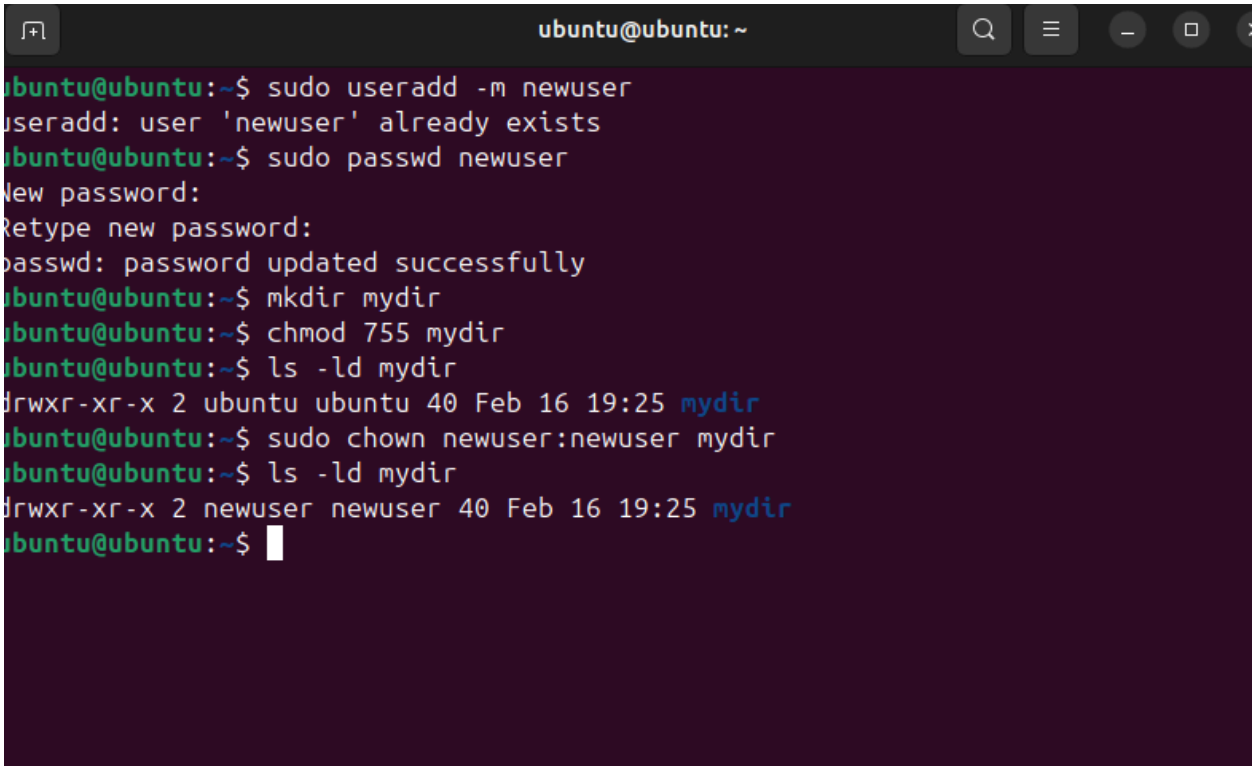
Lucy Njuguna

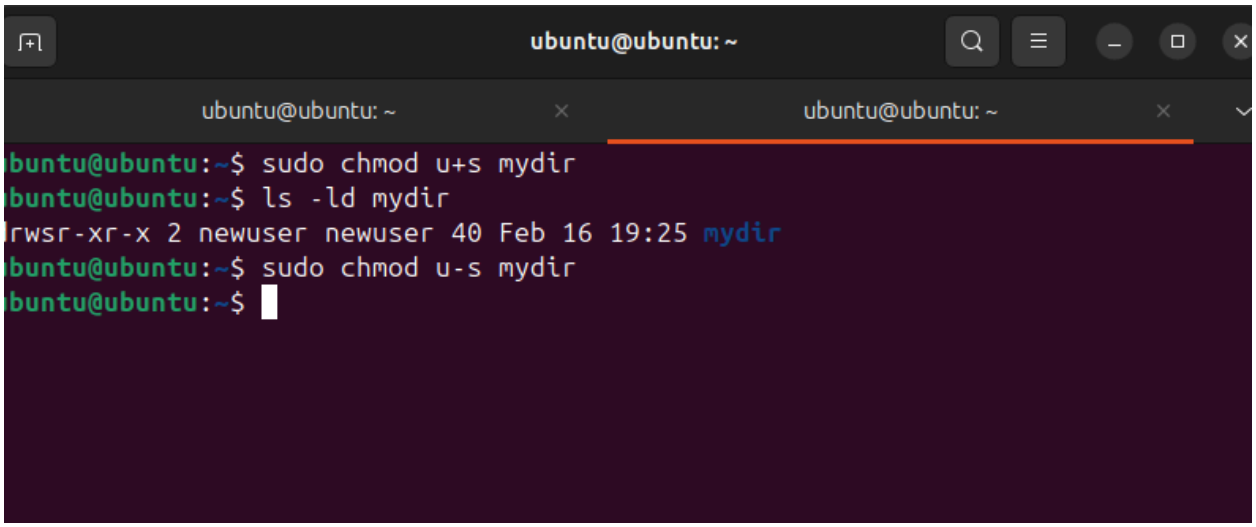
Access Control in Linux

1. **User and Group Permissions:**

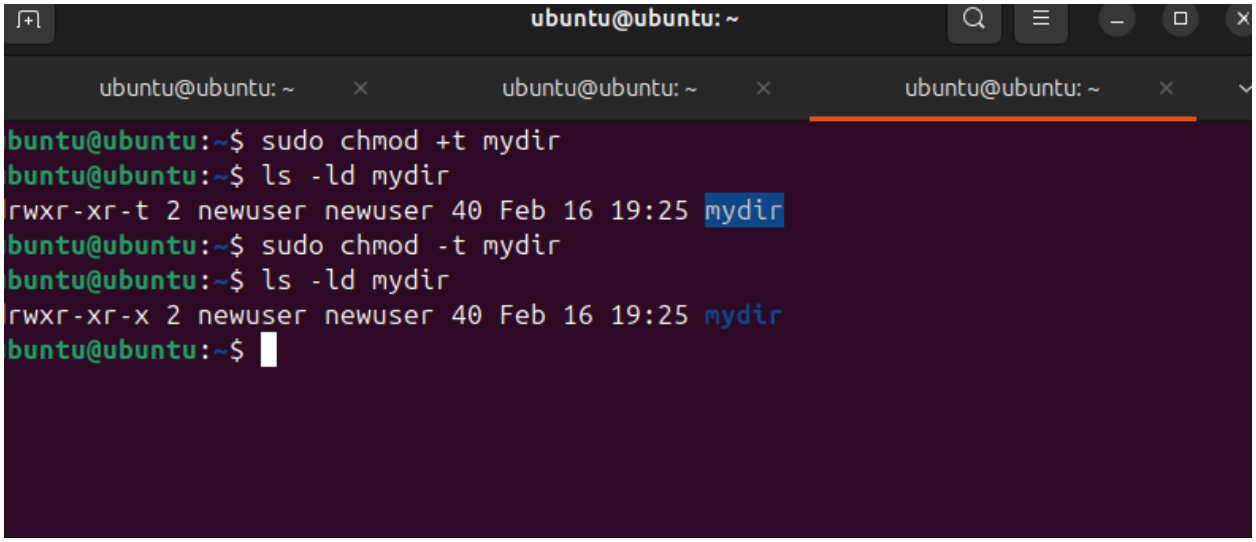


2. File Attributes:

Set and unset the setuid



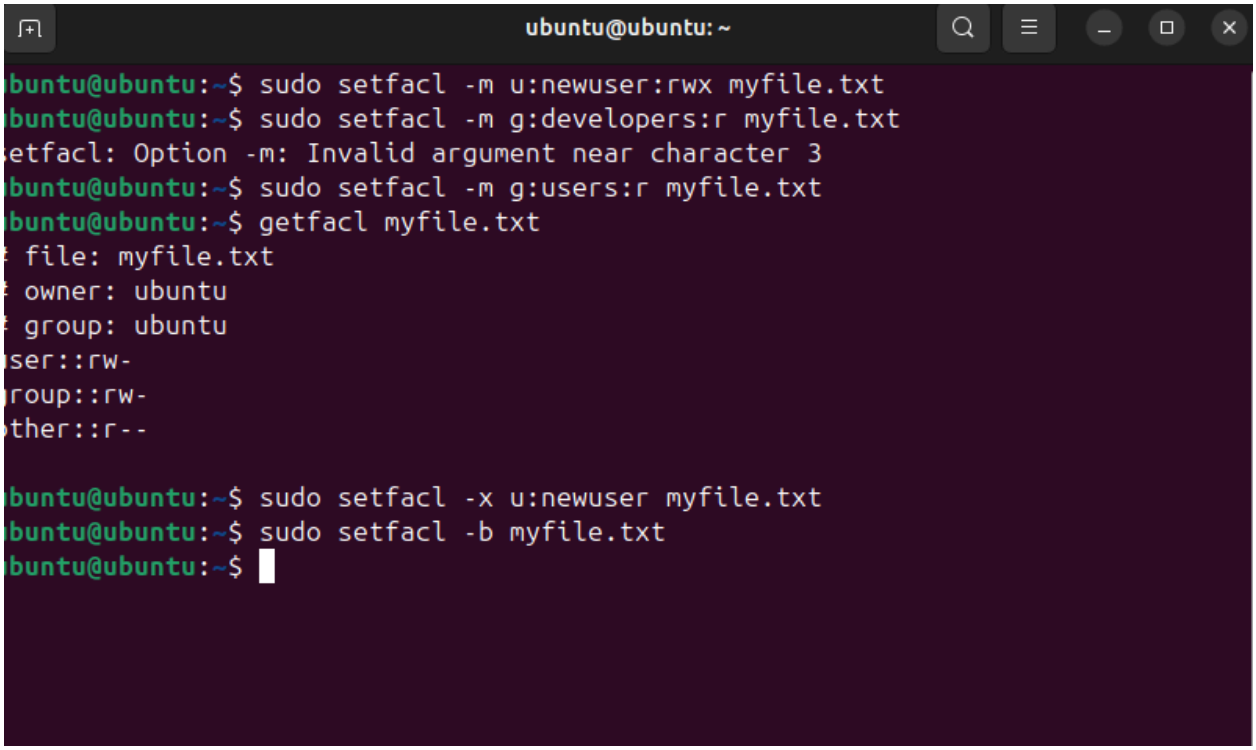
setgid, and sticky bit



3. Access Control Lists (ACLs):

- **Create a new file and assign a basic ACL to it.**

Modify the ACL to grant or revoke specific permissions for a user or group.



Investigation Report: Access Control in Linux

Introduction

Access control is an important aspect of Linux security, because it ensures that files and directories are protected from unauthorized access and modifications. Properly managing permissions helps maintain system integrity by allowing only authorized users to read, write, or execute files. This report explores key access control methods, including standard file permissions, special attributes, and Access Control Lists (ACLs).

User and Group Permissions

