جامعة القاهرة الجديدة التكنولوجية

Fall 2023

(8)

Linux Essentials

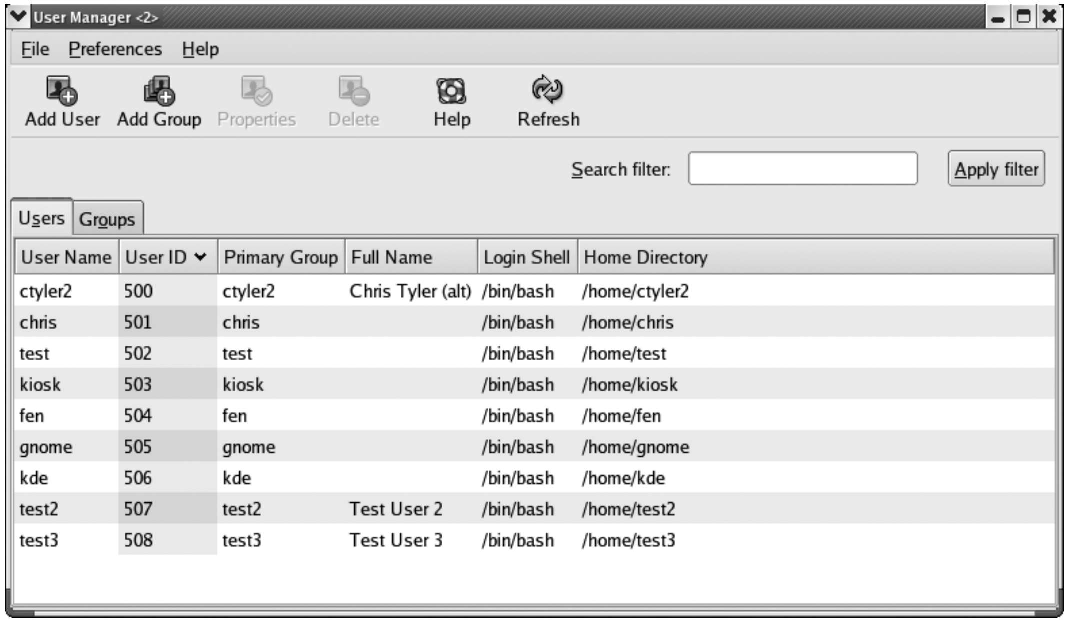
Dr. Hatem Yousry

Agenda

- **Managing Users and Groups.**
  - **User Management in Linux.**
  - **Important commands to manage users in Linux.**
  - **Linux Groups.**
  - **Ubuntu - Packages.**

Linux System Administration: Managing Users and Groups

- Linux is a multiuser OS, meaning that it provides features to help multiple individuals use the computer. Collectively, these features constitute accounts.
- **Understanding accounts.**
- **Using account tools.**
- **Working as root.**


User Manager <2>
File Preferences Help
Add User Add Group Properties Delete Help Refresh
Search filter: Apply filter
Users Groups

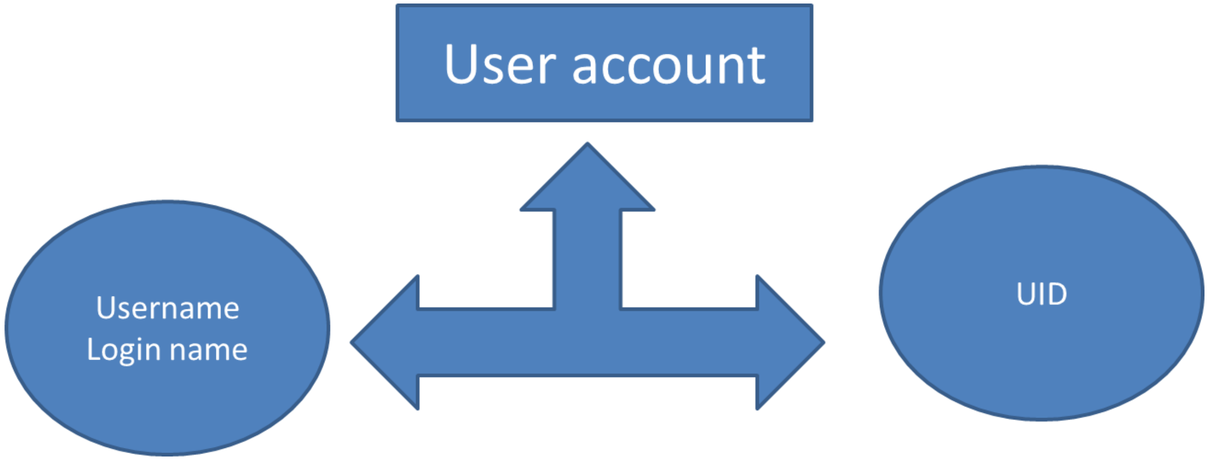
User Name	User ID	Primary Group	Full Name	Login Shell	Home Directory
ctyler2	500	ctyler2	Chris Tyler (alt)	/bin/bash	/home/ctyler2
chris	501	chris		/bin/bash	/home/chris
test	502	test		/bin/bash	/home/test
kiosk	503	kiosk		/bin/bash	/home/kiosk
fen	504	fen		/bin/bash	/home/fen
gnome	505	gnome		/bin/bash	/home/gnome
kde	506	kde		/bin/bash	/home/kde
test2	507	test2	Test User 2	/bin/bash	/home/test2
test3	508	test3	Test User 3	/bin/bash	/home/test3

Understanding Accounts

- Accounts enable multiple users to **share a single computer** without causing each other too much trouble.
- They also **enable system administrators to track who** is using system **resources and, sometimes, who is doing things that they shouldn't be doing.** Account features help users use a computer and administer it. Understanding these features is the basis for enabling you to manage accounts.
- Some account features help you **identify accounts and the files and resources associated** with them. Knowing how to use these features will help you track down account-related problems and manage the computer's users.

What is a user account?

- A user account is a systematic approach to track and monitor the usage of system resources. Each user account contains two unique identifiers; **username and UID.**
- When a user account is created, its username is mapped to a unique UID.

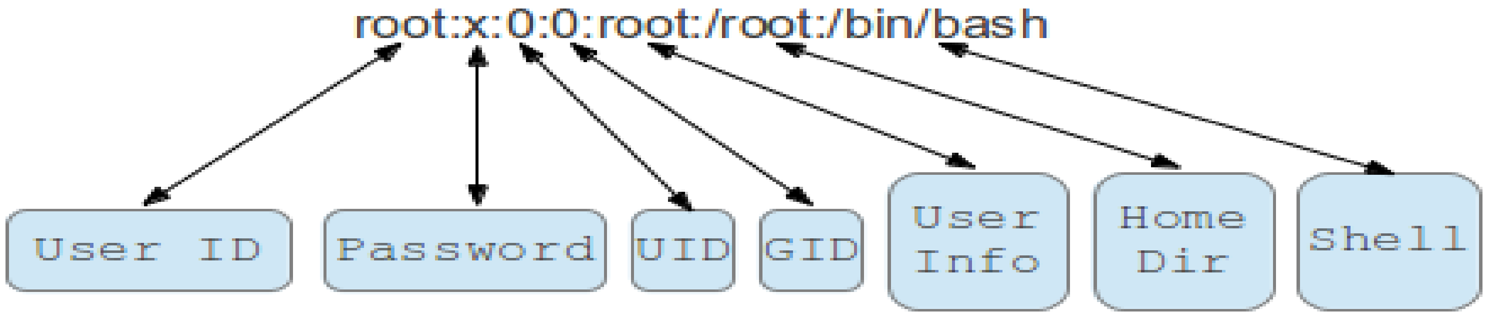


What is a user account?

- Username is flexible. It can be changed as per requirement.
- Regardless it is selected first time or changed later; it must be unique in system. **Two users can’t use the same username.**
- **UID is fixed.** It cannot be changed. Once assigned, it always remains the same for that user account.
- Username is used to access the user account. Username is also known as login name. **UID is used to authenticate**, track and monitor the activity of user account. Username is used by the user while the UID is used by the system.

Understanding Account Features

- Most account features are defined in the /etc/passwd file, which consists of colon-delimited lines, with each line (or record) defining a single account. An entry might resemble the following:
- **rich:x:1003:100:Rich Blum:/home/rich:/bin/bash**
- **⟨username⟩:⟨password⟩:⟨UID⟩:⟨GID⟩:⟨GECOS⟩:⟨home directory⟩:⟨shell⟩**
- The information contained in the fields of this record includes the following:
- **Username, Password, UID, GID, Comment Field, and Home Directory.**



rich:x:1003:100:Rich Blum:/home/rich:/bin/bash

- **Username:** An account’s username is its most relevant feature.
- **Password:** User accounts are typically protected by a password, which is required to log into the computer. The password is stored as a salted hash.
- **UID:** The computer uses **a user identification (UID)** number to track accounts.
- **GID:** Accounts are tied to one or more groups, which are similar to accounts in many ways; however, a group is a collection of accounts. One of the primary purposes of groups is to enable users to give certain users access to their files, while preventing others from accessing them. Each account is tied directly to a primary group via a **group ID (GID)** number (100 in the preceding example). Accounts can be tied to other groups by inclusion in the group’s definition.
- **Comment Field:** The comment field normally holds the user’s full name (Rich Blum in this example), although this field can hold other information instead of or in addition to the user’s name.
- **Home Directory:** User accounts, and some system accounts, have home directories(/home/rich in this example). A home directory is an account’s “home base.” Normally, ownership of an account’s home directory belongs to the account.

user account’s UID

- You can find out a user account’s UID, the primary and secondary groups and the corresponding GIDs by means of the id program:
- ***$ id***
- uid=1000(joe) gid=1000(joe) groups=24(cdrom),29(audio),44(video), 1000(joe)
- ***$ id root***
- uid=0(root) gid=0(root) groups=0(root)

Types of Users

- There are three types of user in Linux: - root, regular and service.

