
Subject: [PATCH 2.6.24-rc8-mm1 06/15] IPC: get rid of the use *_setbuf structure.
Posted by [Pierre Peiffer](#) on Tue, 29 Jan 2008 16:02:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Pierre Peiffer <pierre.peiffer@bull.net>

All IPCs make use of an intermediate *_setbuf structure to handle the IPC_SET command. This is not really needed and, moreover, it complicates a little bit the code.

This patch gets rid of the use of it and uses directly the semid64_ds/msgid64_ds/shmid64_ds structure.

In addition to removing one structure declaration, it also simplifies and improves a little bit the common 64-bits path.
Moreover, this will simplify the code for handling the IPC_SETALL command provided in the next patch.

Signed-off-by: Pierre Peiffer <pierre.peiffer@bull.net>

Acked-by: Serge Hallyn <serue@us.ibm.com>

```
ipc/msg.c | 51 ++++++-----
ipc/sem.c | 40 ++++++-----
ipc/shm.c | 41 ++++++-----
3 files changed, 46 insertions(+), 86 deletions(-)
```

Index: b/ipc/msg.c

=====

--- a/ipc/msg.c

+++ b/ipc/msg.c

```
@@ -351,31 +351,14 @@ copy_msqid_to_user(void __user *buf, str
 }
 }
```

```
-struct msq_setbuf {
- unsigned long qbytes;
- uid_t uid;
- gid_t gid;
- mode_t mode;
-};
```

```
-
static inline unsigned long
-copy_msqid_from_user(struct msq_setbuf *out, void __user *buf, int version)
+copy_msqid_from_user(struct msqid64_ds *out, void __user *buf, int version)
{
    switch(version) {
    case IPC_64:
```

```

- {
- struct msqid64_ds tbuf;
-
- if (copy_from_user(&tbuf, buf, sizeof(tbuf)))
+ if (copy_from_user(out, buf, sizeof(*out)))
    return -EFAULT;
-
- out->qbytes = tbuf.msg_qbytes;
- out->uid = tbuf.msg_perm.uid;
- out->gid = tbuf.msg_perm.gid;
- out->mode = tbuf.msg_perm.mode;
-
    return 0;
- }
case IPC_OLD:
{
    struct msqid_ds tbuf_old;
@@ -383,14 +366,14 @@ copy_msqid_from_user(struct msq_setbuf *
    if (copy_from_user(&tbuf_old, buf, sizeof(tbuf_old)))
        return -EFAULT;

- out->uid = tbuf_old.msg_perm.uid;
- out->gid = tbuf_old.msg_perm.gid;
- out->mode = tbuf_old.msg_perm.mode;
+ out->msg_perm.uid = tbuf_old.msg_perm.uid;
+ out->msg_perm.gid = tbuf_old.msg_perm.gid;
+ out->msg_perm.mode = tbuf_old.msg_perm.mode;

    if (tbuf_old.msg_qbytes == 0)
- out->qbytes = tbuf_old.msg_lqbytes;
+ out->msg_qbytes = tbuf_old.msg_lqbytes;
    else
- out->qbytes = tbuf_old.msg_qbytes;
+ out->msg_qbytes = tbuf_old.msg_qbytes;

    return 0;
}
@@ -408,12 +391,12 @@ static int msgctl_down(struct ipc_namesp
    struct msqid_ds __user *buf, int version)
{
    struct kern_ipc_perm *ipcp;
- struct msq_setbuf setbuf;
+ struct msqid64_ds msqid64;
    struct msg_queue *msq;
    int err;

    if (cmd == IPC_SET) {
- if (copy_msqid_from_user(&setbuf, buf, version))

```

```

+ if (copy_msqid_from_user(&msqid64, buf, version))
    return -EFAULT;
}

@@ -431,8 +414,10 @@ static int msgctl_down(struct ipc_namesp
    goto out_unlock;

    if (cmd == IPC_SET) {
- err = audit_ipc_set_perm(setbuf.qbytes, setbuf.uid, setbuf.gid,
-     setbuf.mode);
+ err = audit_ipc_set_perm(msqid64.msg_qbytes,
+     msqid64.msg_perm.uid,
+     msqid64.msg_perm.gid,
+     msqid64.msg_perm.mode);
    if (err)
        goto out_unlock;
    }
@@ -454,18 +439,18 @@ static int msgctl_down(struct ipc_namesp
    freeque(ns, ipcp);
    goto out_up;
    case IPC_SET:
- if (setbuf.qbytes > ns->msg_ctlmnrb &&
+ if (msqid64.msg_qbytes > ns->msg_ctlmnrb &&
        !capable(CAP_SYS_RESOURCE)) {
    err = -EPERM;
    goto out_unlock;
    }

- msq->q_qbytes = setbuf.qbytes;
+ msq->q_qbytes = msqid64.msg_qbytes;

- ipcp->uid = setbuf.uid;
- ipcp->gid = setbuf.gid;
+ ipcp->uid = msqid64.msg_perm.uid;
+ ipcp->gid = msqid64.msg_perm.gid;
    ipcp->mode = (ipcp->mode & ~S_IRWXUGO) |
-     (S_IRWXUGO & setbuf.mode);
+     (S_IRWXUGO & msqid64.msg_perm.mode);
    msq->q_ctime = get_seconds();
    /* sleeping receivers might be excluded by
     * stricter permissions.
Index: b/ipc/sem.c
=====
--- a/ipc/sem.c
+++ b/ipc/sem.c
@@ -837,28 +837,14 @@ out_free:
    return err;
    }

```

```

-struct sem_setbuf {
- uid_t uid;
- gid_t gid;
- mode_t mode;
-};
-
-static inline unsigned long copy_semid_from_user(struct sem_setbuf *out, void __user *buf, int
version)
+static inline unsigned long
+copy_semid_from_user(struct semid64_ds *out, void __user *buf, int version)
{
    switch(version) {
        case IPC_64:
            {
                struct semid64_ds tbuf;

                if(copy_from_user(&tbuf, buf, sizeof(tbuf)))
+ if (copy_from_user(out, buf, sizeof(*out)))
                    return -EFAULT;

                out->uid = tbuf.sem_perm.uid;
                out->gid = tbuf.sem_perm.gid;
                out->mode = tbuf.sem_perm.mode;

                return 0;
            }
        case IPC_OLD:
            {
                struct semid_ds tbuf_old;
@@ -866,9 +852,9 @@ static inline unsigned long copy_semid_f
                if(copy_from_user(&tbuf_old, buf, sizeof(tbuf_old)))
                    return -EFAULT;

                out->uid = tbuf_old.sem_perm.uid;
                out->gid = tbuf_old.sem_perm.gid;
                out->mode = tbuf_old.sem_perm.mode;
+ out->sem_perm.uid = tbuf_old.sem_perm.uid;
+ out->sem_perm.gid = tbuf_old.sem_perm.gid;
+ out->sem_perm.mode = tbuf_old.sem_perm.mode;

                return 0;
            }
@@ -887,11 +873,11 @@ static int semctl_down(struct ipc_namesp
{
    struct sem_array *sma;
    int err;
- struct sem_setbuf uninitialized_var(setbuf);

```

```

+ struct semid64_ds semid64;
  struct kern_ipc_perm *ipcp;

  if(cmd == IPC_SET) {
- if(copy_semids_from_user (&setbuf, arg.buf, version))
+ if (copy_semids_from_user(&semid64, arg.buf, version))
    return -EFAULT;
  }
  down_write(&sem_ids(ns).rw_mutex);
@@ -908,7 +894,9 @@ static int semctl_down(struct ipc_namesp
  goto out_unlock;

  if (cmd == IPC_SET) {
- err = audit_ipc_set_perm(0, setbuf.uid, setbuf.gid, setbuf.mode);
+ err = audit_ipc_set_perm(0, semid64.sem_perm.uid,
+   semid64.sem_perm.gid,
+   semid64.sem_perm.mode);
  if (err)
    goto out_unlock;
  }
@@ -927,10 +915,10 @@ static int semctl_down(struct ipc_namesp
  freeary(ns, ipcp);
  goto out_up;
  case IPC_SET:
- ipcp->uid = setbuf.uid;
- ipcp->gid = setbuf.gid;
+ ipcp->uid = semid64.sem_perm.uid;
+ ipcp->gid = semid64.sem_perm.gid;
  ipcp->mode = (ipcp->mode & ~S_IRWXUGO)
-   | (setbuf.mode & S_IRWXUGO);
+   | (semid64.sem_perm.mode & S_IRWXUGO);
  sma->sem_ctime = get_seconds();
  break;
  default:
Index: b/ipc/shm.c
=====
--- a/ipc/shm.c
+++ b/ipc/shm.c
@@ -520,28 +520,14 @@ static inline unsigned long copy_shmid_t
 }
 }

-struct shm_setbuf {
- uid_t uid;
- gid_t gid;
- mode_t mode;
-};
-

```

```

-static inline unsigned long copy_shmid_from_user(struct shm_setbuf *out, void __user *buf, int
version)
+static inline unsigned long
+copy_shmid_from_user(struct shmid64_ds *out, void __user *buf, int version)
{
    switch(version) {
    case IPC_64:
-    {
-    struct shmid64_ds tbuf;
-
-    if (copy_from_user(&tbuf, buf, sizeof(tbuf)))
+    if (copy_from_user(out, buf, sizeof(*out)))
        return -EFAULT;
-
-    out->uid = tbuf.shm_perm.uid;
-    out->gid = tbuf.shm_perm.gid;
-    out->mode = tbuf.shm_perm.mode;
-
        return 0;
-    }
    case IPC_OLD:
        {
            struct shmid_ds tbuf_old;
@@ -549,9 +535,9 @@ static inline unsigned long copy_shmid_f
            if (copy_from_user(&tbuf_old, buf, sizeof(tbuf_old)))
                return -EFAULT;

-    out->uid = tbuf_old.shm_perm.uid;
-    out->gid = tbuf_old.shm_perm.gid;
-    out->mode = tbuf_old.shm_perm.mode;
+    out->shm_perm.uid = tbuf_old.shm_perm.uid;
+    out->shm_perm.gid = tbuf_old.shm_perm.gid;
+    out->shm_perm.mode = tbuf_old.shm_perm.mode;

        return 0;
    }
}
@@ -634,12 +620,12 @@ static int shmctl_down(struct ipc_namesp
    struct shmid_ds __user *buf, int version)
{
    struct kern_ipc_perm *ipcp;
-    struct shm_setbuf setbuf;
+    struct shmid64_ds shmid64;
    struct shmid_kernel *shp;
    int err;

    if (cmd == IPC_SET) {
-    if (copy_shmid_from_user(&setbuf, buf, version))
+    if (copy_shmid_from_user(&shmid64, buf, version))

```

```

    return -EFAULT;
}

@@ -657,8 +643,9 @@ static int shmctl_down(struct ipc_namesp
    goto out_unlock;

    if (cmd == IPC_SET) {
-   err = audit_ipc_set_perm(0, setbuf.uid,
-       setbuf.gid, setbuf.mode);
+   err = audit_ipc_set_perm(0, shm64.shm_perm.uid,
+       shm64.shm_perm.gid,
+       shm64.shm_perm.mode);
    if (err)
        goto out_unlock;
    }
@@ -678,10 +665,10 @@ static int shmctl_down(struct ipc_namesp
    do_shm_rmid(ns, ipc);
    goto out_up;
    case IPC_SET:
-   ipc->uid = setbuf.uid;
-   ipc->gid = setbuf.gid;
+   ipc->uid = shm64.shm_perm.uid;
+   ipc->gid = shm64.shm_perm.gid;
    ipc->mode = (ipc->mode & ~S_IRWXUGO)
-   | (setbuf.mode & S_IRWXUGO);
+   | (shm64.shm_perm.mode & S_IRWXUGO);
    shp->shm_ctim = get_seconds();
    break;
    default:

```

--
 Pierre Peiffer

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