Subject: [RFC][PATCH] Static init struct pid for swapper
Posted by Sukadev Bhattiprolu on Sat, 13 Jan 2007 21:57:00 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: Statically initialize struct pid for swapper

Statically initialize a struct pid for the swapper process (pid_t == 0)
and attach it to init_task. This is needed so task_pid(), task_pgrp()
and task_session() interfaces work on the swapper process also.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Cc: Cedric Le Goater <clg@fr.ibm.com>
Cc: Dave Hansen <haveblue@us.ibm.com>
Cc: Serge Hallyn <serue@us.ibm.com>
Cc: containers@lists.osdl.org

---
 arch/alpha/kernel/init_task.c | 1 +
 arch/arm/kernel/init_task.c | 1 +
 arch/arm26/kernel/init_task.c | 1 +
 arch/avr32/kernel/init_task.c | 1 +
 arch/frv/kernel/init_task.c | 1 +
 arch/h8300/kernel/init_task.c | 1 +
 arch/i386/kernel/init_task.c | 1 +
 arch/ia64/kernel/init_task.c | 1 +
 arch/m32r/kernel/init_task.c | 1 +
 arch/m68knommu/kernel/init_task.c | 1 +
 arch/mips/kernel/init_task.c | 1 +
 arch/parisc/kernel/init_task.c | 1 +
 arch/powerpc/kernel/init_task.c | 1 +
 arch/s390/kernel/init_task.c | 1 +
 arch/sh/kernel/init_task.c | 1 +
 arch/sh64/kernel/init_task.c | 1 +
 arch/sparc/kernel/init_task.c | 1 +
 arch/sparc64/kernel/init_task.c | 1 +
 arch/um/kernel/init_task.c | 1 +
 arch/v850/kernel/init_task.c | 1 +
 arch/x86_64/kernel/init_task.c | 1 +
 include/linux/init_task.h | 33 ++++++++++++++++++++++++++++++++++
 22 files changed, 54 insertions(+)

