Course name - Operating System/ Linux

Make Directory - mkdir

Add -v to any command to see the execution print
Remove directory - rmdir
The rmdir will not delete directory if it contains any file or any any sub directory
To do the recursive delete - rm(remove) -R directory name/path

Listing files

Ls - l(long)
Ls -F - this -F lists the file names with the nature like, if a file is a directory its name will come as test/ and if its an executable file then it will come as test@
Ls -a - shows all the files including the hidden ones
Ls - R - this recursive list, lists the entire files and subdirectories in a directory

Moving files

Mv move / rename
Ex mv test.txt desktop/test1.txt
Copy - cp, cp -R recursive copy to copy all files under a directory when copying
Stat - stat filename, gives the details
Touch - modifies

Locate files

GREP - globally search regular expression and print
Grep “Adobe Flash” ~ datafiles/mydir/*
the star/wildcard at the end to signify to search all files in the mydir folder
-I want to search and list a particular keyword ex. Q4 in a file name and list all such filenames
Ls -l -R | grep Q4

Find

If getting an error with find try: *find . -name “*.pdf”* to list all the pdf files in the current directory and its subdirectories.

find ~/Datafiles/ -type f -name “Jan”
The above finds in datafiles root dir, type file only having name Jan
find -type f -name “Jan” -or -name “Jan”

This finds in the current working directory files with names Jan or Aug
find -type f -name “.txt” -not -name “S*”*
Finds file names ending with .txt but not starting with S
find -type f -name “.txt” -not -name “S*” -exec grep -H “Adobe” () \;*
Finds file names ending with .txt but not starting with S and executes grep to find text Adobe in the files result, -H is for printing it. Also the () tells the grep command to search for each file returned from the find result. The \;, the \ is to just tell the terminal to consider the ; as a semicolon

Backup and restore files

The **dump** command “dumps” all files in a filesystem into a tape or another file. It can also be used to dump files modified after a specified date.
The syntax of the **dump** command is **dump {-level #} -f {file} {filesystem/file/directory}**.
The **restore** command enables you to restore files or filesystems from backups made using the **dump** command. This command can be used across networks to restore data.

The **tar** command allows you to create archives of data.

For example, **tar cf filename.tar files*** is equivalent (and more frequently used than) **tar -cf filename.tar files***.
Note: Archives made with **tar** are frequently compressed with **gzip** (resulting in the file extension **.tar.gz**) or bzip2 (resulting in the file extension .tar.bz2).

The command **tar -xvf** will restore the entire contents of the source file or directory structure.
ou can also make restores interactive by using the command **tar -wxvf [destination] [source]**.

Gzip

GNU zip (**gzip**) is a compression utility that reduces the size of selected files. Files compressed with **gzip** frequently have the **.gz** file extension.
The syntax of the **gzip** command is **gzip [options] {file name}**.
Compressed files can be restored to their original form using **gzip -d**, **gunzip**, or **zcat**.

Unzip

The **unzip** command is used to list, test, and extract compressed files in a ZIP archive.
The syntax of the **unzip** command is **unzip [options] {file name} -d [directory]**.

Working with Text Files

Vim

Vim command invokes the Vim editor

:w {*file name*}	Saves a file with a file name if it is being saved for the first time.
:q	Quits when no changes have been made after the last save.
:q!	Quits ignoring the changes made.
:qa	Quits multiple files.
:wq	Saves the current file and exits.
:e!	Reverts to the last saved format without closing the file.
:!{*any Linux command*}	Executes the command and gets the result in the Vim interface.
ZZ	Writes the file only if changes were made and quits the Vim editor.

