Subject: [RFC PATCH 3/5] SUNRPC: pipefs dentry lookup helper introduced
Posted by Stanislav Kinsbursky on Mon, 17 Oct 2011 12:12:30 GMT
View Forum Message <> Reply to Message

In all places, where pipefs dentries are created, only directory inode is
actually required to create new dentry. And all this directories has root
pipefs dentry as their parent. So we actually don't need this pipefs mount
point at all if some pipefs lookup method will be provided.
IOW, all we really need is just superblock and simple lookup method to find
root's dentry child dentry with appropriate name. And this patch introduces
this method.
Note, that no loking implemented in rpc_d_lookup_sb(). So it can be used only
in case of assurance, that pipefs superblock still exist. IOW, we can use this
method only in pipefs mount-umount notificators.

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

---
 include/linux/sunrpc/rpc_pipe_fs.h | 3 +++
 net/sunrpc/rpc_pipe.c | 16 ++++++++++++++++
 2 files changed, 19 insertions(+), 0 deletions(-)

diff --git a/include/linux/sunrpc/rpc_pipe_fs.h b/include/linux/sunrpc/rpc_pipe_fs.h
index c1cdb2f..4a327ad 100644
--- a/include/linux/sunrpc/rpc_pipe_fs.h
+++ b/include/linux/sunrpc/rpc_pipe_fs.h
@@ -52,6 +52,9 @@ enum {
  RPC_PIPEFS_UMOUNT,
 };

+extern struct dentry *rpc_d_lookup_sb(const struct super_block *sb,
+   const unsigned char *dir_name);
+
 extern int rpc_queue_upcall(struct inode *, struct rpc_pipe_msg *);

 struct rpc_clnt;
diff --git a/net/sunrpc/rpc_pipe.c b/net/sunrpc/rpc_pipe.c
index c90a7e0..a8a8812 100644
--- a/net/sunrpc/rpc_pipe.c
+++ b/net/sunrpc/rpc_pipe.c
@@ -1017,6 +1017,22 @@ static const struct rpc_filelist files[] = {
  },
 };

+/*
+ * This call can be used only in RPC pipefs mount notification hooks.
+ */
+struct dentry *rpc_d_lookup_sb(const struct super_block *sb,

```
+          const unsigned char *dir_name)
+{
+ struct qstr dir = {
+  .name = dir_name,
+  .len = strlen(dir_name),
+  .hash = full_name_hash(dir_name, strlen(dir_name)),
+ };
+
+ return d_lookup(sb->s_root, &dir);
+}
+EXPORT_SYMBOL_GPL(rpc_d_lookup_sb);
+
 static int
 rpc_fill_super(struct super_block *sb, void *data, int silent)
 {
```