
Subject: [PATCH COMMIT] diff-merge-2.6.16.18-20060530

Posted by [xemul](#) on Tue, 30 May 2006 10:18:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Added to 026test013

Patch from OpenVZ team <devel@openvz.org>

Merged 2.6.16.18 from /linux/kernel/git/stable/linux-2.6.16.y

From: OpenVZ team <devel@openvz.org>

Date: Tue, 30 May 2006 10:02:30 +0000 (+0400)

Subject: Merged 2.6.16.18 from /linux/kernel/git/stable/linux-2.6.16.y

X-Git-Url:

<http://10.0.101.105/cgi-bin/gitweb.cgi?p=kernel;a=commitdiff;h=d08484cb38711ac57143c7967c7128580a0fb133>

Merged 2.6.16.18 from /linux/kernel/git/stable/linux-2.6.16.y

--- a/block/elevator.c

+++ b/block/elevator.c

@@ -314,6 +314,7 @@ void elv_insert(request_queue_t *q, stru

```
{
    struct list_head *pos;
    unsigned ordseq;
+ int unplug_it = 1;
```

```
    rq->q = q;
```

@@ -378,6 +379,11 @@ void elv_insert(request_queue_t *q, stru

```
    list_add_tail(&rq->queuelist, pos);
```

```
+ /*
+  * most requeues happen because of a busy condition, don't
+  * force unplug of the queue for that case.
+  */
+ unplug_it = 0;
    break;
```

```
default:
```

@@ -386,7 +392,7 @@ void elv_insert(request_queue_t *q, stru

```
- if (blk_queue_plugged(q)) {
+ if (unplug_it && blk_queue_plugged(q)) {
```

```

int nrq = q->rq.count[READ] + q->rq.count[WRITE]
- q->in_flight;

--- a/block/ll_rw_blk.c
+++ b/block/ll_rw_blk.c
@@ -1719,8 +1719,21 @@ void blk_run_queue(struct request_queue

    spin_lock_irqsave(q->queue_lock, flags);
    blk_remove_plug(q);
- if (!elv_queue_empty(q))
- q->request_fn(q);
+
+ /*
+  * Only recurse once to avoid overrunning the stack, let the unplug
+  * handling reinvoke the handler shortly if we already got there.
+  */
+ if (!elv_queue_empty(q)) {
+ if (!test_and_set_bit(Queue_FLAG_REENTER, &q->queue_flags)) {
+ q->request_fn(q);
+ clear_bit(Queue_FLAG_REENTER, &q->queue_flags);
+ } else {
+ blk_plug_device(q);
+ kblockd_schedule_work(&q->unplug_work);
+ }
+ }
+
    spin_unlock_irqrestore(q->queue_lock, flags);
}
EXPORT_SYMBOL(blk_run_queue);
--- a/drivers/block/ub.c
+++ b/drivers/block/ub.c
@@ -704,6 +704,9 @@ static void ub_cleanup(struct ub_dev *sc
    kfree(lun);
}

+ usb_set_intfdata(sc->intf, NULL);
+ usb_put_intf(sc->intf);
+ usb_put_dev(sc->dev);
+ kfree(sc);
+ }

@@ -2428,7 +2431,12 @@ static int ub_probe(struct usb_interface
// sc->ifnum = intf->cur_altsetting->desc.bInterfaceNumber;
usb_set_intfdata(intf, sc);
usb_get_dev(sc->dev);
- // usb_get_intf(sc->intf); /* Do we need this? */
+ /*
+  * Since we give the interface struct to the block level through

