

## Re: Announce: kdb v4.4 x86-64 updates for for kernel 2.6.6

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-06/2068.html>

---

**From:** Andi Kleen ([ak\\_at\\_muc.de](mailto:ak_at_muc.de))

**Date:** 06/09/04

To: Keith Owens <[kaos@sgi.com](mailto:kaos@sgi.com)>, [jfv@bluesong.net](mailto:jfv@bluesong.net), [linux-kernel@vger.kernel.org](mailto:linux-kernel@vger.kernel.org)

Date: Wed, 09 Jun 2004 11:37:18 +0200

Keith Owens <[kaos@sgi.com](mailto:kaos@sgi.com)> writes:

> *KDB (Linux Kernel Debugger) has been updated.*  
>  
> <ftp://oss.sgi.com/projects/kdb/download/v4.4/>  
>  
> *kdb-v4.4-2.6.6-x86-64-2.bz2 is available. The x86-64 patch is still a*  
> *work in progress, use with care. Changelog extract.*  
>  
> *2004-05-15 Jack F. Vogel <[jfv@bluesong.net](mailto:jfv@bluesong.net)>*  
> *\* port to 2.6.6 for x86\_64*

It uses the wrong interfaces. The x86-64 die notifier was exactly designed to avoid putting KDB hooks all over the kernel, but you added them anyways. Please fix that.

In theory with the die hooks interface kdb could be a loadable (although not unloadable) module btw.

Some comments:

```
+#if defined(CONFIG_SMP) && defined(CONFIG_KDB)
+static void do_ack_apic_irq(void)
+{
+ ack_APIC_irq();
+}
+#endif
```

NMIs don't need to be ACKed in the x86 APIC. The only reason to ack something is that a higher priority interrupt can run, but there is no higher priority interrupt than an NMI.

```
+#if defined(CONFIG_SMP) && defined(CONFIG_KDB)
+ /*
+ * Call the debugger to see if this NMI is due
```

```
+ * to a KDB requested IPI. If so, it will handle
+ */
+ if (kdb_ipi(regs, do_ack_apic_irq)){
+ nmi_exit();
+ return;
+ }
+ #endif
```

Please just grab DIE\_NMI\_IPI in the die chain, which you're already using. The NMI handler can veto the NMI then using NOTIFY\_BAD. The ack is also not needed.

```
@@ -367,6 +376,10 @@
    handle_BUG(regs);
    __die(str, regs, err);
    oops_end();
+ #ifdef CONFIG_KDB
+ kdb_diemsg = str;
+ kdb(KDB_REASON_OOPS, err, regs);
+ #endif /* CONFIG_KDB */
```

\_\_die already has a die chain hook for that. Use that instead.

```
+ #if !defined(CONFIG_KDB)
    DO_ERROR( 3, SIGTRAP, "int3", int3);
+ #endif
```

and

```
+ #ifdef CONFIG_KDB
+ /*
+ * KDB Breakpoint vector
+ */
+ asmlinkage int do_int3(struct pt_regs * regs, long error_code)
+ {
+ if (kdb(KDB_REASON_BREAK, error_code, regs))
+ return 0;
+ do_trap(3, SIGTRAP, "int3", regs, error_code, NULL);
+ return 0;
+ }
```

Unnecessary when you have a die chain handler (which you have already). Take a look at how DO\_ERROR is defined.

```
- set_intr_gate_ist(18, &machine_check, MCE_STACK);
+ #ifdef CONFIG_KDB
+ {
+ set_intr_gate(18, &machine_check);
+ }
+ kdb_enablehwfault();
```

and

```
+void
+kdba_enable_mce(void)
+{
+ /* No longer required, arch/x86_64/kernel/bluesmoke.c does the job now */
+ /
+ }
```

kdb\_enablehwfault only calls kdba\_enable\_mce and is a complete nop. Can be just removed.

Overall about 90% of your changes to arch/x86\_64/kernel/traps.c are unnecessary and the rest could be easily moved into arch/x86\_64/kdb, giving an completely independent and low maintainance kdb.

-Andi

-

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/>