Subject: Acks for 3 pid-namespace patches
Posted by Sukadev Bhattiprolu on Tue, 19 Sep 2006 00:01:04 GMT
View Forum Message <> Reply to Message

Wanted to check if we have a consensus on following three patches and
see if we are ready to send them to akpm and LKML.

http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-rename-pspace-to-pid-namespace.patch

http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-pid-namespace-to-nsproxy.patch

http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-child-reaper-to-pid-namespace.patch

For the last patch add-child-reaper-to-pid-namespace, there was discussion
on how to implement killing all processes in a pid-namespace when
the reaper dies, but can we address that in a follow-on patch ?

Suka

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

Subject: [PATCH 1/3]: Rename pspace to pid-namespace (was: Acks for 3 pid-namespace patches)
Posted by Sukadev Bhattiprolu on Tue, 19 Sep 2006 00:25:40 GMT
View Forum Message <> Reply to Message

Sukadev Bhattiprolu [sukadev@us.ibm.com] wrote:
|
| Wanted to check if we have a consensus on following three patches and
| see if we are ready to send them to akpm and LKML.
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-rename-pspace-to-pid-namespace.patch
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-pid-namespace-to-nsproxy.patch
|
|

http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-child-reaper-to-pid-namespace.patch
|
| For the last patch add-child-reaper-to-pid-namespace, there was discussion
| on how to implement killing all processes in a pid-namespace when
| the reaper dies, but can we address that in a follow-on patch ?
|

And here is the first patch, (renmae-pspace).

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

---

Subject: [PATCH 2/3] add pid-namespace to nsproxy (was: Acks for 3 pid-namespace patches)
Posted by Sukadev Bhattiprolu on Tue, 19 Sep 2006 00:28:09 GMT
View Forum Message <> Reply to Message

Sukadev Bhattiprolu [sukadev@us.ibm.com] wrote:
|
| Wanted to check if we have a consensus on following three patches and
| see if we are ready to send them to akpm and LKML.
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-rename-pspace-to-pid-namespace.patch
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-pid-namespace-to-nsproxy.patch
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-child-reaper-to-pid-namespace.patch
|
| For the last patch add-child-reaper-to-pid-namespace, there was discussion
| on how to implement killing all processes in a pid-namespace when
| the reaper dies, but can we address that in a follow-on patch ?
|
| Suka

Here is the second one, add-pid-namespace-to-nsproxy

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: add pid_namespace to nsproxy

Add a notion of pid namespace to nsproxy (and by extension, to task_struct).
Currently there is only one pid namespace, init_pid_ns and all tasks belong
to this pid namespace. When a new task is created, it inherits its parent's
pid namespace (in copy_process()).