Linux file permissions are based on three categories: owner, group, and others, each with three permission types: read (r), write (w), and execute (x). The chmod command is used to modify these permissions, while chown changes file ownership. In the case above, the file owner has full access, while the group and others can only read and execute it. Improper permission settings can either restrict necessary access or expose files to unauthorized users; hence, the correct balance should be met.

File Attributes: setuid, setgid, and Sticky Bit

Beyond standard permissions, Linux provides additional file attributes to control execution and access:

- **setuid (chmod u+s)**: Ensures that a file executes with the permissions of its owner rather than the user running it. This is commonly used in system utilities like passwd.
- **setgid (chmod g+s)**: Ensures files created in a directory inherit the directory's group ownership, making it useful for shared project directories.
- **Sticky Bit (chmod +t)**: Prevents users from deleting files owned by others in shared directories, such as /tmp.

Access Control Lists (ACLs)

ACLs provide a more flexible permission model by allowing specific users or groups to have customized access beyond traditional Linux permissions. The setfacl command assigns ACLs, while getfacl displays current ACL settings.

Example: Granting read and write permissions to newuser in the case above the following command was used: **setfacl -m u:newuser:rw myfile.txt**

ACLs are mostly useful in environments where multiple users require different levels of access to the same files or directories.

Conclusion

Effective access control is essential for maintaining security and organization in Linux systems. Standard file permissions, special attributes, and ACLs work together to provide a layered approach to access management. Understanding and implementation of these methods ensures that data remains protected while allowing necessary access for users and groups.