Subject: [PATCH 2/6] Rename pid_nr function
Posted by Sukadev Bhattiprolu on Fri, 06 Jul 2007 05:53:16 GMT
View Forum Message <> Reply to Message

Subject: [PATCH 2/6] Rename pid_nr function

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

Rename pid_nr() function to pid_to_nr() which is more descriptive
and will hopefully cause less confusion with new structure/functions
being added to support multiple pid namespaces.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
---
 drivers/char/n_r3964.c | 7 ++++---
 drivers/char/tty_io.c | 4 ++--
 fs/autofs/root.c | 2 +-
 fs/fcntl.c | 2 +-
 fs/proc/array.c | 2 +-
 fs/smbfs/proc.c | 2 +-
 include/linux/pid.h | 2 +-
 ipc/mqueue.c | 2 +-
 kernel/container.c | 2 +-
 kernel/fork.c | 2 +-
 kernel/sysctl.c | 2 +-
 11 files changed, 15 insertions(+), 14 deletions(-)

Index: lx26-22-rc6-mm1/include/linux/pid.h
===
--- lx26-22-rc6-mm1.orig/include/linux/pid.h 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/include/linux/pid.h 2007-07-05 18:54:29.000000000 -0700
@@ -98,7 +98,7 @@ extern struct pid *find_ge_pid(int nr);
 extern struct pid *alloc_pid(void);
 extern void FASTCALL(free_pid(struct pid *pid));

-static inline pid_t pid_nr(struct pid *pid)
+static inline pid_t pid_to_nr(struct pid *pid)
 {
  pid_t nr = 0;
  if (pid)
Index: lx26-22-rc6-mm1/drivers/char/n_r3964.c
===
--- lx26-22-rc6-mm1.orig/drivers/char/n_r3964.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/drivers/char/n_r3964.c 2007-07-05 18:54:29.000000000 -0700
@@ -771,7 +771,7 @@ static int enable_signals(struct r3964_i

    if (pClient->pid == pid) {
     TRACE_PS("removing client %d from client list",

```
-      pid_nr(pid));
+      pid_to_nr(pid));
     *ppClient = pClient->next;
     while (pClient->msg_count) {
      pMsg = remove_msg(pInfo, pClient);
@@ -801,7 +801,8 @@ static int enable_signals(struct r3964_i
    if (pClient == NULL)
     return -ENOMEM;

-   TRACE_PS("add client %d to client list", pid_nr(pid));
+   TRACE_PS("add client %d to client list",
+     pid_to_nr(pid));
    spin_lock_init(&pClient->lock);
    pClient->sig_flags = arg;
    pClient->pid = get_pid(pid);
@@ -933,7 +934,7 @@ static void remove_client_block(struct r
 {
  struct r3964_block_header *block;

- TRACE_PS("remove_client_block PID %d", pid_nr(pClient->pid));
+ TRACE_PS("remove_client_block PID %d", pid_to_nr(pClient->pid));

  block = pClient->next_block_to_read;
  if (block) {
Index: lx26-22-rc6-mm1/drivers/char/tty_io.c
===================================================================
--- lx26-22-rc6-mm1.orig/drivers/char/tty_io.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/drivers/char/tty_io.c 2007-07-05 18:54:29.000000000 -0700
@@ -3080,7 +3080,7 @@ static int tiocgpgrp(struct tty_struct *
   */
  if (tty == real_tty && current->signal->tty != real_tty)
   return -ENOTTY;
- return put_user(pid_nr(real_tty->pgrp), p);
+ return put_user(pid_to_nr(real_tty->pgrp), p);
 }

 /**
@@ -3151,7 +3151,7 @@ static int tiocgsid(struct tty_struct *t
   return -ENOTTY;
  if (!real_tty->session)
   return -ENOTTY;
- return put_user(pid_nr(real_tty->session), p);
+ return put_user(pid_to_nr(real_tty->session), p);
 }

 /**
Index: lx26-22-rc6-mm1/fs/autofs/root.c
===================================================================
```

```
--- lx26-22-rc6-mm1.orig/fs/autofs/root.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/fs/autofs/root.c 2007-07-05 18:54:29.000000000 -0700
@@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup

  oz_mode = autofs_oz_mode(sbi);
  DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
-    "oz_mode = %d\n", pid_nr(task_pid(current)),
+    "oz_mode = %d\n", pid_to_nr(task_pid(current)),
     process_group(current), sbi->catatonic,
     oz_mode));

Index: lx26-22-rc6-mm1/fs/fcntl.c
===================================================================
--- lx26-22-rc6-mm1.orig/fs/fcntl.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/fs/fcntl.c 2007-07-05 18:54:29.000000000 -0700
@@ -305,7 +305,7 @@ pid_t f_getown(struct file *filp)
 {
  pid_t pid;
  read_lock(&filp->f_owner.lock);
- pid = pid_nr(filp->f_owner.pid);
+ pid = pid_to_nr(filp->f_owner.pid);
  if (filp->f_owner.pid_type == PIDTYPE_PGID)
   pid = -pid;
  read_unlock(&filp->f_owner.lock);
Index: lx26-22-rc6-mm1/fs/proc/array.c
===================================================================
--- lx26-22-rc6-mm1.orig/fs/proc/array.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/fs/proc/array.c 2007-07-05 18:54:29.000000000 -0700
@@ -398,7 +398,7 @@ static int do_task_stat(struct task_stru
   struct signal_struct *sig = task->signal;

   if (sig->tty) {
-   tty_pgrp = pid_nr(sig->tty->pgrp);
+   tty_pgrp = pid_to_nr(sig->tty->pgrp);
    tty_nr = new_encode_dev(tty_devnum(sig->tty));
   }

Index: lx26-22-rc6-mm1/fs/smbfs/proc.c
===================================================================
--- lx26-22-rc6-mm1.orig/fs/smbfs/proc.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/fs/smbfs/proc.c 2007-07-05 18:54:29.000000000 -0700
@@ -972,7 +972,7 @@ smb_newconn(struct smb_sb_info *server,

  VERBOSE("protocol=%d, max_xmit=%d, pid=%d capabilities=0x%x\n",
   server->opt.protocol, server->opt.max_xmit,
-  pid_nr(server->conn_pid), server->opt.capabilities);
+  pid_to_nr(server->conn_pid), server->opt.capabilities);
```

```
  /* FIXME: this really should be done by smbmount. */
  if (server->opt.max_xmit > SMB_MAX_PACKET_SIZE) {
Index: lx26-22-rc6-mm1/ipc/mqueue.c
===================================================================
--- lx26-22-rc6-mm1.orig/ipc/mqueue.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/ipc/mqueue.c 2007-07-05 18:54:29.000000000 -0700
@@ -336,7 +336,7 @@ static ssize_t mqueue_read_file(struct f
     (info->notify_owner &&
      info->notify.sigev_notify == SIGEV_SIGNAL) ?
      info->notify.sigev_signo : 0,
-    pid_nr(info->notify_owner));
+    pid_to_nr(info->notify_owner));
   spin_unlock(&info->lock);
   buffer[sizeof(buffer)-1] = '\0';
   slen = strlen(buffer)+1;
Index: lx26-22-rc6-mm1/kernel/fork.c
===================================================================
--- lx26-22-rc6-mm1.orig/kernel/fork.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/kernel/fork.c 2007-07-05 18:54:29.000000000 -0700
@@ -1032,7 +1032,7 @@ static struct task_struct *copy_process(
   p->did_exec = 0;
   delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
   copy_flags(clone_flags, p);
- p->pid = pid_nr(pid);
+ p->pid = pid_to_nr(pid);
   INIT_LIST_HEAD(&p->children);
   INIT_LIST_HEAD(&p->sibling);
   p->vfork_done = NULL;
Index: lx26-22-rc6-mm1/kernel/sysctl.c
===================================================================
--- lx26-22-rc6-mm1.orig/kernel/sysctl.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/kernel/sysctl.c 2007-07-05 18:54:29.000000000 -0700
@@ -2313,7 +2313,7 @@ static int proc_do_cad_pid(ctl_table *ta
   pid_t tmp;
   int r;

- tmp = pid_nr(cad_pid);
+ tmp = pid_to_nr(cad_pid);

   r = __do_proc_dointvec(&tmp, table, write, filp, buffer,
           lenp, ppos, NULL, NULL);
Index: lx26-22-rc6-mm1/kernel/container.c
===================================================================
--- lx26-22-rc6-mm1.orig/kernel/container.c 2007-07-05 18:53:43.000000000 -0700
+++ lx26-22-rc6-mm1/kernel/container.c 2007-07-05 18:54:29.000000000 -0700
@@ -1600,7 +1600,7 @@ static int pid_array_load(pid_t *pidarra
   while ((tsk = container_iter_next(cont, &it))) {
    if (unlikely(n == npids))
```

```
    break;
-  pidarray[n++] = pid_nr(task_pid(tsk));
+  pidarray[n++] = pid_to_nr(task_pid(tsk));
  }
  container_iter_end(cont, &it);
  return n;
```