Types of Users

- **The root user account**
- This is the main user account in Linux system. It is **automatically created during the installation.** It has the **highest privilege** in system. It can do **any administrative work** and can **access any service**. This account is intended for system administration and should be used only for this purpose. It should not be used for routine activities. It **can't be deleted**. But if require, it **can be disabled.**
- **The regular user account**
- This is the normal user account. During the installation, one regular user account is created automatically. After the installation, we can create as many regular user accounts as we need. This account has **moderate privilege**. This account is intended for routine works. It can **perform only the tasks for which it is allowed and can access only those files and services for which it is authorized**. As per requirement, it can be disabled or deleted.
- **The service account**
- Service accounts are created by installation packages when they are installed. These accounts are **used by services to run processes and execute functions**. These accounts are neither intended nor should be used for routine work.

Working as root

- Root user, though, needs extraordinary power in order to **manage the features of the computer as a whole.**
- This is the root user, also known as the super user, superuser , or **the administrator.**
- The root account exists to enable you to perform administrative tasks . These tasks include **installing new software, preparing a new disk for use in the computer, and managing ordinary user accounts.** Such tasks require access to system files that ordinary users need not modify, or sometimes even read.
- Log in as root You can log in directly as root at a text-mode shell or by using a remote login tool.

use sudo or su

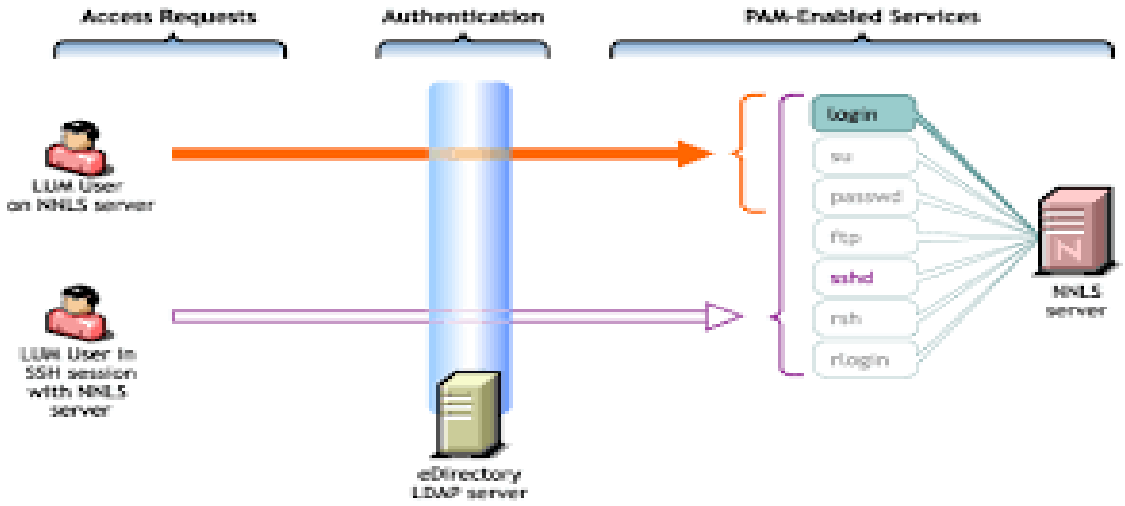
- In Ubuntu distribution Root is disabled so we **use sudo** to execute as a root.
- Other distribution If the root account is enabled, then a regular user can execute the **su command** to switch accounts to the root account.
- ***su [options] [username]*** Command to switch users
- ***$su***
- **You can also use su to run a single command as root by using the -c option, as in su -c command to run command as root.**
- **Use sudo The sudo command is similar to su , but it works for just one command at a time, which you type after sudo , similar to using su -c .**

Superuser Permissions

- Adding a new user involves dealing with an account other than your own which requires superuser (aka root) privileges. The same applies to other user or group management tasks, such as deleting an account, updating accounts, and creating and removing groups.
- These operations are performed using the following commands:
- **adduser:** add a user to the system.
- **userdel:** delete a user account and related files.
- **addgroup:** add a group to the system.
- **delgroup:** remove a group from the system.
- **usermod:** modify a user account.

User Management in Linux

- To list out all the users in Linux, **use the awk command with -F option.**
- Using id command, you can get the **ID** of any username.
- The command to **add** a user.
- Using **passwd** command to assign a password to a user.
- Accessing a user **configuration file**.
- The command to **change the user ID** for a user.



User Management in Linux

- **A user is an entity**, in a Linux operating system, that can manipulate files and perform several other operations.
- Each user is assigned an **ID** that is unique for each user in the operating system.
- After installation of the operating system, the **ID 0** is assigned to the **root user** and the **IDs 1 to 999** (both inclusive) are assigned to the **system users** and hence the ids for local user begins from 1000 onwards.
- We will learn about users and commands which are used to get information about the users.

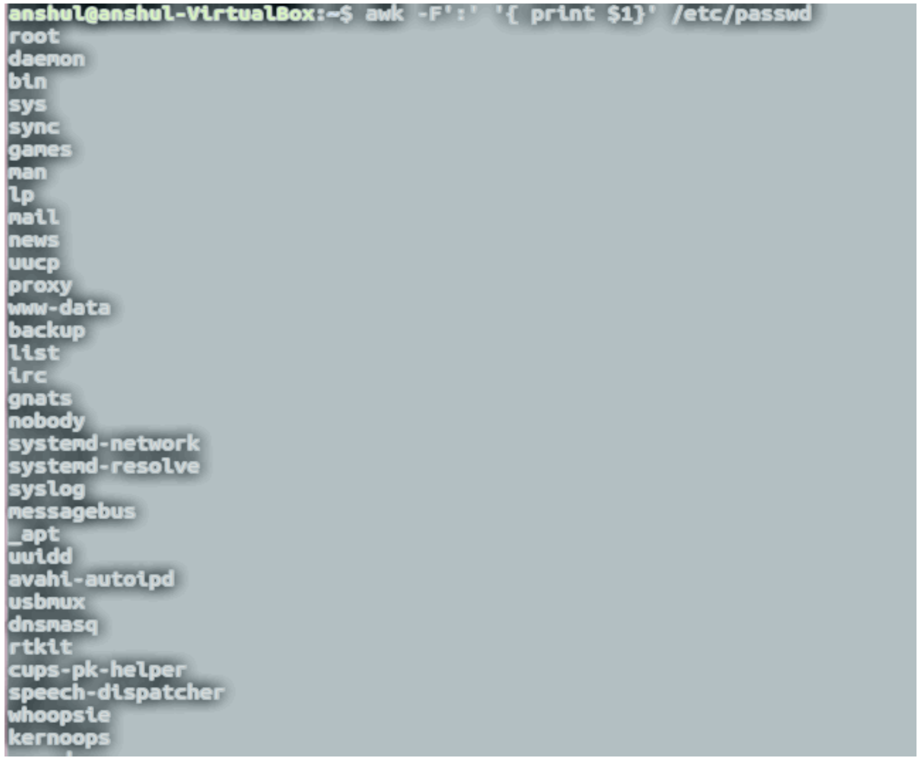
Important commands to manage users in Linux

- AWK command.
- id command.
- useradd command.
- passwd command.
- Accessing a user configuration file command.
- command to change the home directory.
- change the user login name command.
- delete a user name command.
- change the user ID for a user command.
- Modify the group ID of a user command.

AWK command

- To list out all the users in Linux, use the awk command with -F option. Here, we are accessing a file and printing only first column with the help of print $1 and awk.
- awk -F':' '{ print $1}' /etc/passwd

The -F : option tells awk to **use colons** to separate fields. awk refers to fields in a line as $1, $2, $3 ... from left to right. **$0 holds the entire line**, which is often what you want to print out, but not this time.



```
anshul@anshul-VirtualBox:~$ awk -F':' '{ print $1}' /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
syslog
messagebus
_apt
uuidd
avahi-autoipd
usbmux
dnsmasq
rtkit
cups-pk-helper
speech-dispatcher
whoopsie
kernoops
```

AWK command

- Awk is abbreviated from the names of the developers – **Aho, Weinberger, and Kernighan.**
- **Awk is mostly used for pattern scanning and processing.**
- **That define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line.**
- **Syntax:**
- **awk options 'selection _criteria {action }' input-file > output-file**
- **Options:**
  - -f program-file : Reads the AWK program source from the file program-file, instead of from the first command line argument.
  - -F fs : Use fs for the input field separator

AWK command Example:

- Consider the following text file as the input file employee.txt
- **$ awk '{print}' employee.txt**
- **Output:**
- ajay manager account 45000
- sunil clerk account 25000
- varun manager sales 50000
- amit manager account 47000
- tarun peon sales 15000
- deepak clerk sales 23000
- sunil peon sales 13000
- satvik director purchase 80000

```
$cat > employee.txt
ajay manager account 45000
sunil clerk account 25000
varun manager sales 50000
amit manager account 47000
tarun peon sales 15000
deepak clerk sales 23000
sunil peon sales 13000
satvik director purchase 80000
```

