

---

Subject: [PATCH] Use helpers to obtain task pid in printks  
Posted by [Pavel Emelianov](#) on Wed, 22 Aug 2007 09:33:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

The task\_struct->pid member is going to be deprecated, so start using the helpers (task\_pid\_nr/task\_pid\_vnr/task\_pid\_nr\_ns) in the kernel.

The first thing to start with is the pid, printed to dmesg - in this case we may safely use task\_pid\_nr(). Besides, printks produce more (much more) than a half of all the explicit pid usage.

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

---

Andrew, this patch changes many code in drivers and file systems, so it may cause many conflicts with other fixes in -mm tree. Is there any way I can help with it, e.g. split it into pieces, wait till another -mm tree or something else?

```
block/ll_rw_blk.c          | 2 +-
drivers/block/nbd.c        | 2 +-
drivers/cdrom/cdrom.c      | 2 +-
drivers/char/drm/drm_bufs.c | 2 +-
drivers/char/drm/drm_drv.c | 2 +-
drivers/char/drm/drm_fops.c | 4 +++-
drivers/char/drm/drm_lock.c | 6 ++++---
drivers/char/drm/drm_os_linux.h | 2 +-
drivers/char/drm/i810_dma.c | 2 +-
drivers/char/drm/i830_dma.c | 2 +-
drivers/char/sx.c          | 2 +-
drivers/char/tty_io.c      | 6 ++++---
drivers/hid/hidraw.c       | 4 +++-
drivers/md/md.c            | 2 +-
drivers/media/video/zoran_driver.c | 4 +++-
drivers/mtd/ubi/wl.c       | 2 +-
drivers/net/wireless/hostap/hostap_ioctl.c | 2 +-
drivers/scsi/libsas/sas_discover.c | 8 ++++-----
drivers/usb/core/devio.c   | 2 +-
drivers/usb/gadget/file_storage.c | 2 +-
fs/cifs/connect.c         | 2 +-
fs/dlm/user.c             | 2 +-
fs/fs-writeback.c         | 2 +-
fs/jffs2/debug.h          | 8 ++++-----
fs/nfsd/vfs.c             | 6 ++++---
fs/ocfs2/cluster/heartbeat.c | 2 +-
fs/ocfs2/cluster/masklog.h | 2 +-

```

```

fs/ocfs2/dlm/dlmrecovery.c      | 10 +++++-----
fs/reiser4/debug.h              | 2 +-
include/linux/reiserfs_fs.h     | 2 +-
include/net/9p/9p.h             | 4 +++-
kernel/cpu.c                    | 3 ++-
kernel/exit.c                   | 2 +-
kernel/lockdep.c                | 22 ++++++++-----
kernel/rtmutex-debug.c          | 15 ++++++++-----
kernel/rtmutex.c                | 2 +-
kernel/sched.c                  | 7 +++++-
kernel/signal.c                 | 2 +-
kernel/softlockup.c             | 2 +-
kernel/workqueue.c              | 2 +-
lib/spinlock_debug.c            | 8 +++++-
mm/oom_kill.c                   | 5 +++-
net/core/pktgen.c               | 2 +-
net/core/sock.c                 | 2 +-
net/ipv4/ipv4/ip_vs_sync.c      | 4 +++-
net/ipv4/tcp.c                  | 2 +-
net/llc/af_llc.c                | 2 +-
net/sunrpc/sched.c              | 2 +-
48 files changed, 96 insertions(+), 90 deletions(-)

```

```

diff --git a/block/ll_rw_blk.c b/block/ll_rw_blk.c
index 2849ef4..f830872 100644
--- a/block/ll_rw_blk.c
+++ b/block/ll_rw_blk.c
@@ -3303,7 +3303,7 @@ void submit_bio(int rw, struct bio *bio)
 if (unlikely(block_dump)) {
     char b[BDEVNAME_SIZE];
     printk(KERN_DEBUG "%s(%d): %s block %Lu on %s\n",
-    current->comm, current->pid,
+    current->comm, task_pid_nr(current),
     (rw & WRITE) ? "WRITE" : "READ",
     (unsigned long long)bio->bi_sector,
     bdevname(bio->bi_bdev, b));
diff --git a/drivers/block/nbd.c b/drivers/block/nbd.c
index 84d139b..a539eca 100644
--- a/drivers/block/nbd.c
+++ b/drivers/block/nbd.c
@@ -147,7 +147,7 @@ static int sock_xmit(struct socket *sock)
 if (signal_pending(current)) {
     siginfo_t info;
     printk(KERN_WARNING "nbd (pid %d: %s) got signal %d\n",
-    current->pid, current->comm,
+    task_pid_nr(current), current->comm,
     dequeue_signal_lock(current, &current->blocked, &info));
     result = -EINTR;

```

```

break;
diff --git a/drivers/cdrom/cdrom.c b/drivers/cdrom/cdrom.c
index f0c6318..e5f2339 100644
--- a/drivers/cdrom/cdrom.c
+++ b/drivers/cdrom/cdrom.c
@@ -1103,7 +1103,7 @@ int open_for_data(struct cdrom_device_in
    is the default case! */
    cdinfo(CD_OPEN, "bummer. wrong media type.\n");
    cdinfo(CD_WARNING, "pid %d must open device O_NONBLOCK!\n",
-   (unsigned int)current->pid);
+   (unsigned int)task_pid_nr(current));
    ret=-EMEDIUMTYPE;
    goto clean_up_and_return;
}
diff --git a/drivers/char/drm/drm_bufs.c b/drivers/char/drm/drm_bufs.c
index 3d1ec82..eb67480 100644
--- a/drivers/char/drm/drm_bufs.c
+++ b/drivers/char/drm/drm_bufs.c
@@ -1502,7 +1502,7 @@ int drm_freebufs(struct inode *inode, st
    buf = dma->buflist[idx];
    if (buf->filp != filp) {
        DRM_ERROR("Process %d freeing buffer not owned\n",
-   current->pid);
+   task_pid_nr(current));
        return -EINVAL;
    }
    drm_free_buffer(dev, buf);
diff --git a/drivers/char/drm/drm_drv.c b/drivers/char/drm/drm_drv.c
index 19994cd..30f714c 100644
--- a/drivers/char/drm/drm_drv.c
+++ b/drivers/char/drm/drm_drv.c
@@ -469,7 +469,7 @@ int drm_ioctl(struct inode *inode, struc
    ++priv->iocctl_count;

    DRM_DEBUG("pid=%d, cmd=0x%02x, nr=0x%02x, dev 0x%lx, auth=%d\n",
-   current->pid, cmd, nr,
+   task_pid_nr(current), cmd, nr,
    (long)old_encode_dev(priv->head->device),
    priv->authenticated);

diff --git a/drivers/char/drm/drm_fops.c b/drivers/char/drm/drm_fops.c
index 7bc51ba..faaf219 100644
--- a/drivers/char/drm/drm_fops.c
+++ b/drivers/char/drm/drm_fops.c
@@ -234,7 +234,7 @@ static int drm_open_helper(struct inode
    if (!drm_cpu_valid())
        return -EINVAL;

