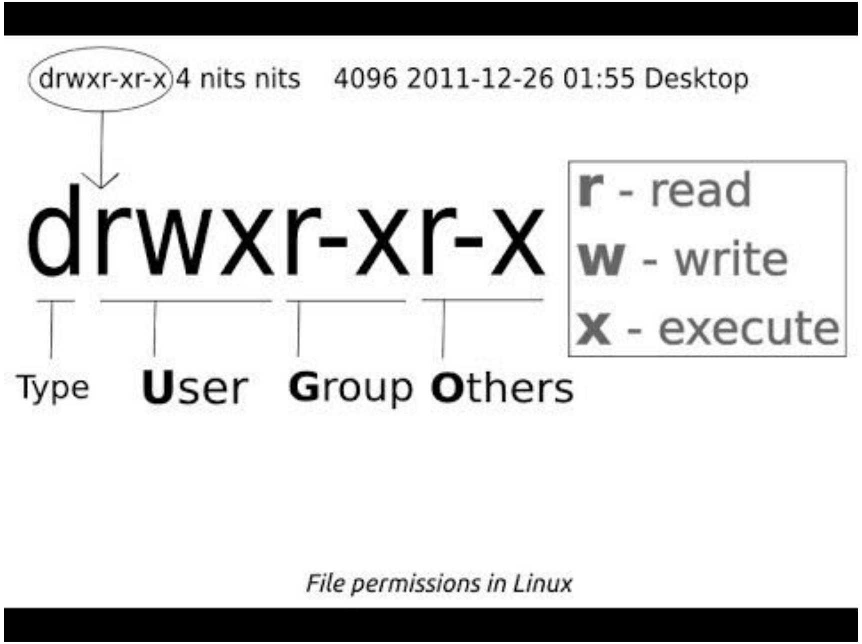
ACL - Access Control List

Optimize way to manage file permissions

Source: devconnected.com

Asst. Prof. Ashwini Mathur

Basic File permissions



```
drwxr-xr-x 4 nits nits    4096 2011-12-26 01:55 Desktop
```

