

---

Subject: [PATCH] pid: Fix mips irix emulation pid usage  
Posted by [ebiederm](#) on Thu, 06 Dec 2007 05:34:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Signed-off-by: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)>

---

```
arch/mips/kernel/irixelf.c | 14 ++++++-----  
arch/mips/kernel/irixsig.c | 16 +++++++-----  
arch/mips/kernel/sysirix.c | 12 ++++++-----  
3 files changed, 23 insertions(+), 19 deletions(-)
```

```
diff --git a/arch/mips/kernel/irixelf.c b/arch/mips/kernel/irixelf.c  
index 7852c7c..290d8e3 100644  
--- a/arch/mips/kernel/irixelf.c  
+++ b/arch/mips/kernel/irixelf.c  
@@ -591,9 +591,9 @@ static void irix_map_prda_page(void)  
    return;  
  
    pp = (struct prda *) v;  
- pp->prda_sys.t_pid = current->pid;  
+ pp->prda_sys.t_pid = task_pid_vnr(current);  
    pp->prda_sys.t_prid = read_c0_prid();  
- pp->prda_sys.t_rpid = current->pid;  
+ pp->prda_sys.t_rpid = task_pid_vnr(current);  
  
    /* We leave the rest set to zero */  
}  
@@ -1170,11 +1170,11 @@ static int irix_core_dump(long signr, struct pt_regs *regs, struct file  
*file, u  
    prstatus.pr_info.si_signo = prstatus.pr_cursig = signr;  
    prstatus.pr_sigpend = current->pending.signal.sig[0];  
    prstatus.pr_sighold = current->blocked.sig[0];  
- psinfo.pr_pid = prstatus.pr_pid = current->pid;  
- psinfo.pr_ppid = prstatus.pr_ppid = current->parent->pid;  
- psinfo.pr_pgrp = prstatus.pr_pgrp = task_pgrp_nr(current);  
- psinfo.pr_sid = prstatus.pr_sid = task_session_nr(current);  
- if (current->pid == current->tgid) {  
+ psinfo.pr_pid = prstatus.pr_pid = task_pid_vnr(current);  
+ psinfo.pr_ppid = prstatus.pr_ppid = task_pid_vnr(current->parent);  
+ psinfo.pr_pgrp = prstatus.pr_pgrp = task_pgrp_vnr(current);  
+ psinfo.pr_sid = prstatus.pr_sid = task_session_vnr(current);  
+ if (thread_group_leader(current)) {  
/*  
 * This is the record for the group leader. Add in the  
 * cumulative times of previous dead threads. This total  
diff --git a/arch/mips/kernel/irixsig.c b/arch/mips/kernel/irixsig.c  
index 5b10ac1..0215c80 100644  
--- a/arch/mips/kernel/irixsig.c
```

```

+++ b/arch/mips/kernel/irixsig.c
@@ -578,10 +578,11 @@ out:

#define W_MASK    (W_EXITED | W_TRAPPED | W_STOPPED | W_CONT | W_NOHANG)

-asmlinkage int irix_waitsys(int type, int pid,
+asmlinkage int irix_waitsys(int type, int upid,
    struct irix5_siginfo __user *info, int options,
    struct rusage __user *ru)
{
+ struct pid *pid = NULL;
    int flag, retval;
    DECLARE_WAITQUEUE(wait, current);
    struct task_struct *tsk;
@@ -604,6 +605,8 @@ asmlinkage int irix_waitsys(int type, int pid,
    if (type != IRIX_P_PID && type != IRIX_P_PGID && type != IRIX_P_ALL)
        return -EINVAL;

+ if (type != IRIX_P_ALL)
+     pid = find_get_pid(upid);
    add_wait_queue(&current->signal->wait_chldexit, &wait);
repeat:
    flag = 0;
@@ -612,9 +615,9 @@ repeat:
    tsk = current;
    list_for_each(_p, &tsk->children) {
        p = list_entry(_p, struct task_struct, sibling);
- if ((type == IRIX_P_PID) && p->pid != pid)
+ if ((type == IRIX_P_PID) && task_pid(p) != pid)
        continue;
- if ((type == IRIX_P_PGID) && task_pgrp_nr(p) != pid)
+ if ((type == IRIX_P_PGID) && task_pgrp(p) != pid)
        continue;
        if ((p->exit_signal != SIGCHLD))
            continue;
@@ -639,7 +642,7 @@ repeat:

    retval = __put_user(SIGCHLD, &info->sig);
    retval |= __put_user(0, &info->code);
-    retval |= __put_user(p->pid, &info->stuff.procinfo.pid);
+    retval |= __put_user(task_pid_vnr(p), &info->stuff.procinfo.pid);
    retval |= __put_user((p->exit_code >> 8) & 0xff,
                        &info->stuff.procinfo.procdata.child.status);
    retval |= __put_user(p->utime, &info->stuff.procinfo.procdata.child.utime);
@@ -657,7 +660,7 @@ repeat:
    getrusage(p, RUSAGE_BOTH, ru);
    retval = __put_user(SIGCHLD, &info->sig);
    retval |= __put_user(1, &info->code); /* CLD_EXITED */

