ESSENTIALS OF RED HAT LINUX

Session 1

Introduction to Linux

Objectives

- Identify the need for using open source software
- Identify how the Linux operating system was originated
- Identify the basic principles of Linux
- Identify Red Hat operating system distributions
- Describe the features of Red Hat Enterprise Linux 6.0
- Explain how to log on to the Linux system
- Explain the root user and shadow passwords
- Explain the Desktop environment and Window’s Managers
- Explain the concept of users and groups
- Explain the use of some basic and advanced Linux

Introduction

- An operating system:
  - Is a software program that acts as an interface between a user and a computer.
  - Consists of instructions that are given to the hardware components of the computer to perform specific tasks, such as creating or copying documents and viewing them.
- Examples of operating systems are:
  - Linux, Windows XP, Windows 7, Solaris, Mac and UNIX.

Linux	Derived from the UNIX operating system
	Offers stability, high security, and reliability
	Runs on a variety of computer hardware, such as mobile phones, tablets, and mainframes.
	Source code is freely available to everyone
	Popular distributions are Debian, Fedora, Red Hat, SUSE, and Ubuntu.

Defining Open Source

- Open source implies free software. The software allows users to freely use the source code and customize it to meet individual requirements.

Open source allows users to perform the following tasks:

- Access the source code of the software
- Run the software for commercial or personal use
- Customize the source code to meet specific requirements
- Redistribute copies of the software
- Release the customized software to the public in general

Defining Closed Source

- The proprietary software source code in closed source is not released to end users and is only available in a compiled executable state.
- The end users have to purchase the software to use it for personal or commercial purposes.
- Copying or redistributing of the software can be restricted by the license agreement between the vendor and the end user.
- Owned, developed, and distributed by a single vendor to end users who purchase the software for a license fee.
- Services, such as upgrades, add-ons, and support are defined by the license agreement between the vendor and the end user.

Open Source Operating Systems

Open Solaris	• By Sun Microsystems
Fedora Linux	• Linux operating system by Red Hat • Release cycle of six months
Red Hat Enterprise Linux (RHEL)	• By Red Hat • End users have to pay subscription fee for support
Open SuSE	• By Novell • Contains StarOffice and Word Perfect
Mandrake Linux	• Open to both individual and enterprise users • Contains complete Microsoft office
Debian Linux	• Sponsored by Debian community • Available free of cost
Apache Web Server	• By Apache Software Foundation • Runs on various platforms
Sendmail	• By Sendmail consortium • General purpose Internetwork e-mail routing application
MySQL	• By Sun Microsystems • Database application

Licensing

- An open source license is a copyright license that allows the source code of the software to be available for everyone to use and modify.

General Public License (GPL)

- Allows users to package and distribute the original or modified source code of the software only with other GPL licensed software.

Lesser General Public License (LGPL)

- Allows users to package and redistribute the original or modified source code of the software only with other LGPL Licensed software.
- Also allows the use of proprietary software with LGPL Licensed open source software.

Berkeley Software Distribution (BSD)

- Allows distribution of original source code after modification even if it uses some other software not covered under the BSD license.
- Also allows the use of proprietary software with BSD covered software.

Origin and History of Linux

- Derived from UNIX.
- Was developed on Minimal UNIX (MINIX), a version of UNIX, using GNU Not UNIX's (GNU) compiler.
- The name Linux was coined by Linus Torvalds in 1991.
- The Linux source code was made available to everyone on the Internet for study and modification.
- Official mascot for Linux is the Linus Penguin, Tux.

