

Re: Linux 2.6.15.7

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-03/msg09497.html>

- *From:* Greg KH <gregkh@xxxxxxx>
 - *Date:* Mon, 27 Mar 2006 23:59:52 -0800
-

```
diff --git a/Makefile b/Makefile
index 3a85dd1..acf0fe5 100644
--- a/Makefile
+++ b/Makefile
@@ -1,7 +1,7 @@
VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 15
-EXTRAVERSION = .6
+EXTRAVERSION = .7
NAME=Sliding Snow Leopard
```

DOCUMENTATION

```
diff --git a/drivers/infiniband/ulp/srp/ib_srp.c b/drivers/infiniband/ulp/srp/ib_srp.c
index ee9fe22..4eec4d7 100644
--- a/drivers/infiniband/ulp/srp/ib_srp.c
+++ b/drivers/infiniband/ulp/srp/ib_srp.c
@@ -1154,6 +1154,12 @@ static int srp_send_tsk_mgmt(struct scsi
```

```
spin_lock_irq(target->scsi_host->host_lock);
```

```
+ if (target->state == SRP_TARGET_DEAD ||
+ target->state == SRP_TARGET_REMOVED) {
+ scmnd->result = DID_BAD_TARGET << 16;
+ goto out;
+ }
+
+ if (scmnd->host_scribble == (void *) -1L)
+ goto out;
```

```
diff --git a/drivers/media/video/Kconfig b/drivers/media/video/Kconfig
index fc87efc..e4d12fa 100644
--- a/drivers/media/video/Kconfig
+++ b/drivers/media/video/Kconfig
@@ -340,6 +340,7 @@ config VIDEO_AUDIO_DECODER
config VIDEO_DECODER
tristate "Add support for additional video chipsets"
depends on VIDEO_DEV && I2C && EXPERIMENTAL
+ select FW_LOADER
```

---help---

Say Y here to compile drivers for SAA7115, SAA7127 and CX25840 video decoders.

diff --git a/fs/compat_ioctl.c b/fs/compat_ioctl.c

index 43a2508..8b19b33 100644

--- a/fs/compat_ioctl.c

+++ b/fs/compat_ioctl.c

@@ -687,7 +687,7 @@ static int dev_ifconf(unsigned int fd, u

ifr = ifc.ifc_req;

ifr32 = compat_ptr(ifc32.ifcbuf);

for (i = 0, j = 0;

- i + sizeof (struct ifreq32) < ifc32.ifc_len && j < ifc.ifc_len;

+ i + sizeof (struct ifreq32) <= ifc32.ifc_len && j < ifc.ifc_len;

i += sizeof (struct ifreq32), j += sizeof (struct ifreq)) {

if (copy_in_user(ifr32, ifr, sizeof (struct ifreq32)))

return -EFAULT;

diff --git a/fs/cramfs/inode.c b/fs/cramfs/inode.c

index 7fe8541..8ad52f5 100644

--- a/fs/cramfs/inode.c

+++ b/fs/cramfs/inode.c

@@ -36,7 +36,7 @@ static DECLARE_MUTEX(read_mutex);

/* These two macros may change in future, to provide better st_ino semantics. */

+#define CRAMINO(x) ((x)->offset?(x)->offset<<2:1)

+#define CRAMINO(x) (((x)->offset && (x)->size)?(x)->offset<<2:1)

#define OFFSET(x) ((x)->i_ino)

@@ -66,8 +66,36 @@ static int cramfs_iget5_test(struct inod

static int cramfs_iget5_set(struct inode *inode, void *opaque)

{

+ static struct timespec zertime;

struct cramfs_inode *cramfs_inode = opaque;

+ inode->i_mode = cramfs_inode->mode;

+ inode->i_uid = cramfs_inode->uid;

+ inode->i_size = cramfs_inode->size;

+ inode->i_blocks = (cramfs_inode->size - 1) / 512 + 1;