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/6] Rename pid_nr function
Posted by Pavel Emelianov on Fri, 13 Jul 2007 05:26:18 GMT
View Forum Message <> Reply to Message

sukadev@us.ibm.com wrote:
> Subject: [PATCH 2/6] Rename pid_nr function
>
> From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
>
> Rename pid_nr() function to pid_to_nr() which is more descriptive
> and will hopefully cause less confusion with new structure/functions
> being added to support multiple pid namespaces.

Don't we need at least two convertors:
pid_to_nr_as_it_seen_by_current()
and
pid_to_nr_as_it_seen_by_some_other_task()

The first one is essentially the second one with "some other
task" being current, but it can be optimized...

> Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> ---
> drivers/char/n_r3964.c | 7 ++++---
> drivers/char/tty_io.c | 4 ++--
> fs/autofs/root.c | 2 +-
> fs/fcntl.c | 2 +-
> fs/proc/array.c | 2 +-
> fs/smbfs/proc.c | 2 +-
> include/linux/pid.h | 2 +-
> ipc/mqueue.c | 2 +-
> kernel/container.c | 2 +-
> kernel/fork.c | 2 +-
> kernel/sysctl.c | 2 +-
> 11 files changed, 15 insertions(+), 14 deletions(-)
>

```
> Index: lx26-22-rc6-mm1/include/linux/pid.h
> ===================================================================
> --- lx26-22-rc6-mm1.orig/include/linux/pid.h 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/include/linux/pid.h 2007-07-05 18:54:29.000000000 -0700
> @@ -98,7 +98,7 @@ extern struct pid *find_ge_pid(int nr);
>  extern struct pid *alloc_pid(void);
>  extern void FASTCALL(free_pid(struct pid *pid));
>
> -static inline pid_t pid_nr(struct pid *pid)
> +static inline pid_t pid_to_nr(struct pid *pid)
>  {
>   pid_t nr = 0;
>   if (pid)
> Index: lx26-22-rc6-mm1/drivers/char/n_r3964.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/drivers/char/n_r3964.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/drivers/char/n_r3964.c 2007-07-05 18:54:29.000000000 -0700
> @@ -771,7 +771,7 @@ static int enable_signals(struct r3964_i
>
>     if (pClient->pid == pid) {
>      TRACE_PS("removing client %d from client list",
> -      pid_nr(pid));
> +      pid_to_nr(pid));
>      *ppClient = pClient->next;
>      while (pClient->msg_count) {
>       pMsg = remove_msg(pInfo, pClient);
> @@ -801,7 +801,8 @@ static int enable_signals(struct r3964_i
>     if (pClient == NULL)
>      return -ENOMEM;
>
> -   TRACE_PS("add client %d to client list", pid_nr(pid));
> +   TRACE_PS("add client %d to client list",
> +     pid_to_nr(pid));
>     spin_lock_init(&pClient->lock);
>     pClient->sig_flags = arg;
>     pClient->pid = get_pid(pid);
> @@ -933,7 +934,7 @@ static void remove_client_block(struct r
>  {
>   struct r3964_block_header *block;
>
> - TRACE_PS("remove_client_block PID %d", pid_nr(pClient->pid));
> + TRACE_PS("remove_client_block PID %d", pid_to_nr(pClient->pid));
>
>   block = pClient->next_block_to_read;
>   if (block) {
> Index: lx26-22-rc6-mm1/drivers/char/tty_io.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/drivers/char/tty_io.c 2007-07-05 18:53:43.000000000 -0700
```

```
> +++ lx26-22-rc6-mm1/drivers/char/tty_io.c 2007-07-05 18:54:29.000000000 -0700
> @@ -3080,7 +3080,7 @@ static int tiocgpgrp(struct tty_struct *
>     */
>    if (tty == real_tty && current->signal->tty != real_tty)
>     return -ENOTTY;
> - return put_user(pid_nr(real_tty->pgrp), p);
> + return put_user(pid_to_nr(real_tty->pgrp), p);
>  }
>
>  /**
> @@ -3151,7 +3151,7 @@ static int tiocgsid(struct tty_struct *t
>     return -ENOTTY;
>    if (!real_tty->session)
>     return -ENOTTY;
> - return put_user(pid_nr(real_tty->session), p);
> + return put_user(pid_to_nr(real_tty->session), p);
>  }
>
>  /**
> Index: lx26-22-rc6-mm1/fs/autofs/root.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/fs/autofs/root.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/fs/autofs/root.c 2007-07-05 18:54:29.000000000 -0700
> @@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup
>
>    oz_mode = autofs_oz_mode(sbi);
>    DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
> -     "oz_mode = %d\n", pid_nr(task_pid(current)),
> +     "oz_mode = %d\n", pid_to_nr(task_pid(current)),
>       process_group(current), sbi->catatonic,
>       oz_mode));
>
> Index: lx26-22-rc6-mm1/fs/fcntl.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/fs/fcntl.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/fs/fcntl.c 2007-07-05 18:54:29.000000000 -0700
> @@ -305,7 +305,7 @@ pid_t f_getown(struct file *filp)
>  {
>    pid_t pid;
>    read_lock(&filp->f_owner.lock);
> - pid = pid_nr(filp->f_owner.pid);
> + pid = pid_to_nr(filp->f_owner.pid);
>    if (filp->f_owner.pid_type == PIDTYPE_PGID)
>     pid = -pid;
>    read_unlock(&filp->f_owner.lock);
> Index: lx26-22-rc6-mm1/fs/proc/array.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/fs/proc/array.c 2007-07-05 18:53:43.000000000 -0700
```

```
> +++ lx26-22-rc6-mm1/fs/proc/array.c 2007-07-05 18:54:29.000000000 -0700
> @@ -398,7 +398,7 @@ static int do_task_stat(struct task_stru
>    struct signal_struct *sig = task->signal;
>
>    if (sig->tty) {
> -   tty_pgrp = pid_nr(sig->tty->pgrp);
> +   tty_pgrp = pid_to_nr(sig->tty->pgrp);
>     tty_nr = new_encode_dev(tty_devnum(sig->tty));
>    }
>
> Index: lx26-22-rc6-mm1/fs/smbfs/proc.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/fs/smbfs/proc.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/fs/smbfs/proc.c 2007-07-05 18:54:29.000000000 -0700
> @@ -972,7 +972,7 @@ smb_newconn(struct smb_sb_info *server,
>
>   VERBOSE("protocol=%d, max_xmit=%d, pid=%d capabilities=0x%x\n",
>    server->opt.protocol, server->opt.max_xmit,
> -  pid_nr(server->conn_pid), server->opt.capabilities);
> +  pid_to_nr(server->conn_pid), server->opt.capabilities);
>
>   /* FIXME: this really should be done by smbmount. */
>   if (server->opt.max_xmit > SMB_MAX_PACKET_SIZE) {
> Index: lx26-22-rc6-mm1/ipc/mqueue.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/ipc/mqueue.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/ipc/mqueue.c 2007-07-05 18:54:29.000000000 -0700
> @@ -336,7 +336,7 @@ static ssize_t mqueue_read_file(struct f
>     (info->notify_owner &&
>      info->notify.sigev_notify == SIGEV_SIGNAL) ?
>      info->notify.sigev_signo : 0,
> -   pid_nr(info->notify_owner));
> +   pid_to_nr(info->notify_owner));
>   spin_unlock(&info->lock);
>   buffer[sizeof(buffer)-1] = '\0';
>   slen = strlen(buffer)+1;
> Index: lx26-22-rc6-mm1/kernel/fork.c
> ===================================================================
> --- lx26-22-rc6-mm1.orig/kernel/fork.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/kernel/fork.c 2007-07-05 18:54:29.000000000 -0700
> @@ -1032,7 +1032,7 @@ static struct task_struct *copy_process(
>   p->did_exec = 0;
>   delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
>   copy_flags(clone_flags, p);
> - p->pid = pid_nr(pid);
> + p->pid = pid_to_nr(pid);
>   INIT_LIST_HEAD(&p->children);
>   INIT_LIST_HEAD(&p->sibling);
```