Basic Principles of Linux

Stability

- Stable operating system
- Does not need to be restarted periodically
- Provides adequate protection against freezing up or slowing down
- Modular by design
- Multi-user operating system
- Large developer base as it is open source and hence bugs are fixed relatively faster

Security

- File system security
- Firewall security
- Run levels security
- Services
- Security Enhanced Linux (SELinux)
- Discretionary Access Control (DAC)

Advantages of Linux Operating System

Advantages of Linux operating system are:

- Reliability
- Backward compatibility
- Simple installation and upgrade process
- Low cost of ownership
- Support for legacy devices
- Graphic User Interface (GUI)
- Excellent security features
- Support for development libraries
- Support for high user loads

Features of Linux Operating System

- Multiprogramming
- Multitasking
- Virtual memory
- Shared libraries
- POSIX compliance
- Samba
- Network Information Service (NIS)
- Office suites
- Data archiving utilities
- Apache Web Server
- Licensing
- Includes text editors, browsers, and scientific applications

Red Hat Operating System Distributions

Red Hat provides different variants for different types of users. Users are divided into:

Business users

- Red Hat provides the Red Hat Enterprise Linux product suite for business users.
- Users are required to pay a product fee for the software.
- Changes in the operating system happen in a slow pace so that there are no frequent updates.

Other users

- Use Linux for experimental purposes.
- Users do not pay the license fee. They download it from the Internet and use it free of charge.
- Fedora is an example of a freely downloadable operating system.

Linux Variants [1-2]

The various Linux variants provided by Red Hat are:

- **Red Hat Enterprise Linux Suite**
  - Runs on multiple hardware architecture such as IBM PowerPC, AMD64, Intel EM64T, and Intel x86/x64
  - Operating systems of Red Hat Enterprise Linus suite are:
    - Red Hat Enterprise Linux AS
    - Red Hat Enterprise Linux ES
    - Red Hat Enterprise Linux WS
- **Red Hat Desktop**
  - Used for small and medium business groups, supporting a single CPU
  - Compatibles with Red Hat Enterprise Linux AS, Red Hat Enterprise Linux ES, and Red Hat Enterprise Linux WS
  - Suitable for document processing, Web browsing, software development, and instant messaging

Linux Variants [2-2]

- **Fedora Core**
  - General-purpose operating system similar to Windows NT and Macintosh
  - Red Hat sponsored project
  - Can be freely downloaded from Websites

- **Red Hat Enterprise Linux versions released so far are:**
  - Red Hat Enterprise Linux 2.1 (Panama)
  - Red Hat Enterprise Linux 3 (Taroon)
  - Red Hat Enterprise Linux 4 (Nahant)
  - Red Hat Enterprise Linux 5 (Tikanga)
  - Red Hat Enterprise Linux 6 (Santiago)

Desktop Environment and Windows Manager [1-2]

- The X Window System:
  - Provides GUI for Linux.
  - Is portable and is a network transparent client/server interface between the hardware and desktop environment.
- Windows Manager customizes the desktop according to the user requirement and the Desktop Environment is a user interface that runs on the Windows Manager.
- Desktop Environment:
  - Used to create a common graphical user environment by combining various X clients.
  - The two types of desktop environments provided by Red Hat Enterprise Linux 5.0 are:
    - Gnome
    - KDE

Desktop Environment and Windows Manager [2-2]

- Windows Manager:
  - Used to control the working of the desktop
  - Helps in controlling moving, hiding, resizing, iconizing, and closing windows
- The different types of Windows Managers are:
  - **kwin**:
    - Default Window Manager for KDE that supports custom themes
  - **metacity**:
    - Default Window Manager for Gnome that supports custom themes
    - Requires installation of the metacity package
  - **mwm**:
    - Motif Window Manager is a stand-alone Window Manager
    - Requires installation of the open motif package
  - **twm**:
    - Tab Window Manager can be used as a stand-alone Window Manager as well as with the desktop environment
    - Available with X11R7.1 release

Logging into and Logging out of the Linux System

- The user has to be authenticated to use a Linux system. The authentication is done through a login process.
- Log in to the system console is through one of the following ways:
  - Text based login
  - GUI based login
- For a text based system, a command prompt ending with a $ sign is displayed.
- Example:

```
localhost login: student
Password:
Last login: Mon Sep 15 14:
30: 46
```

