AFS Basics

Stephen Quinney <squinney@inf.ed.ac.uk>

Outline

- ACLs
- Groups
- Useful commands
- Differences from Unix
- Basic troubleshooting
- Getting more information

AFS Tools

- **fs**
  - Use to manage ACLs
  - Client cache manager admin interface
- **fsr**
  - Recursive version of **fs** command.
- **pts**
  - Use to manage groups
  - Protection Server admin interface

Finding Information

- Built-in short help for each command:
  - `pts help`
  - `fs help`
- Manual pages for full information:
  - `man command_subcommand`
  - `man pts_membership`
  - `man fs_listquota`

ACLs

Unix Permissions

- Set on both files and directories
- 3 permissions – (r)ead, (w)rite, e(x)ecute
- Access controls for user, group, others
- Limitations:
  - single group
  - new groups require admin involvement

AFS Permissions

- Only set on directories not individual files.
- Greater flexibility:
    - Many groups
    - Can allow and deny rights
    - Users can create their own groups
- New directories automatically inherit ACLs of parent.

AFS Permissions – Full List

- 7 permissions:
  - (r)ead
  - (l)ist
  - (i)nsert
  - (d)elete
  - (w)rite
  - loc(k)
  - (a)dminister

Directory Permissions

- (l)ist – list files within the directory
- (i)nsert – add new files to the directory
- (d)elete – remove files from the directory
- (a)dminister – alter ACLs for the directory

File Permissions

- (r)ead – read the contents of files
- (w)rite – alter the contents of files
  (and run chmod)
- loc(k) – use syscalls to lock files

AFS Permissions - Shorthand

- Can refer to groups of permissions:
  - read – (r)ead and (l)ist
  - write – (r)ead, (l)ist, (i)nsert, (d)elete, (w)rite, loc(k)
  - all – everything
  - none – used to remove permission from an ACL

Unix Permissions under AFS

- Unix directory permissions are ignored.
- Unix file permissions have limited effect.
- setuid, setgid and sticky bits are restricted to members of the system:administrators group.

Unix File Permissions under AFS

- Unix “group” and “other” modes are totally ignored on files.
- Read a file only if Unix owner “read” mode is set.
- Write to a file only if Unix “read” and “write” modes are set.
- Execute a file only if Unix “read” and “execute” modes are set.

Viewing ACLs

Use the **fs listacl** command:

```
% fs listacl /afs/inf.ed.ac.uk/user/j/jbloggs
Access list for /afs/inf.ed.ac.uk/user/j/jbloggs is
Normal rights:
  system:administrators rlidwka
  jbloggs rlidwka
```

Setting ACLs

- Set with user/group – permission pairs
- Use the **fs setacl** command:

```
% fs setacl -dir ~/project \
       -acl inf:sysman read jsmith write
% fsr setacl -dir ~/project \
       -acl fbloggs:team write
```

Removing an ACL

- Removing an ACL is simply done by setting the permissions to **none**.
- Again use the **fs setacl** command:

```
% fs setacl -dir ~/project \
            -acl jsmith none
```

Setting ACLs – Negative Rights

- It is possible to allow access to a group and then deny access to individual users.
- Use the **fs setacl** command with the **-negative** option:

```
% fs setacl -dir ~/project2 \
     -acl inf:sysman write
% fs setacl -dir ~/project2 \
     -acl jsmith idwk -negative
```

Copying ACLs

- If the correct ACLs are already set on another directory then just copy them.
- Use the **fs copyacl** command:

```
% fs copyacl -fromdir ~/project \
             -todir ~/project2
% fs copyacl -fromdir ~/project \
             -todir ~/project2 -clear
```

Example: Creating a public directory

```
% mkdir ~/Public
% fs setacl -dir ~/ \
          -acl system:authuser l
% fs setacl -dir ~/Public \
          -acl system:authuser read
```

Groups

Special Groups

- 'system' prefix
- Important two are:
  - system:anyuser – can access AFS
  - system:authuser – has authenticated to local AFS

Local Groups

- 'inf:' prefix - equivalent AFS group for every Unix group
- 'afs-' prefix – other locally maintained groups
- Find your own groups:
  - `pts membership jbloggs`

Viewing Groups

- Membership of a group:
  - `pts membership jbloggs:friends`
  - `pts membership inf:people`
- Detailed information for a group:
  - `pts examine jbloggs:friends`
- List your own groups:
  - `pts listowned jbloggs`

Creating a Group

- Groups are created using the **pts creategroup** command:

```
% pts creategroup jbloggs:friends
```

Adding Members to a Group

- Once a group has been created members are added using the **pts adduser** command:

```
% pts adduser -user bob carol ted alice \
              -group jbloggs:friends
```

Removing Members from a Group

- Use the **pts removeuser** command:

```
% pts removeuser -user bob \
                 -group jbloggs:friends
```

Removing a Group

- Use the **pts delete** command:

```
% pts delete jbloggs:friends
```

- If the group was in any ACLs they must be cleaned:

```
% fs cleanacl ~/project
```

Useful Commands

- Check your quota:

  -
    ```
    fs listquota ~/
    ```
- Where are files stored?

  -
    ```
    fs whereis ~/path
    ```
- Are servers responding?

  -
    ```
    fs checkservers
    ```

Differences from Unix

- `chown/chgrp` – system:administrators only
- Hard links – only within the same directory

No Special Files

- None of the following are supported:
  - Pipes
  - Fifos
  - Devices
  - Unix Sockets
- Use a suitable local filesystem instead.

Basic Troubleshooting

Can't access a file?

1) Check you are still authenticated

- `tokens`

2) Check your ACLs

- `fs listacl ~/path`

3) Check your quota

- `fs listquota ~/path`

4) Check your server

- `fs whereis ~/path`
- `fs checkservers`

Documentation

- Main page -

  http://computing.help.inf.ed.ac.uk/informatics-filesystem
- “Top Tips” -

  http://computing.help.inf.ed.ac.uk/afs-top-ten-tips
- OpenAFS User Guide -

  http://docs.openafs.org/UserGuide/