Cat command

Display contents of a file

Cat -n mytext.txt
shows the text with line number

Cat mytext.txt > some.txt
Copies(overwrites) the contents of mytext.txt to some.txt

Cat mytext.txt >> some.txt
This will add the contents of mytext.txt to some.txt at the end, will not overwrite

Cat >> some.txt
This is the text ro add
I am captain awesome^C(control C to close)
This will add at the end of some.txt the text we mentioned in above commend

Cat mytext.txt thistext.txt text.txt > merge.txt
Merges contents of all three text files into merge.txt file

Diff, vimdiff and wc commands

Diff mytext.txt test.txt -c
Gives the difference in two files,
The lines start with either -, + or ! , = means the line was added to the file and ! mans the line was removed and so on

Vimdiff mytext.txt test.txt
This opens up the vim editor and shows the difference

Wc finename.txt
Just shows the word count

aspell, tr and Redirecting Output

Aspell -c filename.txt
Spell check and fix

Tr
Translate

Redirecting any shell command o/p o/p is to redirect the o/p to not show in the command prompt but to show or save it elsewhere

Ls > lsoutput.txt
This command moves the output of ls command to a text file and does not show in the terminal or cmd prompt

Searching text

Regex

Regular expressions are strings of characters that form a pattern for searching another string, often a word, a set of words, or a sentence. Finding and replacing text and manipulating strings are the main uses of regular expressions.
For example, to find all of the lines in the **/etc/services** file that contain either “**apple**”, “**Microsoft**”, or “**IBM**”, the command would be**grep -E ‘apple|Microsoft|IBM’ /etc/services**. The regular expression in this command is **apple|Microsoft|IBM**. This specifies that any of those three words may match the text string. If you do not want to use the **-E** option, you can use the **egrep** command. In that case, the command would be **egrep ‘apple|Microsoft|IBM’ /etc/services**.
A regular expression with notational elements is a search string formed by combining wildcards, numbers, and characters.
For example, **[^e]?b[1-9]** as a whole is called a regular expression with notational elements. The notational elements are **^**,**?**, and **[1-9]**. This expression searches for a line that starts with the letter “e” followed by a character/number, then by the letter “b”, and finally by a number ranging between 1 and 9. When you’re working with “wildcards” the **?** can represents any one character, a letter or a number. The ***** represents any number of characters (even no characters) which can include numbers or letters.
Another example of a regular expression with notational elements is **1{5}**, where **{**and **}** are the notational elements. This expression searches for the occurrence of the number “1” repeated consecutively five times. A regular expression is often referred to as a **regexp**.

Regular Expression	**Description**
.*	Zero or more of any number of characters in a row.
.+	One or more of any number of characters in a row.
?	Zero or one character, letters or numbers when used with grep. Exactly one character when used with other commands like ls.
.	One character, letters or numbers.
\d	A metacharacter that specifies any numeric digit (0-9).
\w	A metacharacter that specifies any word character (a-z, A-Z, 0-9, including underscore).
[a-zA-Z0-9_]	A character class that equals the \w regular expression metacharacter (a-z, A-Z, 0-9, including underscore).

/<*character*	Matches any instance of the character at the beginning of a word. (NOTE: The beginning of the word is signified by either a space or a period coming directly before the character.
Character/>	Matches any instance of the character at the end of a word. (NOTE: The beginning of the word is signified by either a space or a period coming directly after the character.
^\d{5}	A 5-digit number (i.e., a US ZIP code).
http:\/\/[A-Za-z0-9.]{1,}	An HTTP web address (URL). (NOTE: The \ is used to “escape” the “/” that follows meaning it will be treated as literally the character “/” instead of being interpreted as part of the expression.)
https?:\/\/[A-Za-z0-9.]{1,}	An HTTP or HTTPS web address (URL). (NOTE: The \ is used to “escape” the “/” that follows meaning it will be treated as literally the character “/” instead of being interpreted as part of the expression.)
^*character(s)*	When used at the beginning of an expression, this searches for the character(s) at the beginning of a line. (Example: ^S searches for lines that start with the letter “S”. When used inside of brackets it excludes those characters from the answer. (Example: [^t] searches for matches that do not contain the letter “t”. On a US keyboard, you create this character using shift+6.
character{#}	Searches for instances of the character repeated # of times in a row. Example: 3{4} would search for the string “3333”.
character{#,}	Searches for instances of the character repeated # of times in a row or more. Example: 3{4,} would match any string that has at least four instances of “3” in a row. Therefore it would match “3333”, “33333”, “333333”, etc.
character	Escapes the character causing it to be treated as an instance of the character rather than a symbol that has meaning in the expression. Example: \/ would search for “/” in a file. \. would search for a period in a file.