```

```

- DRM_DEBUG("pid = %d, minor = %d\n", current->pid, minor);
+ DRM_DEBUG("pid = %d, minor = %d\n", task_pid_nr(current), minor);

priv = drm_alloc(sizeof(*priv), DRM_MEM_FILES);
if (!priv)
@@ -339,7 +339,7 @@ int drm_release(struct inode *inode, str
*/

DRM_DEBUG("pid = %d, device = 0x%lx, open_count = %d\n",
- current->pid, (long)old_encode_dev(priv->head->device),
+ task_pid_nr(current), (long)old_encode_dev(priv->head->device),
dev->open_count);

if (dev->driver->reclaim_buffers_locked && dev->lock.hw_lock) {
diff --git a/drivers/char/drm/drm_lock.c b/drivers/char/drm/drm_lock.c
index c0534b5..7f85a0a 100644
--- a/drivers/char/drm/drm_lock.c
+++ b/drivers/char/drm/drm_lock.c
@@ -64,12 +64,12 @@ int drm_lock(struct inode *inode, struct
if (lock.context == DRM_KERNEL_CONTEXT) {
    DRM_ERROR("Process %d using kernel context %d\n",
- current->pid, lock.context);
+ task_pid_nr(current), lock.context);
    return -EINVAL;
}

DRM_DEBUG("%d (pid %d) requests lock (0x%08x), flags = 0x%08x\n",
- lock.context, current->pid,
+ lock.context, task_pid_nr(current),
dev->lock.hw_lock->lock, lock.flags);

if (drm_core_check_feature(dev, DRIVER_DMA_QUEUE))
@@ -162,7 +162,7 @@ int drm_unlock(struct inode *inode, stru
if (lock.context == DRM_KERNEL_CONTEXT) {
    DRM_ERROR("Process %d using kernel context %d\n",
- current->pid, lock.context);
+ task_pid_nr(current), lock.context);
    return -EINVAL;
}

diff --git a/drivers/char/drm/drm_os_linux.h b/drivers/char/drm/drm_os_linux.h
index 0b8d343..6bb8929 100644
--- a/drivers/char/drm/drm_os_linux.h
+++ b/drivers/char/drm/drm_os_linux.h
@@ -12,7 +12,7 @@
#define DRM_IOCTL_ARGS struct inode *inode, struct file *filp, unsigned int cmd, unsigned

```

```

long data
#define DRM_ERR(d) -(d)
/** Current process ID */
-#define DRM_CURRENTPID current->pid
+#define DRM_CURRENTPID task_pid_nr(current)
#define DRM_SUSUSER(p) capable(CAP_SYS_ADMIN)
#define DRM_UDELAY(d) udelay(d)
/** Read a byte from a MMIO region */
diff --git a/drivers/char/drm/i810_dma.c b/drivers/char/drm/i810_dma.c
index cb44999..8cd5433 100644
--- a/drivers/char/drm/i810_dma.c
+++ b/drivers/char/drm/i810_dma.c
@@ -1119,7 +1119,7 @@ static int i810_getbuf(struct inode *ino
    retcode = i810_dma_get_buffer(dev, &d, filp);

    DRM_DEBUG("i810_dma: %d returning %d, granted = %d\n",
-    current->pid, retcode, d.granted);
+    task_pid_nr(current), retcode, d.granted);

    if (copy_to_user((void __user *) arg, &d, sizeof(d)))
        return -EFAULT;
diff --git a/drivers/char/drm/i830_dma.c b/drivers/char/drm/i830_dma.c
index dc20c1a..e94077f 100644
--- a/drivers/char/drm/i830_dma.c
+++ b/drivers/char/drm/i830_dma.c
@@ -1441,7 +1441,7 @@ static int i830_getbuf(struct inode *ino
    retcode = i830_dma_get_buffer(dev, &d, filp);

    DRM_DEBUG("i830_dma: %d returning %d, granted = %d\n",
-    current->pid, retcode, d.granted);
+    task_pid_nr(current), retcode, d.granted);

    if (copy_to_user((void __user *) arg, &d, sizeof(d)))
        return -EFAULT;
diff --git a/drivers/char/sx.c b/drivers/char/sx.c
index 85a2328..a6e1c9b 100644
--- a/drivers/char/sx.c
+++ b/drivers/char/sx.c
@@ -1467,7 +1467,7 @@ static int sx_open(struct tty_struct *tt

    line = tty->index;
    sx_dprintk(SX_DEBUG_OPEN, "%d: opening line %d. tty=%p cttty=%p, "
-    "np=%d)\n", current->pid, line, tty,
+    "np=%d)\n", task_pid_nr(current), line, tty,
    current->signal->tty, sx_nports);

    if ((line < 0) || (line >= SX_NPORTS) || (line >= sx_nports))
diff --git a/drivers/char/tty_io.c b/drivers/char/tty_io.c

```

index 583a2d9..e9da99c 100644

--- a/drivers/char/tty\_io.c

+++ b/drivers/char/tty\_io.c

@@ -3532,7 +3532,7 @@ void \_\_do\_SAK(struct tty\_struct \*tty)

```
do_each_pid_task(session, PIDTYPE_SID, p) {
    printk(KERN_NOTICE "SAK: killed process %d"
           " (%s): task_session_nr(p)==tty->session\n",
```

```
-    p->pid, p->comm);
```

```
+    task_pid_nr(p), p->comm);
```

```
    send_sig(SIGKILL, p, 1);
```

```
} while_each_pid_task(session, PIDTYPE_SID, p);
```

```
/* Now kill any processes that happen to have the
```

@@ -3542,7 +3542,7 @@ void \_\_do\_SAK(struct tty\_struct \*tty)

```
if (p->signal->tty == tty) {
    printk(KERN_NOTICE "SAK: killed process %d"
           " (%s): task_session_nr(p)==tty->session\n",
```

```
-    p->pid, p->comm);
```

```
+    task_pid_nr(p), p->comm);
```

```
    send_sig(SIGKILL, p, 1);
```

```
    continue;
```

```
}
```

