

ESSENTIALS OF RED HAT LINUX

Session 4

Using Command-Line Interface



Objectives



- Explain the process of installing Linux operating system
- Understand the command line interface
- Access command line from X Window System
- Explain the booting process using GRUB
- Explain different run-levels
- Explain basic Linux commands
- Identifying current users
- Using Filters and Pipe

For Aptech Centre Use Only

Understanding Command Line from X Window System [1-2]



- X Window System is based on the client-server architecture that includes two primary components – X server and X client.
- X server is a program that enables various applications, called the X clients, which run on a Linux system to display their GUI components.
- X clients send the display request to the X server.
- Other important functions performed by the X server are:

Displays the errors that occur when the X client makes a request for display.

Controls the I/O devices.

Directs the keyboard and mouse input to the desired X clients or X events on the network.

Manages the windows displayed on the screen by creating, mapping, and closing the windows.

Understanding Command Line from X Window System [2-2]



- Key functions performed by X client are:

Requests the server for various services

Accepts events in response to the requests from server

Accepts errors messages from the server

- The potential of the client-server architecture is realized when it is used across the network on multiple computers.
- Advantages of using the client-server architecture of X Window System are:
 - Install the client and server software on different computers.
 - Port the X client applications to other platforms.
 - Multiple X servers running on more than one computer at a time.

Accessing Command Line from X Window System



- Terminal: An instance of a command-line shell accessed from the GUI mode.
- A user can:

Access command-line in the GUI mode by using a terminal.

Create multiple terminals as per the requirement to execute commands.

- To create a terminal, select, from the GNOME desktop:



- Invoke a new instance of the GNOME terminal by executing the following command from a GNOME terminal:

```
# gnome-terminal
```

Working with Linux Shell [1-3]



- Shell: A term used for the interactive user interface of an operating system. It is the layer of programming that understands and executes the commands.
- Shell script: A file that contains the sequence of commands to perform a particular operation on the computer.
 - Execution of such shell scripts saves effort and time.
 - Allows users to handle a system with flexibility and to automate tasks in the form of shell scripts.

In a multiuser environment, the shell has to isolate the user interface from the kernel.

- The shell acts as an interface between the user and the kernel.

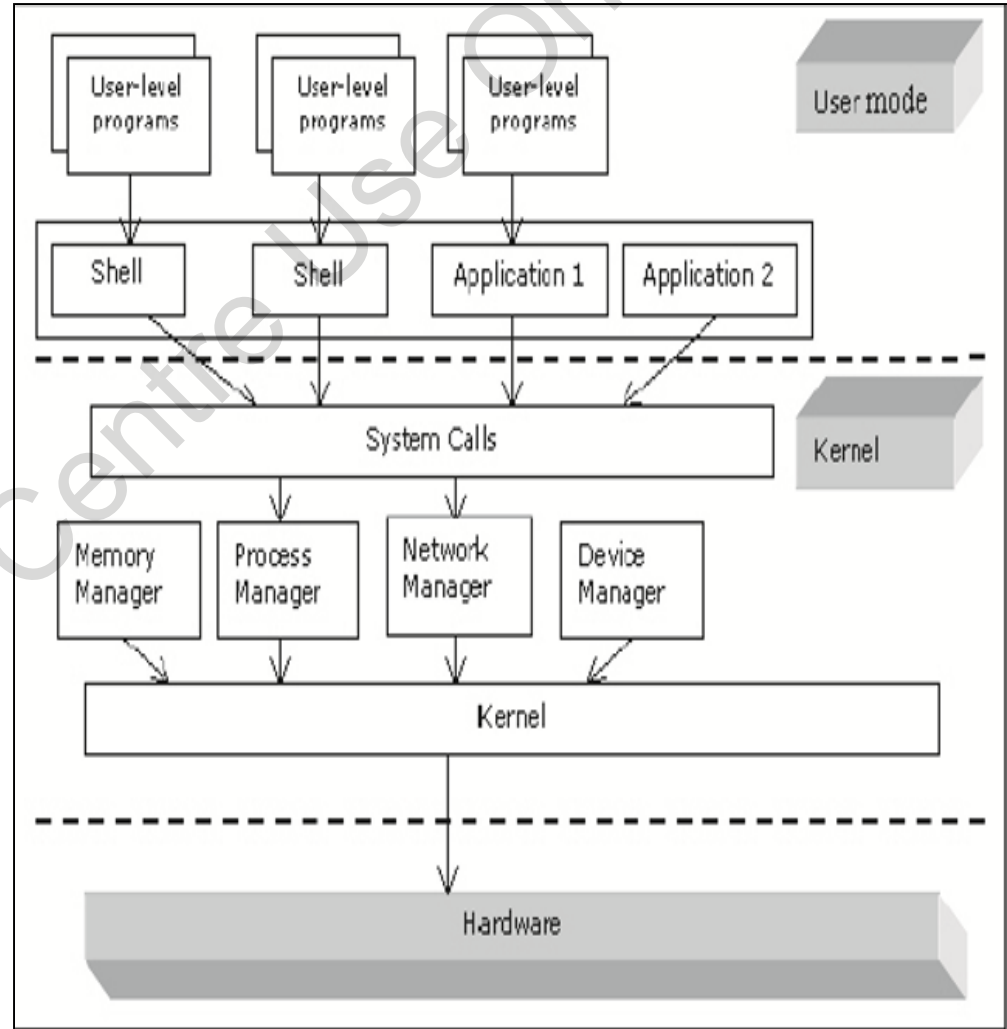
The kernel is in charge of:

- The user interaction with the hardware.
- Scheduling tasks.
- Allocating resources to the running processes.

Working with Linux Shell [2-3]



- The figure is the complete representation of the functioning of a Linux system that explains the importance of shell and its role in the system.
- The shell interacts with the kernel through the system calls – a set of routines that allow an application to access kernel services.
- Different applications or daemons, manage each of the services.



Functioning of the Linux System

Working with Linux Shell [3-3]



- Daemons run at the kernel level and are always active.
- Shells are called command processors or command interpreters in MS-DOS. In a Microsoft Windows platform, `cmd.exe` is a shell.

Login shells

- Interactive shells that set up a certain pre-defined environment when a user logs on.
- Set up a pre-defined user environment before being launched.

Non-login shells

- Interactive shells that do not set any pre-defined environment and inherit the environment of the parent shell under which they are created.
- Created when users execute a command or a shell script, invoke a graphical terminal or run the `bash` or `sh` command.

GRUB Boot Loader



- The booting process is the first process that is executed when a Linux system is started:

1

- The processor searches for the BIOS program in the memory and executes it.

2

- BIOS runs the POST program that checks the system hardware by verifying the hardware configuration information.

3

- BIOS checks the system memory for any hardware-related errors and searches for a bootable device. Specify the booting sequence by modifying the booting sequence information in the BIOS setup program. The boot sequence can be specified as CD-ROM, C, A.

4

- BIOS locates and invokes the boot loader by passing the control of the system to the IPL, which is present in the MBR.

5

- IPL loads the boot loader into the memory of the system. The boot loader takes the control of the system from the IPL when it is loaded into the memory of the system.

Understanding GRUB Boot Loader [1-3]



- Boot loader: A program that loads the operating system into the system memory while booting.
- Made up of two parts:

Primary boot loader

- Part of the boot loader that is loaded into the memory of the system by BIOS at the start of the booting process.

Secondary boot loader

- Part of the boot loader that is loaded into the memory of the system by the primary boot loader.

- Secondary boot loader is either directly loaded into the memory or stage 1.5 boot loader.
- Boot partition: A partition on the hard disk that contains the files required for booting.
- This interface is used to:
 - Customize the booting process by modifying the boot-loader commands.
 - Specifying the parameters to be passed to the kernel at the time of booting.