- Login session can be ended by typing the logout command at the shell prompt.
- This logs the user out of the system and a new login prompt appears on the terminal.
- Example:

```
student@localhost~]$  logout
localhost login:
```

Users, Groups, and Root Users

- The accounts that exist on the system are termed as users.
- The concept of Groups in Linux allow groups of users in an organization to share, read, write, and execute common files amongst them.
- Every user and group logging into the system is uniquely identified using unique numerical identification numbers called userid and groupid.
- The user who creates the file is called the owner and group owner of the file. Each file is assigned different read, write, and execute permissions for individual users and groups.
- The access permissions for the file are assigned by the root user or the file owner.
- Root user:

  Is a special administrative account which has unrestrictive access to all files, devices, and programs in the system.

  Is also known as the super user and guards the system against accidental damages.

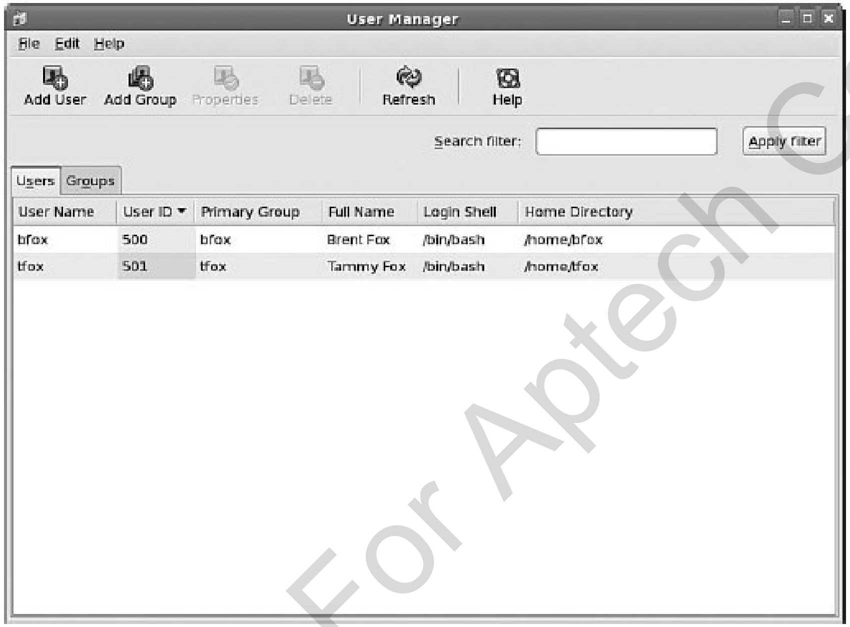
Managing Users [1-2]

- When a user logs into the system, Linux automatically places the user in a directory called the home directory.

  Pathname: **/usr/<username>** or **/home/<username>**.
- The home directory is decided by the system administrator at the time of creating a user account, and the details are stored in the file **/etc/passwd**.
- When a new user is added to the system, a private user group, having the same name as the user name, is create with the same name as the user's name.
- A graphical program, called user manager, is included in the Red Hat Enterprise Linux for managing users and groups.

Managing Users [2-2]

- User Manager is started by selecting the users and Groups option from the administration menu. This invokes the user manager window.
- All the existing users are listed in the users tab of this window as shown.