> p->vfork_done = NULL;
> Index: lx26-22-rc6-mm1/kernel/sysctl.c
> ===
> --- lx26-22-rc6-mm1.orig/kernel/sysctl.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/kernel/sysctl.c 2007-07-05 18:54:29.000000000 -0700
> @@ -2313,7 +2313,7 @@ static int proc_do_cad_pid(ctl_table *ta
> pid_t tmp;
> int r;
>
> - tmp = pid_nr(cad_pid);
> + tmp = pid_to_nr(cad_pid);
>
> r = __do_proc_dointvec(&tmp, table, write, filp, buffer,
> lenp, ppos, NULL, NULL);
> Index: lx26-22-rc6-mm1/kernel/container.c
> ===
> --- lx26-22-rc6-mm1.orig/kernel/container.c 2007-07-05 18:53:43.000000000 -0700
> +++ lx26-22-rc6-mm1/kernel/container.c 2007-07-05 18:54:29.000000000 -0700
> @@ -1600,7 +1600,7 @@ static int pid_array_load(pid_t *pidarra
> while ((tsk = container_iter_next(cont, &it))) {
> if (unlikely(n == npids))
> break;
> - pidarray[n++] = pid_nr(task_pid(tsk));
> + pidarray[n++] = pid_to_nr(task_pid(tsk));
> }
> container_iter_end(cont, &it);
> return n;
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers
>

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/6] Rename pid_nr function
Posted by Sukadev Bhattiprolu on Fri, 13 Jul 2007 05:55:43 GMT
View Forum Message <> Reply to Message

Pavel Emelianov [xemul@openvz.org] wrote:
| sukadev@us.ibm.com wrote:
| > Subject: [PATCH 2/6] Rename pid_nr function
| >
| > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
| >

| > Rename pid_nr() function to pid_to_nr() which is more descriptive
| > and will hopefully cause less confusion with new structure/functions
| > being added to support multiple pid namespaces.
|
| Don't we need at least two convertors:
| pid_to_nr_as_it_seen_by_current()
| and
| pid_to_nr_as_it_seen_by_some_other_task()

Can you give me an example of where you would use this latter
(pid_to_nr_as_it_seen_by_some_other_task()) interface ?

I have a similar confusion with following two sets of interfaces
in your patchset and have been meaning to discuss it further.

 pid_nr() and pid_vnr()
 find_pid() and find_vpid()

I can understand the use of pid_vnr() and find_vpid().

Where do you see us using pid_nr() and find_pid() - i.e the values
as seen in init_pid_ns ?

Like I pointed out in another thread, sunos_killpg() for instance
needs to used find_vpid(). Similarly, kill_proc_info(),
__set_special_pids() etc need find_vpid().