```

```

+ * disk->driverfs_dev, we have to pin it. Otherwise, block_uevent
+ * oopses on close after a disconnect (kernels 2.6.16 and up).
+ */
+ usb_get_intf(sc->intf);

    snprintf(sc->name, 12, DRV_NAME "(%d.%d)",
             sc->dev->bus->busnum, sc->dev->devnum);
@@ -2509,7 +2517,7 @@ static int ub_probe(struct usb_interface
err_diag:
err_dev_desc:
    usb_set_intfdata(intf, NULL);
- // usb_put_intf(sc->intf);
+ usb_put_intf(sc->intf);
    usb_put_dev(sc->dev);
    kfree(sc);
err_core:
@@ -2688,12 +2696,6 @@ static void ub_disconnect(struct usb_int
    */

    device_remove_file(&sc->intf->dev, &dev_attr_diag);
- usb_set_intfdata(intf, NULL);
- // usb_put_intf(sc->intf);
- sc->intf = NULL;
- usb_put_dev(sc->dev);
- sc->dev = NULL;
-
    ub_put(sc);
}

--- a/drivers/char/pcmcia/cm4000_cs.c
+++ b/drivers/char/pcmcia/cm4000_cs.c
@@ -2010,10 +2010,6 @@ static int __init cmm_init(void)
    if (!cmm_class)
        return -1;

- rc = pcmcia_register_driver(&cm4000_driver);
- if (rc < 0)
-     return rc;
-
    major = register_chrdev(0, DEVICE_NAME, &cm4000_fops);
    if (major < 0) {
        printk(KERN_WARNING MODULE_NAME
@@ -2021,6 +2017,12 @@ static int __init cmm_init(void)
        return -1;
    }

+ rc = pcmcia_register_driver(&cm4000_driver);
+ if (rc < 0) {

```

```

+ unregister_chrdev(major, DEVICE_NAME);
+ return rc;
+ }
+
+ return 0;
+ }

--- a/drivers/char/pcmcia/cm4040_cs.c
+++ b/drivers/char/pcmcia/cm4040_cs.c
@@ -769,16 +769,19 @@ static int __init cm4040_init(void)
    if (!cmx_class)
        return -1;

- rc = pcmcia_register_driver(&reader_driver);
- if (rc < 0)
- return rc;
-
    major = register_chrdev(0, DEVICE_NAME, &reader_fops);
    if (major < 0) {
        printk(KERN_WARNING MODULE_NAME
            ": could not get major number\n");
        return -1;
    }
+
+ rc = pcmcia_register_driver(&reader_driver);
+ if (rc < 0) {
+ unregister_chrdev(major, DEVICE_NAME);
+ return rc;
+ }
+
+ return 0;
+ }

--- a/drivers/i2c/busses/scx200_acb.c
+++ b/drivers/i2c/busses/scx200_acb.c
@@ -440,7 +440,6 @@ static int __init scx200_acb_create(int
    struct scx200_acb_iface *iface;
    struct i2c_adapter *adapter;
    int rc = 0;
- char description[64];

    iface = kzalloc(sizeof(*iface), GFP_KERNEL);
    if (!iface) {
@@ -459,8 +458,7 @@ static int __init scx200_acb_create(int

    init_MUTEX(&iface->sem);

- snprintf(description, sizeof(description), "NatSemi SCx200 ACCESS.bus [%s]", adapter->name);