Understanding GRUB Boot Loader [2-3]

- Default boot loader in RHEL.
- Enables to select an operating system from the various operating systems installed on the computer at the time of booting.
- Displays a menu interface that lists the various operating systems installed on the system.
- GRUB offers the following advantages:

Direct and Chain-Loading Booting
LBA Mode
Command-Based, Pre-Operating System Environment
ext2 Partition Support
Menu Interface
Flexible Command-Line Interface
Automatic File Decompression
Network Booting

Understanding GRUB Boot Loader [3-3]



- The various interfaces provided by GRUB are:

Menu Interface

- Appears by default, if the installation program configures GRUB, called the graphical splash screen.
- Displays a list of various operating systems that can be installed using GRUB in the alphabetical order.
- User can select a particular operating system from the list.

Menu-Entry-Editor Interface

- Used to edit the GRUB commands, required to be executed while booting.
- To switch to this interface, press the 'e' key on the menu interface.
- Shows information, required by the GRUB at the time of booting.
- Commands to modify the GRUB configuration file are displayed in the menu-entry-editor interface.

Command-Line Interface

- Used to execute the GRUB commands by providing a command-line.
- To switch to the command-line interface, press the 'c' key on the menu-entry-editor interface or the menu interface.
- Type a GRUB command in this interface, press the ENTER key to execute the command.

GRUB Commands and its Description [1-2]



- The table describes the various GRUB commands.

Command	Description
boot	Boots the operating system or the chain loader that has been specified.
chainloader </path/to/file>	Loads the specified file as a chain loader. If the file is located on the first sector of the specified partition, use the block list notation instead of the filename.
Displaymem	Displays the amount of RAM that a system has prior to booting.
initrd </path/to/initrd>	Specifies an initial RAM disk, initrd, to be used while booting. An initrd is necessary when the root partition is formatted using the ext3 file system.
install <stage-1><install-disk><stage-2><config-file>	Installs GRUB in the MBR. The parameter, <stage-1>, signifies a device, partition or file where the primary boot loader is stored. The parameter, <install-disk>, specifies the disk where the primary boot loader should be installed. The parameter, <stage- 2>, specifies the location of the secondary boot loader to the primary boot loader. The parameter, <config-file>, specifies the path of the GRUB configuration file.

GRUB Commands and its Description [2-2]



Command	Description
<code>kernel</code> <code></path/to/kernel><option-1> <option-N></code>	Specifies the kernel file to be loaded while booting the operating system. The parameter, <code></path/to/kernel></code> , specifies the path of the root command. The parameter, <code><option-1> <option-N></code> , specifies the options for the Linux kernel, such as <code>root=/ dev/ hda5</code> , to indicate the device on which the root partition of the system is located.
<code>root</code> <code>(<devicetype><devicenum ber>, <partition>)</code>	Configures and mounts the root partition, such as <code>(hd0, 0)</code> , for GRUB.
<code>rootnoverify(<devicetype><devicenum ber>, <partition>)</code>	Configures the root partition for GRUB but does not mount the partition.

Identifying Different Runlevels [1-2]



Runlevels are the various modes in which a user can work on a Linux system.

Help in assigning privileges such as access to the network or GUI.

Helps reduce the potential attack surface area of a Linux system, as they do not allow unnecessary services to run.

Each runlevel executes certain services to enable the mailing and Web services.

Services have symbolic links that are present in the rc directory that correspond to the runlevels.

Identifying Different Runlevels [2-2]



Each symbolic link has a name that begins with the letters K or S.

- K specifies that the service is terminated.
- S specifies that the corresponding service is started in a runlevel.

The scripts that correspond to a particular runlevel are executed when the user switches to that runlevel.

Each script is associated with a service that must run in a particular runlevel.

SysV init runlevels: Used to control the services that are started or halted by the init program at the time of initializing a runlevel.