Thanks,

Suka

|
| The first one is essentially the second one with "some other
| task" being current, but it can be optimized...
|
| > Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
| > ---
| > drivers/char/n_r3964.c | 7 ++++---
| > drivers/char/tty_io.c | 4 ++--
| > fs/autofs/root.c | 2 +-
| > fs/fcntl.c | 2 +-
| > fs/proc/array.c | 2 +-
| > fs/smbfs/proc.c | 2 +-
| > include/linux/pid.h | 2 +-
| > ipc/mqueue.c | 2 +-
| > kernel/container.c | 2 +-
| > kernel/fork.c | 2 +-

```
| >  kernel/sysctl.c        |    2 +-
| >  11 files changed, 15 insertions(+), 14 deletions(-)
| >
| > Index: lx26-22-rc6-mm1/include/linux/pid.h
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/include/linux/pid.h 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/include/linux/pid.h 2007-07-05 18:54:29.000000000 -0700
| > @@ -98,7 +98,7 @@ extern struct pid *find_ge_pid(int nr);
| >  extern struct pid *alloc_pid(void);
| >  extern void FASTCALL(free_pid(struct pid *pid));
| >
| > -static inline pid_t pid_nr(struct pid *pid)
| > +static inline pid_t pid_to_nr(struct pid *pid)
| >  {
| >   pid_t nr = 0;
| >   if (pid)
| > Index: lx26-22-rc6-mm1/drivers/char/n_r3964.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/drivers/char/n_r3964.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/drivers/char/n_r3964.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -771,7 +771,7 @@ static int enable_signals(struct r3964_i
| >
| >     if (pClient->pid == pid) {
| >      TRACE_PS("removing client %d from client list",
| > -      pid_nr(pid));
| > +      pid_to_nr(pid));
| >      *ppClient = pClient->next;
| >      while (pClient->msg_count) {
| >       pMsg = remove_msg(pInfo, pClient);
| > @@ -801,7 +801,8 @@ static int enable_signals(struct r3964_i
| >     if (pClient == NULL)
| >      return -ENOMEM;
| >
| > -   TRACE_PS("add client %d to client list", pid_nr(pid));
| > +   TRACE_PS("add client %d to client list",
| > +     pid_to_nr(pid));
| >     spin_lock_init(&pClient->lock);
| >     pClient->sig_flags = arg;
| >     pClient->pid = get_pid(pid);
| > @@ -933,7 +934,7 @@ static void remove_client_block(struct r
| >  {
| >   struct r3964_block_header *block;
| >
| > - TRACE_PS("remove_client_block PID %d", pid_nr(pClient->pid));
| > + TRACE_PS("remove_client_block PID %d", pid_to_nr(pClient->pid));
| >
| >   block = pClient->next_block_to_read;
| >   if (block) {
```

```
| > Index: lx26-22-rc6-mm1/drivers/char/tty_io.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/drivers/char/tty_io.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/drivers/char/tty_io.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -3080,7 +3080,7 @@ static int tiocgpgrp(struct tty_struct *
| >    */
| >   if (tty == real_tty && current->signal->tty != real_tty)
| >    return -ENOTTY;
| > - return put_user(pid_nr(real_tty->pgrp), p);
| > + return put_user(pid_to_nr(real_tty->pgrp), p);
| >  }
| >
| >  /**
| > @@ -3151,7 +3151,7 @@ static int tiocgsid(struct tty_struct *t
| >    return -ENOTTY;
| >   if (!real_tty->session)
| >    return -ENOTTY;
| > - return put_user(pid_nr(real_tty->session), p);
| > + return put_user(pid_to_nr(real_tty->session), p);
| >  }
| >
| >  /**
| > Index: lx26-22-rc6-mm1/fs/autofs/root.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/fs/autofs/root.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/fs/autofs/root.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup
| >
| >   oz_mode = autofs_oz_mode(sbi);
| >   DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
| > -    "oz_mode = %d\n", pid_nr(task_pid(current)),
| > +    "oz_mode = %d\n", pid_to_nr(task_pid(current)),
| >      process_group(current), sbi->catatonic,
| >      oz_mode));
| >
| > Index: lx26-22-rc6-mm1/fs/fcntl.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/fs/fcntl.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/fs/fcntl.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -305,7 +305,7 @@ pid_t f_getown(struct file *filp)
| >  {
| >   pid_t pid;
| >   read_lock(&filp->f_owner.lock);
| > - pid = pid_nr(filp->f_owner.pid);
| > + pid = pid_to_nr(filp->f_owner.pid);
| >   if (filp->f_owner.pid_type == PIDTYPE_PGID)
| >    pid = -pid;
| >   read_unlock(&filp->f_owner.lock);
```

```
| > Index: lx26-22-rc6-mm1/fs/proc/array.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/fs/proc/array.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/fs/proc/array.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -398,7 +398,7 @@ static int do_task_stat(struct task_stru
| >    struct signal_struct *sig = task->signal;
| >
| >    if (sig->tty) {
| > -   tty_pgrp = pid_nr(sig->tty->pgrp);
| > +   tty_pgrp = pid_to_nr(sig->tty->pgrp);
| >     tty_nr = new_encode_dev(tty_devnum(sig->tty));
| >    }
| >
| > Index: lx26-22-rc6-mm1/fs/smbfs/proc.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/fs/smbfs/proc.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/fs/smbfs/proc.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -972,7 +972,7 @@ smb_newconn(struct smb_sb_info *server,
| >
| >   VERBOSE("protocol=%d, max_xmit=%d, pid=%d capabilities=0x%x\n",
| >    server->opt.protocol, server->opt.max_xmit,
| > -  pid_nr(server->conn_pid), server->opt.capabilities);
| > +  pid_to_nr(server->conn_pid), server->opt.capabilities);
| >
| >   /* FIXME: this really should be done by smbmount. */
| >   if (server->opt.max_xmit > SMB_MAX_PACKET_SIZE) {
| > Index: lx26-22-rc6-mm1/ipc/mqueue.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/ipc/mqueue.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/ipc/mqueue.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -336,7 +336,7 @@ static ssize_t mqueue_read_file(struct f
| >     (info->notify_owner &&
| >      info->notify.sigev_notify == SIGEV_SIGNAL) ?
| >      info->notify.sigev_signo : 0,
| > -   pid_nr(info->notify_owner));
| > +   pid_to_nr(info->notify_owner));
| >   spin_unlock(&info->lock);
| >   buffer[sizeof(buffer)-1] = '\0';
| >   slen = strlen(buffer)+1;
| > Index: lx26-22-rc6-mm1/kernel/fork.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/kernel/fork.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/kernel/fork.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -1032,7 +1032,7 @@ static struct task_struct *copy_process(
| >   p->did_exec = 0;
| >   delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
| >   copy_flags(clone_flags, p);
| > - p->pid = pid_nr(pid);
```

```
| > + p->pid = pid_to_nr(pid);
| >   INIT_LIST_HEAD(&p->children);
| >   INIT_LIST_HEAD(&p->sibling);
| >   p->vfork_done = NULL;
| > Index: lx26-22-rc6-mm1/kernel/sysctl.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/kernel/sysctl.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/kernel/sysctl.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -2313,7 +2313,7 @@ static int proc_do_cad_pid(ctl_table *ta
| >   pid_t tmp;
| >   int r;
| >
| > - tmp = pid_nr(cad_pid);
| > + tmp = pid_to_nr(cad_pid);
| >
| >   r = __do_proc_dointvec(&tmp, table, write, filp, buffer,
| >           lenp, ppos, NULL, NULL);
| > Index: lx26-22-rc6-mm1/kernel/container.c
| > ===================================================================
| > --- lx26-22-rc6-mm1.orig/kernel/container.c 2007-07-05 18:53:43.000000000 -0700
| > +++ lx26-22-rc6-mm1/kernel/container.c 2007-07-05 18:54:29.000000000 -0700
| > @@ -1600,7 +1600,7 @@ static int pid_array_load(pid_t *pidarra
| >   while ((tsk = container_iter_next(cont, &it))) {
| >    if (unlikely(n == npids))
| >     break;
| > -  pidarray[n++] = pid_nr(task_pid(tsk));
| > +  pidarray[n++] = pid_to_nr(task_pid(tsk));
| >   }
| >   container_iter_end(cont, &it);
| >   return n;
| > _____________________________________________
| > Containers mailing list
| > Containers@lists.linux-foundation.org
| > https://lists.linux-foundation.org/mailman/listinfo/containers
| >
| > _____________________________________________
| > Devel mailing list
| > Devel@openvz.org
| > https://openvz.org/mailman/listinfo/devel
| >

_____________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers
```