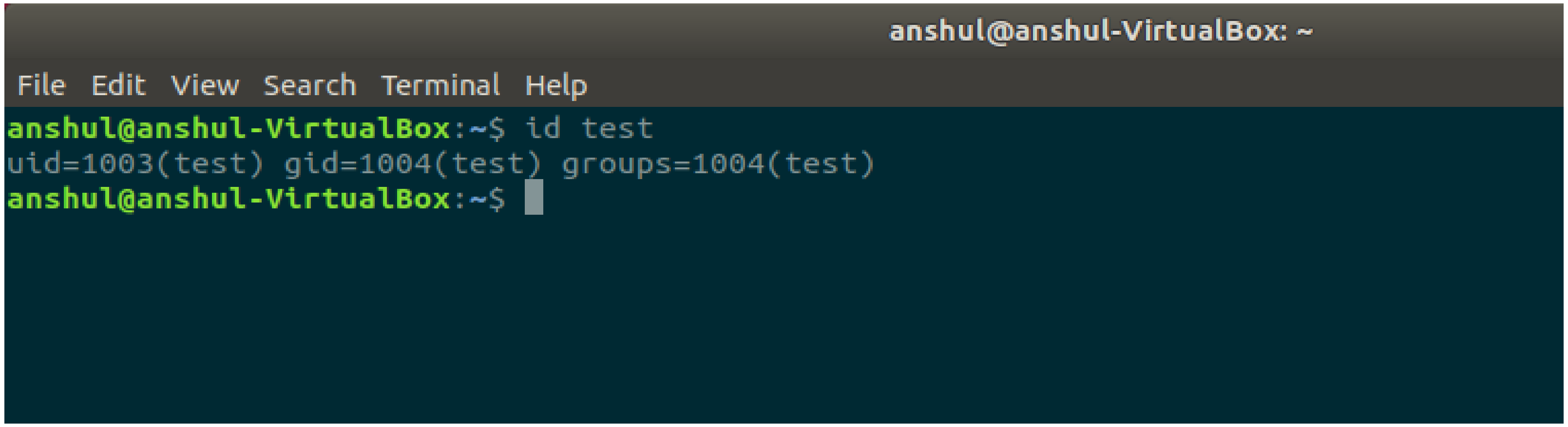
AWK command Example:

- Splitting a Line Into Fields
- **$ awk '{print $1,$4}' employee.txt**
- **Output:**
- ajay 45000
- sunil 25000
- varun 50000
- amit 47000
- tarun 15000
- deepak 23000
- sunil 13000
- satvik 80000

```
$cat > employee.txt
ajay manager account 45000
sunil clerk account 25000
varun manager sales 50000
amit manager account 47000
tarun peon sales 15000
deepak clerk sales 23000
sunil peon sales 13000
satvik director purchase 80000
```

id command

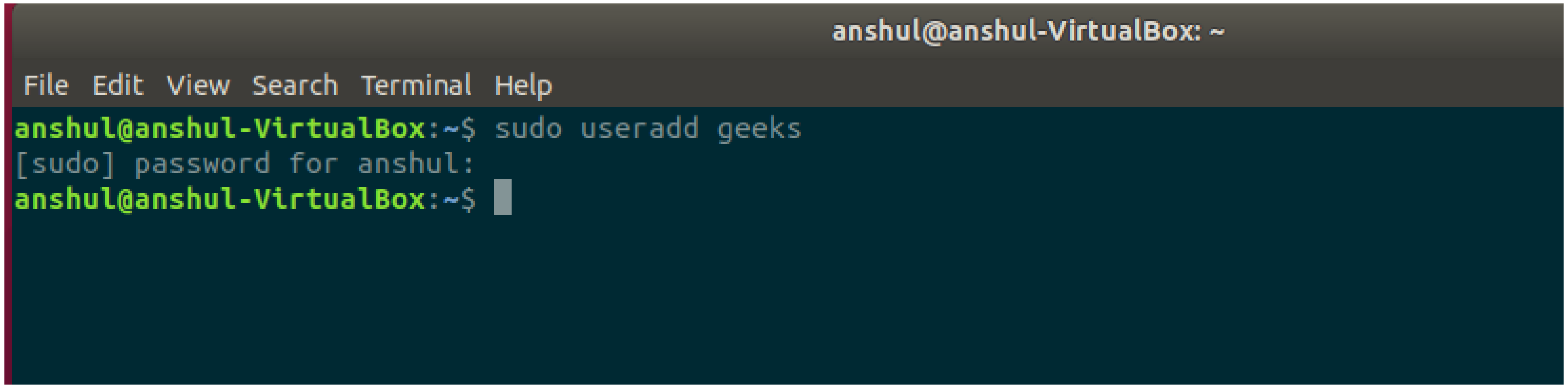
- Using id command, you can **get the ID of any username**.
- Every user has an id assigned to it and the user is identified with the help of this id. By default, this id is also the group id of the user.
- **Syntax:**
- **id username**
- Example: id test


anshul@anshul-VirtualBox: ~
File Edit View Search Terminal Help

```
anshul@anshul-VirtualBox:~$ id test
uid=1003(test) gid=1004(test) groups=1004(test)
anshul@anshul-VirtualBox:~$
```

useradd command

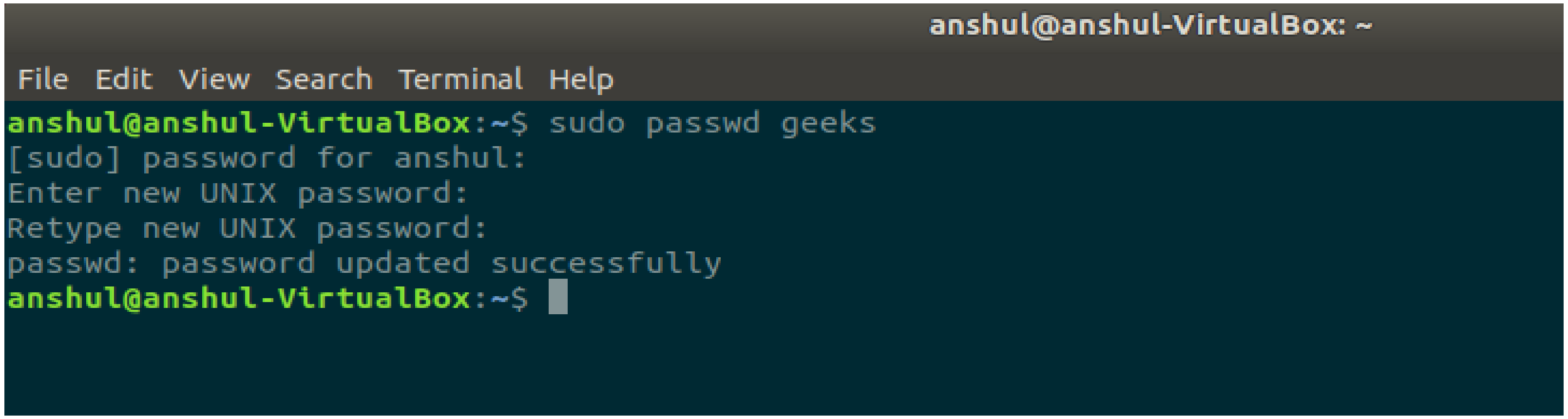
- The command to add a user. useradd command adds a new user to the directory. The user is given the ID automatically depending on which category it falls in. The username of the user will be as provided by us in the command.
- **Syntax:**
- **sudo useradd username**
- Example: sudo useradd geeks


anshul@anshul-VirtualBox: ~
File Edit View Search Terminal Help

```
anshul@anshul-VirtualBox:~$ sudo useradd geeks
[sudo] password for anshul:
anshul@anshul-VirtualBox:~$
```

passwd command

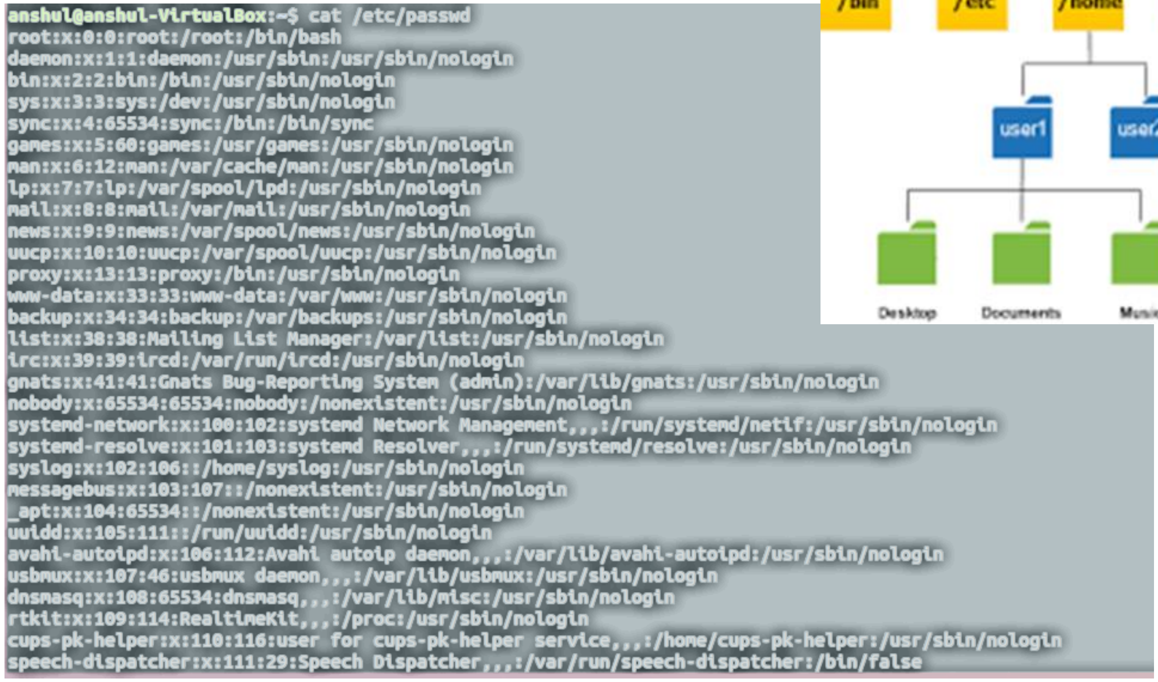
- Using passwd command to assign a password to a user. After using this command we have to enter the new password for the user and then the password gets updated to the new password.
- Syntax:
- **passwd username**
- Example: passwd geeks