To boot the system in a particular runlevel other than the default runlevel:

- Modify the GRUB configuration file.
- Reset the value of the default runlevel when user boot the system.

Runlevels Description



- The table describes the various runlevels defined by Linux.

Runlevel	Description
0	Halts the Linux system.
1	Starts the Linux system in the single-user text mode.
2	Starts the Linux system in the full multiuser text mode without enabling a networking service. In this case, the multiple users can log on to the system at a time and work in the command-line interface mode.
3	Starts the Linux system in the full multiuser text mode with the networking service enabled.
4	Allows the user to define this runlevel.
5	Starts the Linux system in the full multiuser graphical mode.
6	Reboots the Linux system.



Working with Runlevels [1-3]

- Several utilities are offered by Linux, such as:

Chkconfig Utility
The ntsysv Utility
Services Configuration Utility
Service Command

- Chkconfig Utility:** Adds, removes and manipulates the various services that should be started in a particular runlevel.

To check the status of service use command **syntax:**

```
# /sbin/chkconfig --list <service_name>
```

To turn off a service for a particular runlevel, use the command **syntax:**

```
/sbin/chkconfig --level <level_number> <service_name>  
<on|off|reset>
```



Working with Runlevels [2-3]

- The `ntsysv` Utility: Configures the current runlevel by default and provides an interactive text-based interface. To start the `ntsysv` utility, use the command, `ntsysv`.

The screenshot shows a terminal window titled 'Caesor@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal output shows the command `who` being executed, resulting in the following table:

Caesor	tty1	2011-04-29 12:16	(:0)
Caesor	pts/0	2011-04-29 12:16	(:0.0)

The prompt `[Caesor@localhost ~]$` is shown at the end of the output.

Output of `ntsysv` Utility



Working with Runlevels [3-3]

- Services Configuration Utility: Configures various services in different runlevels by using an interactive graphical interface.
- Service Command: Starts or stops stand-alone services that run independent of other services.
 - The service command can be used with various arguments to customize its action. The **syntax** of the *service* command is `service <Service_name> <arguments>`

Argument	Function
Start	Starts a particular service.
Stop	Stops a particular service.
Reload	Updates the configuration file of the service and implements the changes to the service without stopping the service.
Restart	Restarts a particular service that is already running.
Condrestart	Restarts the service when a particular condition is met.
Status	Shows the status of a particular service at a particular time.



Basic Commands [1-3]

- RHEL provides some basic commands and command line expansions that help in configuring the system.
- The `date` Command – Used to display the current system date and time:
 - The **syntax** of the date command is `# date .`
 - The options available with the data command are listed in the table.

Option	Description	Option	Description
%m	Displays the month (in digits)	%S	Displays the seconds (00 to 59)
%d	Displays the day (in digits)	%T	Displays the time as HH:MM:SS
%y	Displays the year (last two digits)	%a	Displays the abbreviated weekday (Sun to Sat)
%D	Displays the date as mm/dd/yy	%h	Displays the abbreviated month (Jan to Dec)
%H	Displays the hour (00 to 23)	%r	Displays the time in the A.M./P.M. notation
%M	Displays the minutes (00 to 59)		



Basic Commands [2-3]

- The `clear` Command –Used to clear the terminal screen. RHEL allows users some measure of screen manipulation with the following commands:

```
tput clear
```

Clears the standard output device and positions the cursor at the top left corner of the screen.

```
tput cup
```

Followed by the screen coordinates, positions the cursor at the specified row and column.

```
tput smso
```

Sets the screen to reverse video.

```
tput rmso
```

Sets the screen back to normal.

```
tput blink
```

Displays a blinking output.

```
tput reset
```

Resets the screen back to the default settings.



Basic Commands [3-3]

- Some of the different command-line expansions are:

~ (Tilde)

- The working directory of the user after logging on to the Linux system is the user's home directory.
- The path for the user home directory can be specified with the tilde (~) sign.