@@ -3562,7 +3562,7 @@ void \_\_do\_SAK(struct tty\_struct \*tty)

```
    filp->private_data == tty) {
```

```
    printk(KERN_NOTICE "SAK: killed process %d"
```

```
           " (%s): fd#%d opened to the tty\n",
```

```
-    p->pid, p->comm, i);
```

```
+    task_pid_nr(p), p->comm, i);
```

```
    force_sig(SIGKILL, p);
```

```
    break;
```

```
}
```

diff --git a/drivers/hid/hidraw.c b/drivers/hid/hidraw.c

index 20ebba1..bb940cb 100644

--- a/drivers/hid/hidraw.c

+++ b/drivers/hid/hidraw.c

@@ -113,13 +113,13 @@ static ssize\_t hidraw\_write(struct file

```
if (count > HID_MIN_BUFFER_SIZE) {
    printk(KERN_WARNING "hidraw: pid %d passed too large report\n",
```

```
-    current->pid);
```

```
+    task_pid_nr(current));
```

```
    return -EINVAL;
```

```
}
```

```
if (count < 2) {
```

```
    printk(KERN_WARNING "hidraw: pid %d passed too short report\n",
```

```
-    current->pid);
```

```
+    task_pid_nr(current));
```

```
    return -EINVAL;
```

```

}

diff --git a/drivers/md/md.c b/drivers/md/md.c
index f883b7e..55d120b 100644
--- a/drivers/md/md.c
+++ b/drivers/md/md.c
@@ -4721,7 +4721,7 @@ mdk_thread_t *md_register_thread(void (*

void md_unregister_thread(mdk_thread_t *thread)
{
- dprintk("interrupting MD-thread pid %d\n", thread->tsk->pid);
+ dprintk("interrupting MD-thread pid %d\n", task_pid_nr(thread->tsk));

    kthread_stop(thread->tsk);
    kfree(thread);
diff --git a/drivers/media/video/zoran_driver.c b/drivers/media/video/zoran_driver.c
index d831ca1..84ee338 100644
--- a/drivers/media/video/zoran_driver.c
+++ b/drivers/media/video/zoran_driver.c
@@ -1288,7 +1288,7 @@ zoran_open (struct inode *inode,
}

dprintk(1, KERN_INFO "%s: zoran_open(%s, pid=[%d]), users(-)=%d\n",
- ZR_DEVNAME(zr), current->comm, current->pid, zr->user);
+ ZR_DEVNAME(zr), current->comm, task_pid_nr(current), zr->user);

/* now, create the open()-specific file_ops struct */
fh = kzalloc(sizeof(struct zoran_fh), GFP_KERNEL);
@@ -1361,7 +1361,7 @@ zoran_close (struct inode *inode,
struct zoran *zr = fh->zr;

dprintk(1, KERN_INFO "%s: zoran_close(%s, pid=[%d]), users(+)=%d\n",
- ZR_DEVNAME(zr), current->comm, current->pid, zr->user);
+ ZR_DEVNAME(zr), current->comm, task_pid_nr(current), zr->user);

/* kernel locks (fs/device.c), so don't do that ourselves
 * (prevents deadlocks) */
diff --git a/drivers/mtd/ubi/wl.c b/drivers/mtd/ubi/wl.c
index 12b25e5..ff64545 100644
--- a/drivers/mtd/ubi/wl.c
+++ b/drivers/mtd/ubi/wl.c
@@ -1352,7 +1352,7 @@ static int ubi_thread(void *u)
struct ubi_device *ubi = u;

ubi_msg("background thread \"%s\" started, PID %d",
- ubi->bgt_name, current->pid);
+ ubi->bgt_name, task_pid_nr(current));

```

```

    set_freezable();
    for (;;) {
diff --git a/drivers/net/wireless/hostap/hostap_ioctl.c b/drivers/net/wireless/hostap/hostap_ioctl.c
index d58ac84..05c123d 100644
--- a/drivers/net/wireless/hostap/hostap_ioctl.c
+++ b/drivers/net/wireless/hostap/hostap_ioctl.c
@@ -2922,7 +2922,7 @@ static int prism2_ioctl_priv_monitor(str

    printk(KERN_DEBUG "%s: process %d (%s) used deprecated iwpriv monitor "
           "- update software to use iwconfig mode monitor\n",
-       dev->name, current->pid, current->comm);
+       dev->name, task_pid_nr(current), current->comm);

/* Backward compatibility code - this can be removed at some point */

diff --git a/drivers/scsi/libsas/sas_discover.c b/drivers/scsi/libsas/sas_discover.c
index 7ef0afc..5f3a0d7 100644
--- a/drivers/scsi/libsas/sas_discover.c
+++ b/drivers/scsi/libsas/sas_discover.c
@@ -285,7 +285,7 @@ static void sas_discover_domain(struct w
    dev = port->port_dev;

    SAS_DPRINTK("DOING DISCOVERY on port %d, pid:%d\n", port->id,
-       current->pid);
+       task_pid_nr(current));

    switch (dev->dev_type) {
    case SAS_END_DEV:
@@ -320,7 +320,7 @@ static void sas_discover_domain(struct w
    }

    SAS_DPRINTK("DONE DISCOVERY on port %d, pid:%d, result:%d\n", port->id,
-       current->pid, error);
+       task_pid_nr(current), error);
    }

static void sas_revalidate_domain(struct work_struct *work)
@@ -334,12 +334,12 @@ static void sas_revalidate_domain(struct
    &port->disc.pending);

    SAS_DPRINTK("REVALIDATING DOMAIN on port %d, pid:%d\n", port->id,
-       current->pid);
+       task_pid_nr(current));
    if (port->port_dev)
        res = sas_ex_revalidate_domain(port->port_dev);

    SAS_DPRINTK("done REVALIDATING DOMAIN on port %d, pid:%d, res 0x%x\n",
-       port->id, current->pid, res);

