:28.000000000 +0100
@@ -33,7 +33,6 @@
}
#endif

-#ifdef CONFIG_4KSTACKS
/*
* per-CPU IRQ handling contexts (thread information and stack)
*/
@@ -44,7 +43,6 @@

static union irq_ctx *hardirq_ctx[NR_CPUS];
static union irq_ctx *softirq_ctx[NR_CPUS];
-#endif

/*
```

Linux-Kernel: [2.6 patch] i386: always use 4k stacks

```
* do_IRQ handles all normal device IRQ's (the special
@@ -58,10 +56,8 @@
#ifdef CONFIG_DEBUG_PREEMPT
    u32 count = preempt_count() & PREEMPT_MASK;
#endif
-#ifndef CONFIG_4KSTACKS
    union irq_ctx *curctx, *irqctx;
    u32 *isp;
-#endif

    irq_enter();
#ifdef CONFIG_DEBUG_STACKOVERFLOW
@@ -79,8 +75,6 @@
    }
#endif

-#ifndef CONFIG_4KSTACKS
-
    curctx = (union irq_ctx *) current_thread_info();
    irqctx = hardirq_ctx[smp_processor_id()];

@@ -118,7 +112,6 @@
        irqctx->tinfo.preempt_count -= count;
    #endif
    } else
-#endif
        __do_IRQ(irq, regs);

    irq_exit();
@@ -128,8 +121,6 @@
    return 1;
}

-#ifndef CONFIG_4KSTACKS
-
/*
 * These should really be __section__(".bss.page_aligned") as well, but
 * gcc's 3.0 and earlier don't handle that correctly.
@@ -228,7 +219,6 @@
}

EXPORT_SYMBOL(do_softirq);
-#endif

/*
 * Interrupt statistics:
-
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/
```