anshul@anshul-VirtualBox: ~
File Edit View Search Terminal Help

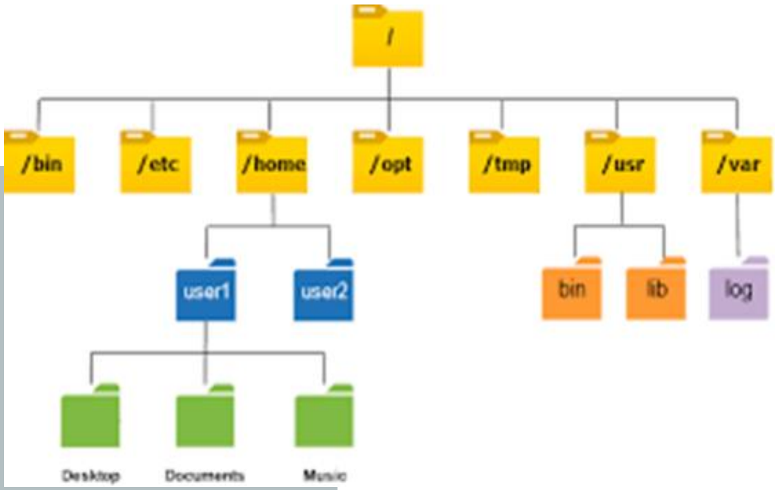
```
anshul@anshul-VirtualBox:~$ sudo passwd geeks
[sudo] password for anshul:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
anshul@anshul-VirtualBox:~$
```

Accessing a user configuration file

- This commands prints the data of the configuration file. This file contains information about the user in the format.
- **username : x : user id : user group id : : /home/username : /bin/bash**
- **cat /etc/passwd**

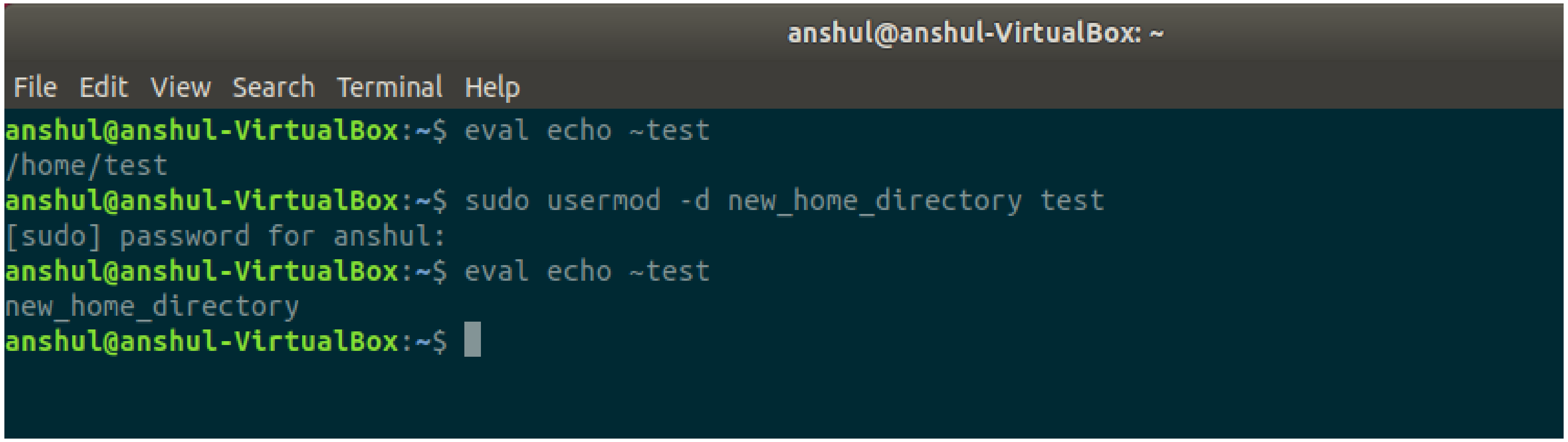


```
anshul@anshul-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:105:111::/run/uuidd:/usr/sbin/nologin
avahi-autoipd:x:106:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
```



command to change the home directory

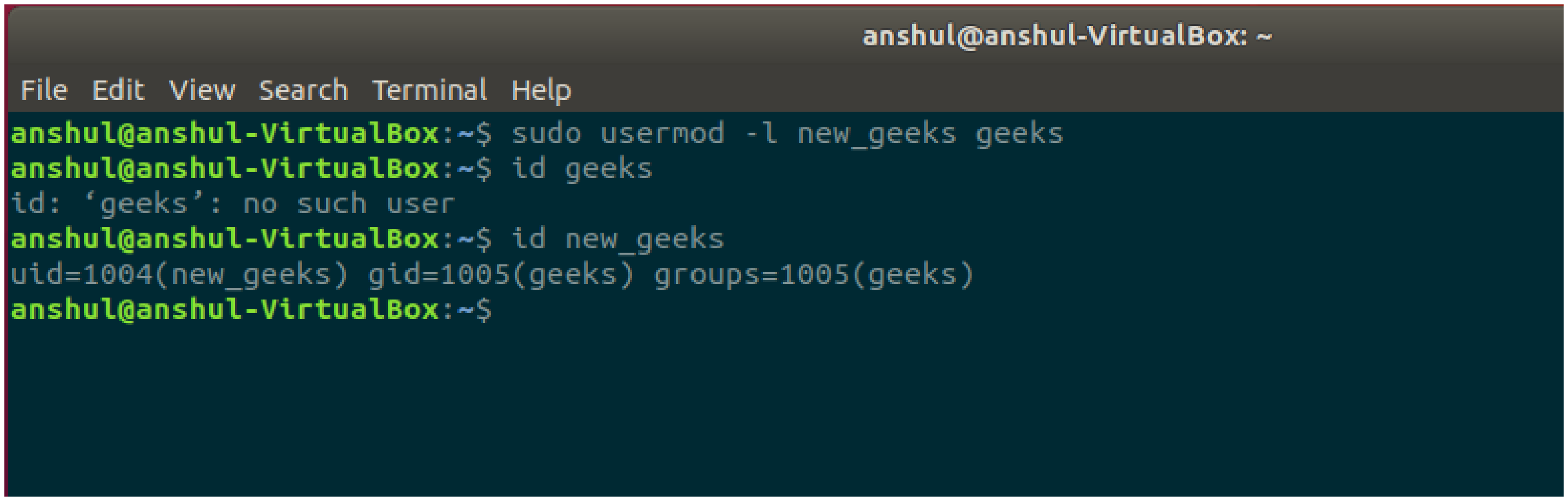
- The command to change the home directory. The below command change the home directory of the user whose username is given and sets the new home directory as the directory whose path is provided.
- **usermod -d new_home_directory_path username**
- Example: usermod -d new_home_directory test


anshul@anshul-VirtualBox: ~
File Edit View Search Terminal Help

```
anshul@anshul-VirtualBox:~$ eval echo ~test
/home/test
anshul@anshul-VirtualBox:~$ sudo usermod -d new_home_directory test
[sudo] password for anshul:
anshul@anshul-VirtualBox:~$ eval echo ~test
new_home_directory
anshul@anshul-VirtualBox:~$
```

change the user login name

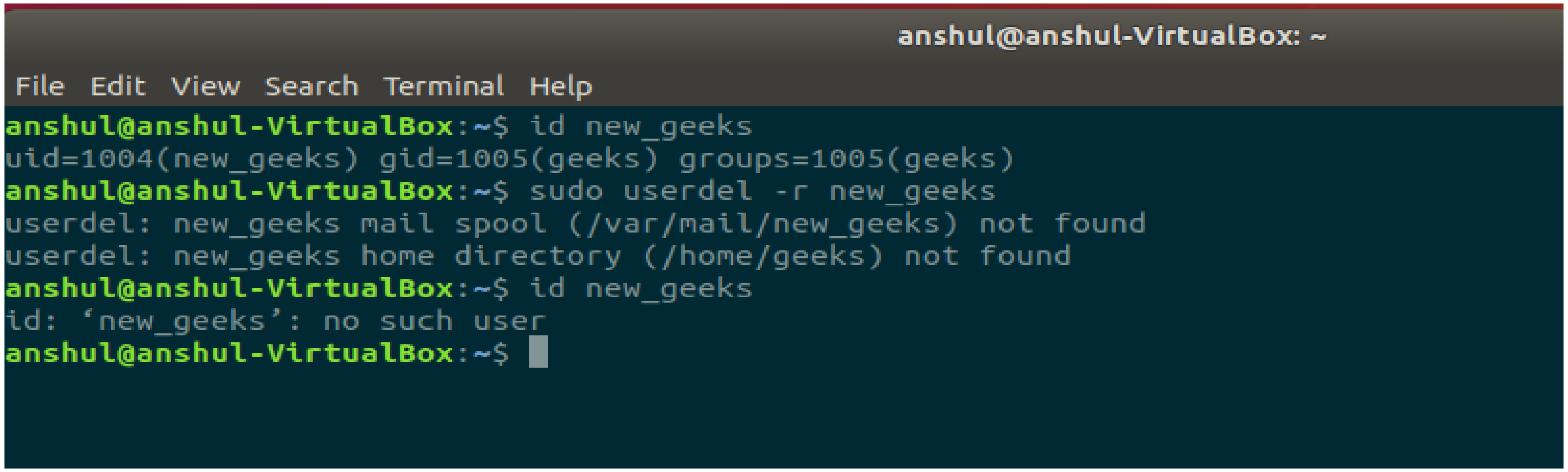
- You can change the user login name using usermod command. The below command is used to change the login name of the user. The old login name of the user is changed to the new login name provided.
- sudo usermod -l new_login_name old_login_name
- Example: sudo usermod -c new_geeks geeks



```
anshul@anshul-VirtualBox:~$ sudo usermod -l new_geeks geeks
anshul@anshul-VirtualBox:~$ id geeks
id: ‘geeks’: no such user
anshul@anshul-VirtualBox:~$ id new_geeks
uid=1004(new_geeks) gid=1005(geeks) groups=1005(geeks)
anshul@anshul-VirtualBox:~$
```

