

Linux Administration

Azza Khalel

khalelazza@gmail.com

[Azza Khalel | LinkedIn](#)

Day2 contents

- User and group administration.
- Permissions.
- Switching to other accounts.
- User and group ownership
- sudo

Listing directory content

- `-rwxr-xr-x` 1 root root 20 512 May 21 16:06 file1
- `drwxr-xr-x` 2 fatma fatma 20 512 May 21 16:06 dir2

type **Permission** nuOfHL**Owner** **Group** Size Creation_time Name_of_file

User accounts

- Root user (super).
- Normal user.
- Service user.

Users passwd file

- The /etc/passwd file

login-name:x:uid:gid:comment:home-directory:login-shell

- Included fields are:
 - Login name.
 - Encrypted password.
 - User Id (uid).
 - Group Id (gid).
 - Comment about the user.
 - Home Directory.
 - Login shell.

```
root:x:0:0:root:/root:/bin/bash
```

```
1 2 3 4 5 6 7
```

```
1.root: username
```

```
2.x: password (saved in /etc/shadow in encrypted form)
```

```
3.0: UID (0 is for root)
```

```
4.0: GID (0 is for root)
```

```
5.root: comments
```

```
6./root: Home directory
```

```
7./bin/bash: Login Shell
```

Users shadow file

- The /etc/shadow file

username:encrypted passwd:last_changed:min:max:warn:inactive:expire:future-use

- Included fields are
 - Login name.
 - Encrypted password.
 - Days since Jan 1, 1970 that password was last changed.
 - Days before password may not be changed.
 - Days after which password must be changed.
 - Days before password is to expire that user is warned.
 - Days after password expires that account is disabled.
 - Days since Jan 1, 1970 that account is disabled.

groups

- The /etc/group file

groupname:x:gid: members_who_this_group_is_secondary_for_them

- The /etc/gshadow

groupname:encrypted_pass:group_admin:members_who_this_group_is_secondary_for_them

Creating new user

- **#useradd [options] username**
 - The useradd command populates user home directories from the /etc/skel directory.
 - #useradd -c → to make comment
 - #useradd -md → to put home dir
 - #useradd -s → to put login shell
 - #useradd -g → to set primary group
 - #useradd -G → to set secondary groups
 - #useradd -u → to set uid
 - #useradd -f <days> <user> → to set when the user will be inactive (will be locked)
 - #useradd -e <days> <user> → to set when the user will be expired (can't login again)
 - #useradd -p <encrypted_pass> → to give password to user during creation
- **#passwd username**
 - #passwd -d <username> → first login without password then must create pass for himself
- **View and modify default setting of user and password**
 - #useradd -D → read useradd defaults from /etc/default/useradd
 - Default password settings are from /etc/login.defs
- **Adding multiple user accounts**
 - #vi file
 - User-name1:password:uid:gid:comment:homedir:loginshell
 - User-name2:password:uid:gid:comment:homedir:loginshell
 -etc
 - #newusers filename

Modifying user accounts

- To change a user's account information, you can:
 - Edit the `/etc/passwd` or `/etc/shadow` files manually. (not recommended)
 - Use the **usermod** or **chage** commands.
- **usermod**
 - The **usermod** command can be used to set all properties of users as stored in `/etc/passwd` and `/etc/shadow`, plus some additional tasks, such as managing group membership.
 - `#usermod -l` → to change login name of a created user
 - `#usermod -c` → to change the comment of the user
 - `#usermod -g` → to change the primary group of the user
 - `#usermod -aG` → to add user in a secondary group (a to append to current groups)
 - `#usermod -s` → to change the default shell
 - `#usermod -md` → to change the home dir of the user
 - `#usermod -u` → change userid
 - `#usermod -U` → to unlock this user
 - `#usermod -L` → to lock this user

Password aging policies

- Change password age
 - `#chage -m` → to change the min number of days between password changes
 - `#chage -M` → to change the max number of days between password changes
 - `#chage -W` → to change the number of days to start warning before a password change will be required
 - `#chage -l` → to change inactive time (if the user don't access the system for this period, it will be automatically locked)
 - `#chage -E` → to change the expiration date for the account (calculated from 1/1/1970) if it set to -1 means will never expire
 - `#chage -E 2026-12-31 john`
 - `#chage -l <user>` → to list all info about user's password; read them from `/etc/shadow`
- Passwd
 - `#passwd -n` → change the min number of days between password changes
 - `#passwd -x` → change the max number of days between password changes
 - `#passwd -w` → To change the number of days to start warning before a password change will be required
 - `#passwd -i` → to change the expiration date for the account
 - `#passwd -l` → to lock the password [the same job of `usermod -L`]
 - `#passwd -u` → to unlock password [the same job of `usermod -U`]

Deleting user accounts

- To delete a user account, you can
 - Manually remove the user from:
 - /etc/passwd file.
 - /etc/shadow files.
 - /etc/group file.
 - remove the user's home directory (/home/username).
 - and mail spool file (/var/spool/mail/username).
 - `#userdel <user_name>`
 - `#userdel -r <user_name>` ➔ It will remove user's home dir and the user's mail spool.

