Subject: Re: [RFC][PATCH 5/5] Add a Signal Control Group Subsystem
Posted by Paul Menage on Fri, 25 Apr 2008 06:01:46 GMT
View Forum Message <> Reply to Message

I don't think you need cgroup_signal.h. It's only included in
cgroup_signal.c, and doesn't really contain any useful definitions
anyway. You should just use a cgroup_subsys_state object as your state
object, since you'll never need to do anything with it anyway.

```
>+static struct cgroup_subsys_state *signal_create(
>+ struct cgroup_subsys *ss, struct cgroup *cgroup)
>+{
>+ struct stateless *dummy;
>+
>+ if (!capable(CAP_SYS_ADMIN))
>+  return ERR_PTR(-EPERM);
```

This is unnecessary.

```
>+
+ dummy = kzalloc(sizeof(struct stateless), GFP_KERNEL);
+ if (!dummy)
+  return ERR_PTR(-ENOMEM);
+ return  &dummy->css;
+}
```

This function could be simplified to:

```
struct cgroup_subsys_state *css;
css = kzalloc(sizeof(*css), GFP_KERNEL);
return css ?: ERR_PTR(-ENOMEM);
```

```
>+static int signal_can_attach(struct cgroup_subsys *ss,
>+       struct cgroup *new_cgroup,
>+       struct task_struct *task)
>+{
>+ return 0;
>+}
```

No need for a can_attach() method if it just returns 0 - that's the default.

```
>+static int signal_kill(struct cgroup *cgroup, int signum)
>+{
>+ struct cgroup_iter it;
>+ struct task_struct *task;
>+ int retval = 0;
>+
>+ cgroup_iter_start(cgroup, &it);
```

```
>+ while ((task = cgroup_iter_next(cgroup, &it))) {
>+  retval = send_sig(signum, task, 1);
>+  if (retval)
>+   break;
>+ }
>+ cgroup_iter_end(cgroup, &it);
>+
>+ return retval;
>+}
```

cgroup_iter_start() takes a read lock - is send_sig() guaranteed not to sleep?

```
>+static ssize_t signal_write(struct cgroup *cgroup,
>+        struct cftype *cft,
>+        struct file *file,
>+        const char __user *userbuf,
>+        size_t nbytes, loff_t *unused_ppos)
```

This should just be a write_u64() method - cgroups will handle the
copying/parsing for you. See e.g.
kernel/sched.c:cpu_shares_write_u64()

```
>+static struct cftype kill_file = {
>+ .name = "kill",
>+ .write = signal_write,
>+ .private = 0,
>+};
```

I agree with PaulJ that "signal.send" would be a nicer name for this
than "signal.kill"