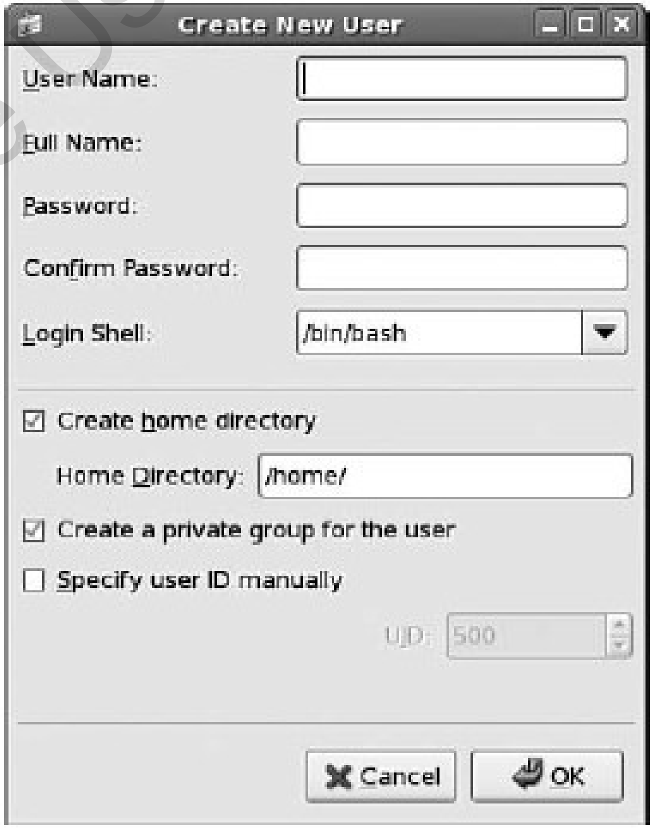


Search filter:

User Name	User ID	Primary Group	Full Name	Login Shell	Home Directory
bfox	500	bfox	Brent Fox	/bin/bash	/home/bfox
tfox	501	tfox	Tammy Fox	/bin/bash	/home/tfox

List of Existing Users

- Click the **Add User** button in the 'User Manager' window to add a new user.



User Name:

Full Name:

Password:

Confirm Password:

Login Shell: /bin/bash

☑ Create home directory

Home Directory: /home/

☑ Create a private group for the user

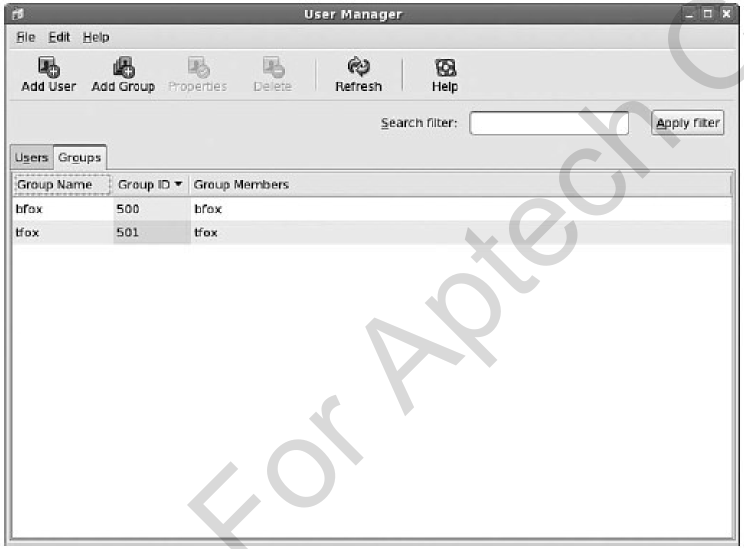
□ Specify user ID manually

UID: 500

Adding a User

Managing Groups

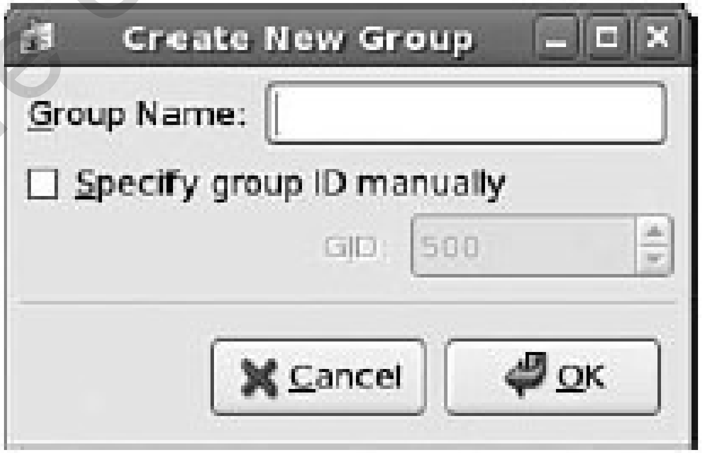
- Every user is assigned to a default private group with the same name as the user. The user can also become a member of other groups.
- The graphical program to manage the groups can be invoked by selecting the Users and Groups option from the administration menu. The user manager window, with the existing groups being listed in the Groups tab, is displayed as shown.



