

```

*****
26470 Mon Dec 1 21:05:21 2014
new/usr/src/uts/common/fs/dev/sdev_profile.c
5360 Race condition in devfs upgrades reader to writer incidentally and causes p
*****
_____unchanged_portion_omitted_____

657 /*
658  * Return True if directory cache is out of date and should be updated.
659  */
660 static boolean_t
661 prof_dev_needupdate(sdev_node_t *ddv)
662 {
663     sdev_node_t *gdir = ddv->sdev_origin;

665     /*
666      * Caller can have either reader or writer lock
667      */
668     ASSERT(RW_LOCK_HELD(&ddv->sdev_contents));

670     /*
671      * We need to rebuild the directory content if
672      * - ddv is not in a SDEV_ZOMBIE state
673      * - SDEV_BUILD is set OR
674      * - The device tree generation number has changed OR
675      * - The corresponding /dev namespace has been updated
676      */
677     return ((ddv->sdev_state != SDEV_ZOMBIE) &&
678             (((ddv->sdev_flags & SDEV_BUILD) != 0) ||
679              (ddv->sdev_devtree_gen != devtree_gen) ||
680              ((gdir != NULL) &&
681               (ddv->sdev_ldir_gen != gdir->sdev_gdir_gen))));
682 }

684 /*
685  * Build directory vnodes based on the profile and the global
686  * dev instance.
687  */
688 void
689 prof_filldir(sdev_node_t *ddv)
690 {
691     sdev_node_t *gdir;
692     int firsttime = 1;
693     struct sdev_node *gdir = ddv->sdev_origin;

695     ASSERT(RW_READ_HELD(&ddv->sdev_contents));

697     if (!prof_dev_needupdate(ddv)) {
698         ASSERT(RW_READ_HELD(&ddv->sdev_contents));
699         return;
700     }
701     /*
702      * Upgrade to writer lock
703      * We need to rebuild the directory content if
704      * - SDEV_BUILD is set
705      * - The device tree generation number has changed
706      * - The corresponding /dev namespace has been updated
707      */
708     if (rw_tryupgrade(&ddv->sdev_contents) == 0) {
709         /*
710          * We need to drop the read lock and re-acquire it as a
711          * write lock. While we do this the condition may change so we
712          * need to re-check condition
713          */
714         check_build:

```

```

676     if ((ddv->sdev_flags & SDEV_BUILD) == 0 &&
677         ddv->sdev_devtree_gen == devtree_gen &&
678         (gdir == NULL || ddv->sdev_ldir_gen
679          == gdir->sdev_gdir_gen))
680         return; /* already up to date */

682     /* We may have become a zombie (across a try) */
683     if (ddv->sdev_state == SDEV_ZOMBIE)
684         return;

686     if (firsttime && rw_tryupgrade(&ddv->sdev_contents) == 0) {
687         rw_exit(&ddv->sdev_contents);
688         firsttime = 0;
689         rw_enter(&ddv->sdev_contents, RW_WRITER);
690         if (!prof_dev_needupdate(ddv)) {
691             /* Downgrade back to the read lock before returning */
692             rw_downgrade(&ddv->sdev_contents);
693             return;
694         }
695         goto check_build;
696     }

698     /* At this point we should have a write lock */
699     ASSERT(RW_WRITE_HELD(&ddv->sdev_contents));

701     sdcmn_err10(("devtree_gen (%s): %ld -> %ld\n",
702                ddv->sdev_path, ddv->sdev_devtree_gen, devtree_gen));

704     gdir = ddv->sdev_origin;

706     if (gdir != NULL)
707         if (gdir)
708             sdcmn_err10(("sdev_dir_gen (%s): %ld -> %ld\n",
709                ddv->sdev_path, ddv->sdev_ldir_gen,
710                gdir->sdev_gdir_gen));

712     /* update flags and generation number so next filldir is quick */
713     if ((ddv->sdev_flags & SDEV_BUILD) == SDEV_BUILD) {
714         ddv->sdev_flags &= ~SDEV_BUILD;
715     }
716     ddv->sdev_devtree_gen = devtree_gen;
717     if (gdir != NULL)
718         if (gdir)
719             ddv->sdev_ldir_gen = gdir->sdev_gdir_gen;

721     prof_make_symlinks(ddv);
722     prof_make_maps(ddv);
723     prof_make_names(ddv);
724     rw_downgrade(&ddv->sdev_contents);
725 }
_____unchanged_portion_omitted_____

```