```



```

+   port->id, task_pid_nr(current), res);
}

/* ----- Events ----- */
diff --git a/drivers/usb/core/devio.c b/drivers/usb/core/devio.c
index b9f1edd..c98eaaf 100644
--- a/drivers/usb/core/devio.c
+++ b/drivers/usb/core/devio.c
@@ -458,7 +458,7 @@ static int checkintf(struct dev_state *p
    return 0;
    /* if not yet claimed, claim it for the driver */
    dev_warn(&ps->dev->dev, "usbfs: process %d (%s) did not claim interface %u before use\n",
-   current->pid, current->comm, ifnum);
+   task_pid_nr(current), current->comm, ifnum);
    return claimintf(ps, ifnum);
}

diff --git a/drivers/usb/gadget/file_storage.c b/drivers/usb/gadget/file_storage.c
index 3c567a8..f401ee7 100644
--- a/drivers/usb/gadget/file_storage.c
+++ b/drivers/usb/gadget/file_storage.c
@@ -4014,7 +4014,7 @@ static int __init fsg_bind(struct usb_ga
    DBG(fsg, "removable=%d, stall=%d, buflen=%u\n",
        mod_data.removable, mod_data.can_stall,
        mod_data.buflen);
-   DBG(fsg, "I/O thread pid: %d\n", fsg->thread_task->pid);
+   DBG(fsg, "I/O thread pid: %d\n", task_pid_nr(fsg->thread_task));

    set_bit(REGISTERED, &fsg->atomic_bitflags);

diff --git a/fs/cifs/connect.c b/fs/cifs/connect.c
index f3d0014..451ef93 100644
--- a/fs/cifs/connect.c
+++ b/fs/cifs/connect.c
@@ -352,7 +352,7 @@ cifs_demultiplex_thread(struct TCP_Serve
    current->flags |= PF_MEMALLOC;
    server->tsk = current; /* save process info to wake at shutdown */
-   cFYI(1, ("Demultiplex PID: %d", current->pid));
+   cFYI(1, ("Demultiplex PID: %d", task_pid_nr(current)));
    write_lock(&GlobalSMBseslock);
    atomic_inc(&tcpSesAllocCount);
    length = tcpSesAllocCount.counter;
diff --git a/fs/dlm/user.c b/fs/dlm/user.c
index 6438941..4f74154 100644
--- a/fs/dlm/user.c
+++ b/fs/dlm/user.c
@@ -456,7 +456,7 @@ static int check_version(struct dlm_writ

```

```

printk(KERN_DEBUG "dlm: process %s (%d) version mismatch "
"user (%d.%d.%d) kernel (%d.%d.%d)\n",
current->comm,
-   current->pid,
+   task_pid_nr(current),
req->version[0],
req->version[1],
req->version[2],
diff --git a/fs/fs-writeback.c b/fs/fs-writeback.c
index eaa2f75..5705c51 100644
--- a/fs/fs-writeback.c
+++ b/fs/fs-writeback.c
@@ -139,7 +139,7 @@ void __mark_inode_dirty(struct inode *in
if (inode->i_ino || strcmp(inode->i_sb->s_id, "bdev"))
printk(KERN_DEBUG
"%s(%d): dirtied inode %lu (%s) on %s\n",
-   current->comm, current->pid, inode->i_ino,
+   current->comm, task_pid_nr(current), inode->i_ino,
name, inode->i_sb->s_id);
}

```

```

diff --git a/fs/gfs2/glock.h b/fs/gfs2/glock.h
diff --git a/fs/jffs2/debug.h b/fs/jffs2/debug.h
index 2a49f2c..4130ada 100644
--- a/fs/jffs2/debug.h
+++ b/fs/jffs2/debug.h
@@ -80,28 +80,28 @@
#define JFFS2_ERROR(fmt, ...)    \
do { \
    printk(JFFS2_ERR_MSG_PREFIX \
-   " (%d) %s: " fmt, current->pid, \
+   " (%d) %s: " fmt, task_pid_nr(current), \
__FUNCTION__ , ##__VA_ARGS__); \
} while(0)

```

```

#define JFFS2_WARNING(fmt, ...) \
do { \
    printk(JFFS2_WARN_MSG_PREFIX \
-   " (%d) %s: " fmt, current->pid, \
+   " (%d) %s: " fmt, task_pid_nr(current), \
__FUNCTION__ , ##__VA_ARGS__); \
} while(0)

```

```

#define JFFS2_NOTICE(fmt, ...) \
do { \
    printk(JFFS2_NOTICE_MSG_PREFIX \
-   " (%d) %s: " fmt, current->pid, \
+   " (%d) %s: " fmt, task_pid_nr(current), \

```

```

__FUNCTION__ , ##__VA_ARGS__); \
} while(0)

#define JFFS2_DEBUG(fmt, ...) \
do { \
    printk(JFFS2_DBG_MSG_PREFIX \
- " (%d) %s: " fmt, current->pid, \
+ " (%d) %s: " fmt, task_pid_nr(current), \
    __FUNCTION__ , ##__VA_ARGS__); \
} while(0)

```

```

diff --git a/fs/nfsd/vfs.c b/fs/nfsd/vfs.c
index fd25697..70f2c86 100644
--- a/fs/nfsd/vfs.c
+++ b/fs/nfsd/vfs.c
@@ -1013,13 +1013,13 @@ nfsd_vfs_write(struct svc_rqst *rqstp, s
    if (EX_WGATHER(exp)) {
        if (atomic_read(&inode->i_writecount) > 1
            || (last_ino == inode->i_ino && last_dev == inode->i_sb->s_dev)) {
-        dprintk("nfsd: write defer %d\n", current->pid);
+        dprintk("nfsd: write defer %d\n", task_pid_nr(current));
            msleep(10);
-        dprintk("nfsd: write resume %d\n", current->pid);
+        dprintk("nfsd: write resume %d\n", task_pid_nr(current));
        }

        if (inode->i_state & I_DIRTY) {
-        dprintk("nfsd: write sync %d\n", current->pid);
+        dprintk("nfsd: write sync %d\n", task_pid_nr(current));
            host_err=nfsd_sync(file);
        }
    }
}
#endif

```

```

diff --git a/fs/ocfs2/cluster/heartbeat.c b/fs/ocfs2/cluster/heartbeat.c
index 2bd7f78..1b85480 100644
--- a/fs/ocfs2/cluster/heartbeat.c
+++ b/fs/ocfs2/cluster/heartbeat.c
@@ -1377,7 +1377,7 @@ static ssize_t o2hb_region_pid_read(stru

    spin_lock(&o2hb_live_lock);
    if (reg->hr_task)
-    pid = reg->hr_task->pid;
+    pid = task_pid_nr(reg->hr_task);
    spin_unlock(&o2hb_live_lock);

    if (!pid)

```

```

diff --git a/fs/ocfs2/cluster/masklog.h b/fs/ocfs2/cluster/masklog.h
index 75cd877..cd04606 100644
--- a/fs/ocfs2/cluster/masklog.h