+ inode->i_blksize = PAGE_CACHE_SIZE;

+ inode->i_gid = cramfs_inode->gid;

+ /* Struct copy intentional */

+ inode->i_mtime = inode->i_atime = inode->i_ctime = zertime;

inode->i_ino = CRAMINO(cramfs_inode);

+ /* inode->i_nlink is left 1 - arguably wrong for directories,

+ but it's the best we can do without reading the directory

+ contents. 1 yields the right result in GNU find, even

+ without -noleaf option. */

+ if (S_ISREG(inode->i_mode)) {

+ inode->i_fop = &generic_ro_fops;

+ inode->i_data.a_ops = &cramfs_aops;

```

+ } else if (S_ISDIR(inode->i_mode)) {
+ inode->i_op = &cramfs_dir_inode_operations;
+ inode->i_fop = &cramfs_directory_operations;
+ } else if (S_ISLNK(inode->i_mode)) {
+ inode->i_op = &page_symlink_inode_operations;
+ inode->i_data.a_ops = &cramfs_aops;
+ } else {
+ inode->i_size = 0;
+ inode->i_blocks = 0;
+ init_special_inode(inode, inode->i_mode,
+ old_decode_dev(cramfs_inode->size));
+ }
return 0;
}

```

```

@@ -77,37 +105,7 @@ static struct inode *get_cramfs_inode(st
struct inode *inode = iget5_locked(sb, CRAMINO(cramfs_inode),
cramfs_iget5_test, cramfs_iget5_set,
cramfs_inode);
- static struct timespec zerotime;
-
if (inode && (inode->i_state & I_NEW)) {
- inode->i_mode = cramfs_inode->mode;
- inode->i_uid = cramfs_inode->uid;
- inode->i_size = cramfs_inode->size;
- inode->i_blocks = (cramfs_inode->size - 1) / 512 + 1;
- inode->i_blksize = PAGE_CACHE_SIZE;
- inode->i_gid = cramfs_inode->gid;
- /* Struct copy intentional */
- inode->i_mtime = inode->i_atime = inode->i_ctime = zerotime;
- inode->i_ino = CRAMINO(cramfs_inode);
- /* inode->i_nlink is left 1 - arguably wrong for directories,
- but it's the best we can do without reading the directory
- contents. 1 yields the right result in GNU find, even
- without -noleaf option. */
- if (S_ISREG(inode->i_mode)) {
- inode->i_fop = &generic_ro_fops;
- inode->i_data.a_ops = &cramfs_aops;
- } else if (S_ISDIR(inode->i_mode)) {
- inode->i_op = &cramfs_dir_inode_operations;
- inode->i_fop = &cramfs_directory_operations;
- } else if (S_ISLNK(inode->i_mode)) {
- inode->i_op = &page_symlink_inode_operations;
- inode->i_data.a_ops = &cramfs_aops;
- } else {
- inode->i_size = 0;
- inode->i_blocks = 0;
- init_special_inode(inode, inode->i_mode,
- old_decode_dev(cramfs_inode->size));
- }
unlock_new_inode(inode);