Group Name	Group ID	Group Members
bfox	500	bfox
tfox	501	tfox

List of Existing Groups

- Click the **Add Group** button in the 'User Manager' window to add a new group. This invokes the create new Group window as shown.



Group Name:

□ Specify group ID manually

GID: 500

Adding a Group

Basic Linux Commands

- Linux is a command line operating system.
- The commands in Linux are case-sensitive and usually in lower case.
- The general syntax for a command is:

  `command [options]`

- **`date`** command: Used to display current system date and time.

  Syntax:

  `student localhost ~]$ date [options]`

- The following code snippet shows the use of date command with different options:

```
[student@localhost ~]$ date "+%T"
21:36:42
[student@localhost ~]$ date "+%y"
08
```

`date` Command

- The `date` command can be used only by the system administrator to change the system date and time.
- The common options that can be used with the `date` command are:

Option	Function
%m	Displays month of the year (in digits)
%d	Displays day of the month (in digits)
%y	Displays year (last two digits)
%D	Displays date as mm/dd/yy
%H	Displays hour (00 to 23)
%M	Displays minutes (00 to 59)
%S	Displays seconds (00 to 59)
%T	Displays time as HH:MM:SS

Options of the `date` Command

who Command [1-2]

- Used to display the names of all the users logged on to the system.

  Syntax:

  ```
  [student@localhost ~]$ who [options]
  ```

- The common options available with the who command are:

Option	Function
-b	Indicated the most recent startup time and date
-l	Lists any login process
-H	Displays a header
-q	Prints only the login names and the number of users logged in

Options of the who Command

who Command [2-2]

- The following code snippet shows the use of the who command:

```
[student@localhost ~]$ who
student pts/0 2008-03-26 17:36(:0.0)
```

- The pts in the output denotes a remote terminal. A remote terminal is a terminal connected from a machine other than the server.
- The output of the who command is explained in the following table:

Column#	Description
1	Displays a login name
2	Displays a terminal type and number
3	Specifies date and time of logging in
4	Specifies the remote host name of the terminal of the users who have not logged in to the server

Output of the who Command

`man` Command

- Used to display pages from the Linux reference manual installed along with the Linux operating system.

  Syntax:

  ```
  [student@localhost ~]$ man [-] [-k keywords] topic
  ```

- The arguments to the `man` command are as follows:

Argument	Function
-	Displays content of the manual without pausing
-k keywords	Searches for the keywords specified within the available manuals
topic	Displays the manual for the command typed in

Arguments of the `man` Command

- The following code lists the help information to the `ls` command:

```
[student@localhost ~]$ man ls
```

Reading Directories and Files: `ls` Command

- Displays the names of files and sub-directories within a directory.

  Syntax:

```
[student@localhost ~]$ ls [options]
```

- The common options available in the `ls` command are:

Option	Function
-a	Lists all the files including hidden files
-F	Displays the file type along with the name
-R	Displays the contents of specified directory and sub-directories
-r	Displays files and sub-directories in the reverse order
-S	Lists all files sorted by file size
-A	Displays hidden files and files beginning with ‘.’
-l	Displays a detailed list of files and directories

Options of the `ls` Command

Access Permissions in Linux

read permission (r)

This means that the contents of the file can be read using the `cat` command.

