

[PATCH][Bug 5132] fix sys_poll() large timeout handling

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-08/8027.html>

From: Nishanth Aravamudan (*nacc_at_us.ibm.com*)

Date: 08/31/05

Date: Wed, 31 Aug 2005 13:01:09 -0700

To: akpm@osdl.org

Sorry everybody, forgot the most important Cc: :)

-Nish

Hi Andrew,

In looking at Bug 5132 and sys_poll(), I think there is a flaw in the current code.

The @timeout parameter to sys_poll() is in milliseconds but we compare it to (MAX_SCHEDULE_TIMEOUT / HZ), which is jiffies/jiffies-per-sec or seconds. That seems blatantly broken. Also, I think we are better served by converting to jiffies first then comparing, as opposed to converting our maximum to milliseconds (or seconds, incorrectly) and comparing.

Comments, suggestions for improvement?

Description: The current sys_poll() implementation does not seem to handle large timeouts correctly. Any value in milliseconds (@timeout) which exceeds the maximum representable jiffy value (MAX_SCHEDULE_TIMEOUT) should result in a MAX_SCHEDULE_TIMEOUT schedule_timeout() call. To achieve this, convert @timeout to jiffies first, then compare to MAX_SCHEDULE_TIMEOUT.

Signed-off-by: Nishanth Aravamudan <nacc@us.ibm.com>

```
---
 fs/select.c | 14 ++++++-----
 1 files changed, 9 insertions(+), 5 deletions(-)
diff -urpN 2.6.13/fs/select.c 2.6.13-dev/fs/select.c
--- 2.6.13/fs/select.c 2005-08-28 17:46:14.000000000 -0700
+++ 2.6.13-dev/fs/select.c 2005-08-31 12:43:52.000000000 -0700
@@ -470,12 +470,16 @@ asmlinkage long sys_poll(struct pollfd _
         return -EINVAL;

         if (timeout) {
-            /* Careful about overflow in the intermediate values */
-            if ((unsigned long) timeout < MAX_SCHEDULE_TIMEOUT / HZ)
```

Linux-Kernel: [PATCH][Bug 5132] fix sys_poll() large timeout handling

```
-             timeout = (unsigned long)(timeout*HZ+999)/1000+1;
-         else /* Negative or overflow */
-             timeout = MAX_SCHEDULE_TIMEOUT;
+         /*
+          * Convert the value from msecs to jiffies - if overflow
+          * occurs we get a negative value, which gets handled by
+          * the next block
+          */
+         timeout = msecs_to_jiffies(timeout) + 1;
    }
+     if (timeout < 0) /* Negative requests result in infinite timeouts */
+         timeout = MAX_SCHEDULE_TIMEOUT;
+     /* 0 case falls through */

    poll_initwait(&table);
```

-
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/>