Ls *Test.txt
This cmd gives the o/ps such as myTest.txt Test.txt AllTest.txt

Ls ??Test.txt

This cmd will give o/p as myTest.txt because there are exactly 2 charaters in this file , as specified by the two ?s

Say I have a chapters directory with 5 chapters namely chapter1, chapter2 respectively
If I do
Ls chapter[1]
Searching for the file with name chaperand a number 1
This will give the o/p chapter1

Ls chapter[135]
This will search for all files with number ending with a 1 or 3 or 5

Grep with regex for finding numbers

I have a file HR.txt and it has a paragraph of text and a pincode 02905 in it
Grep -E '[0-9]{5}' HR.txt
This is an extended search to check for any digit between 0-9 and 5 digits in a row

Grep with regex for finding letters

Grep -E 'blue|gray' HR.text
This will search for blue and gray in the content

Grep '^S' HR.text
We are looking for all the lines which start with a capital S

Grep '\<c' HR.txt
This will search for all the words which start a c

Grep 'r\>' HR.txt
This will search for all the words which end with a r

Grep '\<c.r\>' HR.txt
This will search for all the words which start with a c and end with a r. The dot signifies any letter between c and r. Example this will find the word car

Grep '\<c...r\>' HR.txt
This will search for all the words which start with a c and end with a r. The dot signifies 3 letter3 between c and r. Example this will find the word color

Grep -E '\<c[a-z]{1}r\>' HR.txt
this will work same as and the o/o will be car, as we have specified to check for one letter between a-z and starts with c and r

Grep -E '\<c[a-z]{1,}r\>' HR.txt
The 1, will actually take any amount of letters in between starting letter c and ending letter r

Grep -E '\<c[^o]{1,}r\>' HR.txt
This will search for car, character but wont find color because it has o in it, and the exp above searches for any word starting with c ending with r but not containing o's

Grep with regex for finding special characters

Grep -E 'http:\/\/[A-Za-z0-9.]{1,}' HR.txt
Search for http then the forward slashes, so we need to tell that treat the forward flash not as an expression notation but a thing to search for and that's why we give a backslash with it.
Then any capital letter, any small letter and any number and also dot and the {1,} represents any number of those.
This will search for http://www.google.com

Grep -E 'http?:\/\/[A-Za-z0-9.]{1,}' HR.txt
We added ? , it means any one character after http, it could be numbers, letters
This finds https://www.google.com

Grep -E '1{3}.*SL'' Leave_log.txt
We are looking for lines with for three ones, then there can be any number of characters in between whether digits or letters by using the .* and followed by SL

Filtering text

Cut command

If we have a delimiter separated data such as data in csv are comma separated, then the cut command helps to just cut the separated data, like we can specify if we want the first field or the data after the delimiter.

cut -d ',' -f 1 test.txt
-d specifies what the delimiter is, the comma is the delimiter here, the -f is for the field and followed with 1 meaning we need the first column and then specify the file name.

cut -d ',' -f 1 -s test.txt

The -s added here simply suppresses meaning we are telling that ignore any row where there is no delimiter in the output.

cut -f 1 -s test.txt
If we remove the delimiter, it by default takes tab as a delimiter

cut -d ‘,’ -f 1 -s test.txt > employeenumber.txt
This saves the output in a new employeenumber.txt text file

Sort command

sort -t ‘,’ ActiveEmp.txt
Sorts in ascending order(numbers take precedence)

sort -t ‘,’ -k 1 -r filename.txt
1 is for sorting by first field The -r is for descending