File permissions in Linux

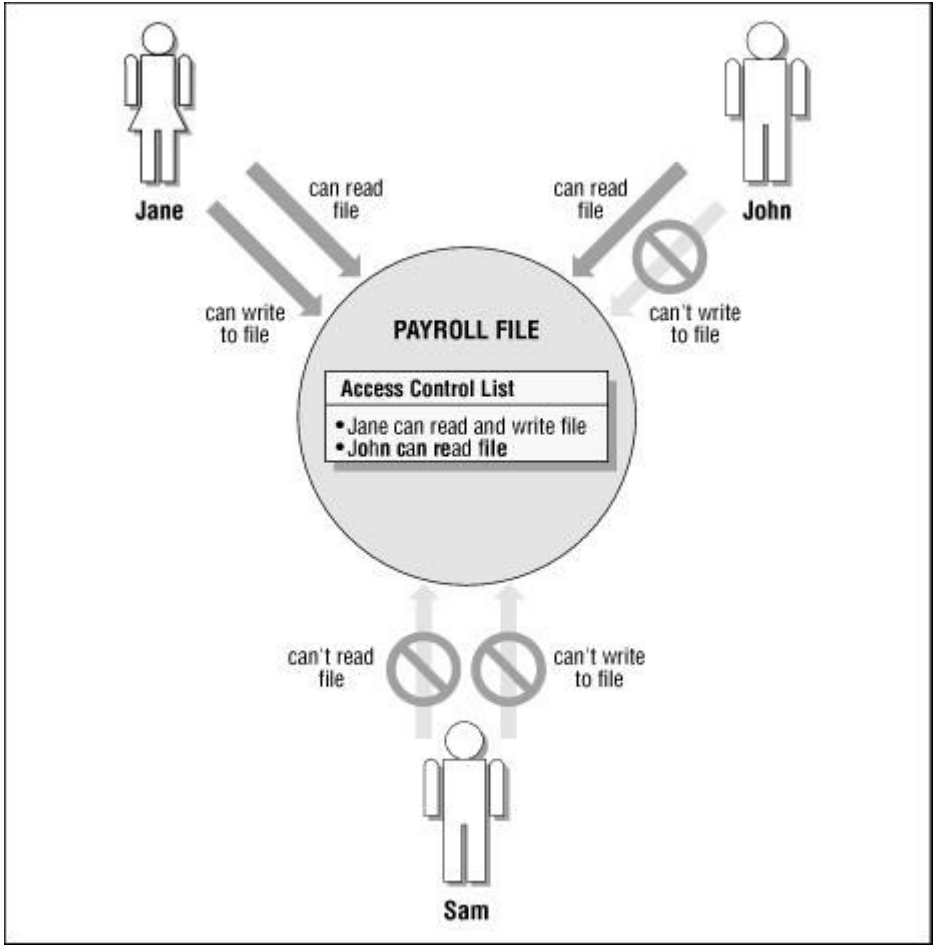
Overview

As a system administrator, you are probably spending quite some time **configuring permissions for user and groups on your system.** File permissions are already quite handy in order to give read, write or execute permissions to directories or files.

But what if we need a more precise way to give permissions to folders or files?

What if I want to give access to a file to a specific user or a specific group, that is not the current owner of the file?

This is exactly what **access control lists, also shortened ACL**, are solving on a Linux system


Jane
can read file
can write to file
John
can read file
can't write to file
PAYROLL FILE
Access Control List
• Jane can read and write file
• John can read file
can't read file
can't write to file
Sam

Objectives

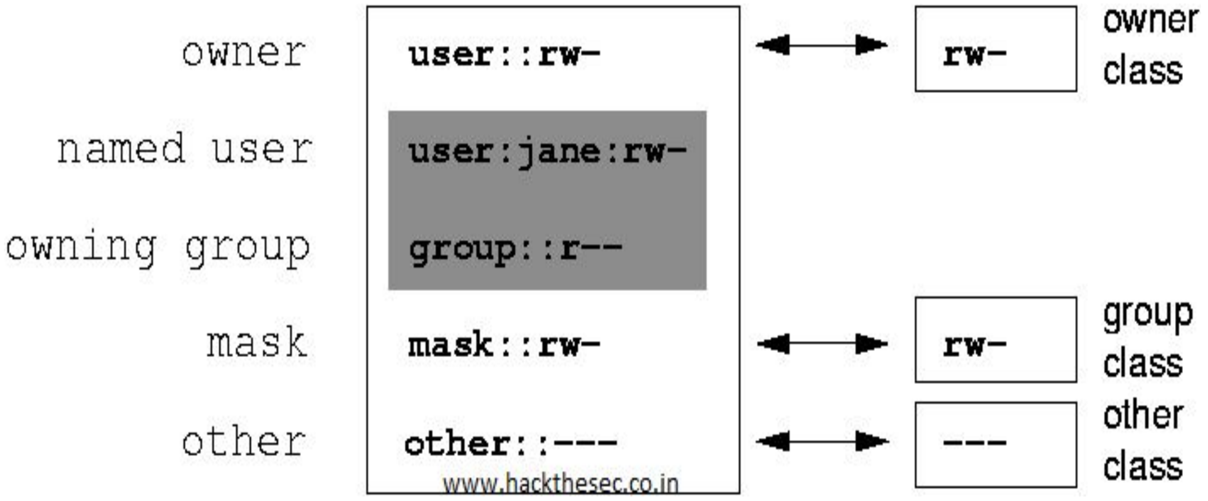
- What **access control lists** are and how they can be read from the ls command;
- How to set basic permissions on a file using the **setfacl** command:
- How to read access control lists using the **getfacl** command;
- What is the **access control list mask** and how it should be read;
- What **access control list defaults** are and how they can be used effectively

Access Control Lists Basics on Linux

On Linux, there are two ways of setting permissions for users and groups : with regular file permissions or with **access control lists.**

Access Control List Definition

Access control lists are used on Linux filesystems to set custom and more personalized permissions on files and folders. ACLs allow file owners or privileged users to grant rights to specific users or to specific groups.



