ESSENTIALS OF RED HAT LINUX

Session 15

User Administration

Objectives

- Explain the need for multiple user accounts
- Explain the need for group accounts
- Explain user accounts and its privileges
- Explain the difference between user and root account

User Accounts [1-3]

Provides a user the capability to access the computer with a specific set of privileges.

- Each user account is assigned a user name that is unique in a specific network.

Primarily, it is an account that is created for a specific user or an application.

- Each user account is identified by a name that can correspond to a user or an application.

- Certain attributes are identified for a user account when it is created:
  - Name
  - Group
  - Home Directory
  - UID

User Accounts [2-3]

- User accounts and UIDs are stored in the **/etc/passwd** file in the operating system.
- Each user is assigned a home directory and a program.

Some permissions configured in a user account are as follows:

- To personalize desktops.
- To restrict access to the user's files and folders to other users.
- To share files with other users over the network.
- To track certain set of activities for a specific user.
- To gain full control over a home directory and its files and folders for a user.

- Multiple user accounts can be created on a computer, considered as local computer accounts.

User Accounts [3-3]

- Each local user account can be used to log on.
- Similarly, network-based computer accounts can be added to a local computer to allow them to log on to the computer.
- Each user that belongs to a specific computer has a dedicated folder, called home folder.
- The settings configured by a user do not affect the settings of other users.
- A user has certain privileges to run applications.
- When a user account is created, it should be protected with a password.

Multiple User Accounts [1-2]

- In an enterprise scenario, thousands of users access and share files over the network.

Sharing of files over the network is restricted by the access given to specific users, who have to access the files.

- It is necessary for each user to have a specific user account.
- It helps the user perform certain set of tasks on the local computer and on the network.

- When there are multiple user accounts, each user has their own specific set of privileges.
- Users with higher privileges have more user rights over the files and directories stored in a computer.
- A user cannot read, write, or execute the files of another user unless superuser rights are provided.

Multiple User Accounts [2-2]

- To get read, write, or execute rights for the files of another user, requires specific permissions.
- Permissions are granted either by the owner of the files or the superuser.
- When a user installs RHEL operating system, few standard users are created.
- These standard users are configured in the **/etc/passwd** file.

User	UID	Home Directory
root	0	/root
bin	1	/bin
daemon	2	/sbin
adm	3	/var/adm
mail	8	/var/spool/mail

Group Accounts [1-3]

- A group is a collection of user accounts.
- A user account is used for assigning permissions to a specific folder on the network.

The factors to be considered to create a group and combine users are as follows:

  - Role
  - Departments
  - Geographies
  - Responsibilities

- Along with the standard users, there are a number of standard groups that are created.

Group Accounts [2-3]

- The table lists some of the standard groups that are created during installation.

User	UID	Home Directory
bin	1	root, bin, daemon
sys	3	root, bin, adm

- The ***id <user account>*** command syntax is used to display UID and GID of specific user.
- Only ***id*** command displays UID and GID of logged on user account.

Group Accounts [3-3]

- The figure shows UID, GID, and groups of the root user.

```
login: root
Password:
Last login: Thu Apr  7 05:03:25 on tty1
[root@localhost ~]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10
(wheel) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost ~]# _
```

- After a user is created, the user is given a private group.

Users can be added to more than one group.

  - Adding users to the relevant group helps in assigning access to a large number of users with ease.

- If a file belongs to a group, all users in that group can share the file with other users.

Changing Identities [1-5]

- A user can execute a command or perform a task using someone else's user account.

Consider a scenario where a user xyz has to perform a task that requires superuser authority.

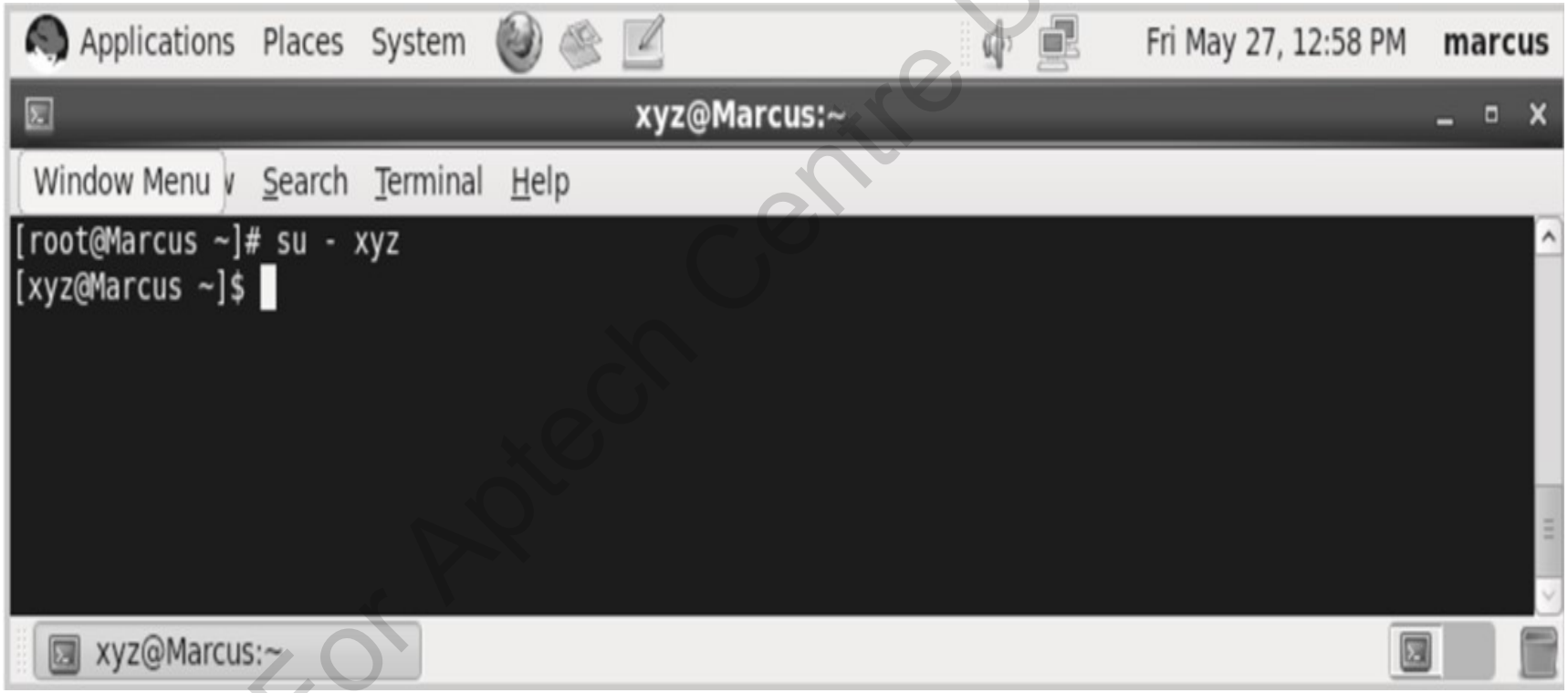
- The user who does not have superuser permissions, can change identity as a root user and create a user account.
- Execute the following command:

```
su - root
```

- When the command is executed, it prompts for the password.
- After the authentication is successful, an administrator can perform tasks that require superuser privileges.
- After performing the task, press `ENTER` key at the command prompt to exit from the superuser account.
- Since the user has root privileges, the user is not prompted for the password while changing to any other user.