For sorting by second field
sort -t ‘,’ -k 2 -r filename.txt

TextUtil command

To see the first 10 lines
head filename.txt

To see last 10 lines
tail filename.txt

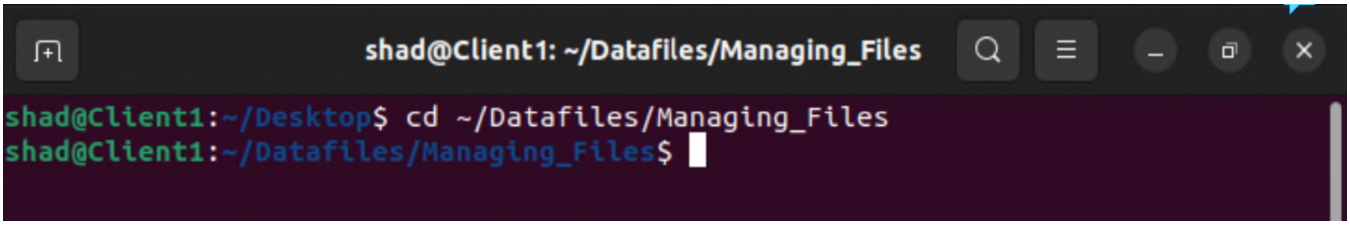
If you want to see the number of lines like a serial number
nl filename.txt
tr ‘,’ ‘\t’ < filename.txt
This like pretty print, tr command followed by what is the current delimiter the ‘\t’ is to replace the comma delimiter by a tab
Now if I still do cat filaname.txt the delimiter will still be a comma.

tr ‘,’ ‘\t’ < filename.txt > Nicefilename.txt
This will save the tab delimiter version data to a new Nicefilename.txt file