Access Control List using ls

1 rw-rw-r-- file

The permission does not contain a "+" character, no ACLs are defined for the file

2 rw-rw-r--+ file

The permission contain a "+" character, ACLs are defined for this file

Setting access control lists using setfacl

With access control lists, there are two main commands that you need to remember : **setfacl and getfacl.**

In this chapter, we are going to take a look at the **setfacl** command as the getfacl one is pretty self explanatory.

The setfacl command is used on Linux to create, modify and remove access control lists on a file or directory.

The setfacl has the following syntax

```
$ setfacl {-m, -x}  {u, g}:<name>:[r, w, x] <file, directory>
```

Where curly brackets mean one of the following options and regular brackets mean one or several items.

- **-m** : means that you want to **modify** one or several ACL entries on the file or directory.
- **-x** : means that you want to **remove** one or several ACL entries on a file or directory.
- **{u, g}** : if you want to modify the ACL for a user or for a group.
- **name** : this is an optional parameter, it can be omitted if you want to set the ACL entries for every user or for every group on your host.
- **[r, w, x]** : in order to set read, write or execute permissions on the file or directory.

The getfacl command is divided into multiple categories :

- **Filename, owner and group** : the information about user and group ownership is shown at the top;
- User permissions : first, you would find regular user permissions, also called the owning user, followed by any user-specific ACL entries (called named users)
- **Group permissions** : owning groups are presented followed by group-specific ACL entries, also called named groups
- **Mask** : that restricts the permissions given to ACL entries, the mask is going to be detailed in the next section;
- **Other permissions** : those permissions are always active and this is the last category explored when no other permissions match with the current user or group.

Creating access control lists defaults on directories

As already mentioned in this article, it is possible to create ACL entries on directories and they work in the same way file access control lists work.

However, there is a small difference when it comes to directories : **you have to option to create access control lists defaults.**

Access control lists defaults are used to create ACL entries on a directory that will be inherited by objects in this directory like files or subdirectories.

When creating default ACL entries :

- Files created in this directory **inherit** the ACL entries specified in the parent directory
- Subdirectories created in this directory inherit the ACL entries **as well as the default ACL entries** from the parent directory.

To create default ACL entries, specify the -d option when setting ACL using the setfacl command.

```
$ setfacl -d -m {u, g}:<name>:[r, w, x] <directory>
```

For example, to assign read permissions to all files created in a directory, you would run the following command

```
$ setfacl -d -m u::r directory
```