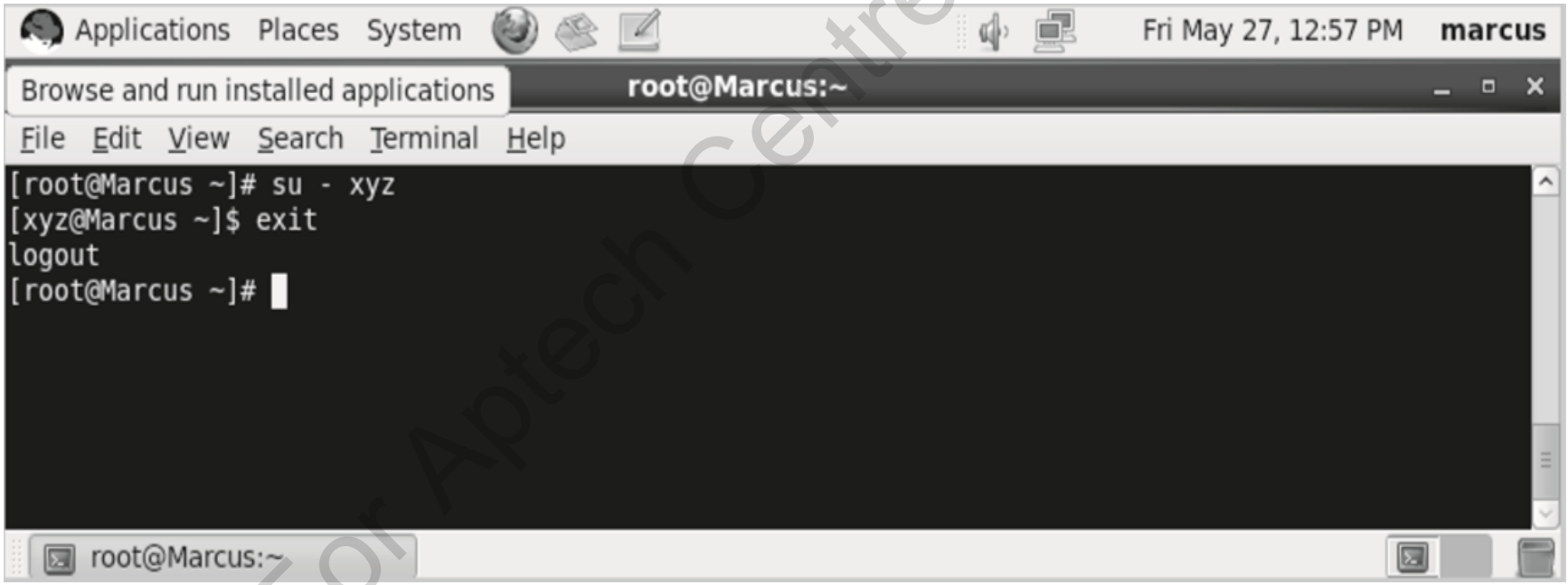
Changing Identities [2-5]

- In the figure, the root user is using `su` command to switch to another user `xyz`.
- The root user is not prompted for any password.



Changing Identities [3-5]

- The `exit` command is used to switch to the previous user account.
- In the figure, the user root switched to user account `xyz` using the `su- xyz` command and switched back to the previous user account using the `exit` command.



Changing Identities [4-5]

- A user can assume to be any other user by using the `su` command with the assumed user account name.

```
# su - xyz
```

- The `su` command is typically used with a - (hyphen).

When used with a - (hyphen), it has two effects:

- Along with the user, it switches to the user's home directory.
- It uses the environmental variables that were being applied to the changed user.

- After execution, the environment is changed according to the specified user.
- The `sudo` command runs the commands as a root user.
- `Sudo` command has a few distinct capabilities while the `su` command does not.

Changing Identities [5-5]

Some of the capabilities of `sudo` command are:

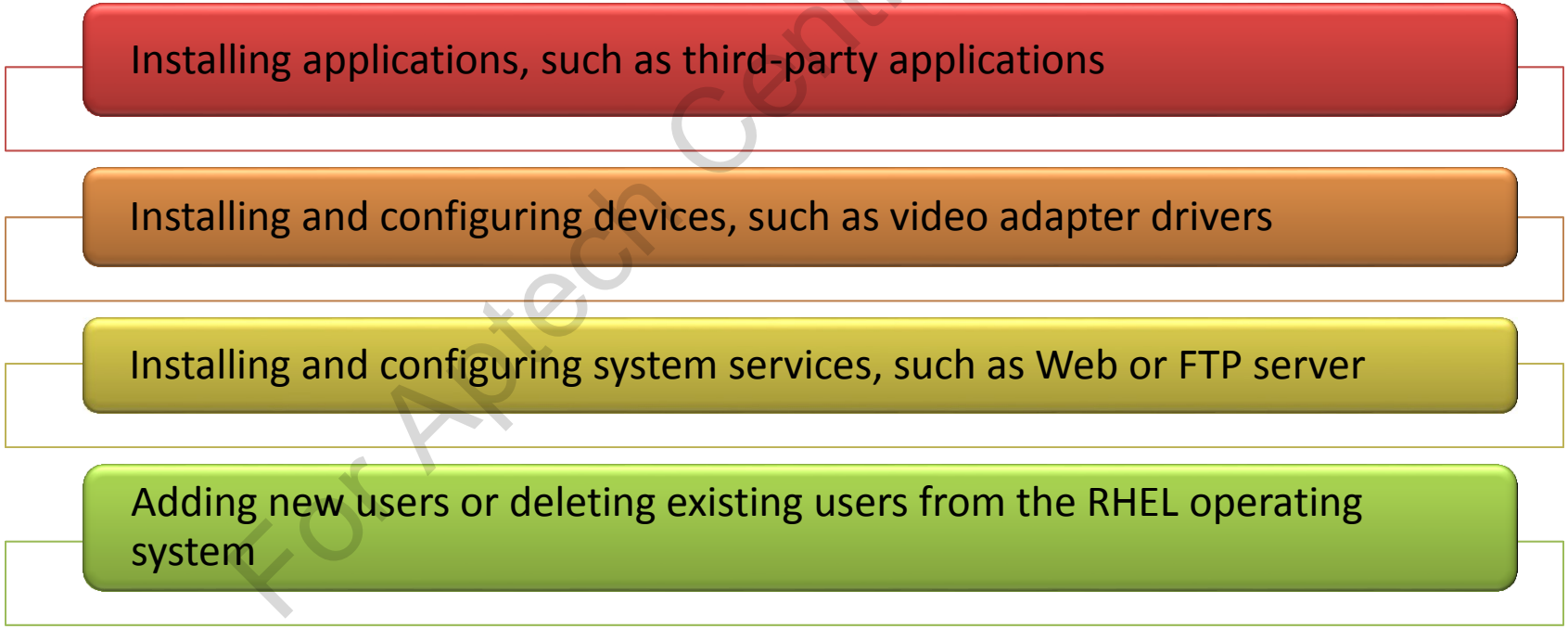
- Can include another command.
- Uses definable constraints.
- Tracks the usage of all commands in a log file.
- Does not require the root password.
- Elevates the privileges of the user account being used with the command.

Privileges

- An attribute is assigned to a user to execute specific tasks and functions.
- An ability that allows a user, an application, or a process to override certain security constraints that otherwise, are applied.
- Helps to bypass certain restrictions and limitations that would have stopped the user from executing certain tasks on a system.
- Can also be assigned to an application or process to gain certain capabilities in a system.
- In context to an application, it can be assigned higher privileges to override certain security constraints.

Root [1-2]

- In Windows operating system, an administrator is the superuser.
- The superuser has complete control over the operating system.
  - Root is the superuser, who has complete control over the operating system.
- The key tasks are mostly system-related tasks that cannot be performed by a normal user, they are as follows:



Root [2-2]

- Root user account has the power to perform all operations and system level tasks.
  - Root account is listed in the **/etc/passwd** file.
- Has the UID marked as 0.
  - If there is any user with UID as 0, it has the same set of privileges as the root user.
- Authentication of a root account is done using the local security files.
- Some users would require the root user account to perform certain system-level operations.

It is not advisable to grant everyone the rights of root user account.

- Instead, each user should log on using their user account and use the `su` command to impersonate root user account.
- This way, a record is maintained in the **/var/adm/syslog** for all the activities performed.

User [1-2]

- Root user has the power to perform any operation in the RHEL operating system.
- A user has only restricted capabilities.
  - The capabilities of a user also depend on the group, they are a part of.
- The table lists the basic differences between a root and a user.

Properties	Root User	Normal User
Security	Low	High
Privileges	Complete operating system	Own home directory
The su command to access another user	Password not required	Password required
UID	0	Dynamically assigned
GID	0	Dynamically assigned
Private group	Root	Username group
Console access	Unlimited	Limited

User [2-2]

- As mentioned in the table, most of the attributes of the root user account is predefined.
  - Not in the case of a user, created by root or another user with superuser privileges.
- The figure highlights some of the differences that are mentioned in the table.