delete a user name

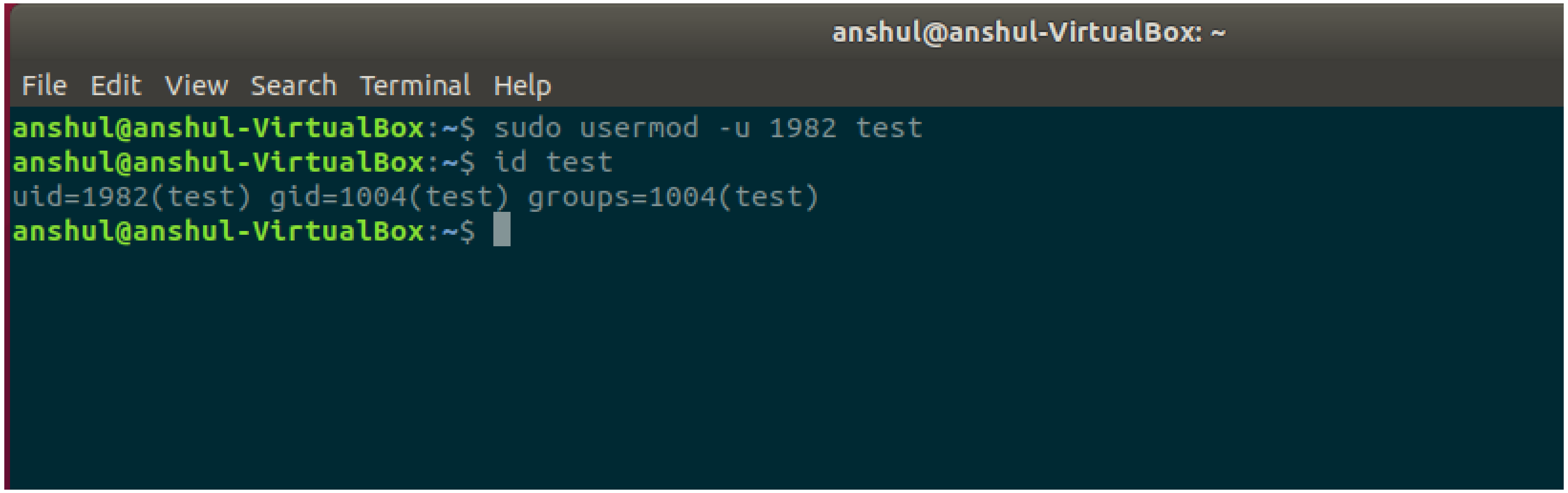
- You can also delete a user name. The below command deletes the user whose username is provided. Make sure that the user is not part of a group. If the user is part of a group then it will not be deleted directly, hence we will have to first remove him from the group and then we can delete him.
- **userdel -r username**
- Example: sudo userdel -r new_geeks


anshul@anshul-VirtualBox: ~
File Edit View Search Terminal Help

```
anshul@anshul-VirtualBox:~$ id new_geeks
uid=1004(new_geeks) gid=1005(geeks) groups=1005(geeks)
anshul@anshul-VirtualBox:~$ sudo userdel -r new_geeks
userdel: new_geeks mail spool (/var/mail/new_geeks) not found
userdel: new_geeks home directory (/home/geeks) not found
anshul@anshul-VirtualBox:~$ id new_geeks
id: 'new_geeks': no such user
anshul@anshul-VirtualBox:~$
```

change the user ID for a user.

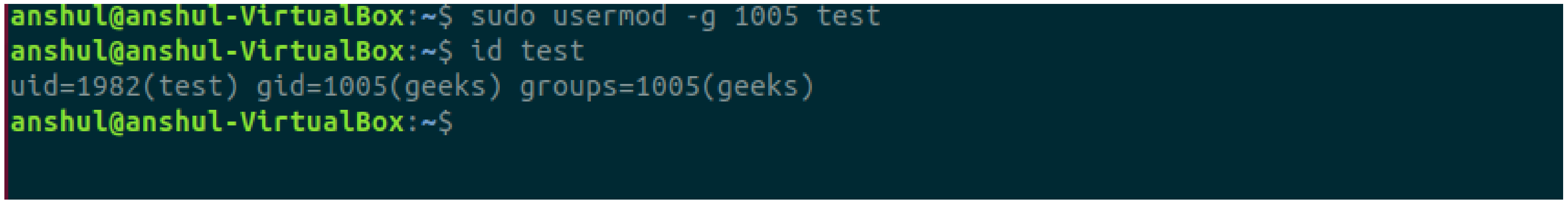
- The command to change the user ID for a user.
- **usermod -u new_id username**
- This command can change the user ID of a user. The user with the given username will be assigned with the new ID given in the command and the old ID will be removed.
- Example: sudo usermod -u 1982 test


anshul@anshul-VirtualBox: ~
File Edit View Search Terminal Help

```
anshul@anshul-VirtualBox:~$ sudo usermod -u 1982 test
anshul@anshul-VirtualBox:~$ id test
uid=1982(test) gid=1004(test) groups=1004(test)
anshul@anshul-VirtualBox:~$
```

Modify the group ID of a user.

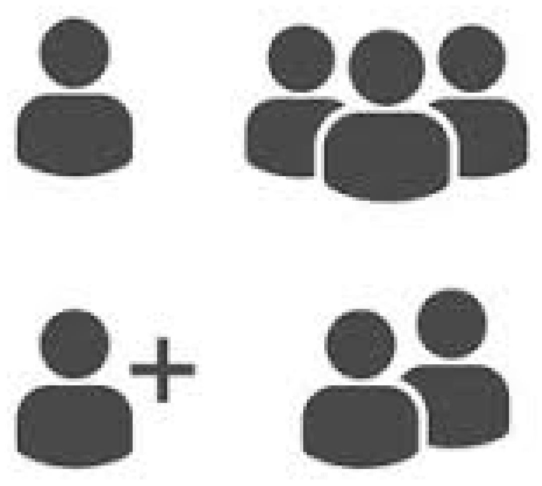
- Command to Modify the group ID of a user.
- **usermod -g new_group_id username**
- This command can change the group ID of a user and hence it can even be used to move a user to an already existing group. It will change the group ID of the user whose username is given and sets the group ID as the given new_group_id.
- Example: sudo usermod -g 1005 test

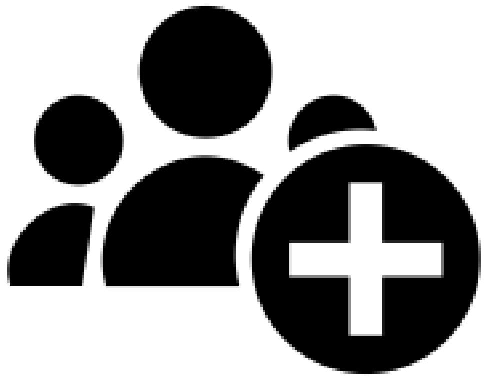


```
anshul@anshul-VirtualBox:~$ sudo usermod -g 1005 test
anshul@anshul-VirtualBox:~$ id test
uid=1982(test) gid=1005(geeks) groups=1005(geeks)
anshul@anshul-VirtualBox:~$
```

Creating and managing groups on Linux

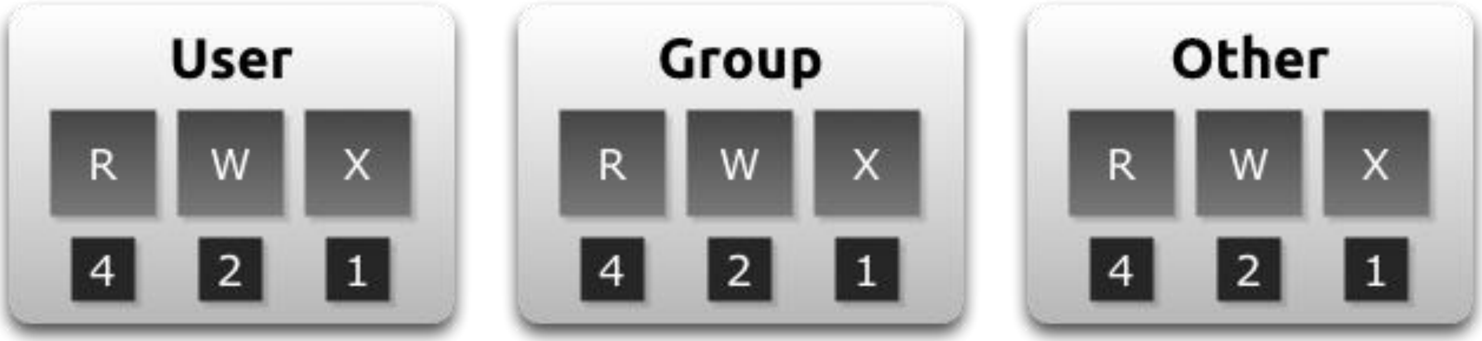
- To create a new group, use the **groupadd command.**
- To add a member to a supplementary group, **use the usermod command** to list the supplementary groups that the user is currently a member of, and the supplementary groups that the user is to become a member of.