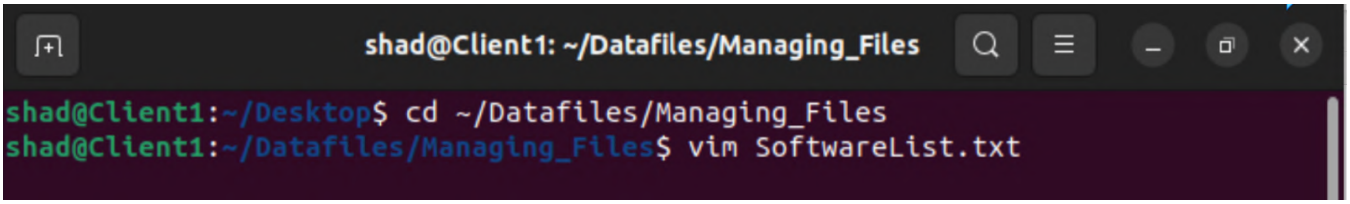
Working with Text Files Lab

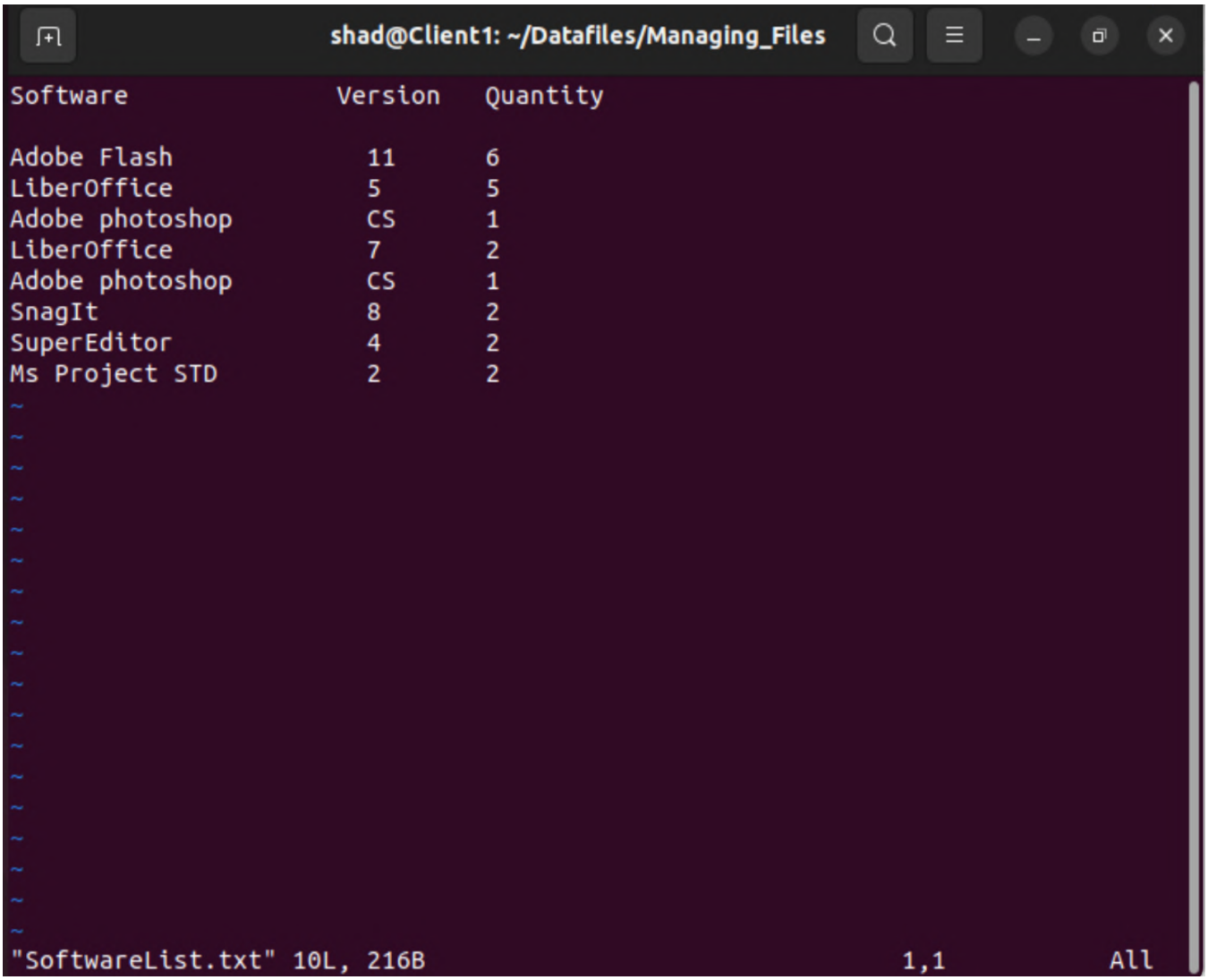
Editing Text Files in Command Mode

1. In the shell, type **cd ~/Datafiles/Managing_Files** and press **Enter**.

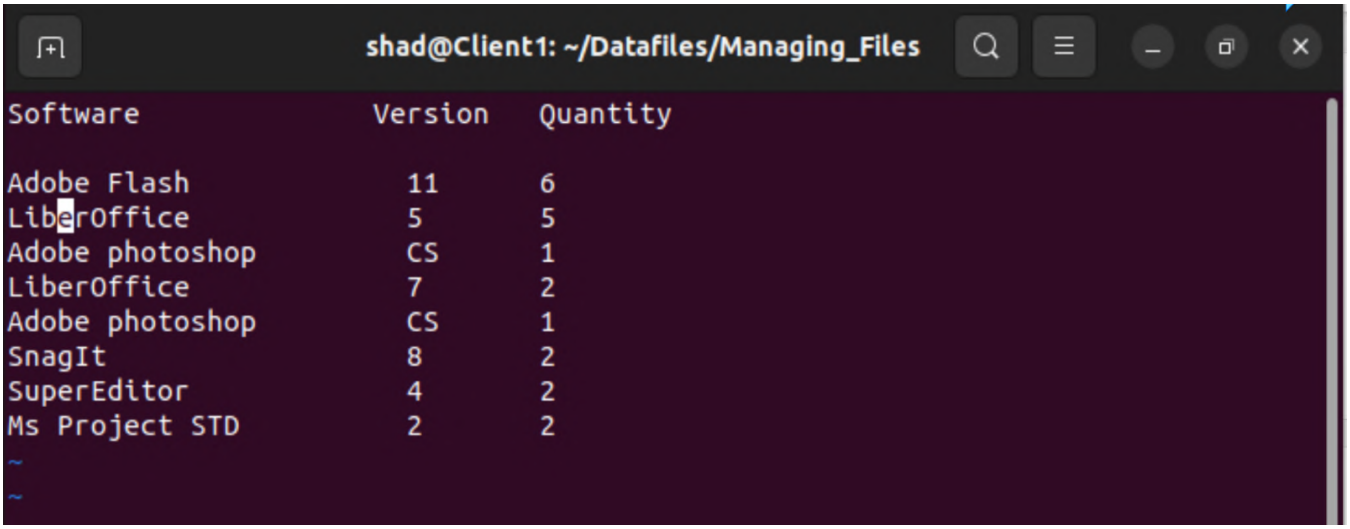


2. Type **vim SoftwareList.txt** and then press **Enter**. This command opens the **SoftwareList.txt** file in the Vim Editor.