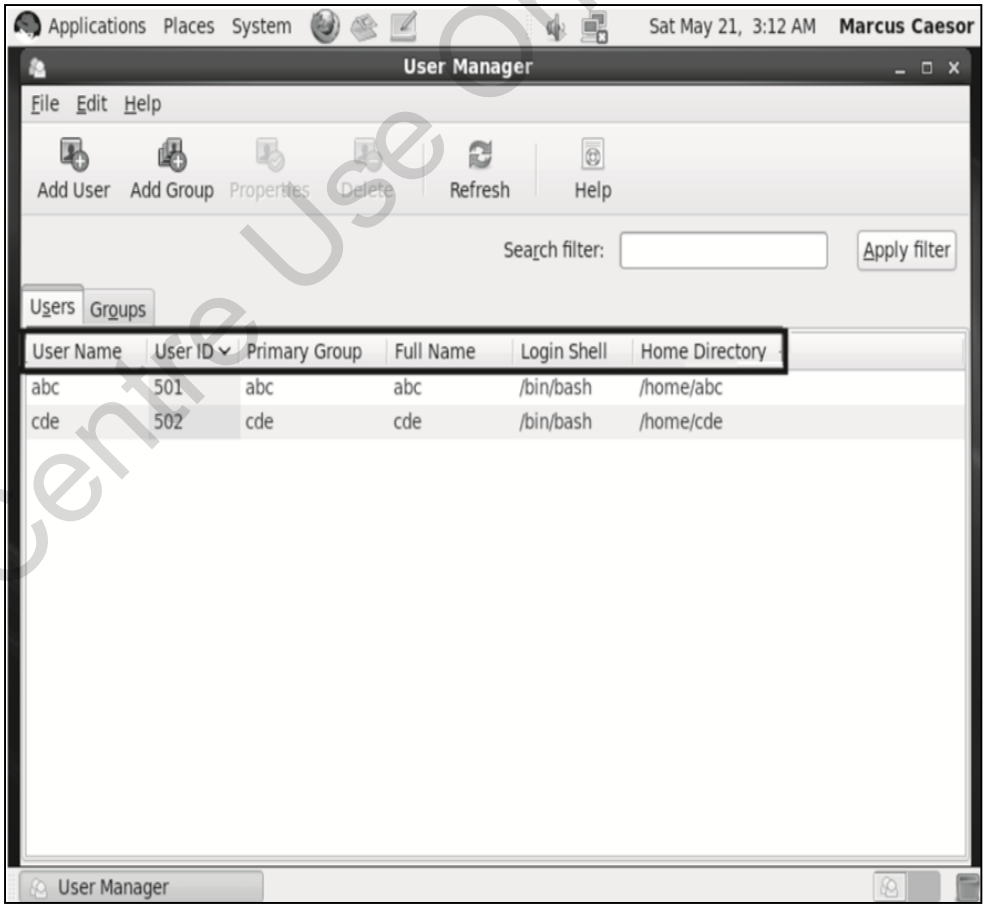
```
[root@localhost ~]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10
(wheel) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost ~]#
```

Root

```
xyz@Marcus ~]$ id
id=581(xyz) gid=501(xyz) groups=581(xyz) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0
c1023
xyz@Marcus ~]$
```

User Manager

- Used in creating a user account in the RHEL graphical environment.
- Displays existing user accounts and details in RHEL as shown in the figure.
- With User Manager, users can add, modify, or delete users and groups.
- Users can be filtered using the built-in search function.



useradd Command [1-3]

- To add a user using the useradd command, perform the following steps:

At the command prompt, type the useradd <username> command; can use username as the input parameter.

Along with this parameter, more parameters can be used.

A user with the specified name is created; important that the user account is created in the locked mode.

To define the password, use the following syntax:

```
passwd <password>
```

After the password is set, the user account is released from locked state and is enabled.

useradd Command [2-3]

- With the useradd command, a number of parameters can be set, they are listed in the table.

Parameter	Description
-c '<comment>	Adds a comment for the user account.
-d <home-dir>	Defines any other home directory. The default home directory is / home/ <username>.
-e <date>	Sets the expiry date of the account. The format is YYYY-MM-DD.
-f <days>	Sets the number of days for which the user account is kept active after the password expires. The value of '0' disables the account immediately after the password expires. The value of '-1' keeps the account active.
-g <group-name>	Adds the default group name or number to a user.
-G <group-list>	Adds any additional group name or number other than the default group to a user.

useradd Command [3-3]

Parameter	Description
-m	Creates the home directory if it does not already exist.
-M	Skips the home directory creation.
-N	Skips the creation of a user's private group.
-p <password>	Encrypts the password with crypt.
-r	Creates a system account that has a UID less than 500. The home directory is not created with this parameter.
-s	Defines the default login shell.
-u <uid>	Defines the UID for the user, which has to be greater than 500.

Files and Directories [1-10]

- A number of operations can be performed at the file system level.

Important Directories

- File System Hierarchy Standard defines the structure and naming convention for directories that should exist within the operating systems similar to UNIX.
- A set of directories and subdirectories are used for specific purposes.

- The table lists the directories and their purposes.

Directory Name	Function
/bin/	Contains essential commands that are used by administrators and users.
/usr/bin/	Contains common commands that are used by administrators and users.
/sbin/	Contains essential commands that are used by administrators.
/usr/sbin/	Contains common commands that are used by administrators

Files and Directories [2-10]

Directory Name	Function
/tmp/	Contains the temporary files for all users.
/usr/local/	Contains locally installed applications.
/usr/share/man/	Contains manual pages.
/usr/src/	Contains source code.
/var/	Contains variable files. Spool and log files are stored here.
/var/log/	Contains log files.
/etc/	Contains configuration files.
/proc/	Contains Kernel virtual file system.
/dev/	Contains device files.

- A normal user would have restriction on some key directories.
- A root user would have permission on all the directories.

Files and Directories [3-10]

Current Working Directory

- In Linux or UNIX, each shell or a system process is assigned a CWD.
- To know the cwd of a process, enter `pwd` at the command prompt.
- The figure displays the output of the `pwd` command.

File and Directory Names

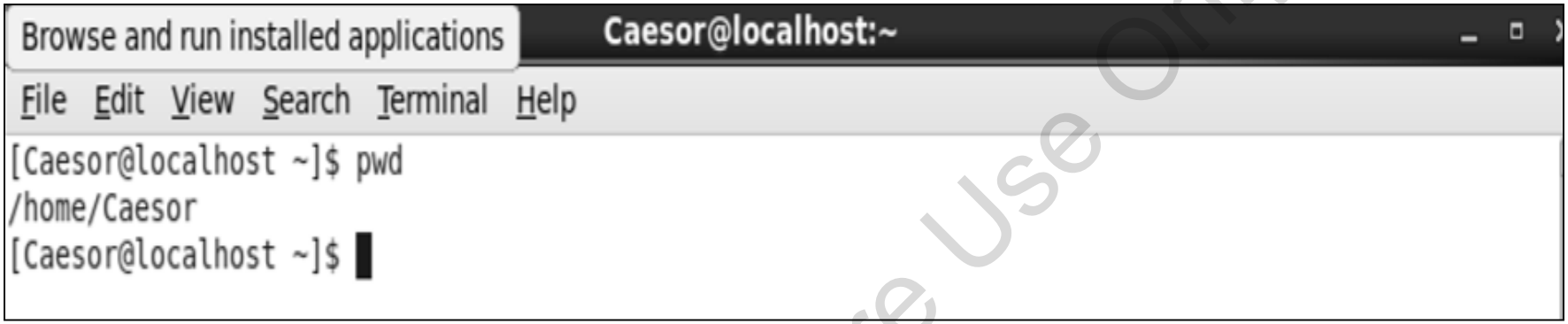
- RHEL, Linux, UNIX operating systems are case sensitive to file and directory names.
- A file name can use all types of characters except the forward slash (/).