This is based on Eric Biederman's patch: http://lkml.org/lkml/2006/2/6/285

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

```
 include/linux/init_task.h     |    2 ++
 include/linux/nsproxy.h       |    2 ++
 include/linux/pid_namespace.h |   15 +++++++++++++++
 kernel/nsproxy.c              |   12 ++++++++++++
 4 files changed, 31 insertions(+)

Index: lx26-18-rc6-mm2/include/linux/init_task.h
===================================================================
--- lx26-18-rc6-mm2.orig/include/linux/init_task.h 2006-09-18 15:46:12.000000000 -0700
+++ lx26-18-rc6-mm2/include/linux/init_task.h 2006-09-18 16:04:44.000000000 -0700
@@ -7,6 +7,7 @@
 #include <linux/utsname.h>
 #include <linux/lockdep.h>
 #include <linux/ipc.h>
+#include <linux/pid_namespace.h>

 #define INIT_FDTABLE \
 {       \
@@ -72,6 +73,7 @@

 extern struct nsproxy init_nsproxy;
 #define INIT_NSPROXY(nsproxy) {      \
+ .pid_ns  = &init_pid_ns,     \
  .count  = ATOMIC_INIT(1),    \
  .nslock  = SPIN_LOCK_UNLOCKED,    \
  .uts_ns  = &init_uts_ns,     \
Index: lx26-18-rc6-mm2/include/linux/nsproxy.h
===================================================================
--- lx26-18-rc6-mm2.orig/include/linux/nsproxy.h 2006-09-18 15:46:12.000000000 -0700
+++ lx26-18-rc6-mm2/include/linux/nsproxy.h 2006-09-18 16:04:44.000000000 -0700
@@ -7,6 +7,7 @@
 struct namespace;
 struct uts_namespace;
 struct ipc_namespace;
```

```
+struct pid_namespace;

 /*
  * A structure to contain pointers to all per-process
@@ -23,6 +24,7 @@ struct ipc_namespace;
 struct nsproxy {
  atomic_t count;
  spinlock_t nslock;
+ struct pid_namespace *pid_ns;
  struct uts_namespace *uts_ns;
  struct ipc_namespace *ipc_ns;
  struct namespace *namespace;
Index: lx26-18-rc6-mm2/include/linux/pid_namespace.h
===================================================================
--- lx26-18-rc6-mm2.orig/include/linux/pid_namespace.h 2006-09-18 15:56:45.000000000 -0700
+++ lx26-18-rc6-mm2/include/linux/pid_namespace.h 2006-09-18 16:04:44.000000000 -0700
@@ -5,6 +5,7 @@
 #include <linux/mm.h>
 #include <linux/threads.h>
 #include <linux/pid.h>
+#include <linux/nsproxy.h>

 struct pidmap {
        atomic_t nr_free;
@@ -20,4 +21,18 @@ struct pid_namespace {

 extern struct pid_namespace init_pid_ns;

+static inline void get_pid_ns(struct pid_namespace *ns)
+{
+}
+
+static inline int copy_pid_ns(int flags, struct task_struct *tsk)
+{
+ tsk->nsproxy->pid_ns = &init_pid_ns;
+ return 0;
+}
+
+static inline void put_pid_ns(struct pid_namespace *ns)
+{
+}
+
 #endif /* _LINUX_PID_NS_H */
Index: lx26-18-rc6-mm2/kernel/nsproxy.c
===================================================================
--- lx26-18-rc6-mm2.orig/kernel/nsproxy.c 2006-09-18 15:46:28.000000000 -0700
+++ lx26-18-rc6-mm2/kernel/nsproxy.c 2006-09-18 16:04:44.000000000 -0700
@@ -19,6 +19,7 @@
```

```
 #include <linux/init_task.h>
 #include <linux/namespace.h>
 #include <linux/utsname.h>
+#include <linux/pid_namespace.h>

 struct nsproxy init_nsproxy = INIT_NSPROXY(init_nsproxy);

@@ -68,6 +69,8 @@ struct nsproxy *dup_namespaces(struct ns
    get_uts_ns(ns->uts_ns);
   if (ns->ipc_ns)
    get_ipc_ns(ns->ipc_ns);
+  if (ns->pid_ns)
+   get_pid_ns(ns->pid_ns);
  }

  return ns;
@@ -111,10 +114,17 @@ int copy_namespaces(int flags, struct ta
  if (err)
   goto out_ipc;

+ err = copy_pid_ns(flags, tsk);
+ if (err)
+  goto out_pid;
+
 out:
  put_nsproxy(old_ns);
  return err;

+out_pid:
+ if (new_ns->ipc_ns)
+  put_ipc_ns(new_ns->ipc_ns);
 out_ipc:
  if (new_ns->uts_ns)
   put_uts_ns(new_ns->uts_ns);
@@ -135,5 +145,7 @@ void free_nsproxy(struct nsproxy *ns)
    put_uts_ns(ns->uts_ns);
   if (ns->ipc_ns)
    put_ipc_ns(ns->ipc_ns);
+  if (ns->pid_ns)
+   put_pid_ns(ns->pid_ns);
   kfree(ns);
 }
```

