1. Useradd command is used to add new users to the system, options: -m (create the user’s home directory), -s shell (set the user’s login shell), -g (set the use’s primary group), -G groups (set additional groups for the user), -p paasword (set the user’s password), -c comment (add a comment for the user), -u UID (set the user’s UID, -e YYYY-MM-DD (set an expiration date for the user account). To create a user: sudo useradd user-name
2. To create a user with a specific home directory and login shell: sudo useradd -m -s /bin/bash user-name; create a user and assign a specific group: sudo useradd -g Develop mike; create a user and assign multiple groups: sudo useradd -G Develop,staff anna; create a user with a password: sudo useradd -m -p $(openssl passwd -1 mypassword) alex
3. The userdel command is used to remove existing user accounts from the system. Option: -r (remove the user’s home directory and mail spool); -f (forcefully remove the user account even if the user is still logged in or has running processes). To delete the user sudo userdel john
4. Delete user and remove the home directory and mail spool: sudo userdel -r sarah
5. To add supplementary group to a user: sudo usermod -G group1, group2 username; to appends groups to the user’s existing group memberships without removing an existing groups : sudo usrmod -aG -group3, group4 username
6. To change the username of an existing user: sudo usermod -l newname oldname
7. To display password status information for a user: sudo passwd -S username
8. To change the home directory of a user: sudo usermod -d /new/home/directory username
9. To lock the user account (disable login): sudo passwd -l username; to unclock a user account (enable login): sudo passwd -u username
10. To unclock a previously locker user account: sudo usermod -Uusername
11. To change the description associated with a user account: sudoo usermod -c “new comment” username
12. To lock the user with usermod command: sudo usermod -L username
13. To force the user to change their password at the next login (expire a user’s password): sudo passwd -e username
14. To remove the user account while preserving the home directory and file: sudo userdel -r username
15. The id command provide details such as the user’s or group’s UIID, GID, and supplementary group memberships.
16. To display information about the current user: id; to the specific use: id username
17. The groups command is used to display the groups a user belongs to. To display the group memberships of the current user: groups; to the specific user: groups username
18. The passwd command is used to change the password of a user account, to set or update their own password: passwd
19. To enforce password expiration, I can set the maximum password age for user, the user will require to change their password after the specific time: sudo passwd –maxdays <days in number> username
20. The sudo command provides the delegation administrative tasks to regular users while maintaining security and control. To enable the users to perform privileged operation I need to modify the sudoers file.
21. The difference between using sudo command and logging in as the root user directly are that using sudo command provides a more secure and controlled approach to perform privileged operations, allowing administrators to delegate specific administrative tasks to trusted users. Logging in as the root user directly should be reserved for exceptional cases when extensive system-wide changes.
22. To grant or revoke the sudo access to a user: logging in as the root user or switch to root user using su -; use the deluser command with the sudo option to remove the user from the sudo group: deluser username sudo
23. To use the visudo command, run sudo visudo, the sudoers file will open, make changes to the file according to the sudoers syntax.
24. Some best security practices to use the sudo command: limit sudo access; use strong passwords; employ the principle of least privilege; use visudo to edit the sudoers file; regularly review sudoers file; implement command logging and auditing; configure sudo session timeouts; enable secure tty requirement; regularly update and patch the system; educate users
25. The use of sudo -I or sudo su andenter the password a new session will be started with the environment variables and settings of the root user.
26. It is a good idea to grant specific user sudo privilege because is perform multiple administrative tasks.
27. To safety grant the sudo privilege
28. The three basic permissions that can be assigned to file and directory are read, write and execute
29. – the read permission allows a user to view the contents of a file or directory; - the write permission allows a user to modify or delete a file or directory; - the execute permission allows a user to execute a file or access the contents of a directory.
30. In the context of file and directory permissions, the owner refers to the user account that created the file or directory; the group refers to a collection of user accounts that share common permissions; the others or world refers to all users who are not the owner or part of the group associated with the file or directory.
31. The chmod command is used to change the permissions of files and directories and allows you to modify the access rights for the owner, group, and others. Option: -c displays a message for each file processed, -R applies the permissions recursively to files and directories within a directory. Example chmod u=rwx,g=rx,o=r file.txt
32. When using the chmod command to change permissions, there are two different modes you can use:

* Symbolic mode: chmod options permissions file/directory
  + Who : u=user/owner ; g=group; o=others (everyone else); a=all (equivalent to ugo)
  + Operator: + (add the specified permissions); - (remove the specified permissions; = (set the specified permissions and remove all others)
  + Permissions: r= read permission; w= write permission; x= execute permission
* Octal mode: read=4, write=2, execute=1, no permissions=0

1. The chown command is used to change the ownership of files and directories, specifically the user owner and group owner. Example chown newuser:newgroup file.txt
2. The difference between chown and chmod command are:

* Ownership modification
  + Chown: you can change the owner and/or group owner of a file or directory
  + Chmod: chmod doesn’t modify ownership directly. It is used to modify the permissions of files and directories for different categories of users
* Permission modification
  + Chown: chown doesn’t modify permission directly. It solely focuses on changing ownership.
  + Chmod: chmod is specifically designed to modify the permissions of files and directories. It allows you to grant or revoke read, write, and execute permisions for the owner, group, and others.

1. The meaning of the file permission symbol displayed by the ls -l command, such as rw-r—r— is: the owner can read and write file; the group can read only the file; and others can read only the file.
2. The difference between rwx notation and numeric notation when representing permission are:

* Rwx notation (symbolic notation)
  + Each permission is represented by its corresponding letter, and the absence of a permission is denoted by the hypen (-)
  + The permissions for the owner, group, and others are represented separately, typically in a sequence of three characters
* Numeric notation (octal notation)
  + Each permission is assigned a numeric value: 4 for read, 2 for write, and 1 for execute.
  + The sum of the numeric values represents the overall permission
  + The permission for the owner, group, and others are represented as a three-digit umber, with each digit representing the sum of the corresponding permissions

1. To change recursively the permissions for all files and directories within a directory using the chmod command, I can use the -R option: chmod -R permissions directory. Example chmod -R u=rwx,g=rw,o= my-directory
2. To change recursively the ownership for all files and directories within a directory using chown command, I can use the -R option: chown -R newuser:newgroup directory/ or chown -R :newgroup directory/
3. – sticky bit (tor T): is commonly used on directories that are shared by multiple users, such as /tmp, to ensure that users can only delete or modify their own files preventing accidental of other user’ files;

- setuid or ser user ID (s or S): is used on executable files tht need elevated privileges to perform certain tasks that regular users can’t execute directly;

- setgid or set group ID: is used on directories to ensure that files created within that directory inherit the group ownership of the parent directory. This is useful when multiple users nee to collaborate on files within a shared directory, allowing them to have the same group ownership for easy file sharing and collaboration.