Files and Directories [4-10]

- **Absolute and Relative Pathnames**
  - User can change the directory either by using the absolute path or the relative path.
  - The difference between the absolute and relative path names are as follows:

Absolute	Relative
• Also known as the full path. • When changing to another directory using `cd <directory>` command, define the absolute path in place of directory as the parameter. • The absolute path starts with the root directory and then defines the complete path.	• Does not require a complete path. • Specify one or more directories with two dots (..). • Without going to a previous directory, easily navigate to another subdirectory within the same parent directory.

Files and Directories [5-10]



Absolute Path

Relative Path

- **Changing Directories**
  - To change directories to accomplish a task at the command prompt, first execute the command:

    `cd <directory>`

Files and Directories [6-10]

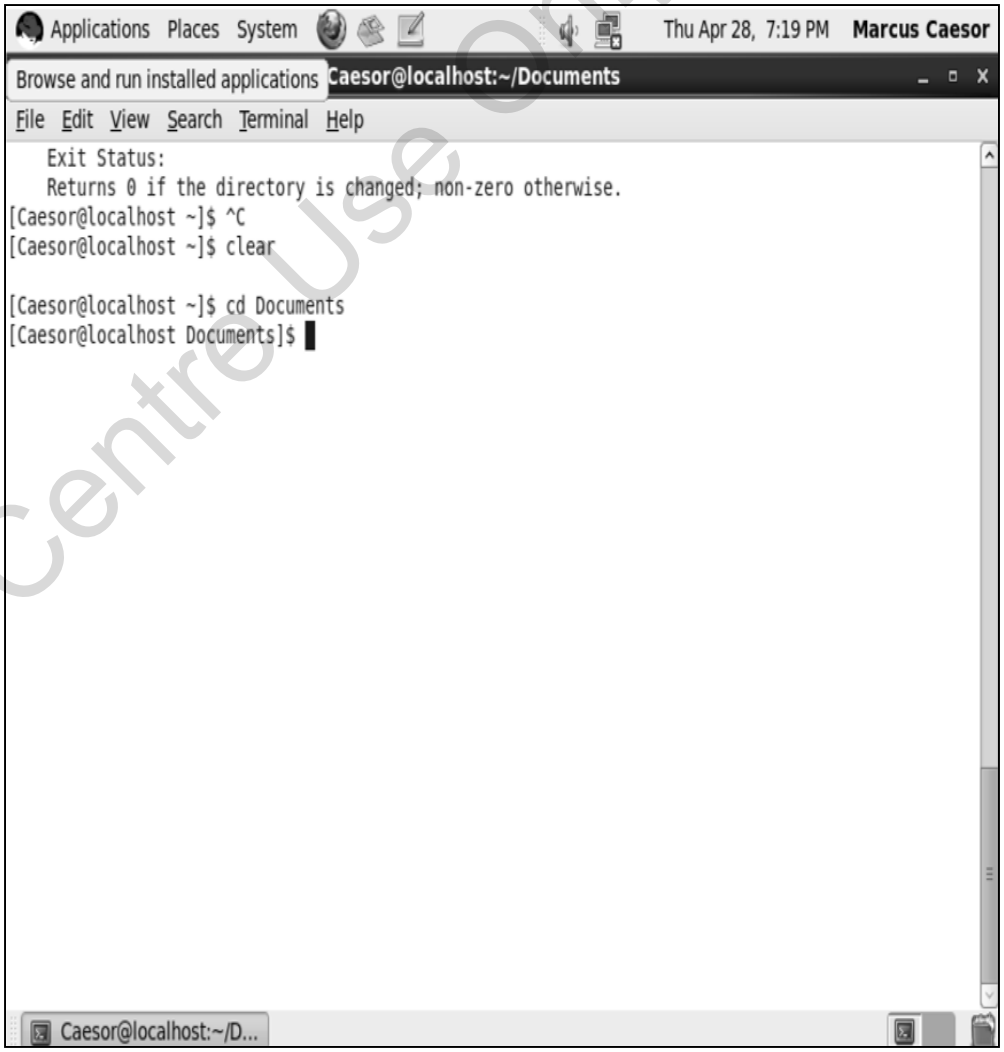
- Execute the command to change directory:

`#cd -L -P <directory>`

- The shell prompt changes when a user changes the directory.
- The directory name is added to the shell prompt.
- The shell prompt shown after changing the directory:

```
[Caesor@localhost
   Documents]
```

- The `cd` command allows a user to navigate between directories using the absolute or relative path.



Files and Directories [7-10]

- To directly move to the home directory, run the command:

```
$ cd
```

- The figure shows the moving to the home directory.

Files and Directories [8-10]

- To move to the previous directory, run the command:

```
# cd -
```

- The figure displays the output of this command.

Files and Directories [9-10]

Listing Directory Contents

- In some cases, a user has to list down the files in a directory.
- The syntax for the `ls` command:
- `ls [options] [files_or_dirs]`
- To list files, run the `ls` command at the command prompt.
- The figure displays an example of `ls` command.

Files and Directories [10-10]

- Some key parameters a user can use with the `ls` command are as follows:
  - `ls -a` - Displays the hidden files in a directory
  - `ls -l` - Displays extra information about the files
  - `ls -R` - Recurses through directories to display files in the subdirectories
  - An example of the `ls -R` command is shown in the figure.

Copying and Moving Files

- The copy operation slightly differs from the move operation.

In a copy operation

The source file is retained as is and another copy of the source file is created at the defined destination.

In a move operation

The source file is deleted from the source location and moved to the destination location.

- Both are used in different situations.
- With the `mv` and `cp` command, a user can use the wildcards to filter out the files that user have to either copy or move.

Copying Files and Directories

cp command

- Used to copy files and directories from source to the destination location.

- The syntax for the cp command:

```
# cp [options] file destination
```

- Copy more than one file to a destination directory by using the command:

```
# cp [options] file1 file2 destination
```

Copying Files and Directories: The Destination

- When copying files and directories, the rules to be observed are as follows:

When copying the file to a directory, the file is copied to the destination directory.

When copying the file and the destination directory does not exist, a file with the defined destination directory name is created.

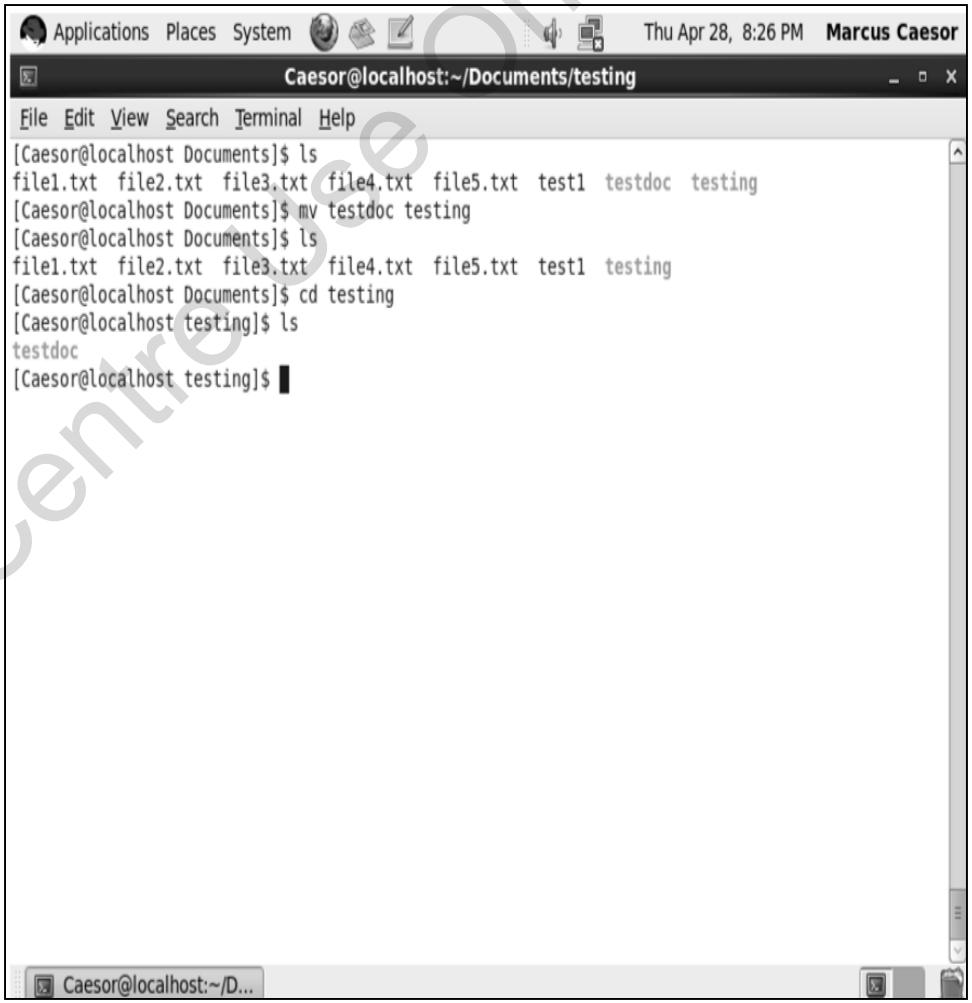
When copying the file and the destination is not a directory but a file, the destination file is overwritten.