---

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

Subject: [PATCH 3/3] Add child-reaper to pid-namespace (was: Acks for 3 pid-namespace patches)
Posted by Sukadev Bhattiprolu on Tue, 19 Sep 2006 00:29:34 GMT
View Forum Message <> Reply to Message

Sukadev Bhattiprolu [sukadev@us.ibm.com] wrote:
|
| Wanted to check if we have a consensus on following three patches and
| see if we are ready to send them to akpm and LKML.
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-rename-pspace-to-pid-namespace.patch
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-pid-namespace-to-nsproxy.patch
|
|
http://www.sr71.net/patches/2.6.18/2.6.18-rc6-mm2/2.6.18-rc6-mm2-lxc1/broken-out/pid-namespace-add-child-reaper-to-pid-namespace.patch
|
| For the last patch add-child-reaper-to-pid-namespace, there was discussion
| on how to implement killing all processes in a pid-namespace when
| the reaper dies, but can we address that in a follow-on patch ?
|
| Suka

Finally, here is the patch to add child-reaper to pid_namespace.

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: add child reaper to pid_namespace

Add per-pid-space child-reaper. This is needed so processes are reaped within
the same pid space and do not spill over to the parent pid space. Its also
needed so containers preserve existing semantic that pid == 1 would reap
orphaned children.

This is based on Eric Biederman's patch: http://lkml.org/lkml/2006/2/6/285

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
Cc: Eric Biederman <ebiederm@xmission.com>
Cc: Dave Hansen <haveblue@us.ibm.com>
Cc: Serge Hallyn <serue@us.ibm.com>