For directories it means that the content of the directory can be listed with the `ls` command.

write permission (w)

This means that the file can be modified and saved.

For directories, it means that files can be created in that directory by the user.

execute permission (x)

This means that files can be executed by the shell when its name is typed at the command prompt.

For directories, the user has the ability to traverse its tree in order to access files or subdirectories.

The files cannot be seen until the permission is set.

Reading Directories and Files: `ls -l` Command

- The following code snippet displays a detailed list of files and directories using the `–l` option with the `ls` command.

```
[student@localhost ~]$ ls -l
total 2
-rw-rw-r-- 1 student student 134 Jun 21 00.18 Desktop
drwxr-xr-x 1 student student 10 Jun 21 13.18 X
```

- The various column outputs of the ls –l command are:

Option	Function
1	Specifies file type and File Access Permissions (FAP)
2	Specifies symbolic links, that is a file that points to another file
3	Specifies the name of the file owner
4	Specifies the name of the group owner
5	Specifies the file size in bytes
6, 7, 8	Specifies the last file modification date and time
9	Specifies the file name

Options of the `ls -l` Command

Reading Directories and Files: dir Command

- The dir command works like the default ls command, as it lists the files in a sorted manner. The following code snippet shows the use of the dir command.

```
[student@localhost ~]$ dir
News axhome nsmail search
author.msg documents reading vultures.msg
auto mail research
```

Reading Directories and Files: `vdir` Command

- The `vdir` command works like the `ls -l` option and presents a long listing by default. The following code snippet demonstrates the use of the `vdir` command.

```
[[student@localhost ~]$ vdir
total 10
drwxr-xr-x 2 bball bball 1024 Nov 12 08:20 News
-rw-rw-r-- 1 bball bball 4766 Nov 12 07:41 author.msg
drwxrwxr-x 2 bball bball 1024 Nov 5 10:04 auto
drwxrwxr-x 3 bball bball 1024 Nov 12 13:54 axhome d
rwxrwxr-x 2 bball bball 1024 Nov 12 15:13 documents
drwx------ 2 bball bball 1024 Nov 12 14:02 mail
drwx------ 2 bball bball 1024 Sep 15 01:57 nsmail
drwxrwxr-x 2 bball bball 1024 Oct 29 20:28 reading d
rwxrwxr-x 5 bball bball 1024 Nov 5 10:03 research
-rwxrwxr-x 1 bball bball 200 Oct 24 13:24 search
```

Reading Directories and Files: `tree` Command [1-2]

- Used to list the contents of the directory in a tree like format.

  Syntax:

  ```
  [student@localhost ~]$ tree [options]
  ```

- The common options available with the `tree` command are:

Option	Function
-a	Displays all files, except the hidden files
-d	Lists the directories only
-f	Displays the full path prefix for each file
-p	Displays the file permissions for each file
-s	Displays the size of each file along with the name
-r	Sorts the output in reverse alphabetical order
-L level	Specifies the maximum display depth of the directory tree

Options of the `tree` Command

Reading Directories and Files: `tree` Command [2-2]

- The following code shows the use of the tree command:

```
[student@localhost ~]$ tree
.
|-- projects
|  |-- current
|  `-- old
|  |-- 1
|  `-- 2
`-- trip
`-- schedule.txt
4 directories, 3 files [
student@localhost ~]$
```

Reading Directories and Files: `touch` Command

- The `touch` command is used to:
  - Create a file
  - Update the file’s modification date

Syntax:

```
[student@localhost ~]$ touch [Options] { File(s)}
```

- The common options available with the `touch` command are:

Option	Function
-a	Changes the access time of the file specified. Does not change the modification time unless –m is also specified.
-c	Does not create a new file if the file already exists.
-f	Attempts to force the execution of touch even if there are read and write restrictions on a file.
-m	Changes only the modification time.
-r file	Uses the access and modification times of the file.