Moving and Renaming Files and Directories [1-2]

- To move files and directories, execute the `mv` command:

  `mv [options] file destination`

- Move multiple files at a time, if the specified destination is a directory.
- The figure displays moving the `testdoc` directory to another directory named `testing`.



Moving and Renaming Files and Directories [2-2]

- The `mv` command can be used for renaming files.

  Syntax:

  ```
  mv filename new_filename
  ```

- The figure displays the output of the `mv` command to rename files.

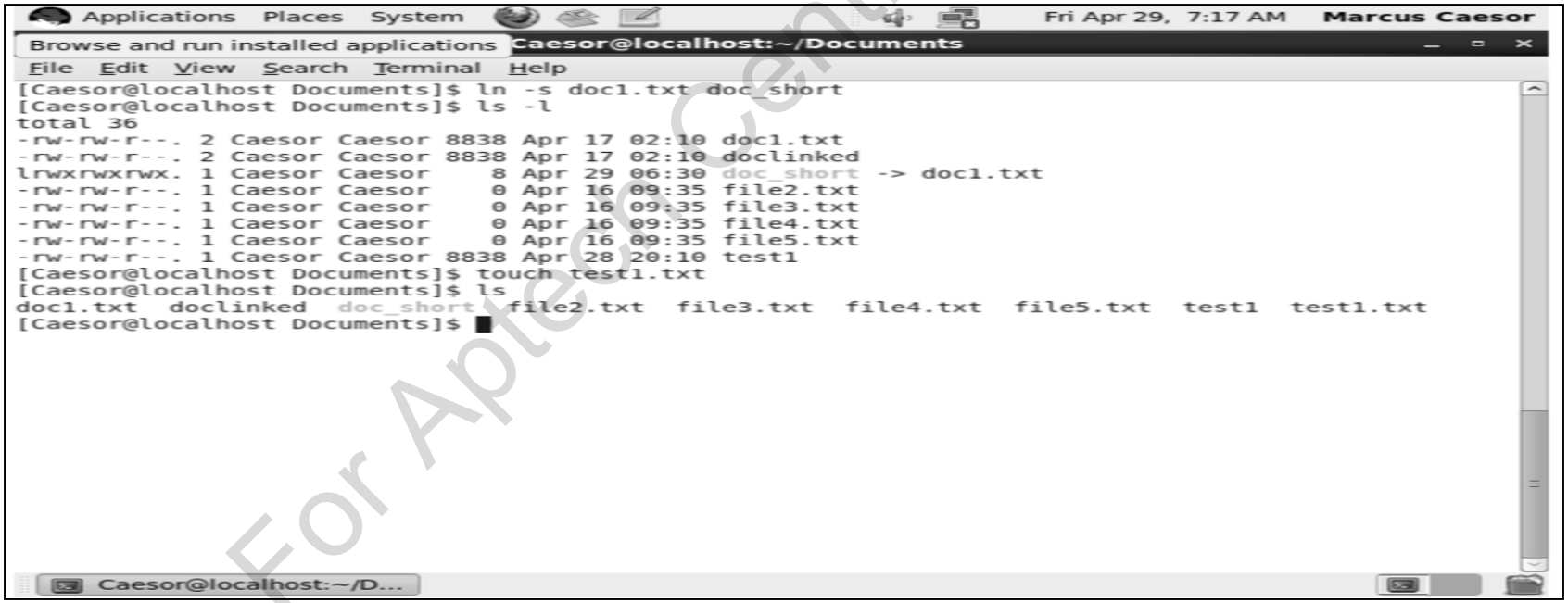
Creating and Removing Files [1-2]

- Create or remove files depending on the requirements.
  - To create a file, use the `touch` command.

Syntax:

```
# touch <filename>
```

- The figure displays the output of the `touch` command.



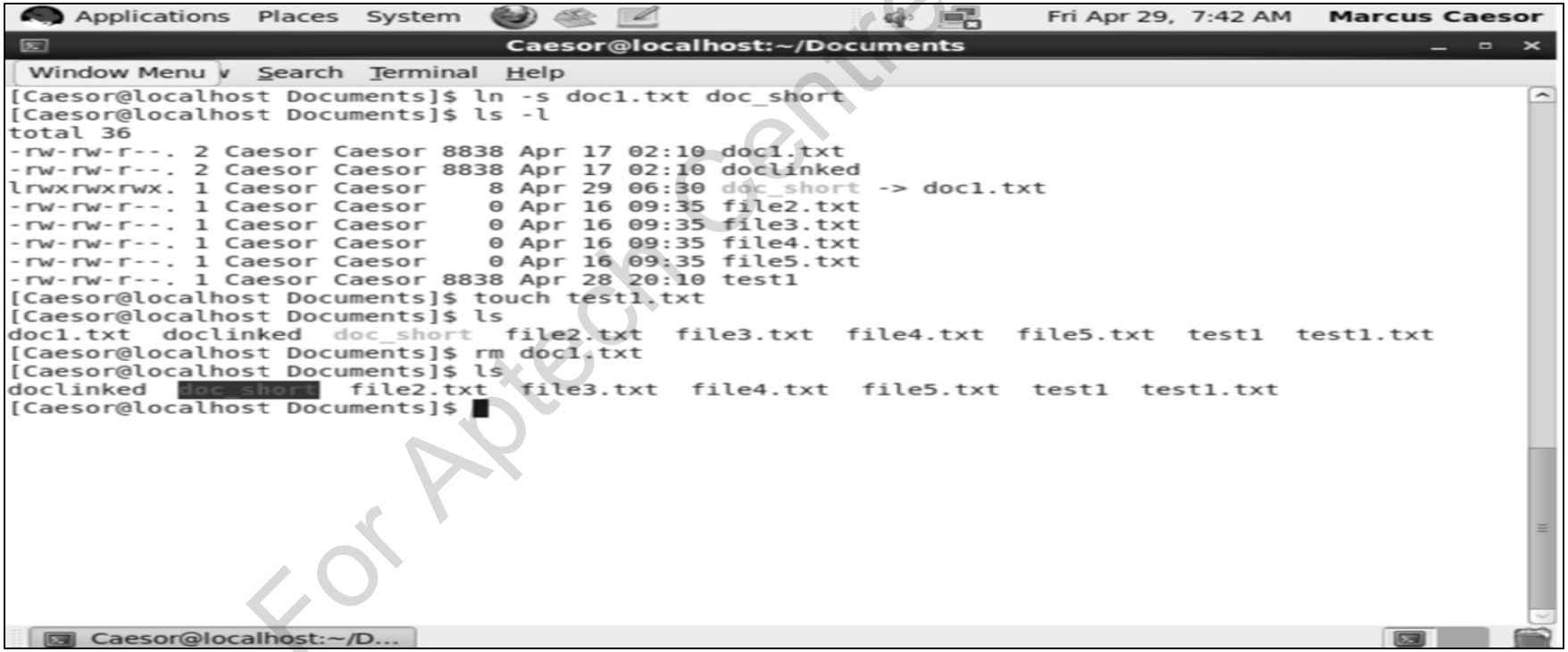
Creating and Removing Files [2-2]

- To remove a file, use the `rm` command.