```

```

- if (request_region(base, 8, description) == 0) {
+ if (!request_region(base, 8, adapter->name)) {
    dev_err(&adapter->dev, "can't allocate io 0x%x-0x%x\n",
        base, base + 8-1);
    rc = -EBUSY;
--- a/drivers/md/raid10.c
+++ b/drivers/md/raid10.c
@@ -1436,9 +1436,9 @@ static void raid10d(mddev_t *mddev)
    sl--;
    d = r10_bio->devs[sl].devnum;
    rdev = conf->mirrors[d].rdev;
-   atomic_add(s, &rdev->corrected_errors);
    if (rdev &&
        test_bit(ln_sync, &rdev->flags)) {
+   atomic_add(s, &rdev->corrected_errors);
    if (sync_page_io(rdev->bdev,
        r10_bio->devs[sl].addr +
        sect + rdev->data_offset,
--- a/drivers/net/tg3.c
+++ b/drivers/net/tg3.c
@@ -7368,21 +7368,23 @@ static int tg3_get_settings(struct net_d
    cmd->supported |= (SUPPORTED_1000baseT_Half |
        SUPPORTED_1000baseT_Full);

- if (!(tp->tg3_flags2 & TG3_FLG2_ANY_SERDES))
+ if (!(tp->tg3_flags2 & TG3_FLG2_ANY_SERDES)) {
    cmd->supported |= (SUPPORTED_100baseT_Half |
        SUPPORTED_100baseT_Full |
        SUPPORTED_10baseT_Half |
        SUPPORTED_10baseT_Full |
        SUPPORTED_MII);
- else
+ cmd->port = PORT_TP;
+ } else {
    cmd->supported |= SUPPORTED_FIBRE;
+ cmd->port = PORT_FIBRE;
+ }

    cmd->advertising = tp->link_config.advertising;
    if (netif_running(dev)) {
        cmd->speed = tp->link_config.active_speed;
        cmd->duplex = tp->link_config.active_duplex;
    }
- cmd->port = 0;
    cmd->phy_address = PHY_ADDR;
    cmd->transceiver = 0;
    cmd->autoneg = tp->link_config.autoneg;
--- a/drivers/net/via-rhine.c

```

```

+++ b/drivers/net/via-rhine.c
@@ -129,6 +129,7 @@
- Massive clean-up
- Rewrite PHY, media handling (remove options, full_duplex, backoff)
- Fix Tx engine race for good
+ - Craig Brind: Zero padded aligned buffers for short packets.

```

```

*/

```

```

@@ -1306,7 +1307,12 @@ static int rhine_start_tx(struct sk_buff
    rp->stats.tx_dropped++;
    return 0;
}

```

```

+
+ /* Padding is not copied and so must be redone. */
+ skb_copy_and_csum_dev(skb, rp->tx_buf[entry]);
+ if (skb->len < ETH_ZLEN)
+   memset(rp->tx_buf[entry] + skb->len, 0,
+     ETH_ZLEN - skb->len);
+   rp->tx_skbuff_dma[entry] = 0;
+   rp->tx_ring[entry].addr = cpu_to_le32(rp->tx_bufs_dma +
+     (rp->tx_buf[entry] -

```

```

--- a/drivers/pci/pci-acpi.c

```

```

+++ b/drivers/pci/pci-acpi.c
@@ -33,13 +33,10 @@ acpi_query_osc (
    acpi_status status;
    struct acpi_object_list input;
    union acpi_object in_params[4];
- struct acpi_buffer output;
- union acpi_object out_obj;
+ struct acpi_buffer output = {ACPI_ALLOCATE_BUFFER, NULL};
+ union acpi_object *out_obj;
    u32 osc_dw0;

```

```

- /* Setting up output buffer */
- output.length = sizeof(out_obj) + 3*sizeof(u32);
- output.pointer = &out_obj;

```

```

/* Setting up input parameters */
input.count = 4;
@@ -61,12 +58,15 @@ acpi_query_osc (
    "Evaluate _OSC Set fails. Status = 0x%04x\n", status);
    return status;
}
- if (out_obj.type != ACPI_TYPE_BUFFER) {
+ out_obj = output.pointer;
+
+ if (out_obj->type != ACPI_TYPE_BUFFER) {