Options of the `touch` Command

Reading Directories and Files: `cat` Command

- Used to display the contents of a file on the screen or store it in another file.
- Also used to create, combine, overwrite, or append files.

Syntax:

```
[student@localhost ~]$ cat filename [options]
```

- The common options available with the `cat` command are:

Option	Function
-n	Precedes each line with a line number
-u	Does not buffer the output
-e	Displays a character $ at the end of each line
-b	Omits line numbers from blank lines
-r file	Displays tabs in the output

Options of the `cat` Command

Reading Directories and Files: `more` Command

- Used to display the contents of a large file in a single screen.

  Syntax:

  ```
  [student@localhost ~]$ more [options]
  ```

- The common options available with the `more` command are:

Option	Function
-c	Clears and redraws the screen before displaying
-d	Displays error message if an unknown command is used
-i	Performs case insensitive pattern matching in searches
+num	Specifies the starting line number
-u	Ignores backspace and underscores

Options of the `more` Command

Reading Directories and Files: `exit` and `shutdown` Commands

- The `exit` command is used to terminate the Linux session.
- The `shutdown` command is used to shutdown the Linux operating system.

Syntax:

```
[student@localhost ~]$ shutdown [options]
```

- The common options available with the `shutdown` command are:

Option	Function
-h	Halts after shutdown
-r	Reboot after shutdown
-c	Cancels a pending shutdown

Options of the `shutdown` Command

Advanced Linux Commands: df Command [1-2]

- The df command stands for Disk Free.
- Used to display the amount of space used and the amount of free disk space available on the currently mounted file systems.

  Syntax:

  ```
  [student@localhost ~]$ df [option] [File_name]
  ```

- The common options available with the df command are:

Option	Function
-h	Displays the size in megabytes and gigabytes and appends the data with M and G respectively
-k	Displays the data in 1KB blocks
-i	Displays the inode usage
-T	Adds type of each file system to the report
-help	Displays a brief help message

Options of the df Command

Advanced Linux Commands: `df` Command [2-2]

- The following code snippet displays the sizes in an easy to read format.

```
[student@localhost ~]$ df -h
Filesystem      Size    Used    Avail   Use%    Mounted on
/dev/hda2       28G     7.6G    19G     29%     /
tmpfs           252M    0       252M    0%      /dev/shm
/dev/hda1       464M    37M     403M    9%      /boot
/dev/hda3       8.3G    429M    7.5G    6%      /var
nfs6: /home     520G    461G    60G     89%     /home
```

Advanced Linux Commands: `ps` Command [1-2]

- The `ps` or Process Status command displays the processes that are currently running on the Linux system.

  Syntax:

  ```
  [student@localhost ~]$ ps [option]
  ```

- The common options available for the `ps` command are:

Options	Function
-f	Generates a full listing of all the processes
-A	Displays information of all processes
-a	Displays information about all frequently requested processes
-e	Displays information about every process that is currently running

Options of the `ps` Command

Advanced Linux Commands: `ps` Command [2-2]

- The following code snippet lists the currently running processes.

```
[student@localhost ~]$ ps
PID TTY                TIME CMD
3511 pts/1             00:00:00
bash 3514 pts/1        00:00:00 ps
```

- The `ps` command displays the process ID, the terminal associated with the process, the CPU time, and the executable shell name.

Advanced Linux Commands: `kill` Command

- The `kill` command is used along with the `ps` command. It stops the execution of one process.

  Syntax:

  ```
  [student@localhost ~]$ kill [options] [pids]
  ```

- The common options available for the kill command are:

Option	Function
-a	Kills all the processes having the name specified
-l	Lists all signals
-p	Prints the process ID of the named processes without sending the signal
-s	Specifies the signal number or name of the signal

Options of the `kill` Command

Advanced Linux Commands: `uname` Command [1-2]

- Used to display the system information.