  Syntax:

  `# rm <filename>`

- The figure displays the output of the `rm` command.



Creating and Removing Directories [1-2]

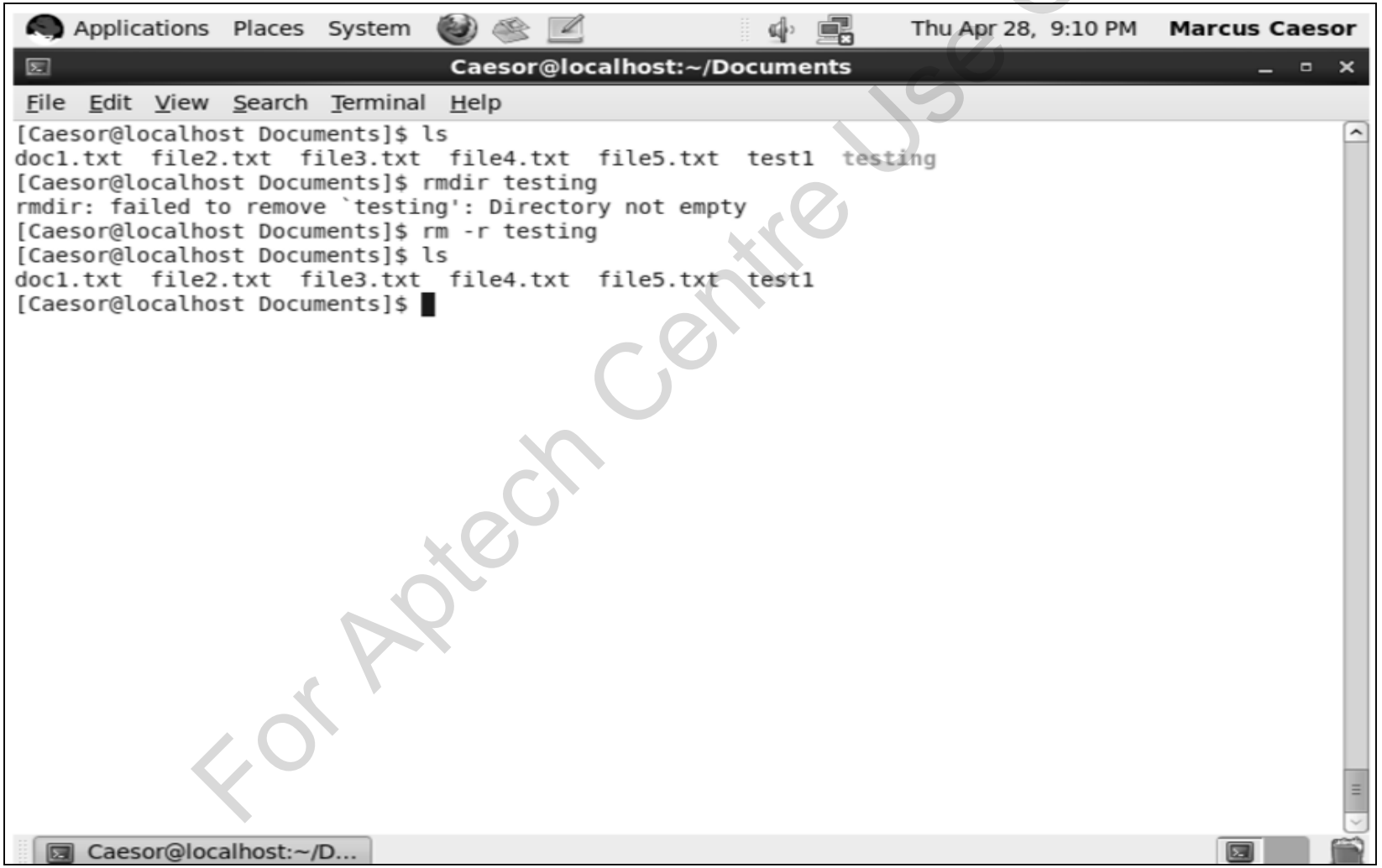
- A user can create and remove directories in RHEL.
- To create a directory, use the `mkdir` command, which can take the absolute or relative path as input.

  Syntax:

  `mkdir [OPTION] directory`
- `rmdir` removes empty directories, but not the directories with the files.
- If a directory which contains files has to be removed, the `rm` command with `-r` parameter must be used.

Creating and Removing Directories [2-2]

- An example of the `rm` command is shown in the figure.



Advanced Operations

- Advanced operations can be performed at the file system level.

  Some operations are automatically performed when a file or directory is created, moved, or deleted.

  - Operations, such as creating links to files are performed by a user.
- Advanced operations can include transferring files from a local system to a remote system.
- Depending on the data criticality, users can choose to transfer the files in a secure or non secure manner.

Inode and Dentries [1-2]

- A `dentry` is a short name for directory entry.
- Linux kernel uses dentry to keep track of the hierarchy of files in directories.
- Each dentry maps an inode number to a file name and a parent directory.

Inode Number

- Is a unique number that is linked with a file name.
- Information maintained in the inode table, contains information of all files that exist on the ext2, ext3, or ext4 file system.

- The inode table maintains metadata information of each file.

Inode and Dentries [2-2]

- The details are included in the metadata are as follows:

UID
GID
Permissions assigned to the file
File type
File size
Time stamps
Pointers that point to the data blocks on the hard disk
Location of the file

- If the file is moved from one directory to another on the same file system, the inode table is updated with minimal changes.
- When a file is moved, a new dentry is created with a new file.
  - When removing files, the inode numbers are updated and the inode count is decremented with the number of files that are deleted.

Create Symbolic Links and Hard Links [1-3]

- A hard link is a physical entry of a file.
- Each file that exists on the file system has one hard link assigned.
- If the hard link is deleted, the reference to the file is deleted and the file no longer exists.

When a hard link is created, the link count is incremented; if deleted, the link count is decremented.

  - If the link count falls to zero, the file is deleted as well.

If hard links to a file that exists in multiple directories, each directory references the same inode number to access the file.

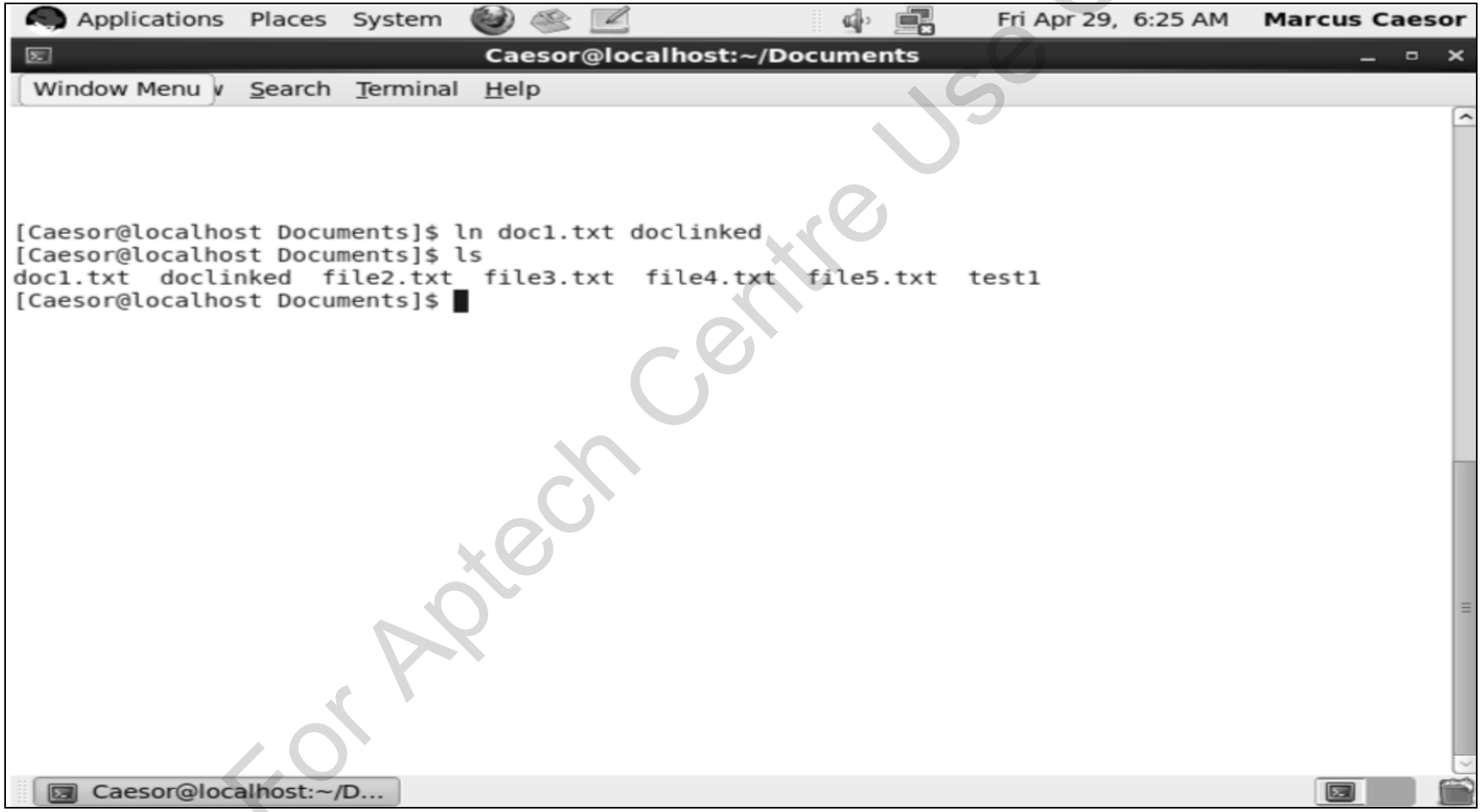
  - No new inodes are created for hard links.