\$

- Used to substitute the value of a variable in a command line.

{ }

- Used to create string pattern of strings.
- Any command, where the {} are used, runs once for each pattern mentioned inside the {}.

Identifying the Current Users Working on the System



- The `who` command displays the names of users that are presently logged on to the system.
- Contains the terminal file name and the date and time the user logged on.
- Possible for multiple users to simultaneously log on from the server using multiple terminals called virtual consoles.

A screenshot of a Linux desktop environment. A terminal window titled 'Caesor@localhost:~' is open. The terminal shows the output of the 'who' command. The output lists two users: 'Caesor' on 'tty1' and 'Caesor' on 'pts/0', both logged in on '2011-04-29 12:16' from '(:0)'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The desktop background is light blue with a taskbar at the bottom showing 'Caesor@localhost:~' and some system icons.

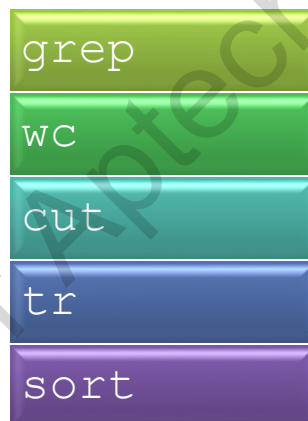
```
[Caesor@localhost ~]$ who
Caesor  tty1          2011-04-29 12:16 (:0)
Caesor  pts/0        2011-04-29 12:16 (:0.0)
[Caesor@localhost ~]$
```

Output of the `who` Command

Using Filter



- Used to restructure the output based on a certain condition.
- Helps in taking the input, modifying it as per requirement, and then passes it as output.
- The functions of filters are:
 - Takes its input from the standard input file
 - Filters the input
 - Sends the output to the standard output file
- RHEL has filters that are used to work on data in an effective way, such as:



The grep Filter [1-4]



- Used to search and display the lines in a file with particular pattern of characters.
- The pattern that is searched in a file is the regular expression.
- The `grep` filter can be used only by specifying a regular expression.

Syntax:

```
# grep [options] pattern [filename]
```

- Regular expressions: Used to specify simple patterns of characters to highly complex one.
- The table lists the simple regular expressions:

Regular	Expression Pattern
'A'	The character A
'F'	The character F
'New'	The word New

The grep Filter [2-4]



- Characters can be used to specify complex regular expressions.
- The table lists the complex regular expressions.

Character	Use	Example	Description
[]	Matches any one of a set of characters	<code>grep "New [abc]"</code>	It specifies the search patterns as 'Newa', 'Newb', or 'Newc'.
[] With Hyphen	Matches any one of a range of characters	<code>grep "New [a-c]"</code>	It specifies the search patterns as 'Newa', 'Newb', or 'Newc'.
^	The pattern following the ^ (caret) must occur at the beginning of each line	<code>grep "^New [abc]"</code>	It specifies the search patterns as 'Newa', 'Newb', or 'Newc', which must occur at the beginning of the line.
^ within []	The pattern that contains the word followed by any character other than the ones specified within brackets	<code>grep "New [^a-c]"</code>	It specifies search patterns that contain the word 'New', followed by any character other than 'a', 'b', or 'c'.

