Paul Menage wrote:
> On Feb 7, 2008 7:37 AM, Pavel Emelyanov <xemul@openvz.org> wrote:
>> The Documentation/cgroups.txt file contains the info on how
>> to write some controller for cgroups subsystem, but even with
>> this, one need to write quite a lot of code before developing
>> the core (or copy-n-paste it from some other place).
>
> Good idea.
>
>> +
>> +static ssize_t foo_bar_read(struct cgroup *cg, struct cftype *cft,
>> +   struct file *file, char __user *userbuf,
>> +   size_t nbytes, loff_t *ppos)
>> +{
>> +   struct foo_cgroup *foo;
>> +
>> +   foo = foo_from_cgroup(cg);
>> +
>> +   /*
>> +   * produce some output
>> +   */
>> +
>> +   return nbytes;
>> +}
>> +
>> +static ssize_t foo_bar_write(struct cgroup *cg, struct cftype *cft,
>> +   struct file *file, const char __user *userbuf,
>> +   size_t nbytes, loff_t *ppos)
>> +{
>> +   struct foo_cgroup *foo;
>> +
>> +   foo = foo_from_cgroup(cg);
>> +
>> +   /*
>> +   * read and tune the foo
>> +   */
>> +
>> +   return nbytes;
>> +}
>> +
>> +static struct cftype foo_files[] = {
>> +   {
>> +   .name = "bar",
>> +   .read = foo_bar_read,

```
>> +                .write = foo_bar_write,
>> +     },
>> +};
>
> Can you structure this example so as to encourage people to use the
> more formatted read/write routines, such as read_int64 and
> write_int64?
```

And one more for ->read_strategy hints :) Well, I thought that
it should be the _minimal_ controller.

```
>> +
>> +static struct cgroup_subsys_state *foo_create(struct cgroup_subsys *cs,
>> +               struct cgroup *cg)
>> +{
>> +     struct foo_cgroup *foo;
>> +
>
> Maybe add a comment here that mentions that if your cgroup needs very
> early initialization, you can check for cg->parent being NULL, and
> return a statically-constructed structure here. (And set
> foo_subsys.early_init = 1)
```

Yet again - this is rather a special feature, that your controller
needs an early initialization - for a very minimal one this is not
required.

Maybe we can have two examples? One is the minimal one and the other
is an advanced one with ->attach callbacks, etc?

```
> Paul
>
```

_____

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