NAME

chGrp, chMod, chOwn – changing permissions of Shore objects

SYNOPSIS

```
(TODO: make sure all these assertions are enforced by the code!)

class Permissions {
      // from svas/include/permissions.h:
      enum {
        SetUid  =04000, SetGid  =02000, Sticky  =01000,
        Rown    =00400, Wown    =00200, Xown    =00100,
        Rgrp    =00040, Wgrp    =00020, Xgrp    =00010,
        Rpub    =00004, Wpub    =00002, Xpub    =00001
      };
      ...
};

VASResult       shore_vas::chGrp(
      const       Path  file,
      gid_t             gid
);

VASResult       shore_vas::chMod(
      const       Path  file,
      mode_t                  mode
);

VASResult       shore_vas::chOwn(
      const       Path  file,
      uid_t             uid
);
```

DESCRIPTION

Each Shore object has an owner, group, and permission bits, analogous to Unix owner, group and permission bits. Only registered objects have their own permissions; anonymous objects inherit the permissions of their pools.

When a registered object is created, its default permissions are determined as follows:

group — if the **Permissions::SetGid** of the parent directory is clear, the object's group is the effective group ID of the client process, otherwise it is the the effective group ID of the directory in which the object is created (BSD file-creation semantics are not supported)

owner — the effective user ID of the client process that creates the object

mode — the mode given as an argument to the create method, modified bye the client process's **umask.**

Each of the above methods takes a pathname for an object to change. If the object identified by *path* is a symbolic link or a cross-reference, the properties of the symbolic link are changed, not the properties of the object to which it refers.

Only the super-user may change the owner of an object. The owner of an object may change the object's group to a group of which he is a member.

The super-user may change an object's group to any group.

The effective user ID of the client process must match the owner of the file (or it must be super-user) in order to change the mode of the file.

If the effective user ID of the client process is not super-user and the process successfully changes the group of an object, the object's **Permissions::SetUid** and **Permissions::SetGid** bits are cleared.

If the effective user ID of the client process is not super-user and the process attempts to change the object's **Permissions::SetGid** bit, and the object's group is not in the process's supplementary groups, the **Permissions::SetGid** bit is cleared.

If a user other than the super-user updates an object, the **Permissions::SetUid** and **Permissions::SetGid** bits are cleared.

ARGUMENTS

The group ID is presumed to be a value taken from the Unix group database.

The user ID is presumed to be a value taken from the Unix password database.

The bits of a *mode_t* value have the following meanings:

SetUid
: Set user ID on execution: the client process's effective user ID is set to that of the owner of the file when the file is executed. For registered objects, this bit is cleared when the object is written.

SetGid
: Set group ID on execution: the client process's effective group ID is set to that of the owner of the file when the file is executed. For registered objects, this bit is cleared when the object is written.

Sticky
: If this bit is set on a directory object, an unprivileged user (the client process's user ID is not super-user) may not delete or rename objects in that directory unless the objects are owned by the user.

Rown
: Read permission bit for the owner of the file or directory.

Wown
: Write permission bit for the owner of the file or directory.

Xown
: Execute permission bit for the owner of a file, search permission bit for the owner of a directory.

Rgrp
: Read permission bit for the group class of the file or directory.

Wgrp
: Write permission bit for the group class of the file or directory.

Xgrp
: Execute permission bit for the group class of a file, search permission bit for the group class of a directory.

Rpub
: Public read permission bit for the file or directory.

Wpub
: Public write permission bit for the file or directory.

Xpub
: Public execute permission bit for the file, public search permission bit for the directory.

The high-order bits of the mode are not used and must be clear.

ERRORS

In order to change any of the system properties of an object, an exclusive lock is acquired for the object. Deadlocks can occur while locks are being acquired. See .SA transaction(svas) for information about deadlocks.

Changing system properties of an object occurs on the server so that the changes are immediate.

A complete list of errors is in **errors(svas).**

ENVIRONMENT

All these methods are available on the client side and the server side.

VERSION

This manual page applies to Version 1.1 of the Shore software.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518.

COPYRIGHT

Copyright © 1994, 1995, 1996, 1997, Computer Sciences Department, University of WisconsinMadison. All Rights Reserved.

SEE ALSO

sysprops(svas)