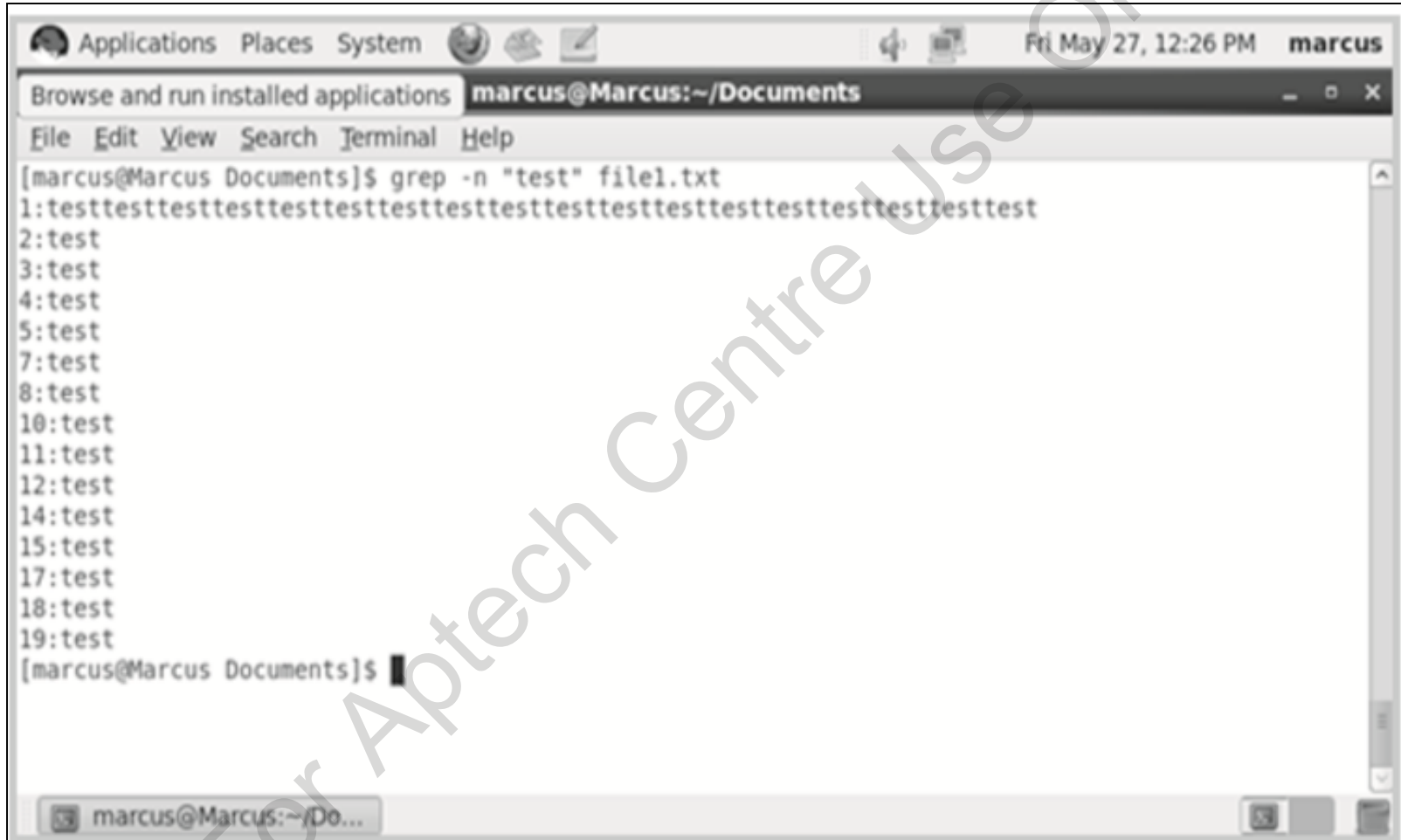
The grep Filter [3-4]



Character	Use	Example	Description
\$	The pattern preceding the dollar sign must occur at the end of each line	grep "New[abc] \$"	It specifies the search patterns as 'Newa', 'Newb', or 'Newc', which must occur at the end of the line.
.(dot)	Matches any one character	grep "New. [abc]"	It specifies patterns that contain the word 'New', followed by either 'a', 'b', or 'c'.
\ (backslash)	Ignores the special meaning of the character that follows the backslash	grep "New\.\ [abc\]"	It specifies a search pattern, New.[abc], where the dot signifies the dot character itself and [abc] signifies itself.

- The `grep` filter has options that alter the output of the command:
 - `-n`: Prints each line that matches the pattern, along with its line number. The number is printed at the beginning of the line.
 - `-c`: Prints a count of the lines that match a pattern.
 - `-v`: Prints all the lines that do not match the pattern specified by the regular expression.