```

```

    printk(KERN_DEBUG
    "Evaluate _OSC returns wrong type\n");
- return AE_TYPE;
+ status = AE_TYPE;
+ goto query_osc_out;
}
- osc_dw0 = *((u32 *) out_obj.buffer.pointer);
+ osc_dw0 = *((u32 *) out_obj->buffer.pointer);
if (osc_dw0) {
    if (osc_dw0 & OSC_REQUEST_ERROR)
        printk(KERN_DEBUG "_OSC request fails\n");
@@ -76,15 +76,21 @@ acpi_query_osc (
    printk(KERN_DEBUG "_OSC invalid revision\n");
    if (osc_dw0 & OSC_CAPABILITIES_MASK_ERROR) {
        /* Update Global Control Set */
- global_ctrlsets = *((u32 *) (out_obj.buffer.pointer+8));
- return AE_OK;
+ global_ctrlsets = *((u32 *) (out_obj->buffer.pointer+8));
+ status = AE_OK;
+ goto query_osc_out;
    }
- return AE_ERROR;
+ status = AE_ERROR;
+ goto query_osc_out;
}

/* Update Global Control Set */
- global_ctrlsets = *((u32 *) (out_obj.buffer.pointer + 8));
- return AE_OK;
+ global_ctrlsets = *((u32 *) (out_obj->buffer.pointer + 8));
+ status = AE_OK;
+
+query_osc_out:
+ kfree(output.pointer);
+ return status;
}

@@ -96,14 +102,10 @@ acpi_run_osc (
    acpi_status status;
    struct acpi_object_list input;
    union acpi_object in_params[4];
- struct acpi_buffer output;
- union acpi_object out_obj;
+ struct acpi_buffer output = {ACPI_ALLOCATE_BUFFER, NULL};
+ union acpi_object *out_obj;
    u32 osc_dw0;

```

```

- /* Setting up output buffer */
- output.length = sizeof(out_obj) + 3*sizeof(u32);
- output.pointer = &out_obj;
-
  /* Setting up input parameters */
  input.count = 4;
  input.pointer = in_params;
@@ -124,12 +126,14 @@ acpi_run_osc (
    "Evaluate _OSC Set fails. Status = 0x%04x\n", status);
    return status;
  }
- if (out_obj.type != ACPI_TYPE_BUFFER) {
+ out_obj = output.pointer;
+ if (out_obj->type != ACPI_TYPE_BUFFER) {
    printk(KERN_DEBUG
      "Evaluate _OSC returns wrong type\n");
- return AE_TYPE;
+ status = AE_TYPE;
+ goto run_osc_out;
  }
- osc_dw0 = *((u32 *) out_obj.buffer.pointer);
+ osc_dw0 = *((u32 *) out_obj->buffer.pointer);
  if (osc_dw0) {
    if (osc_dw0 & OSC_REQUEST_ERROR)
      printk(KERN_DEBUG "_OSC request fails\n");
@@ -139,11 +143,17 @@ acpi_run_osc (
    printk(KERN_DEBUG "_OSC invalid revision\n");
    if (osc_dw0 & OSC_CAPABILITIES_MASK_ERROR) {
      printk(KERN_DEBUG "_OSC FW not grant req. control\n");
- return AE_SUPPORT;
+ status = AE_SUPPORT;
+ goto run_osc_out;
    }
- return AE_ERROR;
+ status = AE_ERROR;
+ goto run_osc_out;
  }
- return AE_OK;
+ status = AE_OK;
+
+run_osc_out:
+ kfree(output.pointer);
+ return status;
  }

/**
--- a/drivers/pci/quirks.c
+++ b/drivers/pci/quirks.c

