Table of Contents

Files.....1
Description of Various Filesystems.....1
Handling Your Files and Directories.....2
Using Access Control Lists for File Sharing.....3
Using 'Giveto' to Copy Files and/or Directories to Another User.....7

Files

Description of Various Filesystems

Follow the links below to find information of each filesystem available to users:

- Pleiades Home Filesystem
- Pleiades Lustre Filesystems
- Columbia Home Filesystems
- Columbia CXFS Filesystems
- Lou Home Filesystems

Handling Your Files and Directories

Follow these guidelines when handling your files and directories under the various filesystems:

- Quota Policy on Disk Space and Files
- Policy on SUID/SGID Scripts

Using Access Control Lists for File Sharing

A common way to share files and/or directories with group members or others is to use the **chmod** command to change the permissions. Yet **chmod** has limitations, so using Access Control Lsts (ACLs) may sometimes be your method of choice.

When you issue the command **chmod g+rx *filename***, for example, all the members in your group (**g**) gain read (**r**) and search/execute (**x**) access to that file, as shown below:

```
% ls -l foo
-rw------- 1 zsmith s0101 9 Jun 10 12:11 foo

% chmod g+rx foo

% ls -l foo
-rw-r-x--- 1 zsmith s0101 9 Jun 10 12:11 foo
```

However, **chmod** does not allow you to select which members of your group or which specific individuals outside of your group can access your files/directories. For this, use ACLs, which provide a mechanism for finer-grain control of file sharing. There are two ACL commands:

- **setfacl** - set file access control lists

  ```
  SYNOPSIS
         setfacl [-bkndRLPvh] [{-m|-x} acl_spec] [{-M|-X} acl_file] file ...

         setfacl --restore=file
  ```

  A detailed usage explanation of **setfacl** and its options can be found via **man setfacl**. Among the options listed:

  1. The **-m** or **-M** option lets you "modify" the ACL, where **-m** expects an ACL on the command line and **-M** expects an ACL from a file or from standard input
  2. The **-x** or **-X** option removes the ACL entries
  3. The **-R** or **--recursive** option applies operations to all files and directories recursively
  4. The **--test** option allows you to test the effect of changing the ACL without actually changing it
  5. The **-b** option removes all extended ACL entries except the base entries of the owner, group, and others
- **getfacl** - get file access control lists

  ```
  SYNOPSIS
         getfacl [-dRLPvh] file ...

         getfacl [-dRLPvh] -
  ```

A detailed usage explanation of **`getfacl`** and its options can be found via **man getfacl**.

Note that **`setfacl`** operations are supported on all Pleiades, Lou and Columbia filesystems except the Columbia home filesystems.

Before you grant another user or group access to certain files or directories, make sure that access to the parent directory (where the files or directories reside) is also allowed.

Example 1

To allow another user (jbrown) to have read/execute (**`rx`**) permission on a file (*foo*) and to view the ACL before and after an ACL change:

```
% ls -l foo
-rw------- 1 zsmith s0101 9 Jun 10 12:11 foo

% getfacl foo
# file: foo
# owner: zsmith
# group: s0101
user::rw-
group::---
other::---

% setfacl -m u:jbrown:rx foo

% getfacl foo
# file: foo
# owner: zsmith
# group: s0101
user::rw-
user:jbrown:r-x
group::---
mask::r-x
other::---

% ls -l foo
-rw-r-x---+ 1 zsmith s0101 9 Jun 10 12:11 foo
```

Example 2

To remove all extended ACLs in Example 1 except the base entries of the owner, group, and others:

```
% setfacl -b foo

% ls -l foo
-rw------- 1 zsmith s0101 9 Jun 10 12:11 foo
```

```
% getfacl foo
# file: foo
# owner: zsmith
# group: s0101
user::rw-
group::---
other::---
```

Example 3

Continuing from Example 1, to test the granting of read/execute (**rx**) access to another group (group id 24176) without actually doing it:

```
% setfacl --test -m g:24176:rx foo foo: u::rw-,u:jbrown:r-x,g::---,g:g24176:r-x,m::r-x,

% getfacl foo
# file: foo
# owner: zsmith
# group: s0101
user::rw-
user:jbrown:r-x
group::---
mask::r-x
other::---
```

Example 4

To allow another user (jbrown) recursive access to a directory (**dir.abc** which contains a file *foo2*):

```
% ls -ld dir.abc
drwx------ 2 zsmith s0101 17 Jun 10 13:19 dir.abc

% ls -l dir.abc
total 0
-rw------- 1 zsmith s0101 0 Jun 10 13:19 foo2

% setfacl -R -m u:jbrown:rx dir.abc

% getfacl dir.abc
# file: dir.abc
# owner: zsmith
# group: s0101
user::rwx
user:jbrown:r-x
group::---
mask::r-x
other::---

% getfacl dir.abc/foo2
```

```
# file: dir.abc/foo2
# owner: zsmith
# group: s0101
user::rw-
user:jbrown:r-x
group::---
mask::r-x
other::---

% ls -ld dir.abc
drwxr-x---+ 2 zsmith s0101 17 Jun 10 13:19 dir.abc

% ls -l dir.abc
total 0
-rw-r-x---+ 1 zsmith s0101 0 Jun 10 13:19 foo2
```