The grep Filter [4-4]

A screenshot of a Linux terminal window. The window title is "Applications Places System" and the user is "marcus". The terminal shows the command "grep -n 'test' file1.txt" being executed. The output lists line numbers and the word "test" for lines 1 through 19. Line 1 contains a long string of "test"s, while lines 2 through 19 contain only the word "test". The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows "marcus@Marcus:~/Do...".

```
marcus@Marcus:~/Documents$ grep -n "test" file1.txt
1:testtesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttest
2:test
3:test
4:test
5:test
7:test
8:test
10:test
11:test
12:test
14:test
15:test
17:test
18:test
19:test
marcus@Marcus Documents]$
```

grep Filter

The `wc` Filter



- Counts the number of lines, words and characters in a disk file or in the standard input file.

Syntax: # `wc` [option] [filename]

- The result of using the `wc` filter is displayed in the figure.

A screenshot of a Linux terminal window. The title bar shows 'Applications Places System' and the user 'marcus'. The terminal prompt is 'marcus@Marcus:~/Documents'. The user has entered the command 'wc -l file1.txt'. The output is '19 file1.txt'. The terminal window has a menu bar with 'File Edit View Search Terminal Help' and a status bar at the bottom showing 'marcus@Marcus:~/Do...'.

```
marcus@Marcus:~/Documents$ wc -l file1.txt
19 file1.txt
marcus@Marcus:~/Documents$
```

- The table lists the options of the `wc` filter.

Option	Description
-l	Used to display the number of lines
-w	Used to display the number of words
-c	Used to display the number of characters

The cut Filter



- Used when specific columns from the output of certain commands such as `ls` and `who` have to be extracted.

Syntax:

```
# cut [options] [filename]
```

- The table lists the options of the cut filter.

Option	Description
-f<column_number(s)>	Displays the specified columns
-c<character_number(s)>	Displays the specified character
-d<column_delimiter>	Specifies the column delimiter

The `tr` Filter



Translates one set of characters to another.

Can be used to compress repeated occurrences of a character into one.

Several commands have multi-column output and the gap between columns is more than one space.

In such cases, to extract a particular column, the cut filter cannot be used as the column separator has to be a single character.

Replace the space between columns with a single space and then use the cut filter to extract the desired columns.

Syntax of the `tr` filter is `tr[Set1] [Set2] .`

The sort Filter



- Used to arrange each line of the standard input in ascending order.
- To use the command, type the sort command, and then enter the data that has to be sorted.
- **Ctrl+D** keys: To sort the data.
- An example:

The user types the names.

After specifying the data, the user must press the Ctrl+D keys.

The sort filter compares the characters on each line.

Displays the lines in the ascending order.

Using Pipes [1-4]



Pipe

- RHEL provides a feature that allows combining multiple commands to perform a task.
- ☐ Using pipes, direct the output of one command, or filter to another command or filter as an input.
- ☐ Following commands, used to display the contents of the current directory, a screen-full at a time.

```
# ls > tempfile
```

```
# more tempfile
```

- ☐ Through the pipe feature, two steps can be combined and executed as a single command, without creating a temporary file.