Subject: Re: [PATCH 2/6] Rename pid_nr function

sukadev@us.ibm.com wrote:
> Pavel Emelianov [xemul@openvz.org] wrote:
> | sukadev@us.ibm.com wrote:
> | > Subject: [PATCH 2/6] Rename pid_nr function
> | >
> | > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> | >
> | > Rename pid_nr() function to pid_to_nr() which is more descriptive
> | > and will hopefully cause less confusion with new structure/functions
> | > being added to support multiple pid namespaces.
> |
> | Don't we need at least two convertors:
> | pid_to_nr_as_it_seen_by_current()
> | and
> | pid_to_nr_as_it_seen_by_some_other_task()
>
> Can you give me an example of where you would use this latter
> (pid_to_nr_as_it_seen_by_some_other_task()) interface ?

Easy. Consider you're observing /proc/<pid>/status file for a task
that lives in two namespaces - init and some created. When making
"cat" on this from init namespace you must get the task's pid as it
is seen from init namespace, but when making "cat" from the created
namespace you must see the pid as it is seen by this namespace. And
so on and so forth.

On the other hand sys_getpid() should simply return the virtual pid
without messing with namespaces.

> I have a similar confusion with following two sets of interfaces
> in your patchset and have been meaning to discuss it further.
>
> pid_nr() and pid_vnr()
> find_pid() and find_vpid()
>
> I can understand the use of pid_vnr() and find_vpid().
>
> Where do you see us using pid_nr() and find_pid() - i.e the values
> as seen in init_pid_ns ?

find_pid() may be used in keventd_create_thread() as eventd thread
lives in init namespace only. Any other place in kernel that is
somehow "global" to all the namespaces and has no user interface
must use global pids not to confuse the kernel.

Of course, if we plan to replace the notion of global numerical pid

with struct pid this is not an option. However, the issues with the
pid_to_nr_as_it_... are still valid.

> Like I pointed out in another thread, sunos_killpg() for instance
> needs to used find_vpid(). Similarly, kill_proc_info(),
> __set_special_pids() etc need find_vpid().

Sure. These places are to be patched as well.

> Thanks,
>
> Suka
>
>
> |
> | The first one is essentially the second one with "some other
> | task" being current, but it can be optimized...
> |
> | > Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> | > ---
> | > drivers/char/n_r3964.c | 7 ++++---
> | > drivers/char/tty_io.c | 4 ++--
> | > fs/autofs/root.c | 2 +-
> | > fs/fcntl.c | 2 +-
> | > fs/proc/array.c | 2 +-
> | > fs/smbfs/proc.c | 2 +-
> | > include/linux/pid.h | 2 +-
> | > ipc/mqueue.c | 2 +-
> | > kernel/container.c | 2 +-
> | > kernel/fork.c | 2 +-
> | > kernel/sysctl.c | 2 +-
> | > 11 files changed, 15 insertions(+), 14 deletions(-)
> | >
> | > Index: lx26-22-rc6-mm1/include/linux/pid.h
> | > ==
> | > --- lx26-22-rc6-mm1.orig/include/linux/pid.h 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/include/linux/pid.h 2007-07-05 18:54:29.000000000 -0700
> | > @@ -98,7 +98,7 @@ extern struct pid *find_ge_pid(int nr);
> | > extern struct pid *alloc_pid(void);
> | > extern void FASTCALL(free_pid(struct pid *pid));
> | >
> | > -static inline pid_t pid_nr(struct pid *pid)
> | > +static inline pid_t pid_to_nr(struct pid *pid)
> | > {
> | >   pid_t nr = 0;
> | >   if (pid)
> | > Index: lx26-22-rc6-mm1/drivers/char/n_r3964.c
> | > ==

```
> | > --- lx26-22-rc6-mm1.orig/drivers/char/n_r3964.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/drivers/char/n_r3964.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -771,7 +771,7 @@ static int enable_signals(struct r3964_i
> | >
> | >     if (pClient->pid == pid) {
> | >      TRACE_PS("removing client %d from client list",
> | > -      pid_nr(pid));
> | > +      pid_to_nr(pid));
> | >      *ppClient = pClient->next;
> | >      while (pClient->msg_count) {
> | >       pMsg = remove_msg(pInfo, pClient);
> | > @@ -801,7 +801,8 @@ static int enable_signals(struct r3964_i
> | >     if (pClient == NULL)
> | >      return -ENOMEM;
> | >
> | > -   TRACE_PS("add client %d to client list", pid_nr(pid));
> | > +   TRACE_PS("add client %d to client list",
> | > +     pid_to_nr(pid));
> | >     spin_lock_init(&pClient->lock);
> | >     pClient->sig_flags = arg;
> | >     pClient->pid = get_pid(pid);
> | > @@ -933,7 +934,7 @@ static void remove_client_block(struct r
> | >  {
> | >   struct r3964_block_header *block;
> | >
> | > - TRACE_PS("remove_client_block PID %d", pid_nr(pClient->pid));
> | > + TRACE_PS("remove_client_block PID %d", pid_to_nr(pClient->pid));
> | >
> | >   block = pClient->next_block_to_read;
> | >   if (block) {
> | > Index: lx26-22-rc6-mm1/drivers/char/tty_io.c
> | > ===================================================================
> | > --- lx26-22-rc6-mm1.orig/drivers/char/tty_io.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/drivers/char/tty_io.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -3080,7 +3080,7 @@ static int tiocgpgrp(struct tty_struct *
> | >    */
> | >   if (tty == real_tty && current->signal->tty != real_tty)
> | >    return -ENOTTY;
> | > - return put_user(pid_nr(real_tty->pgrp), p);
> | > + return put_user(pid_to_nr(real_tty->pgrp), p);
> | >  }
> | >
> | >  /**
> | > @@ -3151,7 +3151,7 @@ static int tiocgsid(struct tty_struct *t
> | >    return -ENOTTY;
> | >   if (!real_tty->session)
> | >    return -ENOTTY;
> | > - return put_user(pid_nr(real_tty->session), p);
```