```



```

@@ -631,6 +631,9 @@ DECLARE_PCI_FIXUP_HEADER(PCI_VENDOR_ID_V
 * non-x86 architectures (yes Via exists on PPC among other places),
 * we must mask the PCI_INTERRUPT_LINE value versus 0xf to get
 * interrupts delivered properly.
+ *
+ * Some of the on-chip devices are actually '586 devices' so they are
+ * listed here.
 */
static void quirk_via_irq(struct pci_dev *dev)
{
@@ -639,13 +642,19 @@ static void quirk_via_irq(struct pci_dev
new_irq = dev->irq & 0xf;
pci_read_config_byte(dev, PCI_INTERRUPT_LINE, &irq);
if (new_irq != irq) {
- printk(KERN_INFO "PCI: Via IRQ fixup for %s, from %d to %d\n",
+ printk(KERN_INFO "PCI: VIA IRQ fixup for %s, from %d to %d\n",
pci_name(dev), irq, new_irq);
udelay(15); /* unknown if delay really needed */
pci_write_config_byte(dev, PCI_INTERRUPT_LINE, new_irq);
}
}
-DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_ANY_ID, quirk_via_irq);
+DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_DEVICE_ID_VIA_82C586_0,
quirk_via_irq);
+DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_DEVICE_ID_VIA_82C586_1,
quirk_via_irq);
+DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_DEVICE_ID_VIA_82C586_2,
quirk_via_irq);
+DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_DEVICE_ID_VIA_82C586_3,
quirk_via_irq);
+DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_DEVICE_ID_VIA_82C686,
quirk_via_irq);
+DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_DEVICE_ID_VIA_82C686_4,
quirk_via_irq);
+DECLARE_PCI_FIXUP_ENABLE(PCI_VENDOR_ID_VIA, PCI_DEVICE_ID_VIA_82C686_5,
quirk_via_irq);

/*
 * VIA VT82C598 has its device ID settable and many BIOSes
@@ -861,6 +870,7 @@ static void __init quirk_eisa_bridge(str
}

DECLARE_PCI_FIXUP_HEADER(PCI_VENDOR_ID_INTEL, PCI_DEVICE_ID_INTEL_82375, qu
irk_eisa_bridge );

+#ifndef CONFIG_ACPI_SLEEP
/*
 * On ASUS P4B boards, the SMBus PCI Device within the ICH2/4 southbridge

```

```

* is not activated. The myth is that Asus said that they do not want the
@@ -872,8 +882,12 @@ DECLARE_PCI_FIXUP_HEADER(PCI_VENDOR_ID_I
* bridge. Unfortunately, this device has no subvendor/subdevice ID. So it
* becomes necessary to do this tweak in two steps -- I've chosen the Host
* bridge as trigger.
+ *
+ * Actually, leaving it unhidden and not redoing the quirk over suspend2ram
+ * will cause thermal management to break down, and causing machine to
+ * overheat.
*/
-static int __initdata asus_hides_smbus = 0;
+static int __initdata asus_hides_smbus;

static void __init asus_hides_smbus_hostbridge(struct pci_dev *dev)
{
@@ -1008,6 +1022,8 @@ static void __init asus_hides_smbus_lpc_
}

DECLARE_PCI_FIXUP_HEADER(PCI_VENDOR_ID_INTEL, PCI_DEVICE_ID_INTEL_ICH6_1,
asus_hides_smbus_lpc_ich6 );

+#endif
+
/*
* SiS 96x south bridge: BIOS typically hides SMBus device...
*/
--- a/fs/compat.c
+++ b/fs/compat.c
@@ -1909,7 +1909,7 @@ asmlinkage long compat_sys_ppoll(struct
}

if (sigmask) {
- if (sigsetsize != sizeof(compat_sigset_t))
+ if (sigsetsize != sizeof(compat_sigset_t))
return -EINVAL;
if (copy_from_user(&ss32, sigmask, sizeof(ss32)))
return -EFAULT;
--- a/fs/locks.c
+++ b/fs/locks.c
@@ -452,15 +452,14 @@ static struct lock_manager_operations le
*/
static int lease_init(struct file *filp, int type, struct file_lock *fl)
{
+ if (assign_type(fl, type) != 0)
+ return -EINVAL;
+
fl->fl_owner = current->files;
fl->fl_pid = current->tgid;