Linux Groups

- How Linux groups work
- Managing groups from the command line

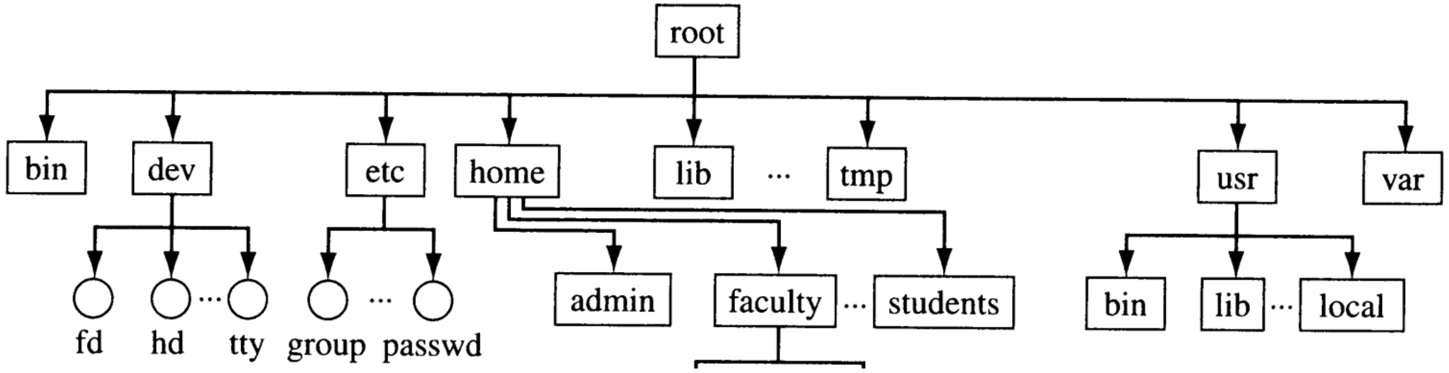


How Linux Groups Work

- If your Linux system has been configured to use local authentication, your groups are defined in the **/etc/group** file.
- Each record is composed of the following four fields:

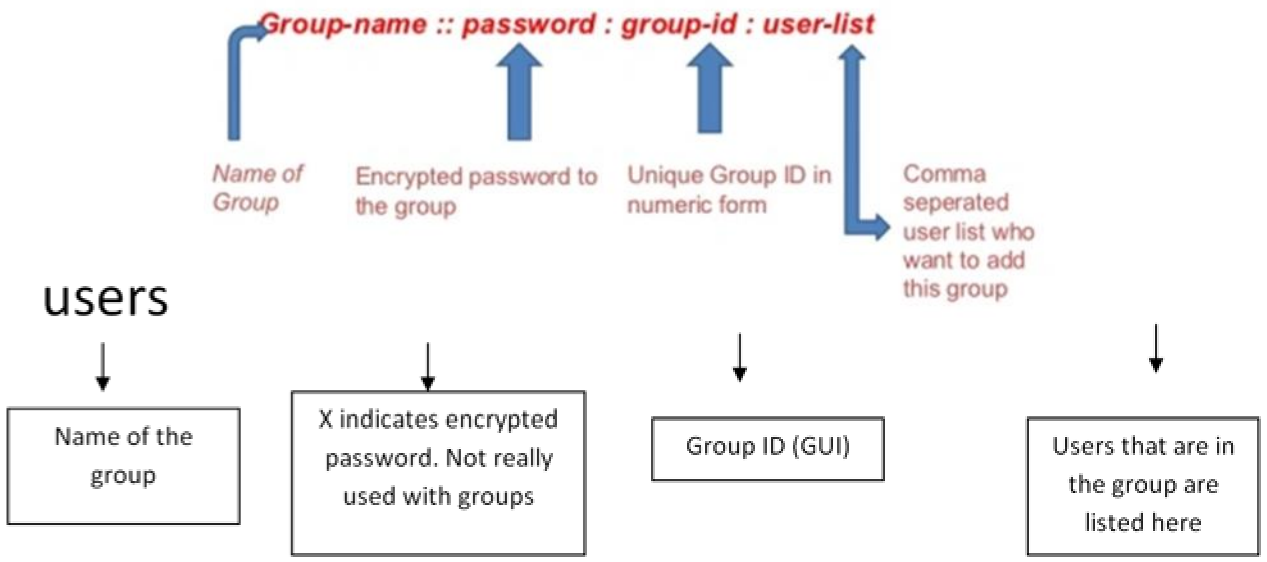
Group:Password:GID:Users

- Group Specifies the name of the group.
- In the example above, the name of the group is video.
- Password Specifies the group password.



How Linux Groups Work

- **GID** Specifies the group ID (GID) number of the group.
- **Users** Lists the members of the group.
- As with **/etc/shadow**, each line in /etc/gshadow represents a record for a single group.
- Each record is composed of the following fields: **Group_Name**:**Password**:Group_Admins:Group_Members


Group-name :: password : group-id : user-list
Name of Group
Encrypted password to the group
Unique Group ID in numeric form
Comma seperated user list who want to add this group
users
Name of the group
X indicates encrypted password. Not really used with groups
Group ID (GUI)
Users that are in the group are listed here

Managing groups from the command line

- Using groupadd.
- Using groupmod.
- Using groupdel.

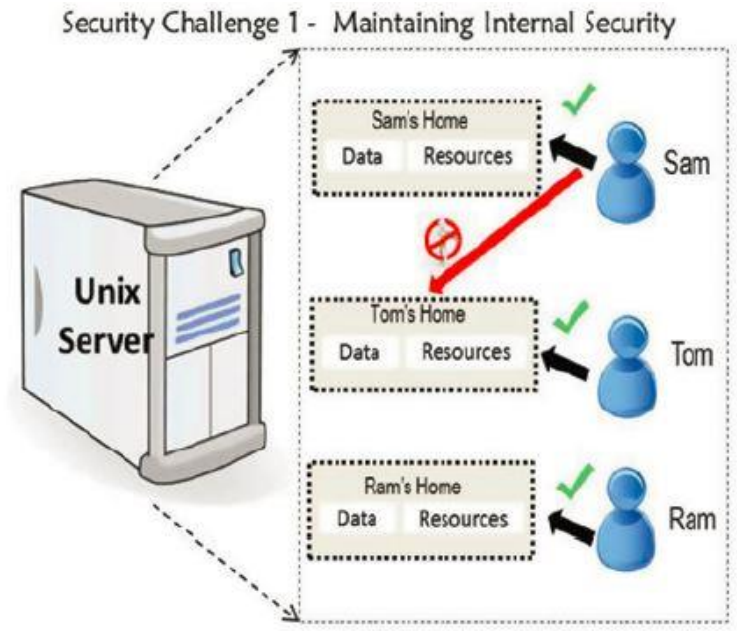


Figure 1. *Maintaining UNIX Internal security*

Using groupadd

- **Syntax:**

groupadd ***options*** ***groupname***

- ***Options:***

–g Specifies a GID for the new group.

–p Specifies a password for the group.

–r Specifies that the group being created is a system group.

Using groupmod

- To modify a group, including adding users to the **group membership**, you use the groupmod utility.
- **Syntax:**

groupmod *options* *group*

- ***Options:***

–g Changes the group’s GID number.

–p Changes the group’s password.

–A Adds a user account to the group.

–R Removes a user account from the group.

Using groupmod

- If we wanted to add ncth to the group, we would enter

  groupmod –A “ncth” *student*

  at the shell prompt.

