2. Student Handout

Linux Users, Groups, and File Permissions: Student Handout

---

1. Linux Users and Groups

Users in Linux

- **Root User**: The most powerful user with unrestricted access.
- **Regular Users**: Users with limited permissions.

Examples:

1. **Creating a User**: Use `sudo adduser username` to create a new user.
2. **Listing Users**: View all users with `cat /etc/passwd`.
3. **Switching Users**: Use `su - username` to switch to another user.

Groups in Linux

- **Primary Group**: Default group assigned to a user.
- **Secondary Group**: Additional groups a user can belong to.

Examples:

1. **Creating a Group**: Use `sudo groupadd groupname` to create a new group.
2. **Adding a User to a Group**: Use `sudo usermod -aG groupname username`.
3. **Listing Groups**: View all groups with `cat /etc/group`.

---

2. File Ownership in Linux

- **User Ownership**: The user who created the file.
- **Group Ownership**: The group associated with the file.

- **Other**: All other users not in the owner or group.

Examples:

1. **Checking File Ownership**: Use `ls -l filename` to view file ownership.
2. **Changing File Ownership**: Use `sudo chown newuser filename`.
3. **Changing Group Ownership**: Use `sudo chown :newgroup filename`.

---

3. Basics of File Permissions: Read, Write, and Execute

- **Read (r)**: View file contents.
- **Write (w)**: Modify or delete file contents.
- **Execute (x)**: Run the file as a program.

Permission Categories

- **Owner (User)**: Permissions for the file owner.
- **Group**: Permissions for the group.
- **Others**: Permissions for all other users.

Examples:

1. **Viewing Permissions**: Use `ls -l` to see file permissions.
2. **Setting Permissions**: Use `chmod u+rwx filename` to set owner permissions.
3. **Removing Permissions**: Use `chmod g-w filename` to remove write permission for the group.

---

4. Changing File Permissions

Using `chmod`

- **Symbolic Notation**: Use letters to modify permissions.
- **Numeric Notation**: Use numbers to set permissions.

Examples:

1. **Adding Execute Permission**: Use `chmod u+x filename`.
2. **Setting Permissions with Numbers**: Use `chmod 755 filename` for full owner permissions and read/execute for others.
3. **Removing Read Permission**: Use `chmod o-r filename` to remove read permission for others.

---

5. Changing File Ownership

Using `chown`

- Change both user and group ownership.

Examples:

1. **Changing User Ownership**: Use `sudo chown newuser filename`.
2. **Changing Group Ownership**: Use `sudo chown :newgroup filename`.
3. **Changing Both User and Group**: Use `sudo chown newuser:newgroup filename`.

Conclusion

- **Users**: Individuals with system access.
- **Groups**: Collections of users for managing permissions.
- **File Ownership**: Divided into user, group, and others.
- **Permissions**: Determine actions on files (read, write, execute).
- **Commands**: Use `chmod` for permissions and `chown` for ownership changes.

By mastering these concepts, you can effectively manage access to files and directories on a Linux system.