Subject: [PATCH 5/5] ipc: cleanup do_msgrcv() aroung MSG_COPY feature
Posted by Stanislav Kinsbursky on Fri, 26 Oct 2012 11:06:17 GMT
View Forum Message <> Reply to Message

MSG_COPY feature was developed for Checkpoint/Restart In User space project
and thus wrapped in CONFIG_CHECKPOINT_RESTORE macro. But code look a bit ugly.
So this patch is an attempt to cleanup do_msgrcv() a bit and make it looks
better.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

```
---
 ipc/msg.c |   79 +++++++++++++++++++++++++++++++++++++++++++++++-----------------------
 1 files changed, 48 insertions(+), 31 deletions(-)

diff --git a/ipc/msg.c b/ipc/msg.c
index 91873df..d20ffc7 100644
--- a/ipc/msg.c
+++ b/ipc/msg.c
@@ -769,6 +769,45 @@ static long do_msg_fill(void __user *dest, struct msg_msg *msg, size_t
bufsz)
  return msgsz;
 }

+#ifdef CONFIG_CHECKPOINT_RESTORE
+static inline struct msg_msg *fill_copy(unsigned long copy_nr,
+     unsigned long msg_nr,
+     struct msg_msg *msg,
+     struct msg_msg *copy)
+{
+ if (copy_nr == msg_nr)
+  return copy_msg(msg, copy);
+ return NULL;
+}
+
+static inline struct msg_msg *prepare_copy(void __user *buf, size_t bufsz,
+       int msgflg, long *msgtyp,
+       unsigned long *copy_number)
+{
+ struct msg_msg *copy;
+
+ *copy_number = *msgtyp;
+ *msgtyp = 0;
+ /*
+  * Create dummy message to copy real message to.
+  */
+ copy = load_msg(buf, bufsz);
+ if (!IS_ERR(copy))
+  copy->m_ts = bufsz;
```

```
+ return copy;
+}
+
+static inline void free_copy(int msgflg, struct msg_msg *copy)
+{
+ if (msgflg & MSG_COPY)
+  free_msg(copy);
+}
+#else
+#define free_copy(msgflg, copy)    do {} while (0)
+#define prepare_copy(buf, sz, msgflg, msgtyp, copy_nr) ERR_PTR(-ENOSYS)
+#define fill_copy(copy_nr, msg_nr, msg, copy)  NULL
+#endif
+
 long do_msgrcv(int msqid, void __user *buf, size_t bufsz, long msgtyp,
        int msgflg,
        long (*msg_handler)(void __user *, struct msg_msg *, size_t))
@@ -777,38 +816,22 @@ long do_msgrcv(int msqid, void __user *buf, size_t bufsz, long msgtyp,
  struct msg_msg *msg;
  int mode;
  struct ipc_namespace *ns;
-#ifdef CONFIG_CHECKPOINT_RESTORE
- struct msg_msg *copy = NULL;
- unsigned long copy_number = 0;
-#endif
+ struct msg_msg *copy;
+ unsigned long __maybe_unused copy_number;

  if (msqid < 0 || (long) bufsz < 0)
   return -EINVAL;
  if (msgflg & MSG_COPY) {
-#ifdef CONFIG_CHECKPOINT_RESTORE
-  copy_number = msgtyp;
-  msgtyp = 0;
-
-  /*
-   * Create dummy message to copy real message to.
-   */
-  copy = load_msg(buf, bufsz);
+  copy = prepare_copy(buf, bufsz, msgflg, &msgtyp, &copy_number);
   if (IS_ERR(copy))
    return PTR_ERR(copy);
-  copy->m_ts = bufsz;
-#else
-  return -ENOSYS;
-#endif
  }
  mode = convert_mode(&msgtyp, msgflg);
```

```
  ns = current->nsproxy->ipc_ns;

  msq = msg_lock_check(ns, msqid);
  if (IS_ERR(msq)) {
-#ifdef CONFIG_CHECKPOINT_RESTORE
-  if (msgflg & MSG_COPY)
-   free_msg(copy);
-#endif
+  free_copy(msgflg, copy);
   return PTR_ERR(msq);
  }

@@ -835,13 +858,12 @@ long do_msgrcv(int msqid, void __user *buf, size_t bufsz, long msgtyp,
     if (mode == SEARCH_LESSEQUAL &&
       walk_msg->m_type != 1) {
      msgtyp = walk_msg->m_type - 1;
-#ifdef CONFIG_CHECKPOINT_RESTORE
     } else if (msgflg & MSG_COPY) {
-     if (copy_number == msg_counter) {
-      msg = copy_msg(walk_msg, copy);
+     msg = fill_copy(copy_number,
+       msg_counter,
+       walk_msg, copy);
+     if (msg)
       break;
-     }
-#endif
     } else
      break;
     msg_counter++;
@@ -857,10 +879,8 @@ long do_msgrcv(int msqid, void __user *buf, size_t bufsz, long msgtyp,
     msg = ERR_PTR(-E2BIG);
     goto out_unlock;
    }
-#ifdef CONFIG_CHECKPOINT_RESTORE
    if (msgflg & MSG_COPY)
     goto out_unlock;
-#endif
    list_del(&msg->m_list);
    msq->q_qnum--;
    msq->q_rtime = get_seconds();
@@ -945,10 +965,7 @@ out_unlock:
   }
  }
  if (IS_ERR(msg)) {
-#ifdef CONFIG_CHECKPOINT_RESTORE
-  if (msgflg & MSG_COPY)
-   free_msg(copy);
```

```
-#endif
+  free_copy(msgflg, copy);
   return PTR_ERR(msg);
  }
```