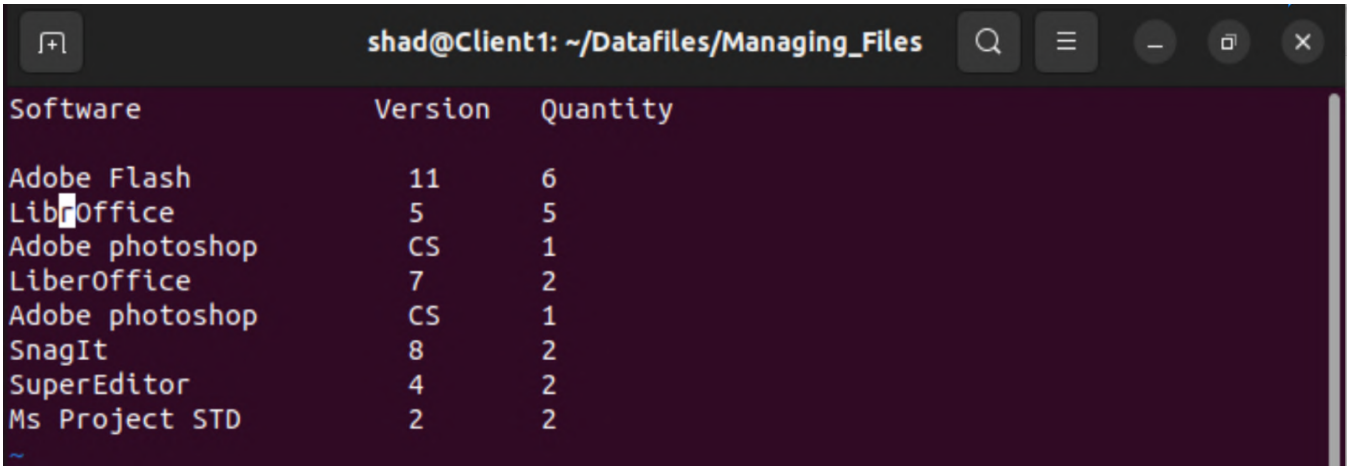




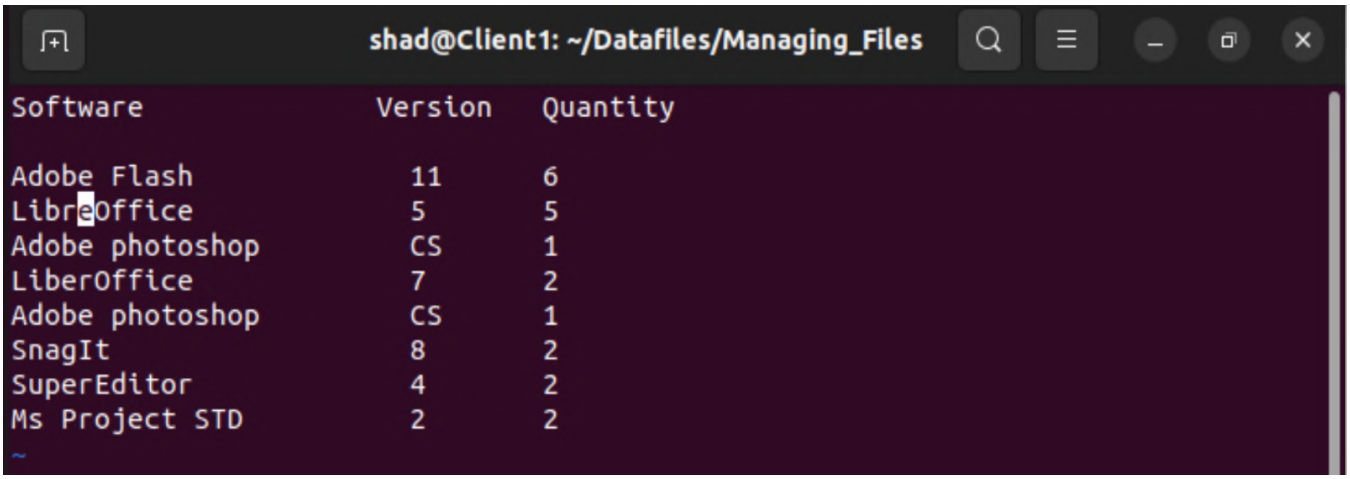
3. Move the cursor down to the first occurrence of the text "**LiberOffice**." The proper spelling of this software is LibreOffice. Position the cursor under the first **e** in **Liber**.