Using groupdel

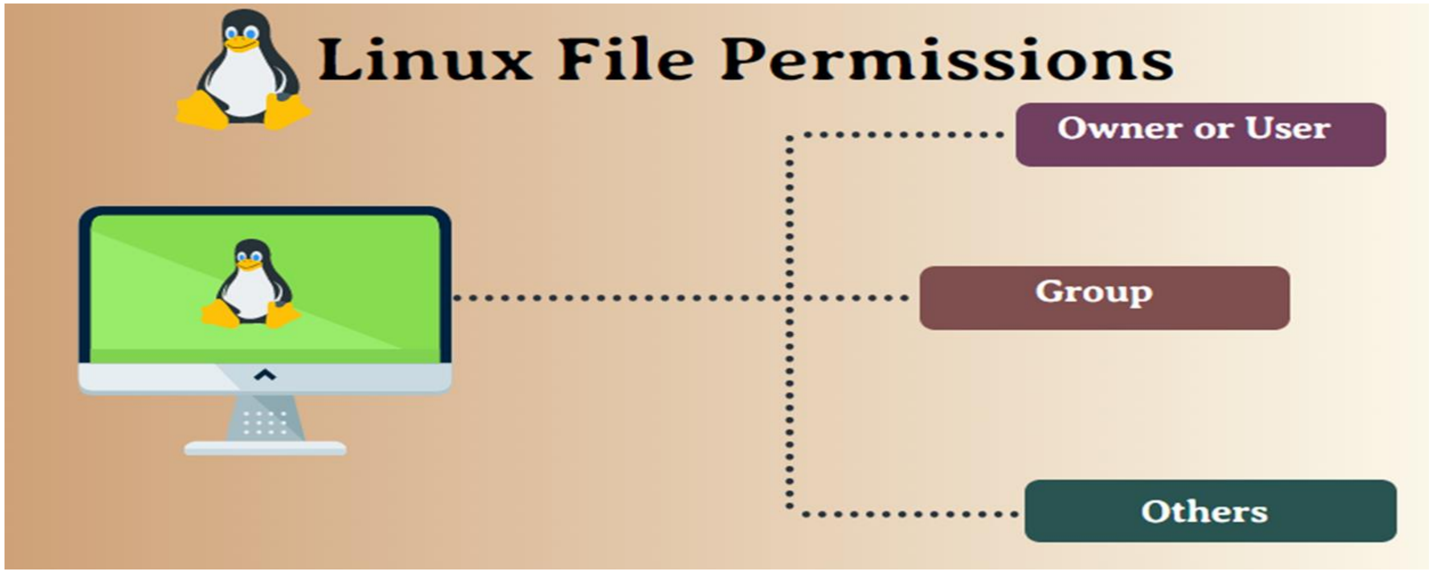
- **Syntax:**

groupdel *group_name*

ex: ***groupdel*** **student**

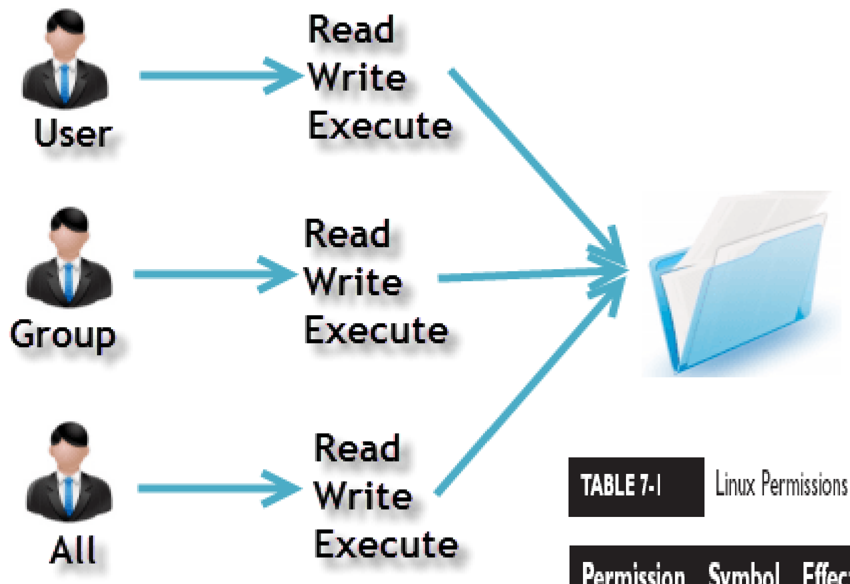
Managing permissions

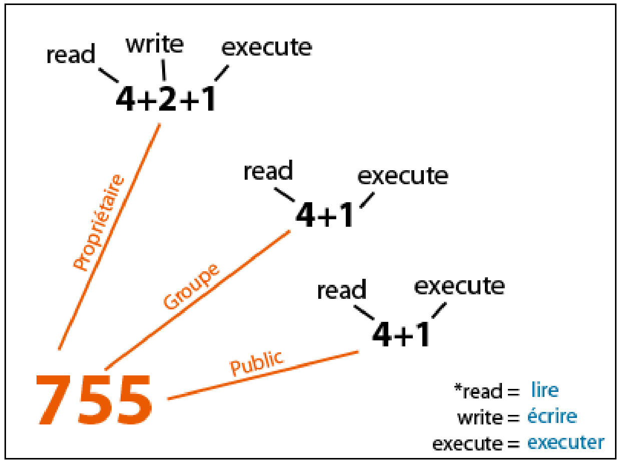
- How permissions work
- Managing permissions from the command line
- Working with default permissions
- Working with special permissions



How permissions work

Owners assigned Permission On Every File and Directory





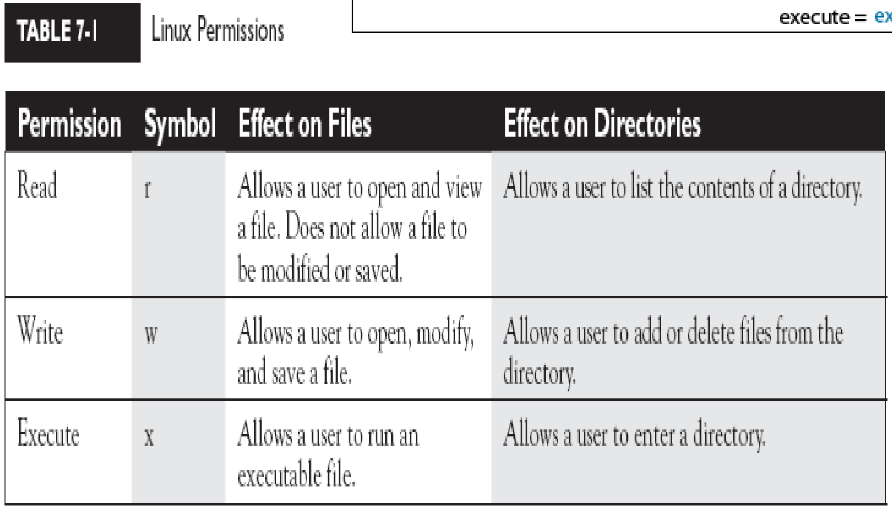


TABLE 7-1 Linux Permissions

Permission	Symbol	Effect on Files	Effect on Directories
Read	r	Allows a user to open and view a file. Does not allow a file to be modified or saved.	Allows a user to list the contents of a directory.
Write	w	Allows a user to open, modify, and save a file.	Allows a user to add or delete files from the directory.
Execute	x	Allows a user to run an executable file.	Allows a user to enter a directory.

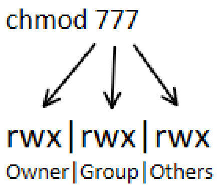
How permissions work

- Each file or directory in your Linux file system stores the specific permissions assigned to it. These permissions together constitute the mode of the file. These permissions are assigned to each of three different entities for each file and directory in the file system:

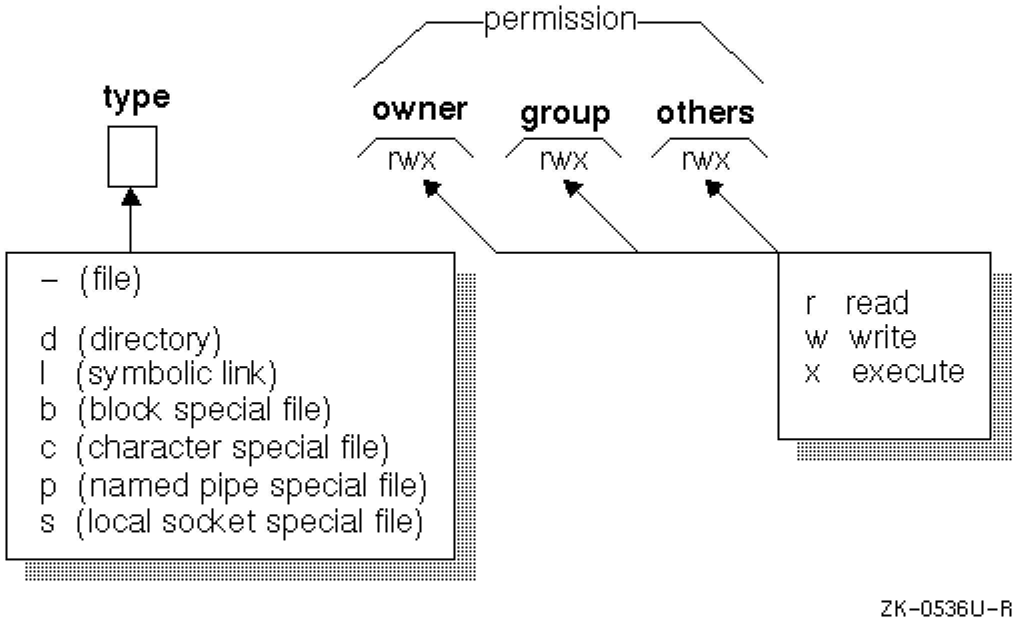
✓ **Owner** This is the user account that has been assigned to be the file or directory's owner. Permissions assigned to the owner apply only to that user account.