> | > + return put_user(pid_to_nr(real_tty->session), p);
> | > }
> | >
> | > /**
> | > Index: lx26-22-rc6-mm1/fs/autofs/root.c
> | > ===
> | > --- lx26-22-rc6-mm1.orig/fs/autofs/root.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/fs/autofs/root.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup
> | >
> | > oz_mode = autofs_oz_mode(sbi);
> | > DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
> | > - "oz_mode = %d\n", pid_nr(task_pid(current)),
> | > + "oz_mode = %d\n", pid_to_nr(task_pid(current)),
> | > process_group(current), sbi->catatonic,
> | > oz_mode));
> | >
> | > Index: lx26-22-rc6-mm1/fs/fcntl.c
> | > ===
> | > --- lx26-22-rc6-mm1.orig/fs/fcntl.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/fs/fcntl.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -305,7 +305,7 @@ pid_t f_getown(struct file *filp)
> | > {
> | > pid_t pid;
> | > read_lock(&filp->f_owner.lock);
> | > - pid = pid_nr(filp->f_owner.pid);
> | > + pid = pid_to_nr(filp->f_owner.pid);
> | > if (filp->f_owner.pid_type == PIDTYPE_PGID)
> | > pid = -pid;
> | > read_unlock(&filp->f_owner.lock);
> | > Index: lx26-22-rc6-mm1/fs/proc/array.c
> | > ===
> | > --- lx26-22-rc6-mm1.orig/fs/proc/array.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/fs/proc/array.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -398,7 +398,7 @@ static int do_task_stat(struct task_stru
> | > struct signal_struct *sig = task->signal;
> | >
> | > if (sig->tty) {
> | > - tty_pgrp = pid_nr(sig->tty->pgrp);
> | > + tty_pgrp = pid_to_nr(sig->tty->pgrp);
> | > tty_nr = new_encode_dev(tty_devnum(sig->tty));
> | > }
> | >
> | > Index: lx26-22-rc6-mm1/fs/smbfs/proc.c
> | > ===
> | > --- lx26-22-rc6-mm1.orig/fs/smbfs/proc.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/fs/smbfs/proc.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -972,7 +972,7 @@ smb_newconn(struct smb_sb_info *server,

```
> | >
> | >   VERBOSE("protocol=%d, max_xmit=%d, pid=%d capabilities=0x%x\n",
> | >    server->opt.protocol, server->opt.max_xmit,
> | > -  pid_nr(server->conn_pid), server->opt.capabilities);
> | > +  pid_to_nr(server->conn_pid), server->opt.capabilities);
> | >
> | >   /* FIXME: this really should be done by smbmount. */
> | >   if (server->opt.max_xmit > SMB_MAX_PACKET_SIZE) {
> | > Index: lx26-22-rc6-mm1/ipc/mqueue.c
> | > ===================================================================
> | > --- lx26-22-rc6-mm1.orig/ipc/mqueue.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/ipc/mqueue.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -336,7 +336,7 @@ static ssize_t mqueue_read_file(struct f
> | >     (info->notify_owner &&
> | >      info->notify.sigev_notify == SIGEV_SIGNAL) ?
> | >      info->notify.sigev_signo : 0,
> | > -   pid_nr(info->notify_owner));
> | > +   pid_to_nr(info->notify_owner));
> | >   spin_unlock(&info->lock);
> | >   buffer[sizeof(buffer)-1] = '\0';
> | >   slen = strlen(buffer)+1;
> | > Index: lx26-22-rc6-mm1/kernel/fork.c
> | > ===================================================================
> | > --- lx26-22-rc6-mm1.orig/kernel/fork.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/kernel/fork.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -1032,7 +1032,7 @@ static struct task_struct *copy_process(
> | >   p->did_exec = 0;
> | >   delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
> | >   copy_flags(clone_flags, p);
> | > - p->pid = pid_nr(pid);
> | > + p->pid = pid_to_nr(pid);
> | >   INIT_LIST_HEAD(&p->children);
> | >   INIT_LIST_HEAD(&p->sibling);
> | >   p->vfork_done = NULL;
> | > Index: lx26-22-rc6-mm1/kernel/sysctl.c
> | > ===================================================================
> | > --- lx26-22-rc6-mm1.orig/kernel/sysctl.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/kernel/sysctl.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -2313,7 +2313,7 @@ static int proc_do_cad_pid(ctl_table *ta
> | >   pid_t tmp;
> | >   int r;
> | >
> | > - tmp = pid_nr(cad_pid);
> | > + tmp = pid_to_nr(cad_pid);
> | >
> | >   r = __do_proc_dointvec(&tmp, table, write, filp, buffer,
> | >             lenp, ppos, NULL, NULL);
> | > Index: lx26-22-rc6-mm1/kernel/container.c
```

> | > ===
> | > --- lx26-22-rc6-mm1.orig/kernel/container.c 2007-07-05 18:53:43.000000000 -0700
> | > +++ lx26-22-rc6-mm1/kernel/container.c 2007-07-05 18:54:29.000000000 -0700
> | > @@ -1600,7 +1600,7 @@ static int pid_array_load(pid_t *pidarra
> | > while ((tsk = container_iter_next(cont, &it))) {
> | > if (unlikely(n == npids))
> | > break;
> | > - pidarray[n++] = pid_nr(task_pid(tsk));
> | > + pidarray[n++] = pid_to_nr(task_pid(tsk));
> | > }
> | > container_iter_end(cont, &it);
> | > return n;
> | > _____
> | > Containers mailing list
> | > Containers@lists.linux-foundation.org
> | > https://lists.linux-foundation.org/mailman/listinfo/containers
> | >
> | > _____
> | > Devel mailing list
> | > Devel@openvz.org
> | > https://openvz.org/mailman/listinfo/devel
> | >
>

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/6] Rename pid_nr function
Posted by serue on Fri, 13 Jul 2007 13:43:43 GMT
View Forum Message <> Reply to Message

Quoting Pavel Emelianov (xemul@openvz.org):
> sukadev@us.ibm.com wrote:
> > Pavel Emelianov [xemul@openvz.org] wrote:
> > | sukadev@us.ibm.com wrote:
> > | > Subject: [PATCH 2/6] Rename pid_nr function
> > | >
> > | > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> > | >
> > | > Rename pid_nr() function to pid_to_nr() which is more descriptive
> > | > and will hopefully cause less confusion with new structure/functions
> > | > being added to support multiple pid namespaces.
> > |
> > | Don't we need at least two convertors:
> > | pid_to_nr_as_it_seen_by_current()

> > | and
> > | pid_to_nr_as_it_seen_by_some_other_task()
> >
> > Can you give me an example of where you would use this latter
> > (pid_to_nr_as_it_seen_by_some_other_task()) interface ?
>
> Easy. Consider you're observing /proc/<pid>/status file for a task
> that lives in two namespaces - init and some created. When making
> "cat" on this from init namespace you must get the task's pid as it
> is seen from init namespace, but when making "cat" from the created
> namespace you must see the pid as it is seen by this namespace. And
> so on and so forth.

But 'current' in that case is the process reading the file, so you do in
fact want to use pid_to_nr_as_it_seen_by_current().

