

Re: [ANNOUNCE] System Inactivity Monitor v1.0

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-01/msg06951.html>

- *From:* Alessandro Di Marco <dmr@xxxxxx>
 - *Date:* Fri, 26 Jan 2007 18:55:53 +0100
-

Pavel Machek <pavel@xxxxxx> writes:

Well, I do not think your kernel code is mergeable. But bits to enable similar functionality in userspace probably would be mergeable.

You said it :-)

This patch exports to the user space the inactivity time (in msec) of a given input device. Example follows:

```
<0> $ cat /proc/bus/input/activity
0011 0001 0001 ab41 1
0011 0002 0008 0000 3160799
0011 0002 0008 7321 549991
0019 0000 0005 0000 3160799
0019 0000 0001 0000 3454901
0010 104d 0000 0000 3160799
0010 104d 0000 0000 2162833
```

The device ordering matches the /proc/bus/input/devices one, anyway I reported also vendor, product, etc. Now the daemon is trivial...

Bye

Ah, the interesting column is the fifth ;-)

```
diff -ur OLD/drivers/input/input.c NEW/drivers/input/input.c
--- OLD/drivers/input/input.c 2007-01-26 16:59:36.000000000 +0100
+++ NEW/drivers/input/input.c 2007-01-26 17:04:38.000000000 +0100
@@ -49,6 +49,8 @@
{
struct input_handle *handle;

+ dev->last_activity = jiffies;
+
if (type > EV_MAX || !test_bit(type, dev->evbit))
return;

@@ -482,6 +484,30 @@
```

Re: [ANNOUNCE] System Inactivity Monitor v1.0

```
return seq_open(file, &input_devices_seq_ops);
}

+static int input_activity_seq_show(struct seq_file *seq, void *v)
+{
+ struct input_dev *dev = container_of(v, struct input_dev, node);
+
+ seq_printf(seq, "%04x %04x %04x %04x\t%u\n",
+ dev->id.bustype, dev->id.vendor,
+ dev->id.product, dev->id.version,
+ jiffies_to_msecs((long) jiffies - (long) dev->last_activity));
+
+ return 0;
+}
+
+static struct seq_operations input_activity_seq_ops = {
+ .start = input_devices_seq_start,
+ .next = input_devices_seq_next,
+ .stop = input_devices_seq_stop,
+ .show = input_activity_seq_show,
+};
+
+static int input_proc_activity_open(struct inode *inode, struct file *file)
+{
+ return seq_open(file, &input_activity_seq_ops);
+}
+
+static struct file_operations input_devices_fileops = {
+ .owner = THIS_MODULE,
+ .open = input_proc_devices_open,
+ @@ -491,6 +517,15 @@
+ .release = seq_release,
+};
+
+static struct file_operations input_activity_fileops = {
+ .owner = THIS_MODULE,
+ .open = input_proc_activity_open,
+ .poll = input_proc_devices_poll,
+ .read = seq_read,
+ .llseek = seq_lseek,
+ .release = seq_release,
+};
+
+static void *input_handlers_seq_start(struct seq_file *seq, loff_t *pos)
+{
+ /* acquire lock here ... Yes, we do need locking, I knowi, I know... */
+ @@ -558,15 +593,23 @@
+ entry->owner = THIS_MODULE;
+ entry->proc_fops = &input_devices_fileops;
+
+ - entry = create_proc_entry("handlers", 0, proc_bus_input_dir);
```

Re: [ANNOUNCE] System Inactivity Monitor v1.0

```
+ entry = create_proc_entry("activity", 0, proc_bus_input_dir);
if (!entry)
goto fail2;

entry->owner = THIS_MODULE;
+ entry->proc_fops = &input_activity_fops;
+
+ entry = create_proc_entry("handlers", 0, proc_bus_input_dir);
+ if (!entry)
+ goto fail3;
+
+ entry->owner = THIS_MODULE;
entry->proc_fops = &input_handlers_fops;

return 0;

+ fail3: remove_proc_entry("activity", proc_bus_input_dir);
fail2: remove_proc_entry("devices", proc_bus_input_dir);
fail1: remove_proc_entry("input", proc_bus);
return -ENOMEM;
diff -ur OLD/include/linux/input.h NEW/include/linux/input.h
--- OLD/include/linux/input.h 2007-01-26 16:59:38.000000000 +0100
+++ NEW/include/linux/input.h 2007-01-26 17:31:29.000000000 +0100
@@ -949,6 +949,8 @@
const char *uniq;
struct input_id id;

+ unsigned long last_activity;
+
+ unsigned long evbit[NBITS(EV_MAX)];
+ unsigned long keybit[NBITS(KEY_MAX)];
+ unsigned long relbit[NBITS(REL_MAX)];
```

--
I don't think anyone should write their autobiography until after they're
dead. – Samuel Goldwyn