Example 5

Continuing from Example 4, to recursively remove all permissions user jbrown for a directory:

```
% setfacl -R -x u:jbrown dir.abc

% getfacl dir.abc
# file: dir.abc
# owner: zsmith
# group: s0101
user::rwx
group::---
mask::---
other::---

% getfacl dir.abc/foo2
# file: dir.abc/foo2
# owner: zsmith
# group: s0101
user::rw-
group::---
mask::---
other::---
```

For more information on ACLs, read **man acl**.

Using 'Giveto' to Copy Files and/or Directories to Another User

NAS's in-house developed **giveto** script is built on the use of Access Control Lists (ACL). It allows one user (the giver) to copy files and/or directories to a **/nobackup** directory of another user (the recipient).

giveto is installed under **/usr/local/bin** on Pleiades, Columbia and Lou.

In the example below, user zsmith gives a copy of his **dir.abc** directory on Pleiades to user jbrown. The steps describe the **giveto** command used by each of them, and the results.

1. User *jbrown* uses the command **giveto -i *zsmith*** to automatically (a) create an INCOMING directory (if it does not already exist) under her **/nobackup/*jbrown*** and (b) grant user *zsmith* read/write/execute permission on this directory.

   ```
   pfe20:/u/jbrown% giveto -i zsmith
   nobackup[1] =  /nobackup/jbrown

   pfe20:/u/jbrown% ls -ld /nobackup/jbrown/INCOMING
   drwxrwx---+ 2 jbrown s0202 4096 Jun 14 12:18 /nobackup/jbrown/INCOMING
   ```
2. User *zsmith* uses the command **giveto *jbrown* dir.abc** to automatically (a) create a subdirectory called ***zsmith*_0** under *jbrown*'s INCOMING directory, (b) copy **dir.abc** to **/nobackup/*jbrown*/INCOMING/*zsmith*_0**, (c) grant user jbrown read/write/execute permission on **/nobackup/*jbrown*/INCOMING/*zsmith*_0**, and (d) send an email to user *jbrown* regarding the copy.

   ```
   pfe20:/home1/zsmith> ls -ld dir.abc
   drwx------ 2 zsmith s0101 17 Jun 14 12:21 dir.abc/

   pfe20:/home1/zsmith> giveto jbrown dir.abc
   setfacl -m u:jbrown:rwx zsmith_0
   setfacl -m u:jbrown:rwx zsmith_0/giveto.log
   setfacl -m u:jbrown:rwx zsmith_0/dir.abc
   setfacl -m u:jbrown:rwx zsmith_0/dir.abc/foo2
   path =  /nobackup/jbrown/INCOMING/zsmith_0
   total 12
   drwxrwx---+ 3 zsmith  s0101 4096 Jun 14 12:29 .
   drwxrwx---+ 3 jbrown  s0202 4096 Jun 14 12:29 ..
   drwxrwx---+ 2 zsmith  s0101 4096 Jun 14 12:21 dir.abc
   -rw-rwx---+ 1 zsmith  s0101   44 Jun 14 12:29 giveto.log
   ```

   Note: If the directory ***zsmith*_0** already exists prior to this step, ***zsmith*_1** would be used instead.
3. User *jbrown* receives an email from user *zsmith* with a subject line "giveto files". They see that the directory **dir.abc** has been copied successfully. Even though the directory **/nobackup/*jbrown*/INCOMING/*zsmith*_0** is still owned by user *zsmith*,

user *jbrown* now has permission to read/write/execute files and directories under **/nobackup/*jbrown*/INCOMING/*zsmith*_0.**

```
pfe20:/u/jbrown% ls -lrt /nobackup/jbrown/INCOMING
total 4
drwxrwx---+ 3 zsmith s0101 4096 Jun 14 12:29 zsmith_0

pfe20:/u/jbrown%ls -lrt /nobackup/jbrown/INCOMING/zsmith_0
total 4
drwxrwx---+ 2 zsmith s0101 4096 Jun 14 12:21 dir.abc

pfe20:/u/jbrown%ls -lrt /nobackup/jbrown/INCOMING/zsmith_0/dir.abc
total 4
-rw-rwx---+ 1 zsmith s0101 8 Jun 14 12:21 foo2

pfe20:/u/jbrown%getfacl /nobackup/jbrown/INCOMING/zsmith_0/dir.abc
# file: /nobackup/jbrown/INCOMING/zsmith_0/dir.abc
# owner: zsmith
# group: s0101
user::rwx
user:jbrown:rwx
group::---
mask::rwx
other::---
```

Read **man giveto** for more information.

The **giveto** script was created by NAS staff member Arthur Lazanoff.