
Subject: [PATCH] Consolidate PTRACE_DETACH
Posted by [Alexey Dobriyan](#) on Thu, 19 Jul 2007 12:38:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Identical handlers of PTRACE_DETACH go into ptrace_request().
Not touching compat code.
Not touching archs that don't call ptrace_request.

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>

```
arch/alpha/kernel/ptrace.c      | 4 ----
arch/arm/kernel/ptrace.c        | 4 ----
arch/arm26/kernel/ptrace.c      | 4 ----
arch/avr32/kernel/ptrace.c      | 5 -----
arch/blackfin/kernel/ptrace.c   | 6 -----
arch/cris/arch-v10/kernel/ptrace.c | 4 ----
arch/cris/arch-v32/kernel/ptrace.c | 4 ----
arch/i386/kernel/ptrace.c       | 5 -----
arch/m32r/kernel/ptrace.c       | 8 -----
arch/m68k/kernel/ptrace.c       | 4 ----
arch/mips/kernel/ptrace.c       | 4 ----
arch/parisc/kernel/ptrace.c     | 4 ----
arch/powerpc/kernel/ptrace.c    | 4 ----
arch/s390/kernel/ptrace.c       | 5 -----
arch/sh/kernel/ptrace.c         | 4 ----
arch/sh64/kernel/ptrace.c       | 4 ----
arch/um/kernel/ptrace.c         | 5 -----
arch/x86_64/kernel/ptrace.c     | 5 -----
arch/xtensa/kernel/ptrace.c     | 4 ----
kernel/ptrace.c                 | 3 +++
20 files changed, 3 insertions(+), 87 deletions(-)
```

--- a/arch/alpha/kernel/ptrace.c

+++ b/arch/alpha/kernel/ptrace.c

```
@ @ -368,10 +368,6 @ @ do_sys_ptrace(long request, long pid, long addr, long data,
    ret = 0;
    goto out;
```

```
- case PTRACE_DETACH: /* detach a process that was attached. */
```

```
- ret = ptrace_detach(child, data);
```

```
- goto out;
```

```
-
```

```
default:
```

```
ret = ptrace_request(child, request, addr, data);
```

```
goto out;
```

--- a/arch/arm/kernel/ptrace.c

+++ b/arch/arm/kernel/ptrace.c

```

@@ -731,10 +731,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = 0;
    break;

- case PTRACE_DETACH:
- ret = ptrace_detach(child, data);
- break;
-
case PTRACE_GETREGS:
    ret = ptrace_getregs(child, (void __user *)data);
    break;
--- a/arch/arm26/kernel/ptrace.c
+++ b/arch/arm26/kernel/ptrace.c
@@ -609,10 +609,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = 0;
    break;

- case PTRACE_DETACH:
- ret = ptrace_detach(child, data);
- break;
-
case PTRACE_GETREGS:
    ret = ptrace_getregs(child, (void *)data);
    break;
--- a/arch/avr32/kernel/ptrace.c
+++ b/arch/avr32/kernel/ptrace.c
@@ -227,11 +227,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = 0;
    break;

- /* Detach a process that was attached */
- case PTRACE_DETACH:
- ret = ptrace_detach(child, data);
- break;
-
case PTRACE_GETREGS:
    ret = ptrace_getregs(child, (void __user *)data);
    break;
--- a/arch/blackfin/kernel/ptrace.c
+++ b/arch/blackfin/kernel/ptrace.c
@@ -373,12 +373,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    break;
}

```

```

- case PTRACE_DETACH:
- { /* detach a process that was attached. */
-   ret = ptrace_detach(child, data);
-   break;
- }
-
case PTRACE_GETREGS:
{

--- a/arch/cris/arch-v10/kernel/ptrace.c
+++ b/arch/cris/arch-v10/kernel/ptrace.c
@@ -177,10 +177,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = 0;
    break;

- case PTRACE_DETACH:
-   ret = ptrace_detach(child, data);
-   break;
-
    /* Get all GP registers from the child. */
    case PTRACE_GETREGS: {
        int i;
--- a/arch/cris/arch-v32/kernel/ptrace.c
+++ b/arch/cris/arch-v32/kernel/ptrace.c
@@ -245,10 +245,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    break;

    }
- case PTRACE_DETACH:
-   ret = ptrace_detach(child, data);
-   break;
-
    /* Get all GP registers from the child. */
    case PTRACE_GETREGS: {
        int i;
--- a/arch/i386/kernel/ptrace.c
+++ b/arch/i386/kernel/ptrace.c
@@ -526,11 +526,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = 0;
    break;

- case PTRACE_DETACH:
-   /* detach a process that was attached. */
-   ret = ptrace_detach(child, data);
-   break;