 fs/exec.c | 5 +++--
 include/linux/pid.h | 5 +++--

```
 include/linux/pid_namespace.h |    9 +++++----
 include/linux/sched.h         |    1 -
 init/main.c                   |    5 ++---
 kernel/exit.c                 |   26 ++++++++++++++++++++--------
 kernel/pid.c                  |    3 ++-
 kernel/signal.c               |   10 ++++++++--
 8 files changed, 41 insertions(+), 23 deletions(-)

Index: lx26-18-rc6-mm2/init/main.c
===================================================================
--- lx26-18-rc6-mm2.orig/init/main.c 2006-09-18 15:46:29.000000000 -0700
+++ lx26-18-rc6-mm2/init/main.c 2006-09-18 16:05:13.000000000 -0700
@@ -49,6 +49,7 @@
 #include <linux/buffer_head.h>
 #include <linux/debug_locks.h>
 #include <linux/lockdep.h>
+#include <linux/pid_namespace.h>

 #include <asm/io.h>
 #include <asm/bugs.h>
@@ -623,8 +624,6 @@ static int __init initcall_debug_setup(c
 }
 __setup("initcall_debug", initcall_debug_setup);

-struct task_struct *child_reaper = &init_task;
-
 extern initcall_t __initcall_start[], __initcall_end[];

 static void __init do_initcalls(void)
@@ -724,7 +723,7 @@ static int init(void * unused)
   * assumptions about where in the task array this
   * can be found.
   */
- child_reaper = current;
+ init_pid_ns.child_reaper = current;

  smp_prepare_cpus(max_cpus);

Index: lx26-18-rc6-mm2/fs/exec.c
===================================================================
--- lx26-18-rc6-mm2.orig/fs/exec.c 2006-09-18 15:46:25.000000000 -0700
+++ lx26-18-rc6-mm2/fs/exec.c 2006-09-18 16:05:13.000000000 -0700
@@ -38,6 +38,7 @@
 #include <linux/binfmts.h>
 #include <linux/swap.h>
 #include <linux/utsname.h>
+#include <linux/pid_namespace.h>
 #include <linux/module.h>
```

```
 #include <linux/namei.h>
 #include <linux/proc_fs.h>
@@ -620,8 +621,8 @@ static int de_thread(struct task_struct
   * Reparenting needs write_lock on tasklist_lock,
   * so it is safe to do it under read_lock.
   */
- if (unlikely(tsk->group_leader == child_reaper))
-  child_reaper = tsk;
+ if (unlikely(tsk->group_leader == tsk->nsproxy->pid_ns->child_reaper))
+  tsk->nsproxy->pid_ns->child_reaper = tsk;

  zap_other_threads(tsk);
  read_unlock(&tasklist_lock);
Index: lx26-18-rc6-mm2/include/linux/pid.h
===================================================================
--- lx26-18-rc6-mm2.orig/include/linux/pid.h 2006-09-18 15:46:12.000000000 -0700
+++ lx26-18-rc6-mm2/include/linux/pid.h 2006-09-18 16:05:13.000000000 -0700
@@ -35,8 +35,9 @@ enum pid_type
  *
  * Holding a reference to struct pid solves both of these problems.
  * It is small so holding a reference does not consume a lot of
- * resources, and since a new struct pid is allocated when the numeric
- * pid value is reused we don't mistakenly refer to new processes.
+ * resources, and since a new struct pid is allocated when the numeric pid
+ * value is reused (when pids wrap around) we don't mistakenly refer to new
+ * processes.
  */

 struct pid
Index: lx26-18-rc6-mm2/include/linux/sched.h
===================================================================
--- lx26-18-rc6-mm2.orig/include/linux/sched.h 2006-09-18 15:46:12.000000000 -0700
+++ lx26-18-rc6-mm2/include/linux/sched.h 2006-09-18 16:05:13.000000000 -0700
@@ -1380,7 +1380,6 @@ extern NORET_TYPE void do_group_exit(int
 extern void daemonize(const char *, ...);
 extern int allow_signal(int);
 extern int disallow_signal(int);
-extern struct task_struct *child_reaper;

 extern int do_execve(char *, char __user * __user *, char __user * __user *, struct pt_regs *);
 extern long do_fork(unsigned long, unsigned long, struct pt_regs *, unsigned long, int __user *,
int __user *);
Index: lx26-18-rc6-mm2/kernel/exit.c
===================================================================
--- lx26-18-rc6-mm2.orig/kernel/exit.c 2006-09-18 15:46:28.000000000 -0700
+++ lx26-18-rc6-mm2/kernel/exit.c 2006-09-18 16:05:13.000000000 -0700
@@ -22,6 +22,7 @@
 #include <linux/file.h>
```

```
 #include <linux/binfmts.h>
 #include <linux/nsproxy.h>
+#include <linux/pid_namespace.h>
 #include <linux/ptrace.h>
 #include <linux/profile.h>
 #include <linux/mount.h>
@@ -48,7 +49,6 @@
 #include <asm/mmu_context.h>

 extern void sem_exit (void);
-extern struct task_struct *child_reaper;

 static void exit_mm(struct task_struct * tsk);

@@ -259,7 +259,8 @@ static int has_stopped_jobs(int pgrp)