(You can use the navigation arrows on the keyboard, or use **h** and **l** to move backward and forward one letter at a time.)



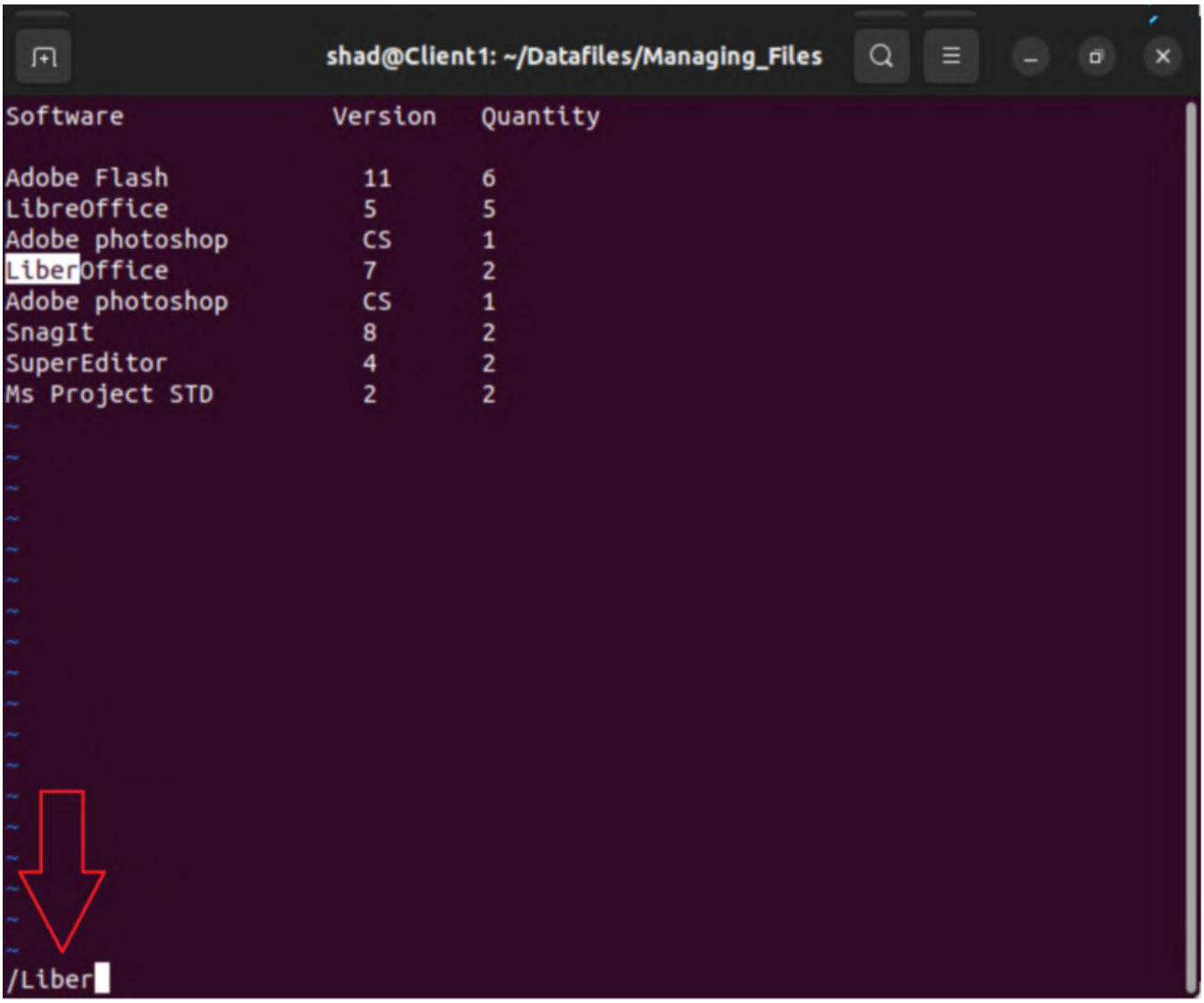
4. Press **x** to cut the letter **e**. When you cut text, the text enters a special place in RAM called the clipboard. It can be pasted from the clipboard in a new location.