```

```

-
case PTRACE_GETREGS: { /* Get all gp regs from the child. */
    if (!access_ok(VERIFY_WRITE, datap, FRAME_SIZE*sizeof(long))) {
        ret = -EIO;
--- a/arch/m32r/kernel/ptrace.c
+++ b/arch/m32r/kernel/ptrace.c
@@ -703,14 +703,6 @@ do_ptrace(long request, struct task_struct *child, long addr, long data)
    break;
}

- /*
-  * detach a process that was attached.
-  */
- case PTRACE_DETACH:
-     ret = 0;
-     ret = ptrace_detach(child, data);
-     break;
-
case PTRACE_GETREGS:
    ret = ptrace_getregs(child, (void __user *)data);
    break;
--- a/arch/m68k/kernel/ptrace.c
+++ b/arch/m68k/kernel/ptrace.c
@@ -227,10 +227,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    wake_up_process(child);
    break;

- case PTRACE_DETACH: /* detach a process that was attached. */
-     ret = ptrace_detach(child, data);
-     break;
-
case PTRACE_GETREGS: /* Get all gp regs from the child. */
    for (i = 0; i < 19; i++) {
        tmp = get_reg(child, i);
--- a/arch/mips/kernel/ptrace.c
+++ b/arch/mips/kernel/ptrace.c
@@ -435,10 +435,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    wake_up_process(child);
    break;

- case PTRACE_DETACH: /* detach a process that was attached. */
-     ret = ptrace_detach(child, data);
-     break;
-
case PTRACE_GET_THREAD_AREA:
    ret = put_user(task_thread_info(child)->tp_value,

```

```

    (unsigned long __user *) data);
--- a/arch/parisc/kernel/ptrace.c
+++ b/arch/parisc/kernel/ptrace.c
@@ -329,10 +329,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    /* give it a chance to run. */
    goto out_wake;

- case PTRACE_DETACH:
-   ret = ptrace_detach(child, data);
-   goto out_tsk;
-
case PTRACE_GETEVENTMSG:
    ret = put_user(child->ptrace_message, (unsigned int __user *) data);
    goto out_tsk;
--- a/arch/powerpc/kernel/ptrace.c
+++ b/arch/powerpc/kernel/ptrace.c
@@ -503,10 +503,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = ptrace_set_debugreg(child, addr, data);
    break;

- case PTRACE_DETACH:
-   ret = ptrace_detach(child, data);
-   break;
-
#ifdef CONFIG_PPC64
case PTRACE_GETREGS64:
#endif
--- a/arch/s390/kernel/ptrace.c
+++ b/arch/s390/kernel/ptrace.c
@@ -683,11 +683,6 @@ do_ptrace(struct task_struct *child, long request, long addr, long data)
    wake_up_process(child);
    return 0;

- case PTRACE_DETACH:
-   /* detach a process that was attached. */
-   return ptrace_detach(child, data);
-
-
    /* Do requests that differ for 31/64 bit */
    default:
#ifdef CONFIG_COMPAT
--- a/arch/sh/kernel/ptrace.c
+++ b/arch/sh/kernel/ptrace.c
@@ -210,10 +210,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    break;

```

```

}

- case PTRACE_DETACH: /* detach a process that was attached. */
- ret = ptrace_detach(child, data);
- break;
-
#ifdef CONFIG_SH_DSP
case PTRACE_GETDSPREGS: {
    unsigned long dp;
--- a/arch/sh64/kernel/ptrace.c
+++ b/arch/sh64/kernel/ptrace.c
@@ -244,10 +244,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    break;
}

- case PTRACE_DETACH: /* detach a process that was attached. */
- ret = ptrace_detach(child, data);
- break;
-
default:
    ret = ptrace_request(child, request, addr, data);
    break;
--- a/arch/um/kernel/ptrace.c
+++ b/arch/um/kernel/ptrace.c
@@ -119,11 +119,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    break;
}

- case PTRACE_DETACH:
- /* detach a process that was attached. */
- ret = ptrace_detach(child, data);
- break;
-
#ifdef PTRACE_GETREGS
case PTRACE_GETREGS: /* Get all gp regs from the child. */
    if (!access_ok(VERIFY_WRITE, p, MAX_REG_OFFSET)) {
--- a/arch/x86_64/kernel/ptrace.c
+++ b/arch/x86_64/kernel/ptrace.c
@@ -505,11 +505,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = 0;
    break;

- case PTRACE_DETACH:
- /* detach a process that was attached. */
- ret = ptrace_detach(child, data);

```

```

- break;
-
case PTRACE_GETREGS: { /* Get all gp regs from the child. */
    if (!access_ok(VERIFY_WRITE, (unsigned __user *)data,
        sizeof(struct user_regs_struct))) {
--- a/arch/xtensa/kernel/ptrace.c
+++ b/arch/xtensa/kernel/ptrace.c
@@ -304,10 +304,6 @@ long arch_ptrace(struct task_struct *child, long request, long addr, long
data)
    ret = put_user(sizeof(elf_fpregset_t), (unsigned long *) data);
    break;

- case PTRACE_DETACH: /* detach a process that was attached. */
- ret = ptrace_detach(child, data);
- break;
-
default:
    ret = ptrace_request(child, request, addr, data);
    goto out;
--- a/kernel/ptrace.c
+++ b/kernel/ptrace.c
@@ -385,6 +385,9 @@ int ptrace_request(struct task_struct *child, long request,
case PTRACE_SETSIGINFO:
    ret = ptrace_setsiginfo(child, (siginfo_t __user *) data);
    break;
+ case PTRACE_DETACH: /* detach a process that was attached. */
+ ret = ptrace_detach(child, data);
+ break;
default:
    break;
}

```

Subject: Re: [PATCH] Consolidate PTRACE_DETACH
Posted by [Christoph Hellwig](#) on Thu, 19 Jul 2007 14:02:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, Jul 19, 2007 at 04:38:01PM +0400, Alexey Dobriyan wrote:
> Identical handlers of PTRACE_DETACH go into ptrace_request().
> Not touching compat code.
> Not touching archs that don't call ptrace_request.

Looks very nice, thanks.