-serge

> On the other hand sys_getpid() should simply return the virtual pid
> without messing with namespaces.
>
> > I have a similar confusion with following two sets of interfaces
> > in your patchset and have been meaning to discuss it further.
> >
> > pid_nr() and pid_vnr()
> > find_pid() and find_vpid()
> >
> > I can understand the use of pid_vnr() and find_vpid().
> >
> > Where do you see us using pid_nr() and find_pid() - i.e the values
> > as seen in init_pid_ns ?
>
> find_pid() may be used in keventd_create_thread() as eventd thread
> lives in init namespace only. Any other place in kernel that is
> somehow "global" to all the namespaces and has no user interface
> must use global pids not to confuse the kernel.
>
> Of course, if we plan to replace the notion of global numerical pid
> with struct pid this is not an option. However, the issues with the
> pid_to_nr_as_it_... are still valid.
>
> > Like I pointed out in another thread, sunos_killpg() for instance
> > needs to used find_vpid(). Similarly, kill_proc_info(),
> > __set_special_pids() etc need find_vpid().
>
> Sure. These places are to be patched as well.
>
> > Thanks,

> >
> > Suka
> >
> >
> > |
> > | The first one is essentially the second one with "some other
> > | task" being current, but it can be optimized...
> > |
> > | > Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> > | > ---
> > | > drivers/char/n_r3964.c | 7 ++++---
> > | > drivers/char/tty_io.c | 4 ++--
> > | > fs/autofs/root.c | 2 +-
> > | > fs/fcntl.c | 2 +-
> > | > fs/proc/array.c | 2 +-
> > | > fs/smbfs/proc.c | 2 +-
> > | > include/linux/pid.h | 2 +-
> > | > ipc/mqueue.c | 2 +-
> > | > kernel/container.c | 2 +-
> > | > kernel/fork.c | 2 +-
> > | > kernel/sysctl.c | 2 +-
> > | > 11 files changed, 15 insertions(+), 14 deletions(-)
> > | >
> > | > Index: lx26-22-rc6-mm1/include/linux/pid.h
> > | > ===
> > | > --- lx26-22-rc6-mm1.orig/include/linux/pid.h 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/include/linux/pid.h 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -98,7 +98,7 @@ extern struct pid *find_ge_pid(int nr);
> > | > extern struct pid *alloc_pid(void);
> > | > extern void FASTCALL(free_pid(struct pid *pid));
> > | >
> > | > -static inline pid_t pid_nr(struct pid *pid)
> > | > +static inline pid_t pid_to_nr(struct pid *pid)
> > | > {
> > | >   pid_t nr = 0;
> > | >   if (pid)
> > | > Index: lx26-22-rc6-mm1/drivers/char/n_r3964.c
> > | > ===
> > | > --- lx26-22-rc6-mm1.orig/drivers/char/n_r3964.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/drivers/char/n_r3964.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -771,7 +771,7 @@ static int enable_signals(struct r3964_i
> > | >
> > | >   if (pClient->pid == pid) {
> > | >   TRACE_PS("removing client %d from client list",
> > | > -   pid_nr(pid));
> > | > +   pid_to_nr(pid));
> > | >   *ppClient = pClient->next;
> > | >   while (pClient->msg_count) {

```
> > | >       pMsg = remove_msg(pInfo, pClient);
> > | > @@ -801,7 +801,8 @@ static int enable_signals(struct r3964_i
> > | >     if (pClient == NULL)
> > | >      return -ENOMEM;
> > | >
> > | > -   TRACE_PS("add client %d to client list", pid_nr(pid));
> > | > +   TRACE_PS("add client %d to client list",
> > | > +     pid_to_nr(pid));
> > | >     spin_lock_init(&pClient->lock);
> > | >     pClient->sig_flags = arg;
> > | >     pClient->pid = get_pid(pid);
> > | > @@ -933,7 +934,7 @@ static void remove_client_block(struct r
> > | >  {
> > | >   struct r3964_block_header *block;
> > | >
> > | > - TRACE_PS("remove_client_block PID %d", pid_nr(pClient->pid));
> > | > + TRACE_PS("remove_client_block PID %d", pid_to_nr(pClient->pid));
> > | >
> > | >   block = pClient->next_block_to_read;
> > | >   if (block) {
> > | > Index: lx26-22-rc6-mm1/drivers/char/tty_io.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/drivers/char/tty_io.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/drivers/char/tty_io.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -3080,7 +3080,7 @@ static int tiocgpgrp(struct tty_struct *
> > | >    */
> > | >   if (tty == real_tty && current->signal->tty != real_tty)
> > | >    return -ENOTTY;
> > | > - return put_user(pid_nr(real_tty->pgrp), p);
> > | > + return put_user(pid_to_nr(real_tty->pgrp), p);
> > | >  }
> > | >
> > | >  /**
> > | > @@ -3151,7 +3151,7 @@ static int tiocgsid(struct tty_struct *t
> > | >    return -ENOTTY;
> > | >   if (!real_tty->session)
> > | >    return -ENOTTY;
> > | > - return put_user(pid_nr(real_tty->session), p);
> > | > + return put_user(pid_to_nr(real_tty->session), p);
> > | >  }
> > | >
> > | >  /**
> > | > Index: lx26-22-rc6-mm1/fs/autofs/root.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/fs/autofs/root.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/fs/autofs/root.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup
> > | >
```

```
> > | >   oz_mode = autofs_oz_mode(sbi);
> > | >   DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
> > | > -    "oz_mode = %d\n", pid_nr(task_pid(current)),
> > | > +    "oz_mode = %d\n", pid_to_nr(task_pid(current)),
> > | >      process_group(current), sbi->catatonic,
> > | >      oz_mode));
> > | >
> > | > Index: lx26-22-rc6-mm1/fs/fcntl.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/fs/fcntl.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/fs/fcntl.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -305,7 +305,7 @@ pid_t f_getown(struct file *filp)
> > | >  {
> > | >   pid_t pid;
> > | >   read_lock(&filp->f_owner.lock);
> > | > - pid = pid_nr(filp->f_owner.pid);
> > | > + pid = pid_to_nr(filp->f_owner.pid);
> > | >   if (filp->f_owner.pid_type == PIDTYPE_PGID)
> > | >    pid = -pid;
> > | >   read_unlock(&filp->f_owner.lock);
> > | > Index: lx26-22-rc6-mm1/fs/proc/array.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/fs/proc/array.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/fs/proc/array.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -398,7 +398,7 @@ static int do_task_stat(struct task_stru
> > | >    struct signal_struct *sig = task->signal;
> > | >
> > | >    if (sig->tty) {
> > | > -   tty_pgrp = pid_nr(sig->tty->pgrp);
> > | > +   tty_pgrp = pid_to_nr(sig->tty->pgrp);
> > | >     tty_nr = new_encode_dev(tty_devnum(sig->tty));
> > | >    }
> > | >
> > | > Index: lx26-22-rc6-mm1/fs/smbfs/proc.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/fs/smbfs/proc.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/fs/smbfs/proc.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -972,7 +972,7 @@ smb_newconn(struct smb_sb_info *server,
> > | >
> > | >   VERBOSE("protocol=%d, max_xmit=%d, pid=%d capabilities=0x%x\n",
> > | >    server->opt.protocol, server->opt.max_xmit,
> > | > -  pid_nr(server->conn_pid), server->opt.capabilities);
> > | > +  pid_to_nr(server->conn_pid), server->opt.capabilities);
> > | >
> > | >   /* FIXME: this really should be done by smbmount. */
> > | >   if (server->opt.max_xmit > SMB_MAX_PACKET_SIZE) {
> > | > Index: lx26-22-rc6-mm1/ipc/mqueue.c
> > | > ===================================================================
```

```
> > | > --- lx26-22-rc6-mm1.orig/ipc/mqueue.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/ipc/mqueue.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -336,7 +336,7 @@ static ssize_t mqueue_read_file(struct f
> > | >      (info->notify_owner &&
> > | >       info->notify.sigev_notify == SIGEV_SIGNAL) ?
> > | >       info->notify.sigev_signo : 0,
> > | > -    pid_nr(info->notify_owner));
> > | > +    pid_to_nr(info->notify_owner));
> > | >    spin_unlock(&info->lock);
> > | >    buffer[sizeof(buffer)-1] = '\0';
> > | >    slen = strlen(buffer)+1;
> > | > Index: lx26-22-rc6-mm1/kernel/fork.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/kernel/fork.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/kernel/fork.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -1032,7 +1032,7 @@ static struct task_struct *copy_process(
> > | >    p->did_exec = 0;
> > | >    delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
> > | >    copy_flags(clone_flags, p);
> > | > - p->pid = pid_nr(pid);
> > | > + p->pid = pid_to_nr(pid);
> > | >    INIT_LIST_HEAD(&p->children);
> > | >    INIT_LIST_HEAD(&p->sibling);
> > | >    p->vfork_done = NULL;
> > | > Index: lx26-22-rc6-mm1/kernel/sysctl.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/kernel/sysctl.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/kernel/sysctl.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -2313,7 +2313,7 @@ static int proc_do_cad_pid(ctl_table *ta
> > | >    pid_t tmp;
> > | >    int r;
> > | >
> > | > - tmp = pid_nr(cad_pid);
> > | > + tmp = pid_to_nr(cad_pid);
> > | >
> > | >    r = __do_proc_dointvec(&tmp, table, write, filp, buffer,
> > | >              lenp, ppos, NULL, NULL);
> > | > Index: lx26-22-rc6-mm1/kernel/container.c
> > | > ===================================================================
> > | > --- lx26-22-rc6-mm1.orig/kernel/container.c 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/kernel/container.c 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -1600,7 +1600,7 @@ static int pid_array_load(pid_t *pidarra
> > | >    while ((tsk = container_iter_next(cont, &it))) {
> > | >     if (unlikely(n == npids))
> > | >      break;
> > | > -   pidarray[n++] = pid_nr(task_pid(tsk));
> > | > +   pidarray[n++] = pid_to_nr(task_pid(tsk));
> > | >    }
```