```

```

-    retval |= __put_user(p->pid, &info->stuff.procinfo.pid);
+    retval |= __put_user(task_pid_vnr(p), &info->stuff.procinfo.pid);
    retval |= __put_user((p->exit_code >> 8) & 0xff,
                         &info->stuff.procinfo.procdata.child.status);
    retval |= __put_user(p->utime,
@@ -665,7 +668,7 @@ repeat:
    retval |= __put_user(p->stime,
                         &info->stuff.procinfo.procdata.child.stime);
    if (retval)
-    return retval;
+    goto end_waitsys;

    if (p->real_parent != p->parent) {
        write_lock_irq(&tasklist_lock);
@@ -698,6 +701,7 @@ repeat:
end_waitsys:
    current->state = TASK_RUNNING;
    remove_wait_queue(&current->signal->wait_chldexit, &wait);
+ put_pid(pid);

    return retval;
}
diff --git a/arch/mips/kernel/sysirix.c b/arch/mips/kernel/sysirix.c
index 6f5446b..13f0f96 100644
--- a/arch/mips/kernel/sysirix.c
+++ b/arch/mips/kernel/sysirix.c
@@ -582,8 +582,8 @@ out:

asmlinkage int irix_getpid(struct pt_regs *regs)
{
- regs->regs[3] = current->real_parent->pid;
- return current->pid;
+ regs->regs[3] = task_pid_vnr(current->real_parent);
+ return task_pid_vnr(current);
}

asmlinkage int irix_getuid(struct pt_regs *regs)
@@ -763,11 +763,11 @@ asmlinkage int irix_setpgrp(int flags)
    printk("[%s:%d] setpgrp(%d)", current->comm, current->pid, flags);
#endif
    if(!flags)
-    error = task_pgrp_nr(current);
+    error = task_pgrp_vnr(current);
    else
        error = sys_setsid();
#ifdef DEBUG_PROCGRPS
-    printk("returning %d\n", task_pgrp_nr(current));
+    printk("returning %d\n", error);

```

```
#endif

    return error;
@@ -1093,10 +1093,10 @@ asmlinkage int irix_BSDsetpgrp(int pid, int pgrp)
    pid, pgrp);
#endif
if(!pid)
- pid = current->pid;
+ pid = task_pid_vnr(current);

/* Wheee, weird sysv thing... */
- if ((pgrp == 0) && (pid == current->pid))
+ if ((pgrp == 0) && (pid == task_pid_vnr(current)))
    error = sys_setsid();
else
    error = sys_setpgid(pid, pgrp);
--
```

1.5.3.rc6.17.g1911

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---