New proc entry under /proc/[pid]

lists.kernelnewbies.org/pipermail/kernelnewbies/2011-January/000475.html

Mauro Romano Trajber trajber at gmail.com
Wed Jan 12 13:58:14 EST 2011

Yes, /proc/[pid] handling is diferent and your solution is perfect just for the other proc directories.

Informations under [pid] directories are about tasks (processes), and most of this information is represented by task_struct structure (defined in include/linux/sched.h).

I think one easy way to solve this problem is:

In include/linux/sched.h file
1 - add a new element in task_struct structure.

```
struct task_struct {
int my_value;
        .......
```

In fs/proc/base.c file
1 - Create a new entry in static pid_entry tgid_base_stuff[] array.
e.g. ONE("MY_FILE", S_IRUSR, proc_pid_my_file),

2 - Create a callback function that will be invoked when the new proc entry is accessed (*task_struct is passed to this function).

```
static int proc_pid_my_file(struct seq_file *m, struct pid_namespace *ns,
 struct pid *pid, struct task_struct *task)
{
seq_printf(m, "%s %d\n", "here we go....", task->my_value);
 return 0;
}
```

Use the task_struct as usual:

```
struct task_struct *task;
 task = pid_task(find_vpid(1), PIDTYPE_PID);
```

I don't know if it is the best solution, but it seems it worked.
If anyone know a more easy or correct solution please let me know.

Thanks again,
Mauro Romano Trajber

On Wed, Jan 12, 2011 at 7:41 AM, Rajat Sharma <fs.rajat at gmail.com> wrote:

> *yes you are right, it gives NULL for <pid> directories under proc but*
> *I tried changing pid to something else, say /proc/sys and it works.*
> *Looks like handling of pid directories is entirely different. I didn't*
> *get time to explore on that, will have to dig more into this.*

```
>
> Rajat
>
> On Wed, Jan 12, 2011 at 12:50 AM, Mauro Romano Trajber
> <trajber at gmail.com> wrote:
> > Following your recommendations parent->pde always returns NULL. You know
> why
> > ?
> > // code...
> > err = kern_path("/proc/1/", LOOKUP_FOLLOW, &path);
> > if (err) {
> > return err;
> > }
> > struct proc_inode *parent = PROC_I(path.dentry->d_inode);
> > struct proc_dir_entry *parent_dir = parent->pde;
> > if (parent_dir == NULL) {
> > printk("parent_dir is NULL\n");
> > } else {
> > create_proc_entry("SOMEFile", 0644, parent_dir);
> > }
> >
> > Mauro!
> > On Tue, Jan 11, 2011 at 4:39 AM, Rajat Sharma <fs.rajat at gmail.com>
> wrote:
> >>
> >> Try this:
> >> 1. do a path_lookup for parent proc dir e.g. /proc/1234 and get its
> inode.
> >> 2. get proc_inode structure for parent from vfs inode like this:
> >>          sruct proc_inode *parent = PROC_I(inode).
> >>    PROC_I is defined in proc_fs.h
> >> 3. get parent proc_dir_entry object:
> >>          struct proc_dir_entry *parent_dir = parent->pde;
> >> 4. now you can call:
> >>          create_proc_entry("SOME file", 0644, parent_dir);
> >> 5. or you can create a directory if you want:
> >>          proc_mkdir("your dir", parent_dir);
> >>
> >> hope this helps.
> >>
> >> Rajat
> >>
> >> On Tue, Jan 11, 2011 at 12:19 AM, Mauro Romano Trajber
> >> <trajber at gmail.com> wrote:
> >> > I think I found:
> >> > static const struct pid_entry tgid_base_stuff[] at fs/proc/base.c has
> >> > all
> >> > /proc/[pid] entries.
> >> > But unfortunately it does not use create_proc_entry function, and I'm
> >> > trying
> >> > to create a new syscall that creates a new proc_entry for the caller
> >> > process.
> >> > Adding a new element in tgid_base_stuff[] makes the things more
> >> > complicated
> >> > than simply call a create_proc_entry function.
> >> > Is there another way to do it ?
> >> > Mauro Romano Trajber
> >> >
> >> > On Mon, Jan 10, 2011 at 3:18 PM, Mauro Romano Trajber
> >> > <trajber at gmail.com>
> >> > wrote:
> >> >>
> >> >> How can I create a new proc entry under /proc/[pid] ?
> >> >> I using create_proc_entry("SOME_FILE", 0644, NULL /* here goes the
> pid
> >> >> proc entry */);
> >> >> Is there any way to get PID directory as a parent proc entry ? How ?
> >> >> Thanks,
> >> >> Mauro
> >> >
> >> > _______________________________________________
> >> > Kernelnewbies mailing list
> >> > Kernelnewbies at kernelnewbies.org
> >> > http://lists.kernelnewbies.org/mailman/listinfo/kernelnewbies
> >> >
> >> >
> >
> >
>
-------------- next part --------------
An HTML attachment was scrubbed...
URL: 
http://lists.kernelnewbies.org/pipermail/kernelnewbies/attachments/20110112/3722a578/attachment.html
```