3. Student Activity

Student Activity: Practicing Linux Users, Groups, and File Permissions

---

Objective

The goal of this activity is to help you practice and reinforce your understanding of Linux users, groups, and file permissions. By the end of this activity, you should be comfortable with creating users and groups, assigning file ownership, and modifying file permissions.

---

1. Linux Users and Groups

Creating and Managing Users

1. **Create a New User**
   - Command: `sudo adduser user1`
   - This command creates a new user named `user1`.
2. **Check User Information**
   - Command: `id user1`
   - This command displays the user ID (UID), primary group, and any secondary groups for `user1`.
3. **Delete a User**
   - Command: `sudo deluser user1`
   - This command removes the user `user1` from the system.

Creating and Managing Groups

1. **Create a New Group**
   - Command: `sudo addgroup group1`
   - This command creates a new group named `group1`.
2. **Add a User to a Group**
   - Command: `sudo usermod -aG group1 user1`

    - This command adds `user1` to `group1` as a secondary group.
3. **Remove a User from a Group**
    - Command: `sudo deluser user1 group1`
    - This command removes `user1` from `group1`.

---

2. File Ownership in Linux

Checking and Changing File Ownership

1. **Check File Ownership**
    - Command: `ls -l File.txt`
    - This command lists the file `File.txt` with its current owner and group.
2. **Change File Owner**
    - Command: `sudo chown user2 File.txt`
    - This command changes the owner of `File.txt` to `user2`.
3. **Change File Group**
    - Command: `sudo chown :group2 File.txt`
    - This command changes the group of `File.txt` to `group2`.

---

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

Viewing and Modifying File Permissions

1. **View File Permissions**
    - Command: `ls -l File.txt`
    - This command shows the current permissions for `File.txt`.
2. **Add Write Permission for Group**
    - Command: `chmod g+w File.txt`
    - This command adds write permission for the group on `File.txt`.
3. **Remove Execute Permission for Others**
    - Command: `chmod o-x File.txt`
    - This command removes execute permission for others on `File.txt`.

---

4. Changing File Permissions

Using `chmod` with Symbolic and Numeric Notation

1. **Change Permissions Using Symbolic Notation**
   - Command: `chmod u+x File.txt`
   - This command adds execute permission for the owner of `File.txt`.
2. **Change Permissions Using Numeric Notation**
   - Command: `chmod 755 File.txt`
   - This command sets the permissions to read, write, and execute for the owner, and read and execute for the group and others.
3. **Set Permissions to Read-Only for Everyone**
   - Command: `chmod 444 File.txt`
   - This command sets the permissions to read-only for the owner, group, and others.

---

5. Changing File Ownership

Using `chown` to Change Ownership

1. **Change Both User and Group Ownership**
   - Command: `sudo chown user2:group2 File.txt`
   - This command changes both the owner to `user2` and the group to `group2` for `File.txt`.
2. **Change Only User Ownership**
   - Command: `sudo chown user3 File.txt`
   - This command changes the owner of `File.txt` to `user3`.
3. **Change Only Group Ownership**
   - Command: `sudo chown :group3 File.txt`
   - This command changes the group of `File.txt` to `group3`.

Conclusion

By completing these exercises, you should now have a practical understanding of how to manage users, groups, and file permissions in Linux. Practice these commands regularly to become proficient in managing Linux systems. If you encounter any issues or have questions, feel free to ask for help!