- To create a hard link, use the command:

```
ln <file name> hard link
```

Create Symbolic Links and Hard Links [2-3]

- The figure displays the creation of a hard link.



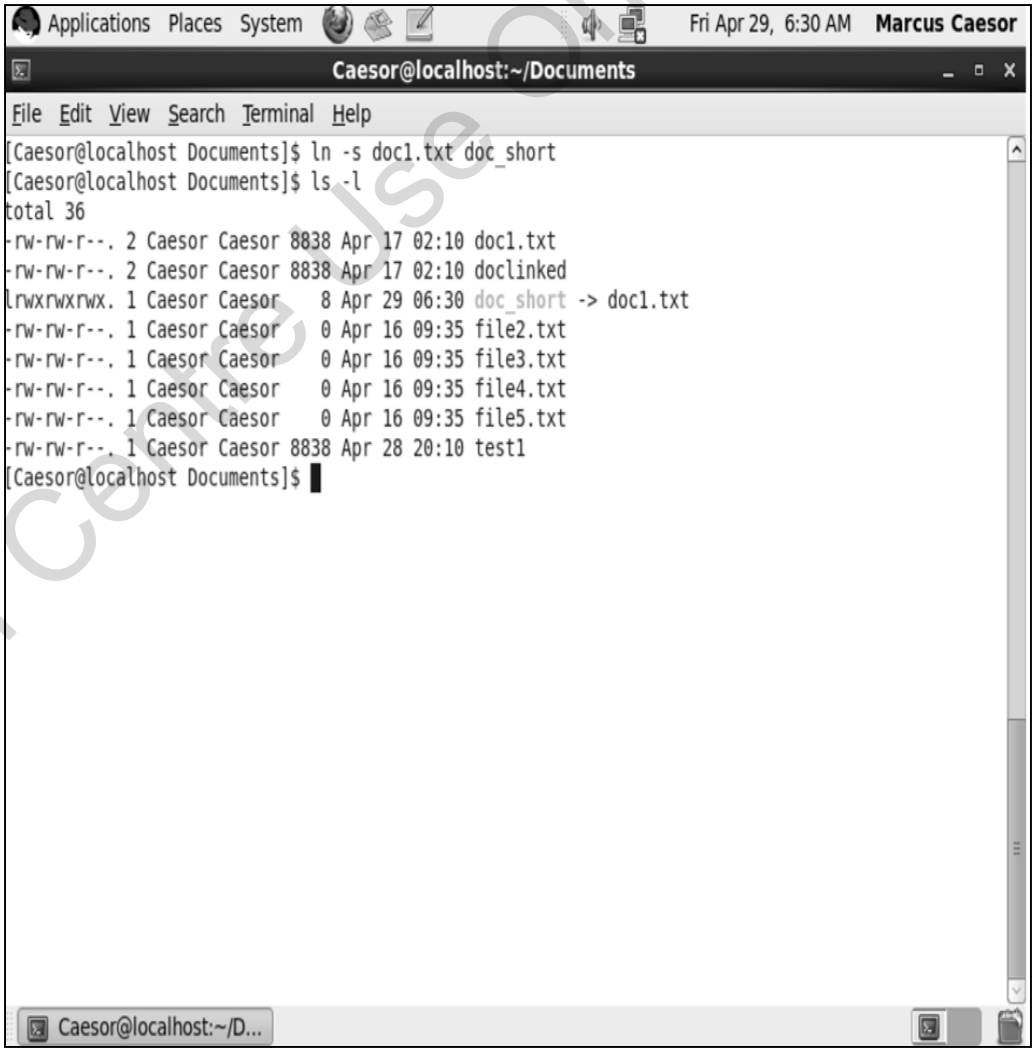
Create Symbolic Links and Hard Links [3-3]

- Symbolic link is similar to a shortcut to a file on the file system.
  - If the original file is deleted, the shortcut is no longer valid or becomes orphan.
  - If the symbolic link is deleted, the original file is retained as is.

  Syntax:

  ```
  ln -s <filename>
     symbolic link name
  ```

- The figure displays symbolic link.



Transfer Files between Systems

- Transfer files by using the `ftp` command or the `scp` command.
- **`ftp` command**: Sends the information to the destination server in an unencrypted form
- **`scp` command**: Secures the information during the transfer.

The advantage of using the `scp` command is, it uses `ssh` protocol for securing data transfers.

- Utilizes the security mechanism of the `ssh` protocol and authenticates users with passwords.

Syntax:

```
scp [-pqrvBC46 ] [-F ssh_config ] [-S program ] [-P port ]
[-c cipher ] [-i identity_file ] [-o ssh_option ] [[user@ ]
host1 : file1 ] [... ] [[user@ ] host2 : file2 ]
```

- Key parameters in this command are as follows:

