Subject: [PATCH] Isolate the explicit usage of signal->pgrp
Posted by Pavel Emelianov on Mon, 20 Aug 2007 11:50:40 GMT
View Forum Message <> Reply to Message

The pgrp field is not used widely around the kernel so it is
now marked as deprecated with appropriate comment.

The initialization of INIT_SIGNALS is trimmed because
a) they are set to 0 automatically;
b) gcc cannot properly initialize two anonymous (the second one
   is the one with the session) unions. In this particular case
   to make it compile we'd have to add some field initialized
   right before the .pgrp.

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

This is the same patch as the 1ec320afdc9552c92191d5f89fcd1ebe588334ca
one (from Cedric), but for the pgrp field.

Some progress report:
We have to deprecate the pid, tgid, session and pgrp fields on
struct task_struct and struct signal_struct. The session and pgrp
are already deprecated. The tgid value is close to being such - the
worst known usage in in fs/locks.c and audit code. The pid field
deprecation is mainly blocked by numerous printk-s around the
kernel that print the tsk->pid to log.

```
 include/linux/init_task.h |    3 ---
 include/linux/sched.h     |   19 ++++++++++++++++++--
 kernel/exit.c             |    2 +-
 kernel/fork.c             |    4 ++--
 kernel/sys.c              |    2 +-
 5 files changed, 21 insertions(+), 9 deletions(-)

diff --git a/include/linux/init_task.h b/include/linux/init_task.h
index 0c8ce4e..1ac10c0 100644
--- a/include/linux/init_task.h
+++ b/include/linux/init_task.h
@@ -66,9 +66,6 @@
  .posix_timers  = LIST_HEAD_INIT(sig.posix_timers),  \
  .cpu_timers = INIT_CPU_TIMERS(sig.cpu_timers),  \
  .rlim  = INIT_RLIMITS,     \
- .pgrp  = 0,      \
- .tty_old_pgrp   = NULL,      \
- { .__session      = 0},      \
 }
```

```
 extern struct nsproxy init_nsproxy;
diff --git a/include/linux/sched.h b/include/linux/sched.h
index 24b669d..a410ac5 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -426,7 +426,17 @@ struct signal_struct {
  cputime_t it_prof_incr, it_virt_incr;

  /* job control IDs */
- pid_t pgrp;
+
+ /*
+  * pgrp and session fields are deprecated.
+  * use the task_session_Xnr and task_pgrp_Xnr routines below
+  */
+
+ union {
+  pid_t pgrp __deprecated;
+  pid_t __pgrp;
+ };
+
  struct pid *tty_old_pgrp;

  union {
@@ -1164,6 +1174,11 @@ static inline void set_task_session(stru
  tsk->signal->__session = session;
 }

+static inline void set_task_pgrp(struct task_struct *tsk, pid_t pgrp)
+{
+ tsk->signal->__pgrp = pgrp;
+}
+
 static inline struct pid *task_pid(struct task_struct *task)
 {
  return task->pids[PIDTYPE_PID].pid;
@@ -1244,7 +1259,7 @@ static inline pid_t task_tgid_vnr(struct

 static inline pid_t task_pgrp_nr(struct task_struct *tsk)
 {
- return tsk->signal->pgrp;
+ return tsk->signal->__pgrp;
 }

 static inline pid_t task_pgrp_nr_ns(struct task_struct *tsk,
diff --git a/kernel/exit.c b/kernel/exit.c
index 872972f..f4783ae 100644
```

```
--- a/kernel/exit.c
+++ b/kernel/exit.c
@@ -315,7 +315,7 @@ void __set_special_pids(pid_t session, p
  }
  if (task_pgrp_nr(curr) != pgrp) {
   detach_pid(curr, PIDTYPE_PGID);
-  curr->signal->pgrp = pgrp;
+  set_task_pgrp(curr, pgrp);
   attach_pid(curr, PIDTYPE_PGID, find_pid(pgrp));
  }
 }
diff --git a/kernel/fork.c b/kernel/fork.c
index d7cff48..03e5ad1 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1278,13 +1278,13 @@ static struct task_struct *copy_process(
    if (clone_flags & CLONE_NEWPID) {
     p->nsproxy->pid_ns->child_reaper = p;
     p->signal->tty = NULL;
-    p->signal->pgrp = p->pid;
+    set_task_pgrp(p, p->pid);
     set_task_session(p, p->pid);
     attach_pid(p, PIDTYPE_PGID, pid);
     attach_pid(p, PIDTYPE_SID, pid);
    } else {
     p->signal->tty = current->signal->tty;
-    p->signal->pgrp = task_pgrp_nr(current);
+    set_task_pgrp(p, task_pgrp_nr(current));
     set_task_session(p, task_session_nr(current));
     attach_pid(p, PIDTYPE_PGID,
       task_pgrp(current));
diff --git a/kernel/sys.c b/kernel/sys.c
index 8333f51..c7c4fa4 100644
--- a/kernel/sys.c
+++ b/kernel/sys.c
@@ -976,7 +976,7 @@ asmlinkage long sys_setpgid(pid_t pid, p
   detach_pid(p, PIDTYPE_PGID);
   pid = find_vpid(pgid);
   attach_pid(p, PIDTYPE_PGID, pid);
-  p->signal->pgrp = pid_nr(pid);
+  set_task_pgrp(p, pid_nr(pid));
  }

  err = 0;
```