> > | > container_iter_end(cont, &it);
> > | > return n;
> > | > _____
> > | > Containers mailing list
> > | > Containers@lists.linux-foundation.org
> > | > https://lists.linux-foundation.org/mailman/listinfo/containers
> > | >
> > | > _____
> > | > Devel mailing list
> > | > Devel@openvz.org
> > | > https://openvz.org/mailman/listinfo/devel
> > | >
> >
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/6] Rename pid_nr function
Posted by Pavel Emelianov on Fri, 13 Jul 2007 14:09:31 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> Quoting Pavel Emelianov (xemul@openvz.org):
>> sukadev@us.ibm.com wrote:
>>> Pavel Emelianov [xemul@openvz.org] wrote:
>>> | sukadev@us.ibm.com wrote:
>>> | > Subject: [PATCH 2/6] Rename pid_nr function
>>> | >
>>> | > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
>>> | >
>>> | > Rename pid_nr() function to pid_to_nr() which is more descriptive
>>> | > and will hopefully cause less confusion with new structure/functions
>>> | > being added to support multiple pid namespaces.
>>> |
>>> | Don't we need at least two convertors:
>>> | pid_to_nr_as_it_seen_by_current()
>>> | and
>>> | pid_to_nr_as_it_seen_by_some_other_task()
>>>
>>> Can you give me an example of where you would use this latter
>>> (pid_to_nr_as_it_seen_by_some_other_task()) interface ?

>> Easy. Consider you're observing /proc/<pid>/status file for a task
>> that lives in two namespaces - init and some created. When making
>> "cat" on this from init namespace you must get the task's pid as it
>> is seen from init namespace, but when making "cat" from the created
>> namespace you must see the pid as it is seen by this namespace. And
>> so on and so forth.
>
> But 'current' in that case is the process reading the file, so you do in
> fact want to use pid_to_nr_as_it_seen_by_current().

Well. I see. There's a misleading in names. Better would be called them
pid_nr_as_it_is_seen_by_the_owning_task_when_he_reads_it() %)
and
pid_nr_as_it_is_seen_by_arbitrary_task()

the first one is pid_vnr() the second is pid_nr_ns().

> -serge

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 2/6] Rename pid_nr function
Posted by Sukadev Bhattiprolu on Fri, 13 Jul 2007 16:11:09 GMT
View Forum Message <> Reply to Message

Pavel Emelianov [xemul@openvz.org] wrote:
| Serge E. Hallyn wrote:
| > Quoting Pavel Emelianov (xemul@openvz.org):
| >> sukadev@us.ibm.com wrote:
| >>> Pavel Emelianov [xemul@openvz.org] wrote:
| >>> | sukadev@us.ibm.com wrote:
| >>> | > Subject: [PATCH 2/6] Rename pid_nr function
| >>> | >
| >>> | > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
| >>> | >
| >>> | > Rename pid_nr() function to pid_to_nr() which is more descriptive
| >>> | > and will hopefully cause less confusion with new structure/functions
| >>> | > being added to support multiple pid namespaces.
| >>> |
| >>> | Don't we need at least two convertors:
| >>> | pid_to_nr_as_it_seen_by_current()
| >>> | and
| >>> | pid_to_nr_as_it_seen_by_some_other_task()
| >>>
| >>> Can you give me an example of where you would use this latter

| >>> (pid_to_nr_as_it_seen_by_some_other_task()) interface ?
| >> Easy. Consider you're observing /proc/<pid>/status file for a task
| >> that lives in two namespaces - init and some created. When making
| >> "cat" on this from init namespace you must get the task's pid as it
| >> is seen from init namespace, but when making "cat" from the created
| >> namespace you must see the pid as it is seen by this namespace. And
| >> so on and so forth.

My patches use pid_nr_in_ns(pid, ns) in that case - where we get 'ns' from
the specific mount of /proc accesed. So the init_pid_ns process would execute
in init_pid_ns and the other would be in child ns.

So, in your case wouldn't it be enough to have pid_vnr() and
pid_nr_in_ns() ?

| >
| > But 'current' in that case is the process reading the file, so you do in
| > fact want to use pid_to_nr_as_it_seen_by_current().
|
| Well. I see. There's a misleading in names. Better would be called them
| pid_nr_as_it_is_seen_by_the_owning_task_when_he_reads_it() %)
| and
| pid_nr_as_it_is_seen_by_arbitrary_task()
|
| the first one is pid_vnr() the second is pid_nr_ns().

yep.

|
| > -serge

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers