User management tools

- Command-line
  - **Useradd**
  - **Usermod**
  - **userdel [-r]**

Examples

- useradd <username> :- adding new user

  # useradd redhat

- passwd <username> :- setting password

  # passwd redhat

- userdel <username> :- deleting user

  # userdel redhat

- userdele -r <username> :- deleting from home directory

  # userdel -r redhat

Group management tools

- System -> Administration -> Users and Groups
- Groupadd
- groupmod
- groupdel
- Eg :-
- groupadd <groupname> :- adding new group

```
# groupadd redhat
```

- groupdel <groupname> :- removing group

```
# groupdel redhat
```

•Adding group with gid 600

groupadd -g 600 linux

•Belongs to secondary group

groupadd admin

useradd local

usermod -aG admin local

•# gpasswd -d user group (remove from group)

•Renaming group

• groupmod -n <new groupname> <old name>

groupmod -n sys admin

To DO enter in sudoers file

- Login into root ----- **su –**
- Edit /etc/sudoers ---- **nano /etc/sudoers**

  At the end of file add below line

  Bob ALL=(ALL) ALL

Secure Linux File Access

Permission Types

- Four symbols are used when displaying permissions:

  r : permission to read a file or list a directory's contents

  w: permission to write to a file or create and remove files from a directory

  x: permission to execute a program or change into a directory and do a long listing of the directory

  -: no permission (in place of the r, w, or x) Slackware

- **File permissions**
  - r = read, view
  - w = write, update
  - x = execute, run
  - - = a permission isn't set
- **Directory permission**
  - r = list contents
  - w = create/delete contents]
  - x = access
  - - = a permission isn't set

Examining Permissions

•File permissions may be viewed using **ls -l file/directory**

•(ls -ld directory)

•Eg :-

ls -l /etc/passwd

-rwxr-xr-x 1 root root 19080 Apr 1 18:26 /etc/passwd

permission field,owner,group,size,date&time,file name

• File type and permissions represented by a 10- character string

• ie., permission are defined for three types of users

Owner	Group	Others

-	- - -	- - -	- - -

• linux file type owner group others

Changing File Ownership

- Only root can change a file's owner
- Only root or the owner can change a file's group
- Ownership is changed with **chown:**

  chown *user_name file|directory*

- Group-Ownership is changed with **chgrp:**

  chgrp *group_name file|directory*

- Eg :-

```
# touch redhat
# ls -l redhat
# useradd linux
# chown linux redhat
# groupadd rhce
# chgrp rhce redhat
# ls -l redhat
```