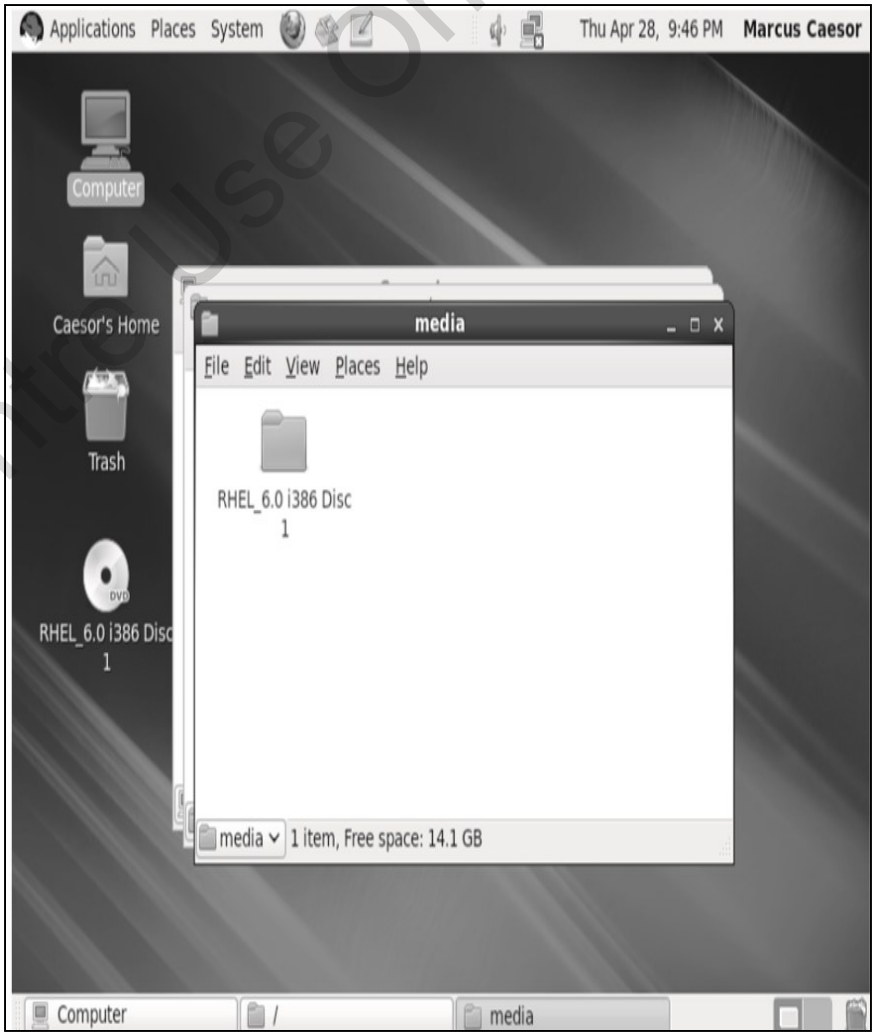
```
scp <local-file> <username>@remote.domain.com:<remote-file>
```

Removable Media [1-2]

- RHEL supports removable media.
- A removable media must be mounted before it can be used.
- Mounting allows the removable media to be accessible on the RHEL operating system.
- GNOME and KDE automatically mount the CD and DVDs.
- The figure displays the mounted DVD.

Removable Media [2-2]

- Mount points for removable media are stored under the **/media** directory.
- The figure displays the mount point for the DVD when users browse the **/media** folder.
- USB devices are detected as SCSI devices by the Linux kernel.
- The **/media** directory contains the USB device information.
- Floppies must be manually mounted or dismounted.



Security Fundamentals

- RHEL allow users to create and manage a user account for each user.
- To ensure security, a system administrator assigns policies for controlling the rights and permissions assigned to a user.

Factors taken into account while creating policies are as follows:

- The login ID and password that should be assigned to a user.
- The group to which a user belongs.
- The services to which a user can have an access.

- These policies are used during the creation and administration of the user accounts.
- By default, the RHEL users have minimum permissions assigned.
- For the system level task, users are required to have root permissions.

Using Policies [1-4]

- Policies can be used to restrict users from logging on to a specific system or disabling a user account after certain number of incorrect login attempts.
- Two types of policies considered while creating user accounts are as follows:
  - User Account Policies
  - Password Aging Policies
- **User Account Policies**
  - Each user has a different account.
  - The user account has unique properties or characteristics.

Factors taken into account while creating a user account are as follows:
• User access to the system files and resources • Periodic password changes required by the users for security reasons • Logins would remain active • CPU and memory limits should be allocated • Disk quota should be enabled

Using Policies [2-4]

Factors considered when assigning login IDs and passwords to users are as follows:

- The login IDs must be unique. It can be decided based on the size, structure or nature of the organization.
- The passwords should be at least eight characters in length. Short passwords are weak passwords and can be easily guessed.
- The passwords should not be easy to decipher.

- **Password Aging Policies**
  - Sets the password age for a user account.
  - Using the policy, the user must set the duration for which a password remains valid.
  - At the end of that duration, the user will be prompted by the operating system to enter a new password.
  - A password set for a longer duration provides less security as the password can get leaked to other users.
  - To configure the password settings, the `chage` command can be used.

Using Policies [3-4]

Syntax:

```
chage [-m mindays] [-M maxdays] [-d lastday] [-I inactive]
[-E expiredate] [-W warndays] user
```

- The table lists the options available with the `chage` command.

Option	Function
-m <mindays>	Specifies the minimum number of days within which a user has to change the password. If the value of the –m parameter is set to zero, it indicates that the password would never expire and the user can change the password at any time.
-M <maxdays>	Specifies the maximum validity period (in days) of the password.
-d <lastday>	Specifies the number of days, when the user last changed the password.
-I <inactive>	Specifies the number of days an account is inactive. An account is inactive after a password has expired and before the account is locked.

Using Policies [4-4]

Option	Function
-E <expiredate>	Sets the date in the YYYY-MM-DD format. After the specified date the account would no longer be accessible by the user.
-W <warndays>	Sets the number of days when a warning would be displayed to intimate the user that the password change is required.

- To determine when the user's password is going to expire, use the `chage` command.

  Syntax:

  ```
  chage -l <user-name>
  ```

Authenticating Users

- When a user logs on to the Linux operating system, it authenticates the ID and password entered by the user.
- Authentication can take place on the local system or on the user database stored on the remote server.
- Local authentication is implemented using the shadow password and MD5 password schemes.

Network authentication is implemented using the NIS, LDAP, or SMB authentication schemes.

- Authentication schemes can be selected during installation of the operating system or after the installation with the help of the `authconfig` command.

Using Local Authentication [1-2]

- In Linux, when a new user account is created, an entry for the user is reflected in the user database file **/etc/passwd**.
- The **/etc/passwd** is a readable file that can be easily changed or can be used to trap the password for any user.

Linux provides a shadow password scheme to encrypt the passwords.

  - If the shadow password scheme is enabled, the encrypted passwords are stored in the **/etc/shadow** file.
  - Only the root user can read these passwords.

The **/etc/shadow** file stores one record for each user account.

  - The owner of the **/etc/shadow** file is a root user; the root user can change the file permissions.

- The format of the **/etc/shadow** file is as follows:

```
username:passwd:last:may:mst:warn:expire:disable:reserved
```

Using Local Authentication [2-2]

- The table lists the fields of the **/etc/shadow** file.

Field	Description
username	Describes the user login name
Password	Describes the password in the encrypted form
last	Describes the number of days, the current date, the date when the password was last changed
may	Describes the number of days before which the password can or cannot be changed
must	Describes the number of days post which the password must be changed
warn	Describes the number of days left for the password to expire, so that the user is warned
expire	Describes the number of days after which the password expires and the account is disabled
disable	Describes the number of days between the current date and the date on which the account was disabled
reserved	Is a reserved field that can be used in future

Summary [1-3]

- To access the RHEL environment a user account is required. Multiple user accounts can be created on a single RHEL system. The user accounts provide an identity for the user. These identities can be changed in different ways such as changing from a user account to root user account or changing from root user account to another user account. The root user has complete control over the RHEL environment. The differences between a root and a normal user account depend on the UIDs of the root and normal user accounts. There is a process to create user accounts and check its properties along with the requirement to create a group by combining multiple users.
- User accounts are created using the User Manager. The user account properties of the user accounts can also be edited in the User Manager. A user account can also be created from the command-line interface. To create user account from the command-line interface, the `useradd` command with various parameters can be executed.

Summary [2-3]

- Users are often required to copy or move files from one directory to another directory. In copy operation, the source file is retained as is and another copy of the source file is created at the defined destination. In a move operation, the source file is deleted from the source location and moved to the destination location. Advanced operations include dentries and inodes, and symbolic and hard links.
- The scp command can be used to securely transfer files from one system to another system. Users can transfer files in secure manner using the scp command.
- Users can use removable media on RHEL. Some examples of removable media are CD, DVD, floppy, and USB. A user must mount the removable media before it can be used.

Summary [3-3]

- To ensure security of the system, a system administrator prepares policies for controlling the rights and the permissions assigned to the user. These policies are used when creating and administering the user accounts. The two types of policies considered while creating user accounts are User account policies and Password aging policies.
- Local authentication is implemented using the Shadow Password and Message Digest 5 (MD5) Password schemes. Network authentication is implemented using the Network Information System (NIS), Lightweight Directory Access Protocol (LDAP) or Session Message Block (SMB) authentication schemes. The account of a new user is locked by default and RHEL enables users to assign one user's privileges to another user.