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

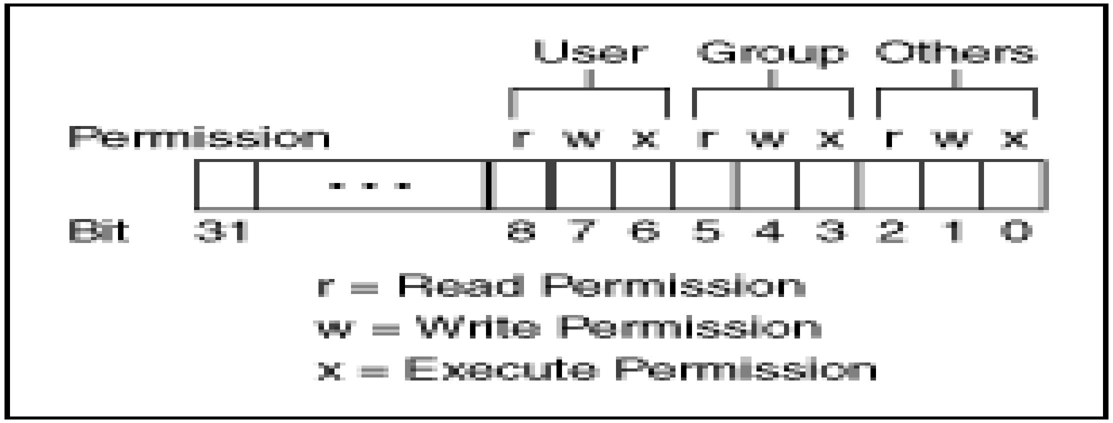


7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000



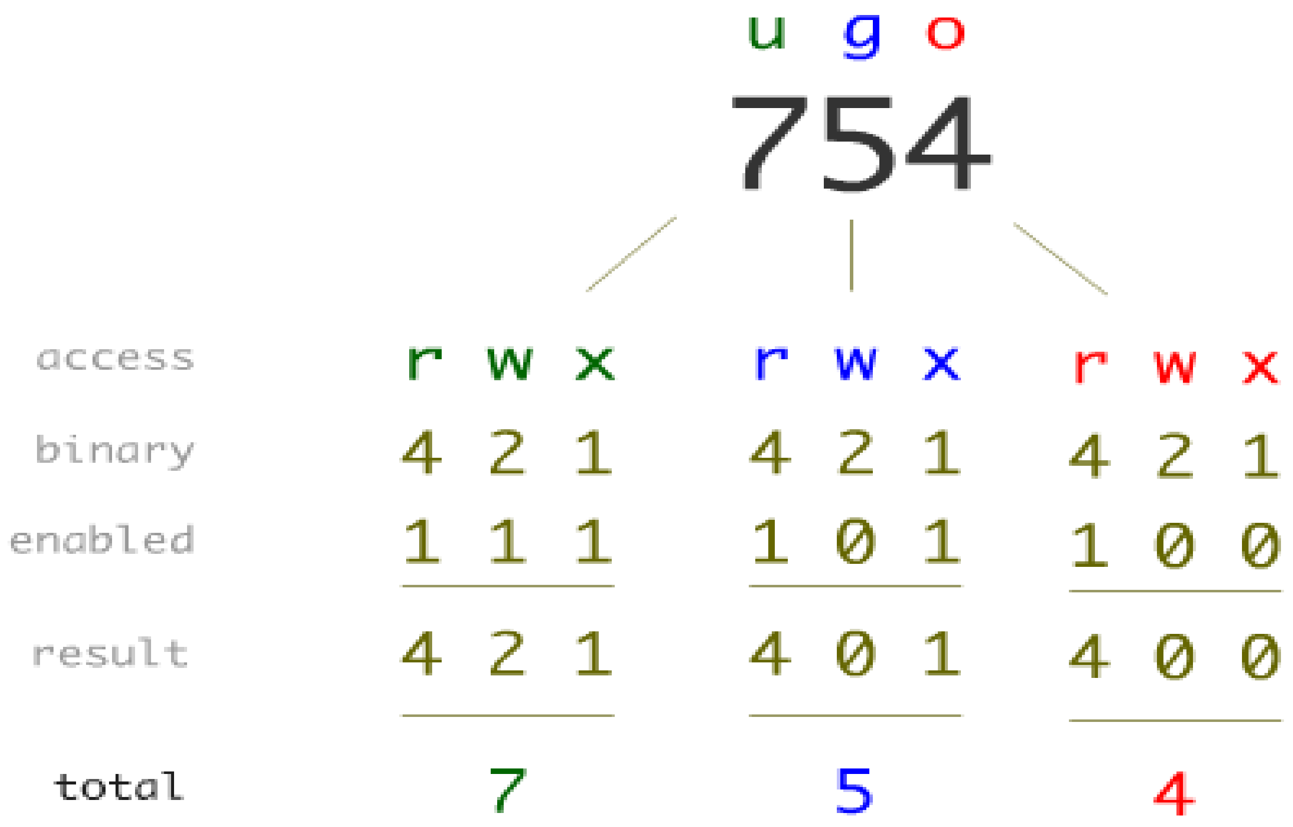
How permissions work

- **Group** This is the group that has been assigned ownership of the file or directory. Permissions assigned to the group apply to **all user accounts** that are members of that group.
- **Others** This entity refers to all other users who have successful authenticated to the system. Permissions assigned to this entity apply to these user accounts.



Managing Permissions from the Command Line with chmod

chmod *entity=permissions* *filename*



	u	g	o
	7	5	4
access	r w x	r w x	r w x
binary	4 2 1	4 2 1	4 2 1
enabled	1 1 1	1 0 1	1 0 0
result	4 2 1	4 0 1	4 0 0
total	7	5	4

Managing Permissions from the Command Line with chmod

Owner, **g for Group**, and o for Others in the entity portion of the command. You substitute **r, w**, and/or **x** for the permissions portion of the command. For example, suppose I wanted to change the mode of contacts.odt to –rw–rw–r– –

```
chmod  u=rw,g=rw,o=r  contacts.odt
```

- You can also use chmod to toggle a particular permission on or off using the + or – signs. For example, suppose I want to turn off the write permission I just gave to Group for the contacts.odt file. I could enter **chmod** g–w **contacts.odt**

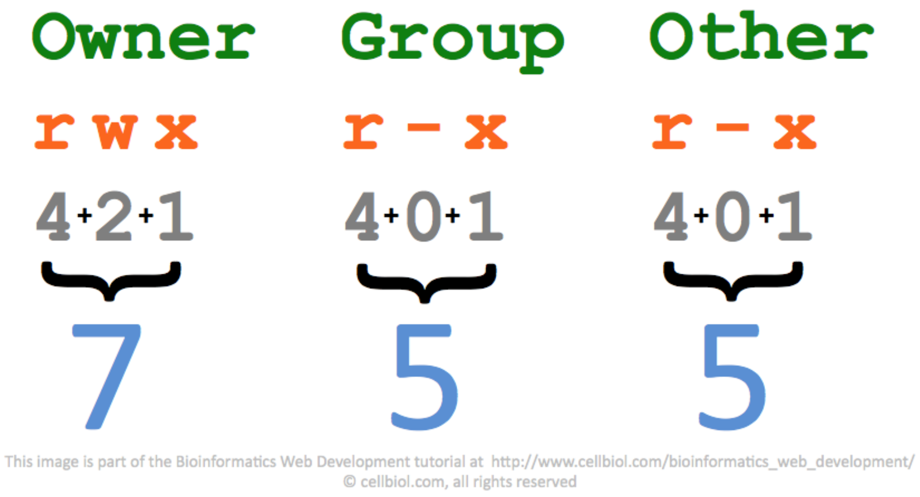
at the shell prompt.

Managing Permissions from the Command Line with chmod

- You can modify all three entities at once with only three characters. To do this, enter

  chmod numeric_permission filename

ex: chmod 755 contacts.odt


Owner
Group
Other
r w x
r - x
r - x
4+2+1
4+0+1
4+0+1
7
5
5
This image is part of the Bioinformatics Web Development tutorial at http://www.cellbiol.com/bioinformatics_web_development/
© cellbiol.com, all rights reserved

Apt

- The apt command is a powerful command-line tool, which works with Ubuntu’s **Advanced Packaging Tool (APT)** performing such functions as installation of new software packages, upgrade of existing software packages, updating of the package list index, and even upgrading the entire Ubuntu system.
- Syntax:
- ***apt-get [options] command***
- or
- ***apt-get [options] install|remove pkg1 [pkg2 ...]***
- or
- ***apt-get [options] source pkg1 [pkg2 ...]***

Ubuntu - Packages

- An Ubuntu package is exactly that: **a collection of items (scripts, libraries, text files, a manifest, license, etc)** that enable you to install a piece of software ordered in such a way that the package manager can unpack it and put it into your system.
- Ubuntu's package management system is derived from the same system used by the **Debian GNU/Linux distribution**.
- The package files contain all of the necessary files, meta-data, and instructions to implement a particular functionality or software application on your Ubuntu computer.
- Many packages use dependencies. **Dependencies are additional packages required by the principal package in order to function properly.**

Update and upgrade

- **update :** This command is used to **synchronize the package index files from their sources again.** You need to perform an update before you upgrade or dist-upgrade.
- ***apt-get update***
- **upgrade :** This command is used **to install the latest versions** of the packages currently installed on the user's system from the sources enumerated in /etc/apt/sources.list. The installed packages which have new packages available are retrieved and installed. **You need to perform an update before the upgrade**, so that apt-get knows that new versions of packages are available.
- ***apt-get upgrade***

install and remove

- **install :** This command is used to install or upgrade packages. It is **followed by one or more package names the user wishes to install**.
- **All the dependencies of the desired packages will also be retrieved and installed.** The user can also select the desired version by following the package name with an ‘equals’ and the desired version number. Also, the user can select a specific distribution by following the package name with a forward slash and the version or **the archive name (e.g. ‘stable’, ‘testing’ or ‘unstable’).** Both of these version selection methods have the potential to downgrade the packages, so must be used with care.
- ***apt-get install [...PACKAGES]***
- **remove :** This is similar to install, with the difference being that it removes the packages instead of installing. **It does not remove any configuration files created by the package.**
- ***apt-get remove [...PACKAGES]***

Check, download, and clean

- **check :** This command is used to update the package cache and checks for broken dependencies.
- ***apt-get check***
- **download :** This command is used to download the given binary package in the current directory.
- ***apt-get download [...PACKAGES]***
- **clean :** This command is used to clear out the local repository of retrieved package files. It removes everything but not the lock file from /var/cache/apt/archives/partial/ and /var/cache/apt/archives/.
- ***apt-get clean***

Some examples of popular uses for the apt utility:

- **Install a Package:** Installation of packages using the apt tool is quite simple. For example, to install the nmap network scanner, type the following:
- ***sudo apt install nmap***
- **Remove a Package:** Removal of a package (or packages) is also straightforward. To remove the package installed in the previous example, type the following:
- ***sudo apt remove nmap***

List all the packages

- This will list all the packages that have been installed using apt.
- ***apt list –installed***
- It will also list the packages that were installed as a dependency. Which means that not only you’ll have the applications you installed, you’ll also have a huge list of libraries and other packages that you didn’t install directly.

Check whether a specific package is installed in Ubuntu

- Since the list of installed packages is a huge one, it would be a better idea to use grep and filter the output for a certain package.
- ***apt list --installed | grep program_name***
- A better way is to use this command:
- ***apt -qq list program_name --installed***
- Both *q options are for quiet mode*. And this way, it only looks for programs that are installed.

Thank You

Linux

Dr. Hatem Yousry
E-mail: Hyousry@nctu.edu.eg