Changing Permissions in CLI

Changing permissions are represented in two ways

1.Symbolic method

2.Numeric method

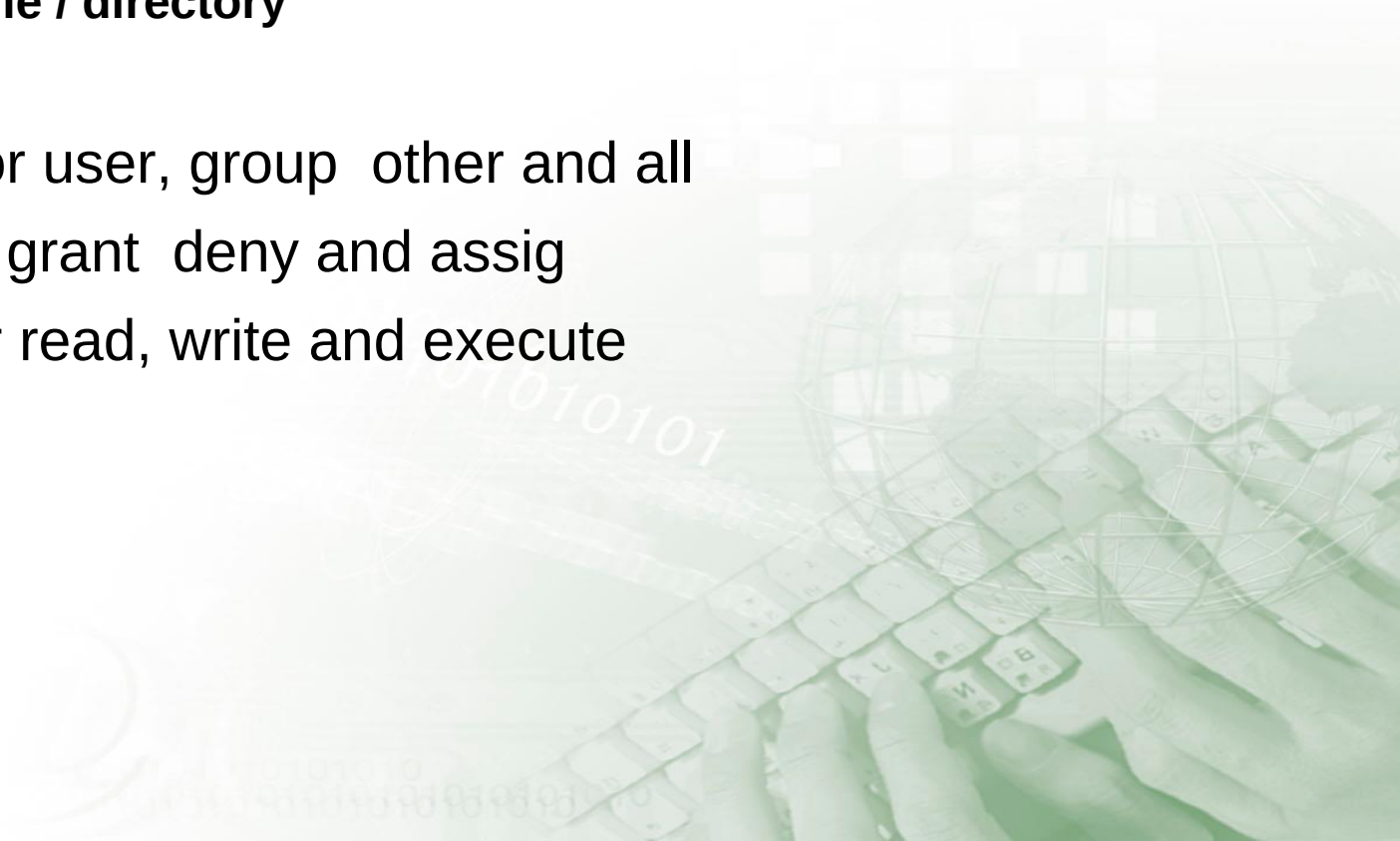
Changing Permissions
Symbolic Method

- To change access modes:
- # **chmod WhoWhatWhich file / directory**

Who = **u,g,o** or **a** for user, group other and all

What = **+ ,-** or **=** for grant deny and assig

Which = **r, w** or **x** for read, write and execute

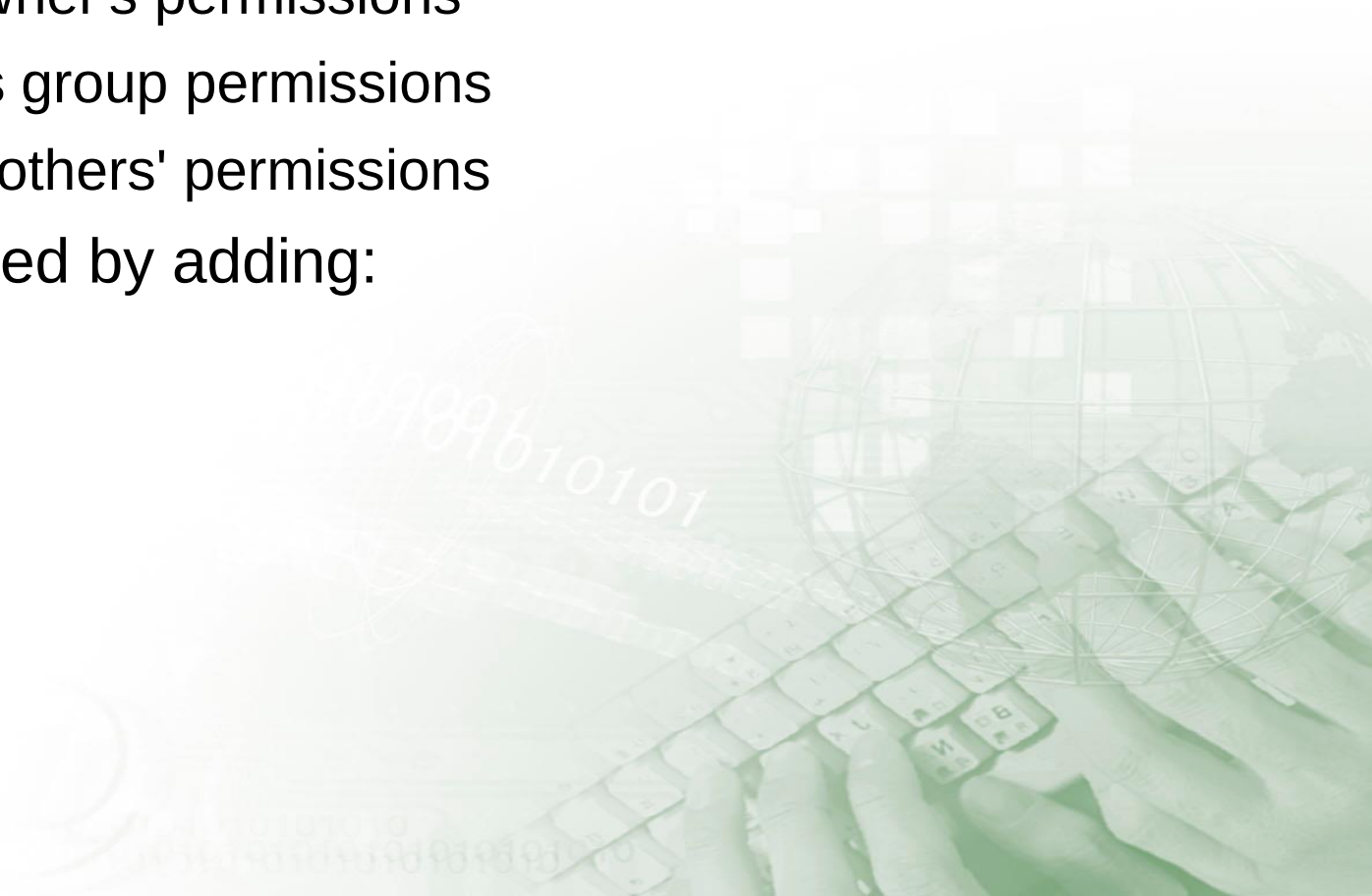


- Eg :-

```
#  touch redhat
#  ls -l redhta
   -rw-r--r– redhat
#  chmod g+w redhat
#  ls -l redhat
   -rw-rw-r-- redhat
#  chmod o-r redhat
#  ls -li redhat
   -rw-rw---- redhat
#  chmod o=w redhat
   -rw-rw--w- redhat
```

Changing Permissions
Numeric Method

- Uses a three-digit mode number
  - first digit specifies owner's permissions
  - second digit specifies group permissions
  - third digit represents others' permissions
- Permissions are calculated by adding:
  - 4 (for read)
  - 2 (for write)
  - 1 (for execute)
- Example:
  - chmod 640 myfile



Permissions	Symbolic	Numeric
Read	r	4
Write	w	2
Execute	x	1
Full Permission	rwx	7

644	**Owner :** read and write permission **Group :** only read permission **Others:** only read permission
755	**Owner** : full permission **Group** : read and execute permission **Others** : read and execute permission

- Example For a user
- User1 and user2
- Log user1

  # touch file1

- Log user2

  # cd /home/user1

  : permission denied

- Log user1

  # chmod 775 /home/user1

- Log user2

  # cd /home/user1

  # ls