```

```

+++ b/fs/ocfs2/cluster/masklog.h
@@ -192,7 +192,7 @@ extern struct mlog_bits mlog_and_bits, m
 * previous token if args expands to nothing.
 */
#define __mlog_printk(level, fmt, args...) \
- printk(level "(%u,%lu):%s:%d " fmt, current->pid, \
+ printk(level "(%u,%lu):%s:%d " fmt, task_pid_nr(current), \
    __mlog_cpu_guess, __PRETTY_FUNCTION__, __LINE__, \
    ##args)

diff --git a/fs/ocfs2/dlm/dlmrecovery.c b/fs/ocfs2/dlm/dlmrecovery.c
index a2c3316..2fde7bf 100644
--- a/fs/ocfs2/dlm/dlmrecovery.c
+++ b/fs/ocfs2/dlm/dlmrecovery.c
@@ -259,7 +259,7 @@ static void dlm_print_reco_node_status(s
 struct dlm_lock_resource *res;

    mlog(ML_NOTICE, "%s(%d): recovery info, state=%s, dead=%u, master=%u\n",
-    dlm->name, dlm->dlm_reco_thread_task->pid,
+    dlm->name, task_pid_nr(dlm->dlm_reco_thread_task),
    dlm->reco.state & DLM_RECO_STATE_ACTIVE ? "ACTIVE" : "inactive",
    dlm->reco.dead_node, dlm->reco.new_master);

@@ -420,7 +420,7 @@ void dlm_wait_for_recovery(struct dlm_ct
 if (dlm_in_recovery(dlm)) {
    mlog(0, "%s: reco thread %d in recovery: "
        "state=%d, master=%u, dead=%u\n",
-    dlm->name, dlm->dlm_reco_thread_task->pid,
+    dlm->name, task_pid_nr(dlm->dlm_reco_thread_task),
    dlm->reco.state, dlm->reco.new_master,
    dlm->reco.dead_node);
 }
@@ -483,7 +483,7 @@ static int dlm_do_recovery(struct dlm_ct
 return 0;
 }
 mlog(0, "%s(%d):recovery thread found node %u in the recovery map!\n",
-    dlm->name, dlm->dlm_reco_thread_task->pid,
+    dlm->name, task_pid_nr(dlm->dlm_reco_thread_task),
    dlm->reco.dead_node);
 spin_unlock(&dlm->spinlock);

@@ -507,7 +507,7 @@ static int dlm_do_recovery(struct dlm_ct
 mlog(0, "another node will master this recovery session.\n");
 }
 mlog(0, "dlm=%s (%d), new_master=%u, this node=%u, dead_node=%u\n",
-    dlm->name, dlm->dlm_reco_thread_task->pid, dlm->reco.new_master,
+    dlm->name, task_pid_nr(dlm->dlm_reco_thread_task), dlm->reco.new_master,
    dlm->node_num, dlm->reco.dead_node);

```

```

/* it is safe to start everything back up here
@@ -520,7 +520,7 @@ static int dlm_do_recovery(struct dlm_ct
master_here:
mlog(0, "(%d) mastering recovery of %s:%u here(this=%u)!\n",
- dlm->dlm_reco_thread_task->pid,
+ task_pid_nr(dlm->dlm_reco_thread_task),
  dlm->name, dlm->reco.dead_node, dlm->node_num);

status = dlm_remaster_locks(dlm, dlm->reco.dead_node);
diff --git a/fs/reiser4/debug.h b/fs/reiser4/debug.h
index 71671b1..6f1cbdb 100644
--- a/fs/reiser4/debug.h
+++ b/fs/reiser4/debug.h
@@ -15,7 +15,7 @@
#define DCALL(lev, fun, reperr, label, format, ...) \
({ \
fun(lev "reiser4[%i]: %s (%s:%i)[%s]:\n" format "\n" , \
- current->comm, current->pid, __FUNCTION__, \
+ current->comm, task_pid_nr(current), __FUNCTION__, \
  __FILE__, __LINE__, label, ## __VA_ARGS__); \
})

diff --git a/include/linux/reiserfs_fs.h b/include/linux/reiserfs_fs.h
index d09e6ed..811210f 100644
--- a/include/linux/reiserfs_fs.h
+++ b/include/linux/reiserfs_fs.h
@@ -85,7 +85,7 @@
@@ -85,7 +85,7 @@ void reiserfs_warning(struct super_block
if( !( cond ) ) \
reiserfs_panic( NULL, "reiserfs[%i]: assertion " scond " failed at " \
  __FILE__ ":%i:%s: " format "\n", \
- in_interrupt() ? -1 : current -> pid, __LINE__, __FUNCTION__, ##args )
+ in_interrupt() ? -1 : task_pid_nr(current), __LINE__, __FUNCTION__, ##args )

#define RASSERT(cond, format, args...) __RASSERT(cond, #cond, format, ##args)

diff --git a/include/net/9p/9p.h b/include/net/9p/9p.h
index 88884d3..96d8295 100644
--- a/include/net/9p/9p.h
+++ b/include/net/9p/9p.h
@@ -44,7 +44,7 @@
@@ -44,7 +44,7 @@ extern unsigned int p9_debug_level;
do { \
if ((p9_debug_level & level) == level) \
printk(KERN_NOTICE "-- %s (%d): " \
- format , __FUNCTION__, current->pid , ## arg); \
+ format , __FUNCTION__, task_pid_nr(current) , ## arg); \
} while (0)

```

```

#define PRINT_FCALL_ERROR(s, fcall) P9_DPRINTK(P9_DEBUG_ERROR, \
@@ -59,7 +59,7 @@ do { \
#define P9_EPRINTK(level, format, arg...) \
do { \
    printk(level "9p: %s (%d): " \
- format, __FUNCTION__, current->pid, ## arg); \
+ format, __FUNCTION__, task_pid_nr(current), ## arg); \
} while (0)

```

```

diff --git a/kernel/cpu.c b/kernel/cpu.c
index a21f71a..ebf6647 100644
--- a/kernel/cpu.c
+++ b/kernel/cpu.c
@@ -98,7 +98,8 @@ static inline void check_for_tasks(int c
    !cputime_eq(p->stime, cputime_zero)))
    printk(KERN_WARNING "Task %s (pid = %d) is on cpu %d\
    (state = %ld, flags = %x) \n",
-    p->comm, p->pid, cpu, p->state, p->flags);
+    p->comm, task_pid_nr(p), cpu,
+    p->state, p->flags);
}
write_unlock_irq(&tasklist_lock);
}