```

```

    fl->fl_file = filp;
    fl->fl_flags = FL_LEASE;
- if (assign_type(fl, type) != 0) {
-     locks_free_lock(fl);
-     return -EINVAL;
- }
    fl->fl_start = 0;
    fl->fl_end = OFFSET_MAX;
    fl->fl_ops = NULL;
@@ -472,16 +471,19 @@ static int lease_init(struct file *filp,
static int lease_alloc(struct file *filp, int type, struct file_lock **flp)
{
    struct file_lock *fl = locks_alloc_lock();
- int error;
+ int error = -ENOMEM;

    if (fl == NULL)
-     return -ENOMEM;
+     goto out;

    error = lease_init(filp, type, fl);
- if (error)
-     return error;
+ if (error) {
+     locks_free_lock(fl);
+     fl = NULL;
+ }
+out:
    *flp = fl;
- return 0;
+ return error;
}

/* Check if two locks overlap each other.
@@ -732,8 +734,9 @@ EXPORT_SYMBOL(posix_locks_deadlock);
 * at the head of the list, but that's secret knowledge known only to
 * flock_lock_file and posix_lock_file.
 */
-static int flock_lock_file(struct file *filp, struct file_lock *new_fl)
+static int flock_lock_file(struct file *filp, struct file_lock *request)
{
+ struct file_lock *new_fl = NULL;
    struct file_lock **before;
    struct inode *inode = filp->f_dentry->d_inode;
    int error = 0;
@@ -748,17 +751,19 @@ static int flock_lock_file(struct file *
    continue;

```

```

    if (filp != fl->fl_file)
        continue;
- if (new_fl->fl_type == fl->fl_type)
+ if (request->fl_type == fl->fl_type)
    goto out;
    found = 1;
    locks_delete_lock(before);
    break;
}
- unlock_kernel();

- if (new_fl->fl_type == F_UNLCK)
- return 0;
+ if (request->fl_type == F_UNLCK)
+ goto out;

+ new_fl = locks_alloc_lock();
+ if (new_fl == NULL)
+ goto out;
/*
 * If a higher-priority process was blocked on the old file lock,
 * give it the opportunity to lock the file.
@@ -766,26 +771,27 @@ static int flock_lock_file(struct file *
    if (found)
        cond_resched();

- lock_kernel();
    for_each_lock(inode, before) {
        struct file_lock *fl = *before;
        if (IS_POSIX(fl))
            break;
        if (IS_LEASE(fl))
            continue;
- if (!flock_locks_conflict(new_fl, fl))
+ if (!flock_locks_conflict(request, fl))
            continue;
        error = -EAGAIN;
- if (new_fl->fl_flags & FL_SLEEP) {
-     locks_insert_block(fl, new_fl);
- }
+ if (request->fl_flags & FL_SLEEP)
+     locks_insert_block(fl, request);
        goto out;
    }
+ locks_copy_lock(new_fl, request);
    locks_insert_lock(&inode->i_flock, new_fl);
- error = 0;
+ new_fl = NULL;

```

```

out:
    unlock_kernel();
+ if (new_fl)
+ locks_free_lock(new_fl);
    return error;
}

@@ -1371,6 +1377,7 @@ static int __setlease(struct file *filp,
    goto out;

    if (my_before != NULL) {
+ *flp = *my_before;
    error = lease->fl_lmops->fl_change(my_before, arg);
    goto out;
    }
@@ -1563,9 +1570,7 @@ asmlinkage long sys_flock(unsigned int f
    error = flock_lock_file_wait(filp, lock);

    out_free:
- if (list_empty(&lock->fl_link)) {
- locks_free_lock(lock);
- }
+ locks_free_lock(lock);

    out_putf:
    fput(filp);
--- a/fs/smbfs/request.c
+++ b/fs/smbfs/request.c
@@ -339,9 +339,11 @@ int smb_add_request(struct smb_request *
/*
 * On timeout or on interrupt we want to try and remove the
 * request from the recvq/xmitq.
+ * First check if the request is still part of a queue. (May
+ * have been removed by some error condition)
 */
    smb_lock_server(server);
- if (!(req->rq_flags & SMB_REQ_RECEIVED)) {
+ if (!list_empty(&req->rq_queue)) {
    list_del_init(&req->rq_queue);
    smb_rput(req);
    }
--- a/include/net/sctp/sctp.h
+++ b/include/net/sctp/sctp.h
@@ -461,12 +461,12 @@ static inline int sctp_frag_point(const
 * there is room for a param header too.
 */
#define sctp_walk_params(pos, chunk, member)\

