



RHSA1
Red Hat System Administration I
Day 2

Day 2 Contents

- User and group administration.
- Permissions.
- Switching to other accounts.
- Shutting down the system.



Listing Directory Contents

- **ls -l dir1**

-rwxr-xr-x 2 root root 20 512 May 21 16:06 file1

d**rwxr-xr-x** **2** **fatma** **fatma** **20 512 May 21 16:06 dir2**

Permission Owner Group



User Accounts

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

Users and Groups

- 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.

Users and Groups

```
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 and Groups

- 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.

Users and Groups

- The `/etc/group` file
groupname:x:gid:comma-separated list of group members
- The `/etc/gshadow` file???

Adding New User

- `useradd [options] username`
- `passwd username`
- The **useradd** command populates user home directories from the **/etc/skel** directory.
- To view and modify default setting:
useradd -D
- Adding multiple user accounts:
newusers filename



Creating New Group

- `groupadd [options] groupname`
 - Linux users can be a member of two different kinds of groups.

Primary group

Secondary group

- Every user must be a member of **only one "private" primary group**.
- This primary group has the same name as the user's username.
- Every user can be a member of **one or more secondary groups**.
- Use the **-r** option to the `groupadd` command avoids using a GID within the range typically assigned to users and their private groups.

Adding New User

Example

```
useradd -u 1003 -g 1003 -c "comment" -md /home/user1 -s /bin/bash user1
```

```
passwd user1
```

```
id ???
```

```
groups ???
```



Red Hat

Modifying User Accounts

- To change a user's account information, you can:
 - Edit the `/etc/passwd` or `/etc/shadow` files manually.
 - Use the `usermod` or `chage` commands.

Modifying User Accounts

- 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**.
- There is just one task it does not do well: setting **passwords**.

Modifying User Accounts

- `usermod [options] username`

Options

- To change the login name use `-l <login name>`
- To lock the password use `-L`
- To unlock the password use `-U`
- To add new secondary group use `-aG`

Modifying an Existing Group

- `groupmod [options] groupname`
- The **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**.

Options

- To changes the groupname use **-n**
- To changes group ID use **-g**

Group membership

- `groupmems -g group1 -l`
- The **groupmems** command can be used to see which users are a member of group1 .

Deleting A 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`).

Deleting A User Accounts

- To delete a user account, you can:
 - `userdel -r username`

Options

- **-r:** It will remove user's home directory and the user's mail spool. Files located in other file systems will have to be searched for and deleted.

Deleting A Certain Group

- To delete a certain group, you can:
 - `groupdel groupname`
- To List all file which are owned by groups not defined in `/etc/group` file
 - `find / -nogroup`

Password Aging Policies

- The **chage** command sets up password aging.
- **chage [options] username**

Options

- To change the min number of days between password changes use **-m**
- To change the max number of days between password changes use **-M**
- To change the expiration date for the account use **-E date**
- To change the number of days to start warning before a password change will be required use **-W**
- To show password expiry information use **-l**

Password Aging Policies

- The **passwd** command updates authentication tokens.
- **passwd [options] username**

Options

- To change the min number of days between password changes use **-n**
- To change the max number of days between password changes use **-x**
- To change the expiration date for the account use **-i**
- To change the number of days to start warning before a password change will be required use **-w**
- To lock the password use **-l**
- To unlock the password use **-u**

Switching Accounts

- `su [-] [username]`
- `su [-] [username] -c command`

The **whoami** Command

- After switching into several users, it is a sever issue to know your current (effective) user
- **whoami**
root

The **id** Command

- **Displays**

- Effective user id.
- Effective user name.
- Effective group id.
- Effective group name.

- **Example**

id user1

uid=101(user1) gid=100(user1) groups=101(user1)

The **who** Command

- Who is on the system.
- Displays
 - User Login name .
 - Login device(tty).
 - Login date and time.
- Example

who

The **w** Command

- The **w** command display a summary of the current activity on the system, including what each user is doing.
- **W [user]**
- **Example**

w

Using `sudo` Command

Ownership and Permissions

- Every file and directory has both **user** and **group** ownership. A newly-created file will be owned by:
 - ▶ The user who creates it.
 - ▶ That user's primary group.

Ownership and Permissions

- File ownership can be changed using **chown** command.

- **Example:**