```

```

diff --git a/kernel/exit.c b/kernel/exit.c
index f4783ae..850a974 100644
--- a/kernel/exit.c
+++ b/kernel/exit.c
@@ -960,7 +960,7 @@ fastcall NORET_TYPE void do_exit(long co

    if (unlikely(in_atomic()))
        printk(KERN_INFO "note: %s[%d] exited with preempt_count %d\n",
-        current->comm, current->pid,
+        current->comm, task_pid_nr(current),
        preempt_count());

```

```

    acct_update_integrals(tsk);

```

```

diff --git a/kernel/fork.c b/kernel/fork.c
diff --git a/kernel/lockdep.c b/kernel/lockdep.c
index db4ae00..d8d020c 100644
--- a/kernel/lockdep.c
+++ b/kernel/lockdep.c
@@ -515,11 +515,11 @@ static void lockdep_print_held_locks(str
    depth = curr->lockdep_depth;

    if (!depth) {
-    printk("no locks held by %s/%d.\n", curr->comm, curr->pid);

```

```

+ printk("no locks held by %s/%d.\n", curr->comm, task_pid_nr(curr));
  return;
}
printk("%d lock%s held by %s/%d:\n",
- depth, depth > 1 ? "s" : "", curr->comm, curr->pid);
+ depth, depth > 1 ? "s" : "", curr->comm, task_pid_nr(curr));

  for (i = 0; i < depth; i++) {
    printk(" #%d: ", i);
@@ -908,7 +908,7 @@ print_circular_bug_header(struct lock_li
    print_kernel_version();
    printk( "-----\n");
    printk("%s/%d is trying to acquire lock:\n",
- curr->comm, curr->pid);
+ curr->comm, task_pid_nr(curr));
    print_lock(check_source);
    printk("\nbut task is already holding lock:\n");
    print_lock(check_target);
@@ -1089,7 +1089,7 @@ print_bad_irq_dependency(struct task_str
    print_kernel_version();
    printk( "-----\n");
    printk("%s/%d [HC%u[%lu]:SC%u[%lu]:HE%u:SE%u] is trying to acquire:\n",
- curr->comm, curr->pid,
+ curr->comm, task_pid_nr(curr),
    curr->hardirq_context, hardirq_count() >> HARDIRQ_SHIFT,
    curr->softirq_context, softirq_count() >> SOFTIRQ_SHIFT,
    curr->hardirqs_enabled,
@@ -1241,7 +1241,7 @@ print_deadlock_bug(struct task_struct *c
    print_kernel_version();
    printk( "-----\n");
    printk("%s/%d is trying to acquire lock:\n",
- curr->comm, curr->pid);
+ curr->comm, task_pid_nr(curr));
    print_lock(next);
    printk("\nbut task is already holding lock:\n");
    print_lock(prev);
@@ -1645,7 +1645,7 @@ print_usage_bug(struct task_struct *curr
    usage_str[prev_bit], usage_str[new_bit]);

    printk("%s/%d [HC%u[%lu]:SC%u[%lu]:HE%u:SE%u] takes:\n",
- curr->comm, curr->pid,
+ curr->comm, task_pid_nr(curr),
    trace_hardirq_context(curr), hardirq_count() >> HARDIRQ_SHIFT,
    trace_softirq_context(curr), softirq_count() >> SOFTIRQ_SHIFT,
    trace_hardirqs_enabled(curr),
@@ -1698,7 +1698,7 @@ print_irq_inversion_bug(struct task_stru
    print_kernel_version();
    printk( "-----\n");

```

```

    printk("%s/%d just changed the state of lock:\n",
-   curr->comm, curr->pid);
+   curr->comm, task_pid_nr(curr));
    print_lock(this);
    if (forwards)
        printk("but this lock took another, %s-irq-unsafe lock in the past:\n", irqclass);
@@ -2491,7 +2491,7 @@ print_unlock_inbalance_bug(struct task_s
    printk( "[ BUG: bad unlock balance detected! ]\n");
    printk( "-----\n");
    printk("%s/%d is trying to release lock (",
-   curr->comm, curr->pid);
+   curr->comm, task_pid_nr(curr));
    print_lockdep_cache(lock);
    printk(") at:\n");
    print_ip_sym(ip);
@@ -2741,7 +2741,7 @@ print_lock_contention_bug(struct task_st
    printk( "[ BUG: bad contention detected! ]\n");
    printk( "-----\n");
    printk("%s/%d is trying to contend lock (",
-   curr->comm, curr->pid);
+   curr->comm, task_pid_nr(curr));
    print_lockdep_cache(lock);
    printk(") at:\n");
    print_ip_sym(ip);
@@ -3076,7 +3076,7 @@ print_freed_lock_bug(struct task_struct
    printk( "[ BUG: held lock freed! ]\n");
    printk( "-----\n");
    printk("%s/%d is freeing memory %p-%p, with a lock still held there!\n",
-   curr->comm, curr->pid, mem_from, mem_to-1);
+   curr->comm, task_pid_nr(curr), mem_from, mem_to-1);
    print_lock(hlock);
    lockdep_print_held_locks(curr);

@@ -3129,7 +3129,7 @@ static void print_held_locks_bug(struct
    printk( "[ BUG: lock held at task exit time! ]\n");
    printk( "-----\n");
    printk("%s/%d is exiting with locks still held!\n",
-   curr->comm, curr->pid);
+   curr->comm, task_pid_nr(curr));
    lockdep_print_held_locks(curr);

    printk("\nstack backtrace:\n");
diff --git a/kernel/rtmutex-debug.c b/kernel/rtmutex-debug.c
index 6b0703d..56d73cb 100644
--- a/kernel/rtmutex-debug.c
+++ b/kernel/rtmutex-debug.c
@@ -87,7 +87,7 @@ static int rt_trace_on = 1;
static void printk_task(struct task_struct *p)