```

```

-_sctp_walk_params((pos), (chunk), WORD_ROUND(ntohs((chunk)->chunk_hdr.length)),
member)
+_sctp_walk_params((pos), (chunk), ntohs((chunk)->chunk_hdr.length), member)

#define _sctp_walk_params(pos, chunk, end, member)\
for (pos.v = chunk->member;\
    pos.v <= (void *)chunk + end - sizeof(sctp_paramhdr_t) &&\
-   pos.v <= (void *)chunk + end - WORD_ROUND(ntohs(pos.p->length)) &&\
+   pos.v <= (void *)chunk + end - ntohs(pos.p->length) &&\
    ntohs(pos.p->length) >= sizeof(sctp_paramhdr_t);\
    pos.v += WORD_ROUND(ntohs(pos.p->length)))

@@ -477,7 +477,7 @@ _sctp_walk_errors((err), (chunk_hdr), nt
for (err = (sctp_errhdr_t *)((void *)chunk_hdr + \
    sizeof(sctp_chunkhdr_t));\
    (void *)err <= (void *)chunk_hdr + end - sizeof(sctp_errhdr_t) &&\
-   (void *)err <= (void *)chunk_hdr + end - WORD_ROUND(ntohs(err->length)) &&\
+   (void *)err <= (void *)chunk_hdr + end - ntohs(err->length) &&\
    ntohs(err->length) >= sizeof(sctp_errhdr_t); \
    err = (sctp_errhdr_t *)((void *)err + WORD_ROUND(ntohs(err->length))))

--- a/kernel/ptrace.c
+++ b/kernel/ptrace.c
@@ -152,12 +152,34 @@ int ptrace_may_attach(struct task_struct
int ptrace_attach(struct task_struct *task)
{
    int retval;
-   task_lock(task);
+
    retval = -EPERM;
    if (task->pid <= 1)
-   goto bad;
+   goto out;
    if (task->tgid == current->tgid)
-   goto bad;
+   goto out;
+
+repeat:
+   /*
+    * Nasty, nasty.
+    *
+    * We want to hold both the task-lock and the
+    * tasklist_lock for writing at the same time.
+    * But that's against the rules (tasklist_lock
+    * is taken for reading by interrupts on other
+    * cpu's that may have task_lock).
+    */
+   task_lock(task);

```

```

+ local_irq_disable();
+ if (!write_trylock(&tasklist_lock)) {
+ local_irq_enable();
+ task_unlock(task);
+ do {
+ cpu_relax();
+ } while (!write_can_lock(&tasklist_lock));
+ goto repeat;
+ }
+
/* the same process cannot be attached many times */
if (task->ptrace & PT_PTRACED)
goto bad;
@@ -170,17 +192,15 @@ int ptrace_attach(struct task_struct *ta
? PT_ATTACHED : 0);
if (capable(CAP_SYS_PTRACE))
task->ptrace |= PT_PTRACE_CAP;
- task_unlock(task);

- write_lock_irq(&tasklist_lock);
__ptrace_link(task, current);
- write_unlock_irq(&tasklist_lock);

force_sig_specific(SIGSTOP, task);
- return 0;

bad:
+ write_unlock_irq(&tasklist_lock);
task_unlock(task);
+out:
return retval;
}

@@ -422,21 +442,22 @@ int ptrace_request(struct task_struct *c
*/
int ptrace_traceme(void)
{
- int ret;
+ int ret = -EPERM;

/*
* Are we already being traced?
*/
- if (current->ptrace & PT_PTRACED)
- return -EPERM;
- ret = security_ptrace(current->parent, current);
- if (ret)
- return -EPERM;

