```
 */
static void __do_notify(struct mqueue_inode_info *info)
{
        /* notification
         * invoked when there is registered process and there isn't process
         * waiting synchronously for message AND state of queue changed from
         * empty to not empty. Here we are sure that no one is waiting
         * synchronously. */
        if (info->notify_owner &&
            info->attr.mq_curmsgs == 1) {
                struct siginfo sig_i;
                switch (info->notify.sigev_notify) {
                case SIGEV_NONE:
                        break;
                case SIGEV_SIGNAL:
                        /* sends signal */

                        sig_i.si_signo = info->notify.sigev_signo;
                        sig_i.si_errno = 0;
                        sig_i.si_code = SI_MESGQ;
                        sig_i.si_value = info->notify.sigev_value;
                        /* map current pid/uid into info->owner's namespaces */
                        rcu_read_lock();
                        sig_i.si_pid = task_tgid_nr_ns(current,
                                                ns_of_pid(info->notify_owner));
                        sig_i.si_uid = from_kuid_munged(info->notify_user_ns, current_uid());
                        rcu_read_unlock();
```

```
625  */
626 static void __do_notify(struct mqueue_inode_
627 {
628         /* notification
629          * invoked when there is                process and there isn't process
630          * waiting synchronous                  sage AND state of queue changed from
631          * empty to not empt                    e are sure that no one is waiting
632          * synchronously.
633         if (info->notify_
634             info->attr.m                     s == 1) {
635                 struct                       sig_i;
636                 switch                       notify.sigev_notify) {
637                 case SI                      E:
638
639                 case SI                      NAL:
640                                              ds signal */
641
642                                              _signo = inf                 sigev_signo;
643                                 s            errno =
644                                 si           de
645                                 sig_                     >notify.sigev_value;
646                                 /* ma                    uid into info->owner's namespaces */
647                                 rcu_read
648                                 sig_i.si_p                   nr_ns(current,
649
650                                 sig_i.si_uid = fro                        , current_uid());
651                                 rcu_read_unlock();
```