Index: lx26-20-rc2-mm1/arch/x86_64/kernel/init_task.c
===
--- lx26-20-rc2-mm1.orig/arch/x86_64/kernel/init_task.c 2007-01-13 04:38:35.892740080 -0800
+++ lx26-20-rc2-mm1/arch/x86_64/kernel/init_task.c 2007-01-13 04:57:58.930931400 -0800
@@ -14,6 +14,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;

```
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/include/linux/init_task.h
===================================================================
--- lx26-20-rc2-mm1.orig/include/linux/init_task.h 2007-01-13 04:38:35.893739928 -0800
+++ lx26-20-rc2-mm1/include/linux/init_task.h 2007-01-13 05:25:52.178559096 -0800
@@ -88,6 +88,19 @@ extern struct nsproxy init_nsproxy;

 extern struct group_info init_groups;

+#define INIT_STRUCT_PID {      \
+ .count   = ATOMIC_INIT(1),    \
+ .nr  = 0,       \
+ /* Do we need to put this struct pid in pid_hash ? */  \
+ .pid_chain = { .next = NULL, .pprev = NULL },  \
+ .tasks   =  {       \
+  { .first = &init_task.pids[PIDTYPE_PID].node },  \
+  { .first = &init_task.pids[PIDTYPE_PGID].node }, \
+  { .first = &init_task.pids[PIDTYPE_SID].node },  \
+ },         \
+ .rcu  = RCU_HEAD_INIT,     \
+}
+
 /*
  *  INIT_TASK is used to set up the first task table, touch at
  * your own risk!. Base=0, limit=0x1fffff (=2MB)
@@ -139,6 +152,26 @@ extern struct group_info init_groups;
  .cpu_timers = INIT_CPU_TIMERS(tsk.cpu_timers),  \
  .fs_excl = ATOMIC_INIT(0),    \
  .pi_lock = SPIN_LOCK_UNLOCKED,    \
+ .pids = {       \
+  { .node = {      \
+    .next = NULL,     \
+     .pprev = &init_struct_pid.tasks[PIDTYPE_PID].first,\
+    },       \
+    .pid = &init_struct_pid,    \
+  },       \
+  { .node = {      \
+    .next = NULL,     \
+     .pprev = &init_struct_pid.tasks[PIDTYPE_PGID].first,\
+    },       \
+    .pid = &init_struct_pid,    \
+  },       \
+  { .node = {      \
```

```
+    .next = NULL,     \
+     .pprev = &init_struct_pid.tasks[PIDTYPE_SID].first,\
+    },       \
+    .pid = &init_struct_pid,    \
+  },       \
+ }       \
  INIT_TRACE_IRQFLAGS      \
  INIT_LOCKDEP       \
 }
Index: lx26-20-rc2-mm1/arch/alpha/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/alpha/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/alpha/kernel/init_task.c 2007-01-13 12:41:56.745940256 -0800
@@ -12,6 +12,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);
 struct task_struct init_task = INIT_TASK(init_task);

Index: lx26-20-rc2-mm1/arch/arm/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/arm/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/arm/kernel/init_task.c 2007-01-13 12:42:18.614615712 -0800
@@ -16,6 +16,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/arm26/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/arm26/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/arm26/kernel/init_task.c 2007-01-13 12:42:12.116603560 -0800
@@ -19,6 +19,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/avr32/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/avr32/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
```

```
+++ lx26-20-rc2-mm1/arch/avr32/kernel/init_task.c 2007-01-13 12:42:25.664543960 -0800
@@ -17,6 +17,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/frv/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/frv/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/frv/kernel/init_task.c 2007-01-13 12:42:37.605728624 -0800
@@ -14,6 +14,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/h8300/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/h8300/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/h8300/kernel/init_task.c 2007-01-13 12:42:42.575973032 -0800
@@ -16,6 +16,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/i386/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/i386/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/i386/kernel/init_task.c 2007-01-13 12:42:52.308493464 -0800
@@ -14,6 +14,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/ia64/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/ia64/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
```

```
+++ lx26-20-rc2-mm1/arch/ia64/kernel/init_task.c 2007-01-13 12:43:03.647769632 -0800
@@ -20,6 +20,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/m32r/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/m32r/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/m32r/kernel/init_task.c 2007-01-13 12:43:08.006107064 -0800
@@ -15,6 +15,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/m68knommu/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/m68knommu/kernel/init_task.c 2006-11-29 13:57:37.000000000
-0800
+++ lx26-20-rc2-mm1/arch/m68knommu/kernel/init_task.c 2007-01-13 12:43:23.268786784 -0800
@@ -16,6 +16,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/mips/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/mips/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/mips/kernel/init_task.c 2007-01-13 12:43:28.325018120 -0800
@@ -13,6 +13,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/parisc/kernel/init_task.c
===================================================================
```

```
--- lx26-20-rc2-mm1.orig/arch/parisc/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/parisc/kernel/init_task.c 2007-01-13 12:43:40.204212208 -0800
@@ -37,6 +37,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/powerpc/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/powerpc/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/powerpc/kernel/init_task.c 2007-01-13 12:43:44.629539456 -0800
@@ -11,6 +11,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/s390/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/s390/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/s390/kernel/init_task.c 2007-01-13 12:43:54.629019304 -0800
@@ -19,6 +19,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/sh/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/sh/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/sh/kernel/init_task.c 2007-01-13 12:45:33.954919472 -0800
@@ -11,6 +11,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/sh64/kernel/init_task.c
===================================================================
```

```
--- lx26-20-rc2-mm1.orig/arch/sh64/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/sh64/kernel/init_task.c 2007-01-13 12:45:26.773011288 -0800
@@ -22,6 +22,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 struct pt_regs fake_swapper_regs;
Index: lx26-20-rc2-mm1/arch/sparc/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/sparc/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/sparc/kernel/init_task.c 2007-01-13 12:45:45.048233032 -0800
@@ -11,6 +11,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);
 struct task_struct init_task = INIT_TASK(init_task);

Index: lx26-20-rc2-mm1/arch/sparc64/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/sparc64/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/sparc64/kernel/init_task.c 2007-01-13 12:45:39.008151264 -0800
@@ -12,6 +12,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM(init_mm);

 EXPORT_SYMBOL(init_mm);
Index: lx26-20-rc2-mm1/arch/um/kernel/init_task.c
===================================================================
--- lx26-20-rc2-mm1.orig/arch/um/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/um/kernel/init_task.c 2007-01-13 12:45:55.279677616 -0800
@@ -19,6 +19,7 @@ struct mm_struct init_mm = INIT_MM(init_
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS(init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 EXPORT_SYMBOL(init_mm);

 /*
Index: lx26-20-rc2-mm1/arch/v850/kernel/init_task.c
===================================================================
```

```
--- lx26-20-rc2-mm1.orig/arch/v850/kernel/init_task.c 2006-11-29 13:57:37.000000000 -0800
+++ lx26-20-rc2-mm1/arch/v850/kernel/init_task.c 2007-01-13 12:46:03.355449912 -0800
@@ -24,6 +24,7 @@ static struct fs_struct init_fs = INIT_F
 static struct files_struct init_files = INIT_FILES;
 static struct signal_struct init_signals = INIT_SIGNALS (init_signals);
 static struct sighand_struct init_sighand = INIT_SIGHAND(init_sighand);
+static struct pid init_struct_pid = INIT_STRUCT_PID;
 struct mm_struct init_mm = INIT_MM (init_mm);

 EXPORT_SYMBOL(init_mm);
```

