
Subject: [PATCH] Brush up task session manipulation routines

Posted by [xemul](#) on Fri, 11 May 2007 16:03:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

The function names `process_session()` and `task_session()` are somewhat discouraging. Functions `set_signal_session()` and `signal_session()` are never used for "orphaned" signals - there's always a task to take it from.

So I propose to:

- * rename `process_session()` to `task_session_nr()` to make it symmetrical to `task_session()`,
- * replace `signal_session()` with `task_session_nr()` and `set_signal_session()` with `set_task_session()` not to make redundant essences.

If this looks reasonable, further proposal would be to make `process_group()` be symmetrical to `task_pgrp()` the same way.

Fits 2.6.21-mm2

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

```
diff --git a/fs/proc/array.c b/fs/proc/array.c
index 74f30e0..1eee446 100644
--- a/fs/proc/array.c
+++ b/fs/proc/array.c
@@ -381,7 +381,7 @@ static int do_task_stat(struct task_stru
     stime = cputime_add(stime, sig->stime);
 }
```

```
- sid = signal_session(sig);
+ sid = task_session_nr(task);
  pgid = process_group(task);
  ppid = rcu_dereference(task->real_parent)->tgid;
```

```
diff --git a/include/linux/sched.h b/include/linux/sched.h
index 40645b4..66b59db 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -1099,19 +1099,14 @@ static inline pid_t process_group(struct
     return tsk->signal->pgrp;
 }
```

```
-static inline pid_t signal_session(struct signal_struct *sig)
+static inline pid_t task_session_nr(struct task_struct *tsk)
```

```

{
- return sig->__session;
+ return tsk->signal->__session;
}

-static inline pid_t process_session(struct task_struct *tsk)
+static inline void set_task_session(struct task_struct *tsk, pid_t session)
{
- return signal_session(tsk->signal);
-}
-
-static inline void set_signal_session(struct signal_struct *sig, pid_t session)
-{
- sig->__session = session;
+ tsk->sig->__session = session;
}

static inline struct pid *task_pid(struct task_struct *task)
diff --git a/kernel/exit.c b/kernel/exit.c
index c6d14b8..e9ad2c3 100644
--- a/kernel/exit.c
+++ b/kernel/exit.c
@@ -310,7 +310,7 @@ void __set_special_pids(pid_t session, p

if (process_session(curr) != session) {
detach_pid(curr, PIDTYPE_SID);
- set_signal_session(curr->signal, session);
+ set_task_session(curr, session);
attach_pid(curr, PIDTYPE_SID, find_pid(session));
}
if (process_group(curr) != pgrp) {
diff --git a/kernel/fork.c b/kernel/fork.c
index 4239de2..090e3a1 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1252,7 +1252,7 @@ static struct task_struct *copy_process(
if (thread_group_leader(p)) {
p->signal->tty = current->signal->tty;
p->signal->pgrp = process_group(current);
- set_signal_session(p->signal, process_session(current));
+ set_task_session(p, task_session_nr(current));
attach_pid(p, PIDTYPE_PGID, task_pgrp(current));
attach_pid(p, PIDTYPE_SID, task_session(current));

```