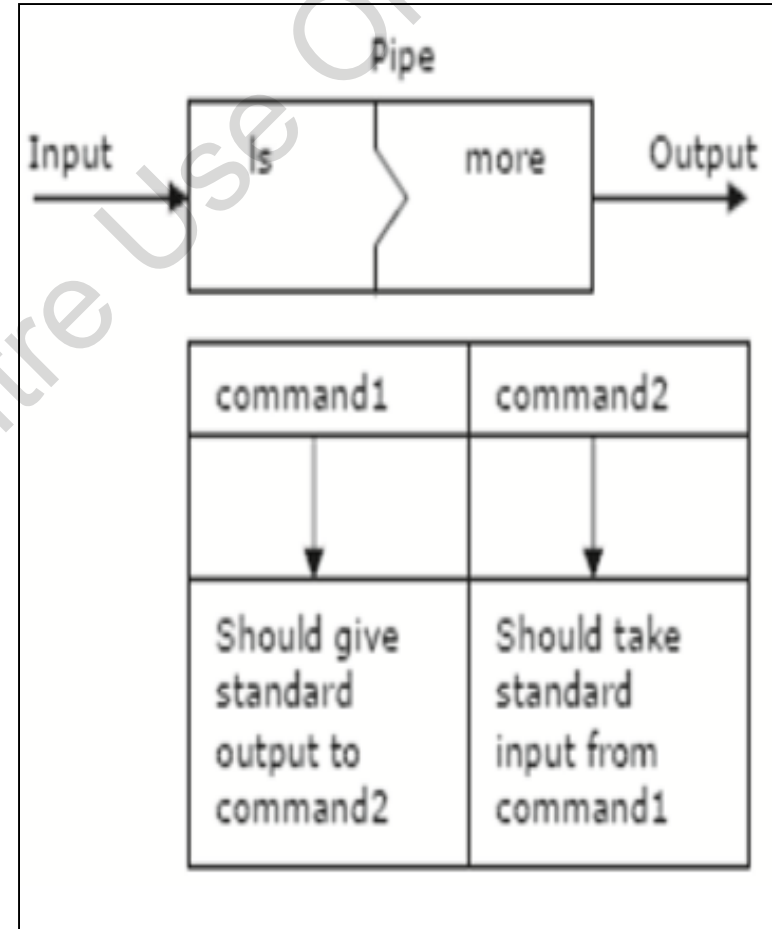
The vertical bar (|)

- The pipe character, indicates to the shell that the output of the command before the '|' is to be sent as input to the command after the '|'.

Using Pipes [2-4]



- An advantage is that utilities are not rewritten to perform complex tasks.
- RHEL tools (commands) can be combined, there is no limit to the number of filters or commands in a pipe.
- The input is processed by the `ls` command, the first command for the pipe.
- After the input is processed, the output from the `ls` command is used as an input for the `more` command, the second command for the pipe.
- The pipe directs the output of one command to another command as input.



Working of a Pipe

Using Pipes [3-4]



- Some uses of the pipe feature are:

Use multiple pipes to display the names of the files in the current directory. The command is,

- `$ ls -l | grep "^-..x" | tr -s " " | cut -d" " -f9`
- The cut command takes this as input and extracts and displays the ninth field.

The command to display the file names in the current directory along with the file sizes, one screen at a time is,

- `$ ls -l | tr -s " " | cut -d" " -f9,5`
- The output of the preceding command is:
1384 DEADJOE
1024 Desktop
1024 Mail
12691 a.out
1024 mail
1024 nsmail 58 program.cc
691 test

Using Pipes [4-4]



The command to display the names of all files in the current directory is,

- `$ ls -l | grep "^-" | tr -s " " | cut -d" " -f9`
- The output of the preceding command is:
 - DEADJOE
 - a.Out
 - program.cc
 - Test

The command to display the total number of files in the current directory is,

- `$ ls | wc -l`
- The output of the preceding command is:
16

The command to display the number of lines where the word 'and', occurs in a file 'test', is,

- `$ grep -c "and" test`
- The output of the preceding command is
1

Summary



- X Window System is based on the client-server architecture that includes two primary components, X server, and X client.
- Shell is a term used for the interactive-user interface of an operating system. In a multiuser environment, the shell has to isolate the user interface from the kernel.
- A boot loader is a program that loads the operating system into the system memory while booting. Runlevels represent the various modes in which the user can work on a Linux system.
- RHEL provides some basic commands and command line expansions that help in configuring the system.
- RHEL provides a feature called pipe that allows users to combine multiple commands to perform a task.