```



```

{
  if (p)
-   printk("%16s:%5d [%p, %3d]", p->comm, p->pid, p, p->prio);
+   printk("%16s:%5d [%p, %3d]", p->comm, task_pid_nr(p), p, p->prio);
  else
    printk("<none>");
}
@@ -152,22 +152,25 @@ void debug_rt_mutex_print_deadlock(struct
  printk( "[ BUG: circular locking deadlock detected! ]\n");
  printk( "-----\n");
  printk("%s/%d is deadlocking current task %s/%d\n\n",
-   task->comm, task->pid, current->comm, current->pid);
+   task->comm, task_pid_nr(task),
+   current->comm, task_pid_nr(current));

  printk("\n1) %s/%d is trying to acquire this lock:\n",
-   current->comm, current->pid);
+   current->comm, task_pid_nr(current));
  printk_lock(waiter->lock, 1);

-   printk("\n2) %s/%d is blocked on this lock:\n", task->comm, task->pid);
+   printk("\n2) %s/%d is blocked on this lock:\n",
+   task->comm, task_pid_nr(task));
  printk_lock(waiter->deadlock_lock, 1);

  debug_show_held_locks(current);
  debug_show_held_locks(task);

-   printk("\n%s/%d's [blocked] stackdump:\n\n", task->comm, task->pid);
+   printk("\n%s/%d's [blocked] stackdump:\n\n",
+   task->comm, task_pid_nr(task));
  show_stack(task, NULL);
  printk("\n%s/%d's [current] stackdump:\n\n",
-   current->comm, current->pid);
+   current->comm, task_pid_nr(current));
  dump_stack();
  debug_show_all_locks();

diff --git a/kernel/rtmutex.c b/kernel/rtmutex.c
index 8cd9bd2..0deef71 100644
--- a/kernel/rtmutex.c
+++ b/kernel/rtmutex.c
@@ -185,7 +185,7 @@ static int rt_mutex_adjust_prio_chain(st
  prev_max = max_lock_depth;
  printk(KERN_WARNING "Maximum lock depth %d reached "
    "task: %s (%d)\n", max_lock_depth,
-   top_task->comm, top_task->pid);
+   top_task->comm, task_pid_nr(top_task));

```

```

}
put_task_struct(task);

diff --git a/kernel/sched.c b/kernel/sched.c
index 1f81803..53a52bd 100644
--- a/kernel/sched.c
+++ b/kernel/sched.c
@@ -3376,7 +3376,7 @@ EXPORT_SYMBOL(sub_preempt_count);
static noinline void __schedule_bug(struct task_struct *prev)
{
    printk(KERN_ERR "BUG: scheduling while atomic: %s/0x%08x/%d\n",
-   prev->comm, preempt_count(), prev->pid);
+   prev->comm, preempt_count(), task_pid_nr(prev));
    debug_show_held_locks(prev);
    if (irqs_disabled())
        print_irqtrace_events(prev);
@@ -4778,7 +4778,8 @@ static void show_task(struct task_struct
    free = (unsigned long)n - (unsigned long)end_of_stack(p);
}
#endif
-   printk("%5lu %5d %6d\n", free, p->pid, p->parent->pid);
+   printk("%5lu %5d %6d\n", free,
+   task_pid_nr(p), task_pid_nr(p->parent));

    if (state != TASK_RUNNING)
        show_stack(p, NULL);
@@ -5096,7 +5097,7 @@ restart:
    if (p->mm && printk_ratelimit())
        printk(KERN_INFO "process %d (%s) no "
            "longer affine to cpu%d\n",
-   p->pid, p->comm, dead_cpu);
+   task_pid_nr(p), p->comm, dead_cpu);
}
if (!__migrate_task(p, dead_cpu, dest_cpu))
    goto restart;
diff --git a/kernel/sched_debug.c b/kernel/sched_debug.c
diff --git a/kernel/signal.c b/kernel/signal.c
index df785a7..1e4f317 100644
--- a/kernel/signal.c
+++ b/kernel/signal.c
@@ -722,7 +722,7 @@ int print_fatal_signals;
static void print_fatal_signal(struct pt_regs *regs, int signr)
{
    printk("%s/%d: potentially unexpected fatal signal %d.\n",
-   current->comm, current->pid, signr);
+   current->comm, task_pid_nr(current), signr);

#ifdef __i386__

```

```

    printk("code at %08lx: ", regs->eip);
diff --git a/kernel/softlockup.c b/kernel/softlockup.c
index e5d0a49..8f6644a 100644
--- a/kernel/softlockup.c
+++ b/kernel/softlockup.c
@@ -119,7 +119,7 @@ void softlockup_tick(void)
    spin_lock(&print_lock);
    printk(KERN_ERR "BUG: soft lockup - CPU#%d stuck for %lus! [%s:%d]\n",
        this_cpu, now - touch_timestamp,
-    current->comm, current->pid);
+    current->comm, task_pid_nr(current));
    if (regs)
        show_regs(regs);
    else
diff --git a/kernel/workqueue.c b/kernel/workqueue.c
index b06137a..b336df7 100644
--- a/kernel/workqueue.c
+++ b/kernel/workqueue.c
@@ -282,7 +282,7 @@ static void run_workqueue(struct cpu_wor
    printk(KERN_ERR "BUG: workqueue leaked lock or atomic: "
        "%s/0x%08x/%d\n",
        current->comm, preempt_count(),
-    current->pid);
+    task_pid_nr(current));
    printk(KERN_ERR "    last function: ");
    print_symbol("%s\n", (unsigned long)f);
    debug_show_held_locks(current);
diff --git a/lib/spinlock_debug.c b/lib/spinlock_debug.c
index 479fd46..9c4b025 100644
--- a/lib/spinlock_debug.c
+++ b/lib/spinlock_debug.c
@@ -60,12 +60,12 @@ static void spin_bug(spinlock_t *lock, c
    owner = lock->owner;
    printk(KERN_EMERG "BUG: spinlock %s on CPU#%d, %s/%d\n",
        msg, raw_smp_processor_id(),
-    current->comm, current->pid);
+    current->comm, task_pid_nr(current));
    printk(KERN_EMERG " lock: %p, .magic: %08x, .owner: %s/%d, "
        ".owner_cpu: %d\n",
        lock, lock->magic,
        owner ? owner->comm : "<none>",
-    owner ? owner->pid : -1,
+    owner ? task_pid_nr(owner) : -1,
        lock->owner_cpu);
    dump_stack();
}
@@ -116,7 +116,7 @@ static void __spin_lock_debug(spinlock_t
    printk(KERN_EMERG "BUG: spinlock lockup on CPU#%d, "

```

```

    "%s/%d, %p\n",
    raw_smp_processor_id(), current->comm,
-   current->pid, lock);
+   task_pid_nr(current), lock);
    dump_stack();
#ifdef CONFIG_SMP
    trigger_all_cpu_backtrace();
@@ -161,7 +161,7 @@ static void rwlock_bug(rwlock_t *lock, c

    printk(KERN_EMERG "BUG: rwlock %s on CPU#%d, %s/%d, %p\n",
    msg, raw_smp_processor_id(), current->comm,
-   current->pid, lock);
+   task_pid_nr(current), lock);
    dump_stack();
}

