

[PATCH 6/6] syslets: add both 32bit and 64bit x86 syslet support

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-12/msg02078.html>

- *From:* Zach Brown <zach.brown@xxxxxxxxxx>
 - *Date:* Thu, 6 Dec 2007 15:20:19 -0800
-

This adds the architecture-specific routines needed by syslets for x86.

The syslet thread creation routines create a new thread which executes a kernel function and then returns to userspace instead of exiting.

`move_user_context()` and `set_user_frame()` let the scheduler modify a child thread so that it returns to userspace at the same place that a blocking system call would have when it finished. This currently performs a very expensive copy of the fpu state. Intel is working on a more robust patch which allocates the i387 state off of `thread_struct`. When that is ready this can just juggle pointers to transfer the fpu state.

The syslets infrastructure needs to work with `ptregs` for the task which is in `sys_indirect()`. So we add a `PTREGSCALL` stub around `sys_indirect()` in `x86_64`.

Finally, we wire up `sys_syslet_ring_wait()`.

Signed-off-by: Zach Brown <zach.brown@xxxxxxxxxx>

```
diff --git a/arch/x86/kernel/entry_32.S b/arch/x86/kernel/entry_32.S
index dc7f938..66a121d 100644
--- a/arch/x86/kernel/entry_32.S
+++ b/arch/x86/kernel/entry_32.S
@@ -1025,6 +1025,30 @@ ENTRY(kernel_thread_helper)
CFI_ENDPROC
ENDPROC(kernel_thread_helper)

+ENTRY(syslet_thread_helper)
+ CFI_STARTPROC
+ /*
+ * Allocate space on the stack for pt-regs.
+ * sizeof(struct pt_regs) == 64, and we've got 8 bytes on the
+ * kernel stack already:
+ */
+ subl $64-8, %esp
+ CFI_ADJUST_CFA_OFFSET 64-8
+ movl %edx,%eax
```

[PATCH 6/6] syslets: add both 32bit and 64bit x86 syslet support

```
+ push %edx
+ CFI_ADJUST_CFA_OFFSET 4
+ call *%ebx
+ addl $4, %esp
+ CFI_ADJUST_CFA_OFFSET -4
+
+ movl %eax, PT_EAX(%esp)
+
+ GET_THREAD_INFO(%ebp)
+
+ jmp syscall_exit
+ CFI_ENDPROC
+ENDPROC(syslet_thread_helper)
+
#ifdef CONFIG_XEN
ENTRY(xen_hypervisor_callback)
CFI_STARTPROC
diff --git a/arch/x86/kernel/entry_64.S b/arch/x86/kernel/entry_64.S
index 3a058bb..08e34f6 100644
--- a/arch/x86/kernel/entry_64.S
+++ b/arch/x86/kernel/entry_64.S
@@ -412,6 +412,7 @@ END(\label)
PTREGSCALL stub_rt_s
```