  Syntax:

  ```
  student@localhost ~]$ uname [options]
  ```

- The common options available for the `uname` command are:

Option	Function
-a	Prints all the basic information currently available from the system
-r	Prints the operating system release level
-s	Prints the operating system name
-m	Prints the machine hardware type
-p	Prints the machine’s processor type
-v	Prints the operating system version

Options of the `uname` Command

Advanced Linux Commands: `uname` Command [2-2]

- The following code snippet prints the basic information of the system along with the OS release level and the OS version.

```
 [student@localhost ~]$ uname -arv
Linux localhost.localhostdomain 2.6.18-8.el5 #1 SMP Fri Jan 26
14:14:15 EST 2007
i686 i686 i286 GNU/Linux
 [student@localhost ~]$
```

Advanced Linux Commands: `tty` Command

- Linux treats even the terminals as files. `tty` (teletype) command is used to display the names of these terminals.

Syntax:

```
[student@localhost ~]$ tty
/dev/pts/0
```

- The terminal filename is `tty01` and is resident in the `/dev` directory.

Printing Commands: `lpr` Command

- This command sends the job to the print queue.

  Syntax:

  ```
  [student@localhost ~]$ lpr [options] filename
  ```

- The common options available with the `lpr` command are:

Option	Function
-P destination	Name of the printer on which to print
-# number	Prints the specified number of copies
-T title	Prints a title on the banner page of the output
-q	Holds a job for printing

Options of the `lpr` Command

- The following code snippet prints 5 copies of the file report.txt on a printer named accounting.

```
[student@localhost ~]$ lpr -P accounting -#5 report.txt
```

Printing Commands: `lpq` Command

- Used to check the print spool queue for the status of print jobs. It displays user name, position in the queue, filenames, job number, and total file size (in bytes) for each job.

  Syntax:

  ```
  [student@localhost ~]$ lpq [option] printer
  ```

- The common options available with the `lpq` command are:

Option	Function
-P destination	Displays information about the printer or class of printers
-a	Reports jobs on all printers
-U username	Specifies an alternate username for the user
-E	Uses encryption to connect to a print server

Options of the `lpq` Command

- The following code snippet displays information about the printer lp0.

```
[student@localhost ~]$ lpq -P lp0
```

Printing Commands: `lprm` Command

- Used to remove a job from a queue.

  Syntax:

  ```
  [student@localhost ~]$ lprm [options]
  ```

- The common options available with the `lprm` command are:

Option	Function
-P destination	Displays information about the printer or class of printers
-h server [:port]	Specifies an alternate server
-U username	Specifies an alternate username

Options of the `lprm` Command

- The following code snippet removes request ID 385 from destination.

```
[student@localhost ~]$ lprm -P kill 385
```

Running a Process in the Background

- When a process starts in the background, a new bash sub shell is created.
- The & is the shell's operator used to run a process in the background.
- To start a process in the background, terminate the process with an & symbol as shown in the following code snippet.

```
[student@localhost ~]$ sort file1.txt file2.txt &
[1] 2975
```

- The shell immediately returns a number which is the PID of the invoked command.
- The prompt is returned, and the shell is ready to accept another command, even though the previous command has not been terminated.

Summary

- An operating system is a software program that acts as an interface between a user and a computer.
- Linux was developed on Minimal UNIX (MINIX), a UNIX version, using GNU Not UNIX’s (GNU’s) C compiler.
- The X Window System is a Graphical User Interface in Linux, which provides the foundation for Window Managers such as GNOME and KDE.
- When a new user is added to the system, a private user group is automatically created with the same name as the user’s name.
- The root user is a special administrative account, that has unrestricted access to all the files, devices, and programs in the system.
- The basic Linux commands include: `date`, `who`, `man`, `dir`, `vdir`, `tree`, `cat`, and `more` commands for reading directories and files command, `exit` command, and `shutdown` command.
- The advanced Linux commands include: `yydf`, `ps`, `kill`, `uname`, `tty`, printing commands such as `lpr`, `lpq`, `lprm`.