```
chown user1 file1
```

```
chown user1:group1 file1
```

```
chown :group1 file1
```

```
chown -R user2 dir1
```

Security Scheme

- Each file has an owner and assigned to a group.
- Linux allows users to set permissions on files and directories to protect them.
- Permissions are assigned to:
 - File owner.
 - Members of the group the file assigned to.
 - All other users.
- The most specific permissions apply.
- Permissions can only be changed by the owner and root.

Listing Directory Contents

- `ls -l dir1`

`-rwxr-xr-x 2 root root 20 512 May 21 16:06 file1`

`drwxrwxrwx 2 fatma fatma 20 512 May 21 16:06 dir2`

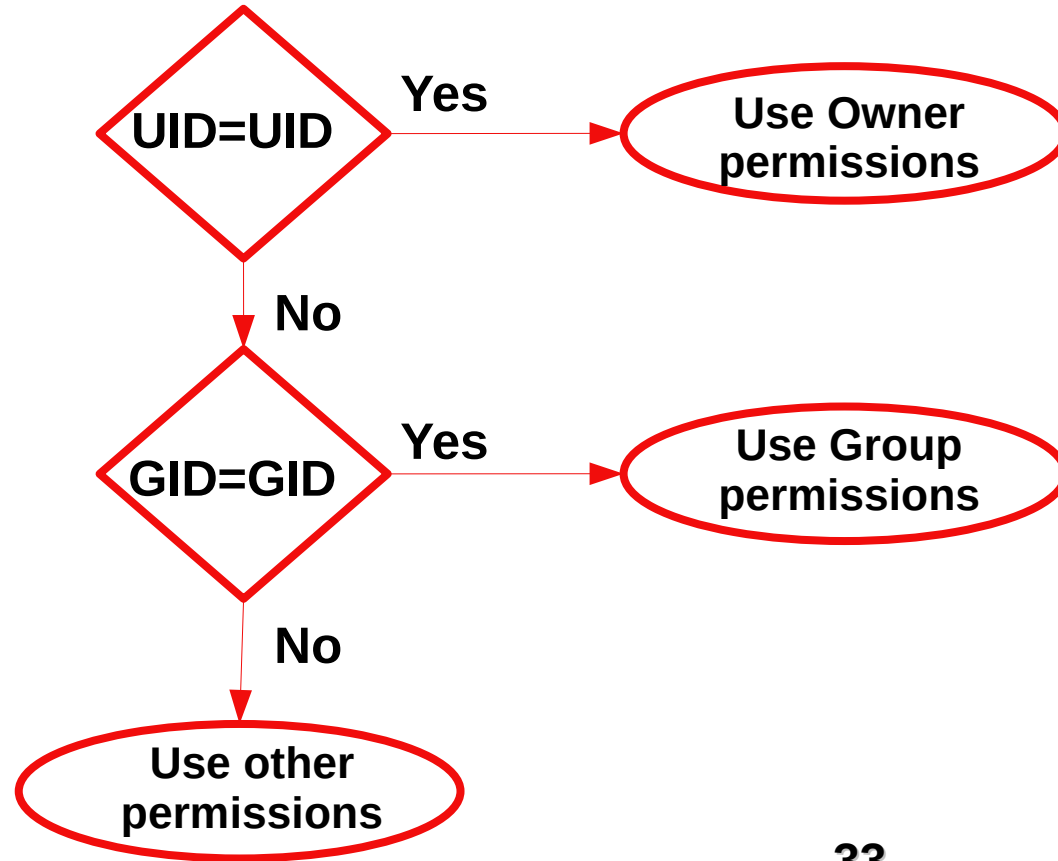
drwxrwxrwx 2 fatma fatma 20 512 May 21 16:06 dir2
User Group others Owner Group
owner
Permission



Permission Notations

Permission	Access for a File	Access for a Directory
Read	You can display file contents and copy the file.	You can list the directory contents with the ls command.
Write	You can modify the file contents.	If you also have execute access, you can add and delete files in the directory.
Execute	You can execute the file if it is an executable. You can execute a shell script if you also have read and execute permissions.	You can use the cd command to access the directory. If you also have read access, you can run the ls -l command on the directory to list contents.

Determining Permissions



Changing Permissions

- `chmod permission filename`
- Permissions are specified in either 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

To change the file permissions for an existing file or directory.

`chmod` u=symbolic_value,g=symbolic_value,o=symbolic_value filename

Changing Permissions

- `chmod permission filename`
- Example:
Change the permissions of `oldpasswd` file to give owner read and write permissions and for group read ,write and execute and execute only for the others.

Changing Permissions

- `chmod permission filename`
- Example:
Change the permissions of `oldpasswd` file to give owner read and execute permissions and add read permission to group and remove execute permission for the others.

Changing Permissions

- `chmod permission filename`
- Permissions are specified in either Octal mode

- ◆ 4 read
- ◆ 2 write
- ◆ 1 execute

To change the file permissions for an existing file or directory.

`chmod` `octal_value` `filename`

Examples

- `ls -l file1`
`-rw-r--r-- 1 user1 staff 1319 Mar 22 14:51 file1`
- `chmod o-r file1`
- `ls -l file1`
`-rw-r----- 1 user1 staff 1319 Mar 22 14:52 file1`
- `chmod g-r file1`
- `ls -l file1`
`-rw----- 1 user1 staff 1319 Mar 22 14:53 file1`

Examples

- `chmod u+x,go+r file1`
- `ls -l file1`
`-rwxr--r-- 1 user1 staff 1319 Mar 22 14:54 file1`
- `chmod a=rw file1`
- `ls -l file1`
`-rw-rw-rw- 1 user1 staff 1319 Mar 22 14:55 file1`
- `chmod 555 file1`
- `ls -l file1`
`-r-xr-xr-x 1 user1 staff 1319 Mar 22 14:56 file1`

Examples

- `chmod 775 file1`
- `ls -l file1`
`-rwxrwxr-x 1 user1 staff 1319 Mar 22 14:54 file1`
- `chmod 755 file1`
- `ls -l file1`
`-rwxr-xr-x 1 user1 staff 1319 Mar 22 14:55 file1`

Default Permissions

- The `umask` command sets the default permissions for files and directories.

Example:

```
umask 002
```

```
umask
```

```
002
```

System Shutdown

- It only requires reboot or shutdown when you need to
 - Add or remove hardware
 - Upgrade to a new version of Ubuntu
 - Or upgrade your kernel
 - shutdown -k now
 - # doesn't really shutdown only send the warning messages and disable logins.
 - shutdown -h time # Halt after shutdown
 - poweroff
 - init 0