```

```

- /*
- * Set the ptrace bit in the process ptrace flags.
- */
- current->ptrace |= PT_PTRACED;
- return 0;
+ task_lock(current);
+ if (!(current->ptrace & PT_PTRACED)) {
+ ret = security_ptrace(current->parent, current);
+ /*
+ * Set the ptrace bit in the process ptrace flags.
+ */
+ if (!ret)
+ current->ptrace |= PT_PTRACED;
+ }
+ task_unlock(current);
+ return ret;
}

/**
--- a/mm/mempolicy.c
+++ b/mm/mempolicy.c
@@ -1796,7 +1796,6 @@ static void gather_stats(struct page *pa
    md->mapcount_max = count;

    md->node[page_to_nid(page)]++;
- cond_resched();
}

#ifdef CONFIG_HUGETLB_PAGE
--- a/mm/shmem.c
+++ b/mm/shmem.c
@@ -2192,6 +2192,7 @@ static struct address_space_operations s
    .prepare_write = shmem_prepare_write,
    .commit_write = simple_commit_write,
#endif
+ .migratepage = migrate_page,
};

static struct file_operations shmem_file_operations = {
--- a/mm/vmscan.c
+++ b/mm/vmscan.c
@@ -949,6 +949,17 @@ redo:
    goto unlock_both;
}

+ /* Make sure the dirty bit is up to date */
+ if (try_to_unmap(page, 1) == SWAP_FAIL) {
+ rc = -EPERM;

```



```

+ goto unlock_both;
+ }
+
+ if (page_mapcount(page)) {
+ rc = -EAGAIN;
+ goto unlock_both;
+ }
+
+ /*
+  * Default handling if a filesystem does not provide
+  * a migration function. We can only migrate clean
--- a/net/ipv4/netfilter/ip_nat_snmp_basic.c
+++ b/net/ipv4/netfilter/ip_nat_snmp_basic.c
@@ -1000,12 +1000,12 @@ static unsigned char snmp_trap_decode(st

return 1;

+err_addr_free:
+ kfree((unsigned long *)trap->ip_address);
+
err_id_free:
 kfree(trap->id);

-err_addr_free:
- kfree((unsigned long *)trap->ip_address);
-
return 0;
}

@@ -1123,11 +1123,10 @@ static int snmp_parse_mangle(unsigned ch
struct snmp_v1_trap trap;
unsigned char ret = snmp_trap_decode(&ctx, &trap, map, check);

- /* Discard trap allocations regardless */
- kfree(trap.id);
- kfree((unsigned long *)trap.ip_address);
-
- if (!ret)
+ if (ret) {
+ kfree(trap.id);
+ kfree((unsigned long *)trap.ip_address);
+ } else
return ret;

} else {
--- a/net/sctp/sm_statefuns.c
+++ b/net/sctp/sm_statefuns.c
@@ -1030,6 +1030,12 @@ sctp_disposition_t sctp_sf_backbeat_8_3(

```

```
commands);
```

```
    hbinfo = (sctp_sender_hb_info_t *) chunk->skb->data;
+ /* Make sure that the length of the parameter is what we expect */
+ if (ntohs(hbinfo->param_hdr.length) !=
+     sizeof(sctp_sender_hb_info_t)) {
+     return SCTP_DISPOSITION_DISCARD;
+ }
+
+ from_addr = hbinfo->daddr;
+ link = sctp_assoc_lookup_paddr(asoc, &from_addr);
```

```
--- a/security/selinux/ss/services.c
```

```
+++ b/security/selinux/ss/services.c
```

```
@@ -592,6 +592,10 @@ int security_sid_to_context(u32 sid, cha
```

```
    *scontext_len = strlen(initial_sid_to_string[sid]) + 1;
    scontextp = kmalloc(*scontext_len, GFP_ATOMIC);
+ if (!scontextp) {
+     rc = -ENOMEM;
+     goto out;
+ }
    strcpy(scontextp, initial_sid_to_string[sid]);
    *scontext = scontextp;
    goto out;
```

File Attachments

1) [diff-merge-2.6.16.18-20060530](#), downloaded 353 times