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [RFC][PATCH] Static init struct pid for swapper
Posted by ebiederm on Mon, 15 Jan 2007 15:16:22 GMT
View Forum Message <> Reply to Message

Sukadev Bhattiprolu <sukadev@us.ibm.com> writes:

> From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> Subject: Statically initialize struct pid for swapper
>
> Statically initialize a struct pid for the swapper process (pid_t == 0)
> and attach it to init_task. This is needed so task_pid(), task_pgrp()
> and task_session() interfaces work on the swapper process also.

This looks encouraging.

We still need to address the fact that we are placing pid == 1
in an invalid process group and session. We need to call something
like setsid() to address this before we start any other threads
or /sbin/init.

All of the other idle threads are created with fork_idle() from
pid == 1 in the smp startup code, so there are still some differences
but that is independent of your patch.

Just please pass the pid as a struct pid into copy_process and
then we can remove the if (likely(p->pid)) case and use attach_pid
for everything. And please call setsid() or the equivalent about
where we recognize we are the child_reaper in init(), before we run
any other threads.

Oh yes. Please fix your whitespace damage in the initializer.

```
>  /*
>   *  INIT_TASK is used to set up the first task table, touch at
>   * your own risk!. Base=0, limit=0x1fffff (=2MB)
> @@ -139,6 +152,26 @@ extern struct group_info init_groups;
>   .cpu_timers = INIT_CPU_TIMERS(tsk.cpu_timers),  \
>   .fs_excl = ATOMIC_INIT(0),    \
>   .pi_lock = SPIN_LOCK_UNLOCKED,    \
> + .pids = {       \
> +  { .node = {      \
> +    .next = NULL,     \
> +     .pprev = &init_struct_pid.tasks[PIDTYPE_PID].first,\
> +    },       \
> +    .pid = &init_struct_pid,    \
> +  },       \
> +  { .node = {      \
> +    .next = NULL,     \
> +     .pprev = &init_struct_pid.tasks[PIDTYPE_PGID].first,\
> +    },       \
> +    .pid = &init_struct_pid,    \
> +  },       \
> +  { .node = {      \
> +    .next = NULL,     \
> +     .pprev = &init_struct_pid.tasks[PIDTYPE_SID].first,\
> +    },       \
> +    .pid = &init_struct_pid,    \
> +  },       \
> + }       \
>   INIT_TRACE_IRQFLAGS      \
>   INIT_LOCKDEP       \
>  }
```

Say something like:

```
#define INIT_PID_LINK(TYPE) =      \
{       \
 .node = {      \
  .next = NULL,     \
  .pprev = &init_struct_pid.tasks[TYPE].first, \
 },       \
 .pid = &init_struct_pid,    \
}
```

And then:

```
 .pids = {
  [PIDTYPE_PID]  = INIT_PID_LINK(PIDTYPE_PID),
  [PIDTYPE_PGID] = INIT_PID_LINK(PIDTYPE_PGID),
              [PIDTYPE_SID]  = INIT_PID_LINK(PIDTYPE_SID),
```

}

As for the question below. It would be very bad if pid 0 every gets into the pid hash table. We have a number of cases that explicitly

```
+#define INIT_STRUCT_PID {      \
+ .count   = ATOMIC_INIT(1),    \
+ .nr  = 0,       \
+ /* Do we need to put this struct pid in pid_hash ? */  \
+ .pid_chain = { .next = NULL, .pprev = NULL },  \
+ .tasks   =  {       \
+  { .first = &init_task.pids[PIDTYPE_PID].node },  \
+  { .first = &init_task.pids[PIDTYPE_PGID].node }, \
+  { .first = &init_task.pids[PIDTYPE_SID].node },  \
+ },         \
+ .rcu  = RCU_HEAD_INIT,     \
+}
+
```

And one final thing. Please place init_pid in pid.c. We can just put an "extern struct pid init_pid;" in pid.h

There is no reason to put a separate copy in every architecture and it will be easier to have just a single copy in the code.