Creating new group

- `#groupadd [options] <group_name>`
 - `#groupadd -g` → to set a specific group id not the default one
- Linux users can be a member of two different kinds of groups
 - Primary group
 - Every user must be a member of only one "private" primary group.
 - Secondary group
 - Every user can be a member of one or more secondary groups
- Give password to a group
 - `#passwd <group_name>`
 - `#passwd -A <username> <group_name>` → set an admin for group
 - `#passwd -a a7med ca` → admin of ca add a7med to the group
 - `#passwd -d a7med ca` → admin of ca delete a7med from the group
 - `#passwd -r ca` → admin of ca remove password of this group
 - `#passwd -R ca` → restrict the access to this group to it's members
 - `#passwd -M a7md ca` → to make a7med member of ca group[by root]
- print all my groups
 - `#groups`
- print all user's groups
 - `#groups <group_name>`
 - `#groupmems -g group1 -l` → see which users are a member of group1
- Switch to a group
 - `#newgrp <group_name>`
 - I can switch to a certain group if I am member in it or I have it's password.
 - when I switch to group and create any files or dirs, the permissions on group part for the group that I switch not for my primary group.

Modifying and deleting an existing group

- `groupmod [options] <group_name>`
 - `groupmod` command can be used to change the name or group ID of the group,
 - but it does not allow you to add group members.
 - `#groupmod -n →` to change group name
 - `#groupmod -g →` to change group ID
- **Deleting a certain group**
 - `#groupdel <group_name>`
- To list all file which are owned by groups not defined in `/etc/group` file
 - `#find / -nogroup`

Switching accounts

- `#su <account>`
- `#su - <account>`
- `#su - <account> -c <command>`

Ownership

- To change user and group ownership of a file or dir
 - `chown <user_name> <file_path>`
 - `chown -R <user_name> <dir_path>`
 - `chown <user_name>:<group_name> <file_path>`
 - `chown -R <user_name>:<group_name> <dir_path>`
 - `chown :<group_name> <file_path> === chgrp <group_name> <file_path>`

Basic permissions

permission	Read	write	execute
File	cat,more,less,head,tail,cp,wc	gedit,vi	If it's a program
directory	ls	mkdir,touch,rm,rmdir,mv	cd

Symbolic mode

Who

- ♦ u: Owner permissions
- ♦ g: Group permissions
- ♦ o: Other permissions
- ♦ a: all permissions

Operator

- ♦ + Add permissions
- ♦ - Remove permissions
- ♦ = Assign permissions absolutely

Permissions

- ♦ r: read
- ♦ w: write
- ♦ x: execute

Octal mode

- ♦ 4 read
- ♦ 2 write
- ♦ 1 execute

To change the permission use

- #chmod permission <filename>
- #chmod u=symbolic_value,g=symbolic_value,o=symbolic_value <filename>
- #chmod octal_value <filename>

Default permissions

- The `#umask` command shows and sets the default permissions for files and directories
- `#umask` → to show default permissions in octal mode
- `#umask -S` → to show default permissions in symbolic mode
- `#umask <permissions>` → to change the default permissions of files and dir (`#umask u=rwx,g=rwx,o=rwx`)
- Note: File don't have execute permissions by default anyway.

sudo

- vi /etc/sudoers and go to the end of the file
- Create a new file with the authorization rules in the /etc/sudoers.d directory, all files under this dir included in /etc/sudoers file (#includedir /etc/sudoers.d)
 - #echo "username ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/username
- visudo will open the file directly

```
<user_name>      ALL=(ALL)      /usr/sbin/useradd, /usr/sbin/usermod
%<group_name>    ALL=(ALL)      ALL
<user_name>      ALL=(ALL)      NOPASSWD: /usr/sbin/useradd, /usr/sbin/usermod
<username> <hostname.example.com> = (<run_as_user>:<run_as_group>) <path/to/command>
```
- Permit a sudo for a user on all system commands
 - #visudo

```
<user_name>      ALL=(ALL)      ALL
```
 - #usermod -aG wheel <username>
- Use sudo as a normal user
 - #sudo -u <username> <command>