 }

 /**
- * reparent_to_init - Reparent the calling kernel thread to the init task.
+ * reparent_to_init - Reparent the calling kernel thread to the init task
+ * of the pid space that the thread belongs to.
  *
  * If a kernel thread is launched as a result of a system call, or if
  * it ever exits, it should generally reparent itself to init so that
@@ -277,8 +278,8 @@ static void reparent_to_init(void)
  ptrace_unlink(current);
  /* Reparent to init */
  remove_parent(current);
- current->parent = child_reaper;
- current->real_parent = child_reaper;
+ current->parent = current->nsproxy->pid_ns->child_reaper;
+ current->real_parent = current->nsproxy->pid_ns->child_reaper;
  add_parent(current);

  /* Set the exit signal to SIGCHLD so we signal init on exit */
@@ -662,7 +663,8 @@ reparent_thread(struct task_struct *p, s
  * When we die, we re-parent all our children.
  * Try to give them to another thread in our thread
  * group, and if no such member exists, give it to
- * the global child reaper process (ie "init")
+ * the child reaper process (ie "init") in our pid
+ * space.
  */
 static void
 forget_original_parent(struct task_struct *father, struct list_head *to_release)
@@ -673,7 +675,10 @@ forget_original_parent(struct task_struc
  do {
   reaper = next_thread(reaper);
```

```
   if (reaper == father) {
-   reaper = child_reaper;
+   /*
+    * FIXME: which reaper to use ?
+    */
+   reaper = init_pid_ns.child_reaper;
    break;
   }
  } while (reaper->exit_state);
@@ -861,8 +866,13 @@ fastcall NORET_TYPE void do_exit(long co
   panic("Aiee, killing interrupt handler!");
  if (unlikely(!tsk->pid))
   panic("Attempted to kill the idle task!");
- if (unlikely(tsk == child_reaper))
-  panic("Attempted to kill init!");
+ if (unlikely(tsk == tsk->nsproxy->pid_ns->child_reaper)) {
+  if (tsk->nsproxy->pid_ns != &init_pid_ns)
+   tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
+  else
+   panic("Attempted to kill init!");
+ }
+

  if (unlikely(current->ptrace & PT_TRACE_EXIT)) {
   current->ptrace_message = code;
Index: lx26-18-rc6-mm2/kernel/signal.c
===================================================================
--- lx26-18-rc6-mm2.orig/kernel/signal.c 2006-09-18 15:46:28.000000000 -0700
+++ lx26-18-rc6-mm2/kernel/signal.c 2006-09-18 16:05:13.000000000 -0700
@@ -24,6 +24,8 @@
 #include <linux/ptrace.h>
 #include <linux/signal.h>
 #include <linux/capability.h>
+#include <linux/pid_namespace.h>
+#include <linux/nsproxy.h>
 #include <asm/param.h>
 #include <asm/uaccess.h>
 #include <asm/unistd.h>
@@ -1994,8 +1996,12 @@ relock:
   if (sig_kernel_ignore(signr)) /* Default is nothing. */
    continue;

-  /* Init gets no signals it doesn't want.  */
-  if (current == child_reaper)
+  /*
+   * Init of a pid space gets no signals it doesn't want from
+   * within that pid space. It can of course get signals from
+   * its parent pid space.
```

```
+   */
+  if (current == current->nsproxy->pid_ns->child_reaper)
     continue;

   if (sig_kernel_stop(signr)) {
Index: lx26-18-rc6-mm2/include/linux/pid_namespace.h
===================================================================
--- lx26-18-rc6-mm2.orig/include/linux/pid_namespace.h 2006-09-18 16:04:44.000000000 -0700
+++ lx26-18-rc6-mm2/include/linux/pid_namespace.h 2006-09-18 16:05:13.000000000 -0700
@@ -8,15 +8,16 @@
 #include <linux/nsproxy.h>

 struct pidmap {
-       atomic_t nr_free;
-       void *page;
+ atomic_t nr_free;
+ void *page;
 };

 #define PIDMAP_ENTRIES         ((PID_MAX_LIMIT + 8*PAGE_SIZE - 1)/PAGE_SIZE/8)

 struct pid_namespace {
-       struct pidmap pidmap[PIDMAP_ENTRIES];
-       int last_pid;
+ struct pidmap pidmap[PIDMAP_ENTRIES];
+ int last_pid;
+ struct task_struct * child_reaper;
 };

 extern struct pid_namespace init_pid_ns;
Index: lx26-18-rc6-mm2/kernel/pid.c
===================================================================
--- lx26-18-rc6-mm2.orig/kernel/pid.c 2006-09-18 15:56:45.000000000 -0700
+++ lx26-18-rc6-mm2/kernel/pid.c 2006-09-18 16:05:13.000000000 -0700
@@ -62,7 +62,8 @@ struct pid_namespace init_pid_ns = {
   .pidmap = {
    [ 0 ... PIDMAP_ENTRIES-1] = { ATOMIC_INIT(BITS_PER_PAGE), NULL }
   },
- .last_pid = 0
+ .last_pid = 0,
+ .child_reaper = &init_task
 };

 /*
```

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