Eric

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [RFC][PATCH] Static init struct pid for swapper
Posted by Sukadev Bhattiprolu on Tue, 16 Jan 2007 03:19:14 GMT
View Forum Message <> Reply to Message

Eric W. Biederman [ebiederm@xmission.com] wrote:
| Sukadev Bhattiprolu <sukadev@us.ibm.com> writes:
|
| > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
| > Subject: Statically initialize struct pid for swapper
| >
| > Statically initialize a struct pid for the swapper process (pid_t == 0)
| > and attach it to init_task. This is needed so task_pid(), task_pgrp()
| > and task_session() interfaces work on the swapper process also.

|
| This looks encouraging.
|
| We still need to address the fact that we are placing pid == 1
| in an invalid process group and session. We need to call something
| like setsid() to address this before we start any other threads
| or /sbin/init.

Maybe I am missing something. INIT_SIGNALS sets pgid = 1, sid = 1 for
the swapper and this is passed on to /sbin/init in copy_process.
i.e after copy_process() the thread that would become /sbin/init has
pgid = sid = 1 - isn't that what we want ?

| All of the other idle threads are created with fork_idle() from
| pid == 1 in the smp startup code, so there are still some differences
| but that is independent of your patch.
|
| Just please pass the pid as a struct pid into copy_process and
| then we can remove the if (likely(p->pid)) case and use attach_pid
| for everything.

Done. Will send out the patch. I have kept it separate from the static
init patch for now.

| And please call setsid() or the equivalent about
| where we recognize we are the child_reaper in init(), before we run
| any other threads.

I need to understand this better, pls see my question above.

|
| Oh yes. Please fix your whitespace damage in the initializer.

I think I fixed it now.

|
| > /*
| >   * INIT_TASK is used to set up the first task table, touch at
| >   * your own risk!. Base=0, limit=0x1fffff (=2MB)
| > @@ -139,6 +152,26 @@ extern struct group_info init_groups;
| >   .cpu_timers = INIT_CPU_TIMERS(tsk.cpu_timers), \
| >   .fs_excl = ATOMIC_INIT(0),   \
| >   .pi_lock = SPIN_LOCK_UNLOCKED,   \
| > + .pids = {   \
| > + { .node = {   \
| > +   .next = NULL,   \
| > +   .pprev = &init_struct_pid.tasks[PIDTYPE_PID].first,\
| > +   },   \

```
| > +    .pid = &init_struct_pid,    \
| > +  },       \
| > +  { .node = {      \
| > +    .next = NULL,     \
| > +     .pprev = &init_struct_pid.tasks[PIDTYPE_PGID].first,\
| > +    },       \
| > +    .pid = &init_struct_pid,    \
| > +  },       \
| > +  { .node = {      \
| > +    .next = NULL,     \
| > +     .pprev = &init_struct_pid.tasks[PIDTYPE_SID].first,\
| > +    },       \
| > +    .pid = &init_struct_pid,    \
| > +  },       \
| > + }       \
| >   INIT_TRACE_IRQFLAGS      \
| >   INIT_LOCKDEP       \
| >  }
|
| Say something like:
|
| #define INIT_PID_LINK(TYPE) =      \
| {       \
|  .node = {      \
|   .next = NULL,     \
|   .pprev = &init_struct_pid.tasks[TYPE].first, \
|  },       \
|  .pid = &init_struct_pid,    \
| }
|
| And then:
|
|  .pids = {
|   [PIDTYPE_PID]  = INIT_PID_LINK(PIDTYPE_PID),
|   [PIDTYPE_PGID] = INIT_PID_LINK(PIDTYPE_PGID),
|               [PIDTYPE_SID]  = INIT_PID_LINK(PIDTYPE_SID),
|  }
```

Yes. Good idea.

```
|
| As for the question below.  It would be very bad if pid 0 every gets
| into the pid hash table.   We have a number of cases that explicitly
|
```

Ok.

```
| +#define INIT_STRUCT_PID {      \
```

| + .count   = ATOMIC_INIT(1),   \
| + .nr = 0,   \
| + /* Do we need to put this struct pid in pid_hash ? */ \
| + .pid_chain = { .next = NULL, .pprev = NULL }, \
| + .tasks   = {   \
| + { .first = &init_task.pids[PIDTYPE_PID].node }, \
| + { .first = &init_task.pids[PIDTYPE_PGID].node }, \
| + { .first = &init_task.pids[PIDTYPE_SID].node }, \
| + },   \
| + .rcu = RCU_HEAD_INIT,   \
| +}
| +
|
|
| And one final thing. Please place init_pid in pid.c. We can just put
| an "extern struct pid init_pid;" in pid.h
|
| There is no reason to put a separate copy in every architecture and it
| will be easier to have just a single copy in the code.

Good idea. I have updated my patch. I was wondering about it myself.

|
|
| Eric

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers