
Subject: [PATCH][IPC] Uninline some code from util.h
Posted by [Pavel Emelianov](#) on Wed, 21 Nov 2007 13:02:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

ipc_lock_check_down(), ipc_lock_check() and ipcget() seem too large to be inline. Besides, they give no optimization being inline as they perform calls inside in any case.

Moving them into ipc/util.c saves 500 bytes of vmlinux and shortens IPC internal API.

```
$ ./scripts/bloat-o-meter vmlinux-orig vmlinux
add/remove: 3/2 grow/shrink: 0/10 up/down: 490/-989 (-499)
function          old    new  delta
ipcget             -    392  +392
ipc_lock_check_down      -    49   +49
ipc_lock_check        -    49   +49
sys_semget         119   105   -14
sys_shmget         108    86   -22
sys_msgget         100    78   -22
do_msgsnd         665   631   -34
do_msgrcv         680   644   -36
do_shmat          771   733   -38
sys_msgctl       1302  1229   -73
ipcget_new         80     -   -80
sys_semtimedop      1534  1452   -82
sys_semctl       2034  1922  -112
sys_shmctl       1919  1765  -154
ipcget_public     322     -  -322
```

The ipcget() growth is the result of gcc in-lining of currently static ipcget_new/_public.

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

```
diff --git a/ipc/util.c b/ipc/util.c
index 5432b8e..910db77 100644
--- a/ipc/util.c
+++ b/ipc/util.c
@@ -248,7 +248,7 @@ int ipc_addid(struct ipc_ids* ids, struct kern_ipc_perm* new, int size)
 * This routine is called by sys_msgget, sys_semget() and sys_shmget()
 * when the key is IPC_PRIVATE.
 */
-int ipcget_new(struct ipc_namespace *ns, struct ipc_ids *ids,
+static int ipcget_new(struct ipc_namespace *ns, struct ipc_ids *ids,
 struct ipc_ops *ops, struct ipc_params *params)
```

```

{
    int err;
@@ -312,7 +312,7 @@ static int ipc_check_perms(struct kern_ipc_perm *ipcp, struct ipc_ops
*ops,
*
* On success, the ipc id is returned.
*/
-int ipcget_public(struct ipc_namespace *ns, struct ipc_ids *ids,
+static int ipcget_public(struct ipc_namespace *ns, struct ipc_ids *ids,
    struct ipc_ops *ops, struct ipc_params *params)
{
    struct kern_ipc_perm *ipcp;
@@ -710,6 +710,57 @@ struct kern_ipc_perm *ipc_lock_down(struct ipc_ids *ids, int id)
    return out;
}

+struct kern_ipc_perm *ipc_lock_check_down(struct ipc_ids *ids, int id)
+{
+ struct kern_ipc_perm *out;
+
+ out = ipc_lock_down(ids, id);
+ if (IS_ERR(out))
+ return out;
+
+ if (ipc_checkid(out, id)) {
+ ipc_unlock(out);
+ return ERR_PTR(-EIDRM);
+ }
+
+ return out;
+}
+
+struct kern_ipc_perm *ipc_lock_check(struct ipc_ids *ids, int id)
+{
+ struct kern_ipc_perm *out;
+
+ out = ipc_lock(ids, id);
+ if (IS_ERR(out))
+ return out;
+
+ if (ipc_checkid(out, id)) {
+ ipc_unlock(out);
+ return ERR_PTR(-EIDRM);
+ }
+
+ return out;
+}
+

```

```

+/**
+ * ipcget - Common sys_*get() code
+ * @ns : namespace
+ * @ids : IPC identifier set
+ * @ops : operations to be called on ipc object creation, permission checks
+ *       and further checks
+ * @params : the parameters needed by the previous operations.
+ *
+ * Common routine called by sys_msgget(), sys_semget() and sys_shmget().
+ */
+int ipcget(struct ipc_namespace *ns, struct ipc_ids *ids,
+ struct ipc_ops *ops, struct ipc_params *params)
+{
+ if (params->key == IPC_PRIVATE)
+ return ipcget_new(ns, ids, ops, params);
+ else
+ return ipcget_public(ns, ids, ops, params);
+}
+
+ #ifdef __ARCH_WANT_IPC_PARSE_VERSION

```

```

diff --git a/ipc/util.h b/ipc/util.h
index fc6b729..ca245fa 100644

```

```

--- a/ipc/util.h
+++ b/ipc/util.h
@@ -131,10 +131,6 @@ int ipc_parse_version (int *cmd);
extern void free_msg(struct msg_msg *msg);
extern struct msg_msg *load_msg(const void __user *src, int len);
extern int store_msg(void __user *dest, struct msg_msg *msg, int len);
-extern int ipcget_new(struct ipc_namespace *, struct ipc_ids *,
- struct ipc_ops *, struct ipc_params *);
-extern int ipcget_public(struct ipc_namespace *, struct ipc_ids *,
- struct ipc_ops *, struct ipc_params *);

static inline int ipc_buildid(int id, int seq)
{
@@ -163,57 +159,9 @@ static inline void ipc_unlock(struct kern_ipc_perm *perm)
rcu_read_unlock();
}

-static inline struct kern_ipc_perm *ipc_lock_check_down(struct ipc_ids *ids,
- int id)
-{
- struct kern_ipc_perm *out;
-
- out = ipc_lock_down(ids, id);
- if (IS_ERR(out))

```

```

- return out;
-
- if (ipc_checkid(out, id)) {
- ipc_unlock(out);
- return ERR_PTR(-EIDRM);
- }
-
- return out;
-}
-
-static inline struct kern_ipc_perm *ipc_lock_check(struct ipc_ids *ids,
- int id)
-{
- struct kern_ipc_perm *out;
-
- out = ipc_lock(ids, id);
- if (IS_ERR(out))
- return out;
-
- if (ipc_checkid(out, id)) {
- ipc_unlock(out);
- return ERR_PTR(-EIDRM);
- }
-
- return out;
-}
-
-/**
- * ipcget - Common sys_get() code
- * @ns : namespace
- * @ids : IPC identifier set
- * @ops : operations to be called on ipc object creation, permission checks
- *       and further checks
- * @params : the parameters needed by the previous operations.
- *
- * Common routine called by sys_msgget(), sys_semget() and sys_shmget().
- */
-static inline int ipcget(struct ipc_namespace *ns, struct ipc_ids *ids,
- struct ipc_ops *ops, struct ipc_params *params)
-{
- if (params->key == IPC_PRIVATE)
- return ipcget_new(ns, ids, ops, params);
- else
- return ipcget_public(ns, ids, ops, params);
-}
+struct kern_ipc_perm *ipc_lock_check_down(struct ipc_ids *ids, int id);
+struct kern_ipc_perm *ipc_lock_check(struct ipc_ids *ids, int id);
+int ipcget(struct ipc_namespace *ns, struct ipc_ids *ids,

```

```
+ struct ipc_ops *ops, struct ipc_params *params);  
  
#endif
```
