Files, Users, and Permissions

Suppose you have a file and obtain a detailed listing of its properties:

```
$ ls -lF file

-rwxr-x--x 1 coop coop 42 Jun 18 13:59 some_file*
```

After the initial dash, there are nine letters, in three groups of three, that indicate read, write and execute permissions for owner, group, and world. In the above example, the owner of the file can read, write and execute, all members of the group can read or execute, and all others (in the world) can only execute.

These file access permissions are a critical part of the Linux security system. Any request to access a file requires comparison of the credentials and identity of the requesting user to those of the owner of the file.

This authorization is granted depending on one of these three sets of permissions, in the following order:

- If the requester is the file owner, the file owner permissions are used.
- Otherwise, if the requester is in the group that owns the files, the group permissions are examined.
- If that does not succeed, the world permissions are examined.

Note that permissions can be changed with **chmod** and ownership with **chown**.

One user is special; the superuser or root user, who has access to all files on the system. This is essentially the equivalent to the administrator account, or privilege, in other operating systems.

Linux contains a full implementation of POSIX ACLs (Access Control Lists) which extends the simpler user, group, world and read, write, execute model.

Particular privileges can be granted to specific users or groups of users when accessing certain objects or classes of objects.

While the Linux kernel enables the use of ACLs, it still must be implemented as well in the particular filesystem. All major filesystems used in modern Linux distributions incorporate the ACL extensions.