```

```

diff --git a/mm/oom_kill.c b/mm/oom_kill.c
index 1a2f687..dab3295 100644

```

```

--- a/mm/oom_kill.c
+++ b/mm/oom_kill.c
@@ -282,7 +282,8 @@ static void __oom_kill_task(struct task_
}

```

```

if (verbose)
-   printk(KERN_ERR "Killed process %d (%s)\n", p->pid, p->comm);
+   printk(KERN_ERR "Killed process %d (%s)\n",
+   task_pid_nr(p), p->comm);

```

```

/*
 * We give our sacrificial lamb high priority and access to
@@ -352,7 +353,7 @@ static int oom_kill_process(struct task_
}

```

```

    printk(KERN_ERR "%s: kill process %d (%s) score %li or a child\n",
-   message, p->pid, p->comm, points);
+   message, task_pid_nr(p), p->comm, points);

```

```

/* Try to kill a child first */
list_for_each_entry(c, &p->children, sibling) {
diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index 7bae576..3a3154e 100644

```

```

--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -3464,7 +3464,7 @@ static int pktgen_thread_worker(void *ar

```

```

    t->pid = current->pid;

-   pr_debug("pktgen: starting pktgen/%d: pid=%d\n", cpu, current->pid);

```

```
+ pr_debug("pktgen: starting pktgen/%d: pid=%d\n", cpu, task_pid_nr(current));
```

```
max_before_softirq = t->max_before_softirq;
```

```
diff --git a/net/core/sock.c b/net/core/sock.c
```

```
index cfed7d4..a8e9db3 100644
```

```
--- a/net/core/sock.c
```

```
+++ b/net/core/sock.c
```

```
@@ -231,7 +231,7 @@ static int sock_set_timeout(long *timeo_  
    warned++;  
    printk(KERN_INFO "sock_set_timeout: `%s' (pid %d) "  
           "tries to set negative timeout\n",  
-   current->comm, current->pid);  
+   current->comm, task_pid_nr(current));  
    return 0;  
}
```

```
*timeo_p = MAX_SCHEDULE_TIMEOUT;
```

```
diff --git a/net/ipv4/ipvs/ip_vs_sync.c b/net/ipv4/ipvs/ip_vs_sync.c
```

```
index 356f067..959c08d 100644
```

```
--- a/net/ipv4/ipvs/ip_vs_sync.c
```

```
+++ b/net/ipv4/ipvs/ip_vs_sync.c
```

```
@@ -877,7 +877,7 @@ int start_sync_thread(int state, char *m  
    if (!tinfo)  
        return -ENOMEM;
```

```
- IP_VS_DBG(7, "%s: pid %d\n", __FUNCTION__, current->pid);
```

```
+ IP_VS_DBG(7, "%s: pid %d\n", __FUNCTION__, task_pid_nr(current));
```

```
IP_VS_DBG(7, "Each ip_vs_sync_conn entry need %Zd bytes\n",  
    sizeof(struct ip_vs_sync_conn));
```

```
@@ -917,7 +917,7 @@ int stop_sync_thread(int state)
```

```
(state == IP_VS_STATE_BACKUP && !sync_backup_pid))
```

```
return -ESRCH;
```

```
- IP_VS_DBG(7, "%s: pid %d\n", __FUNCTION__, current->pid);
```

```
+ IP_VS_DBG(7, "%s: pid %d\n", __FUNCTION__, task_pid_nr(current));
```

```
IP_VS_INFO("stopping sync thread %d ... \n",
```

```
(state == IP_VS_STATE_MASTER) ?
```

```
sync_master_pid : sync_backup_pid);
```

```
diff --git a/net/ipv4/tcp.c b/net/ipv4/tcp.c
```

```
index 7e74011..5e71925 100644
```

```
--- a/net/ipv4/tcp.c
```

```
+++ b/net/ipv4/tcp.c
```

```
@@ -1333,7 +1333,7 @@ do_prequeue:
```

```
if ((flags & MSG_PEEK) && peek_seq != tp->copied_seq) {
```

```
if (net_ratelimit())
```

```
    printk(KERN_DEBUG "TCP(%s:%d): Application bug, race in MSG_PEEK.\n",
```

```
-   current->comm, current->pid);
```

```

+     current->comm, task_pid_nr(current));
    peek_seq = tp->copied_seq;
}
    continue;
diff --git a/net/llc/af_llc.c b/net/llc/af_llc.c
index 6b8a103..2d6a099 100644
--- a/net/llc/af_llc.c
+++ b/net/llc/af_llc.c
@@ -759,7 +759,7 @@ static int llc_ui_recvmmsg(struct kiocb *
    if (net_ratelimit())
        printk(KERN_DEBUG "LLC(%s:%d): Application "
               "bug, race in MSG_PEEK.\n",
-           current->comm, current->pid);
+           current->comm, task_pid_nr(current));
    peek_seq = llc->copied_seq;
}
    continue;
diff --git a/net/sunrpc/sched.c b/net/sunrpc/sched.c
index 954d7ec..c0908b2 100644
--- a/net/sunrpc/sched.c
+++ b/net/sunrpc/sched.c
@@ -845,7 +845,7 @@ void rpc_init_task(struct rpc_task *task
    task->tk_start = jiffies;

    dprintk("RPC:    new task initialized, procpid %u\n",
-   current->pid);
+   task_pid_nr(current));
}

static struct rpc_task *

```

---

Subject: Re: [PATCH] Use helpers to obtain task pid in printks  
 Posted by [Andrew Morton](#) on Wed, 22 Aug 2007 22:16:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, 22 Aug 2007 13:33:22 +0400  
 Pavel Emelyanov <xemul@openvz.org> wrote:

```

> The task_struct->pid member is going to be deprecated, so start
> using the helpers (task_pid_nr/task_pid_vnr/task_pid_nr_ns) in
> the kernel.
>
> The first thing to start with is the pid, printed to dmesg - in
> this case we may safely use task_pid_nr(). Besides, printks produce
> more (much more) than a half of all the explicit pid usage.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```

>  
> ---  
>  
> Andrew, this patch changes many code in drivers and file systems,  
> so it may cause many conflicts with other fixes in -mm tree. Is  
> there any way I can help with it, e.g. split it into pieces, wait  
> till another -mm tree or something else?

I think we just slam it all into the tree as-is. There will be a few places which ended up getting missed, but we can pick those up on a second round and then we rename task\_struct.pid to task\_struct.dont\_use\_me\_directly\_pid so that any remaining, future or out-of-tree usages of ->pid get reliably broken.

---