Linux Access Control Lists (ACLs) - Practice Demonstration

Sharing One Directory with Specific Users and Group - ACL

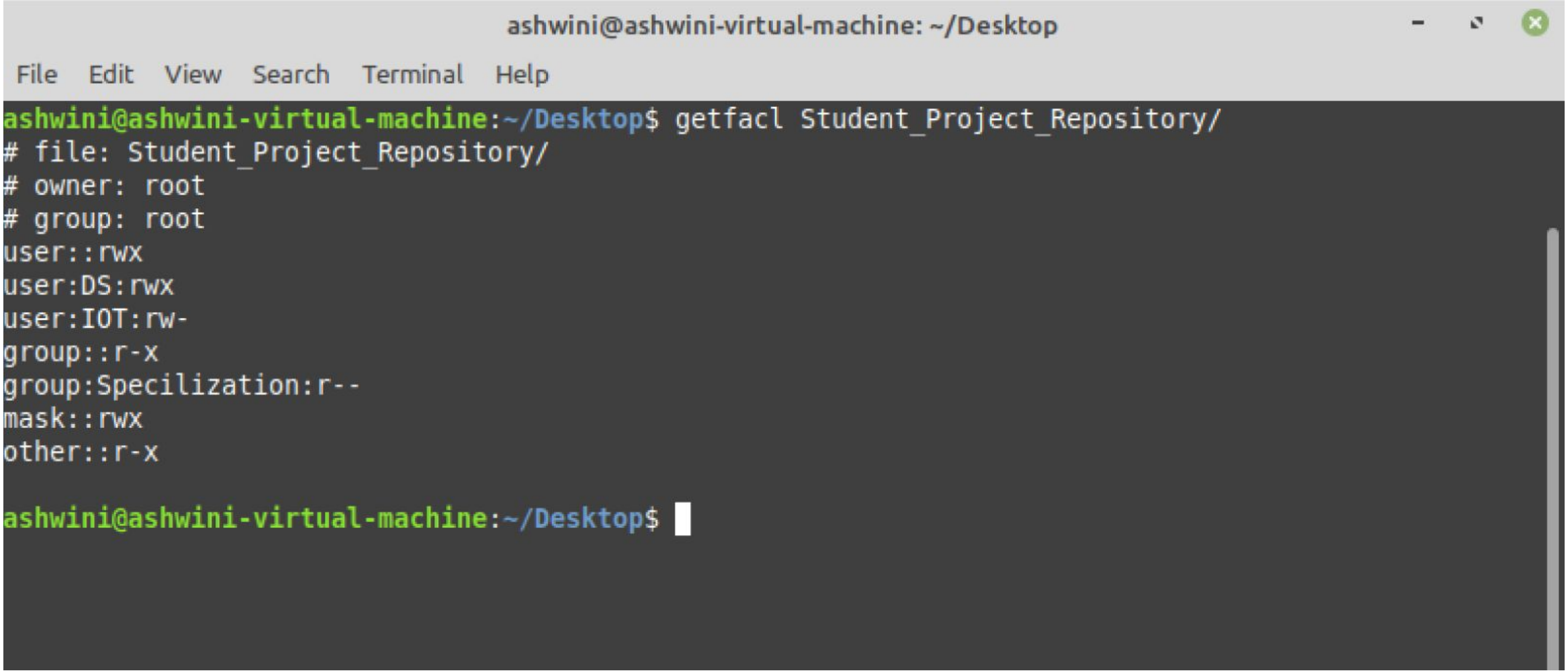
Following ACL Tasks:

Create a Directory in root user- **[Students_Project_Repository]**

Distribute across Specific User and Group with Different Permission

- **Create 2 New users “DS” and “IOT”**
- **Create 1 New Group “Specialization”.**

1. **Assigned the Student_Project_Repository to DS with rwx permission and IOT with rw- permission only**.
2. Assigned the Student_Project_Repository to Specialization Group with r-- permission only.
3. Also check and verify the following Permissions via linux command.



ashwini@ashwini-virtual-machine: ~/Desktop
File Edit View Search Terminal Help

```
ashwini@ashwini-virtual-machine:~/Desktop$ getfacl Student_Project_Repository/
# file: Student_Project_Repository/
# owner: root
# group: root
user::rwx
user:DS:rwx
user:IOT:rw-
group::r-x
group:Specilization:r--
mask::rwx
other::r-x

ashwini@ashwini-virtual-machine:~/Desktop$ 
```

Additional Assignment

Assign the given following specific permission for accessing the department directories to each individual
Via Linux ACL utility tool

Take a Case Study :

Linux sysadmin basics: User account management

System Administrator

- Devesh

Accounting Department : (Group)

1. Ashish - Regular Employee (Head) - Access all the four directories **A1, A2, A3, A4 and F1** with **rwx**
2. Deepti - Regular Employee - Access the directories **A1, A2, A3** with **rw-**
3. Shreyas - Intern **Access A4 and F4 Directories** with **r - -**

Finance Department :(Group)

4. Akash - Regular Employee Access the directories **F1, F2, F3** with **rw-**
5. Navdeep - Regular Employee (Head) - Access all the four directories **F1, F2, F3, F4 and A1** with **rwx**
6. Shristi - Intern **Access A4 and F4 Directories** with **r - -**

Each Department have 4 Important Directories (Workspace folders). (Create at root level)

For Accounting - A1,A2,A3,A4
and Finance - F1,F2,F3,F4