```

```

}
return inode;
diff --git a/fs/ext2/dir.c b/fs/ext2/dir.c
index 5b5f528..e7a1286 100644
--- a/fs/ext2/dir.c
+++ b/fs/ext2/dir.c
@@ -256,11 +256,10 @@ ext2_readdir (struct file * filp, void *
unsigned long npages = dir_pages(inode);
unsigned chunk_mask = ~(ext2_chunk_size(inode)-1);
unsigned char *types = NULL;
- int need_revalidate = (filp->f_version != inode->i_version);
- int ret;
+ int need_revalidate = filp->f_version != inode->i_version;

if (pos > inode->i_size - EXT2_DIR_REC_LEN(1))
- goto success;
+ return 0;

if (EXT2_HAS_INCOMPAT_FEATURE(sb, EXT2_FEATURE_INCOMPAT_FILETYPE))
types = ext2_filetype_table;
@@ -275,12 +274,15 @@ ext2_readdir (struct file * filp, void *
"bad page in %#lu",
inode->i_ino);
filp->f_pos += PAGE_CACHE_SIZE - offset;
- ret = -EIO;
- goto done;
+ return -EIO;
}
kaddr = page_address(page);
- if (need_revalidate) {
- offset = ext2_validate_entry(kaddr, offset, chunk_mask);
+ if (unlikely(need_revalidate)) {
+ if (offset) {
+ offset = ext2_validate_entry(kaddr, offset, chunk_mask);
+ filp->f_pos = (n<<PAGE_CACHE_SHIFT) + offset;
+ }
+ filp->f_version = inode->i_version;
need_revalidate = 0;
}
de = (ext2_dirent *) (kaddr+offset);
@@ -289,9 +291,8 @@ ext2_readdir (struct file * filp, void *
if (de->rec_len == 0) {
ext2_error(sb, __FUNCTION__,
"zero-length directory entry");
- ret = -EIO;
ext2_put_page(page);
- goto done;
+ return -EIO;
}
if (de->inode) {
int over;

```

```

@@ -306,19 +307,14 @@ ext2_readdir (struct file * filp, void *
le32_to_cpu(de->inode), d_type);
if (over) {
ext2_put_page(page);
- goto success;
+ return 0;
}
}
filp->f_pos += le16_to_cpu(de->rec_len);
}
ext2_put_page(page);
}
-
-success:
- ret = 0;
-done:
- filp->f_version = inode->i_version;
- return ret;
+ return 0;
}

/*
diff --git a/net/core/sock.c b/net/core/sock.c
index 13cc3be..2a3e235 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -403,8 +403,9 @@ set_rcvbuf:
if (!valbool) {
sk->sk_bound_dev_if = 0;
} else {
- if (optlen > IFNAMSIZ)
- optlen = IFNAMSIZ;
+ if (optlen > IFNAMSIZ - 1)
+ optlen = IFNAMSIZ - 1;
+ memset(devname, 0, sizeof(devname));
if (copy_from_user(devname, optval, optlen)) {
ret = -EFAULT;
break;
diff --git a/net/ipv4/ip_output.c b/net/ipv4/ip_output.c
index eba64e2..9844d9e 100644
--- a/net/ipv4/ip_output.c
+++ b/net/ipv4/ip_output.c
@@ -1237,11 +1237,7 @@ int ip_push_pending_frames(struct sock *
iph->tos = inet->tos;
iph->tot_len = htons(skb->len);
iph->frag_off = df;
- if (!df) {
- __ip_select_ident(iph, &rt->u.dst, 0);
- } else {
- iph->id = htons(inet->id++);
- }

```

Re: Linux 2.6.15.7

```
+ ip_select_ident(iph, &rt->u.dst, sk);
iph->ttl = ttl;
iph->protocol = sk->sk_protocol;
iph->saddr = rt->rt_src;
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index 36339eb..08f80e2 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -524,7 +524,7 @@ ipq_rcv_skb(struct sk_buff *skb)
write_unlock_bh(&queue_lock);
```

```
status = ipq_receive_peer(NLMSG_DATA(nlh), type,
- skblen - NLMSG_LENGTH(0));
+ nmsglen - NLMSG_LENGTH(0));
if (status < 0)
RCV_SKB_FAIL(status);
```

```
diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index 5027bbe..af06350 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -522,7 +522,7 @@ ipq_rcv_skb(struct sk_buff *skb)
write_unlock_bh(&queue_lock);
```

```
status = ipq_receive_peer(NLMSG_DATA(nlh), type,
- skblen - NLMSG_LENGTH(0));
+ nmsglen - NLMSG_LENGTH(0));
if (status < 0)
RCV_SKB_FAIL(status);
```

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>

Re: Linux 2.6.15.7