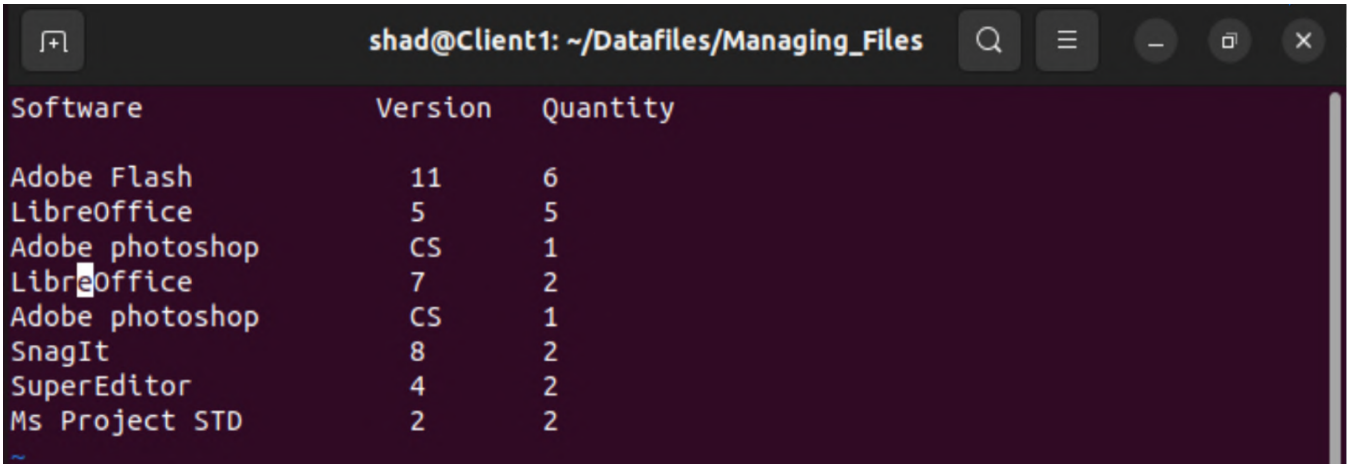
5. With the cursor under the letter **r**, press **p** to paste the **e** you previously cut. The line should now show LibreOffice.



6. Type **/Liber** and press **Enter**. This command searches for the next occurrence of the word “Liber." In a small document like this, you could easily navigate to the next occurrence. However, if this was a long document, searching would be useful.

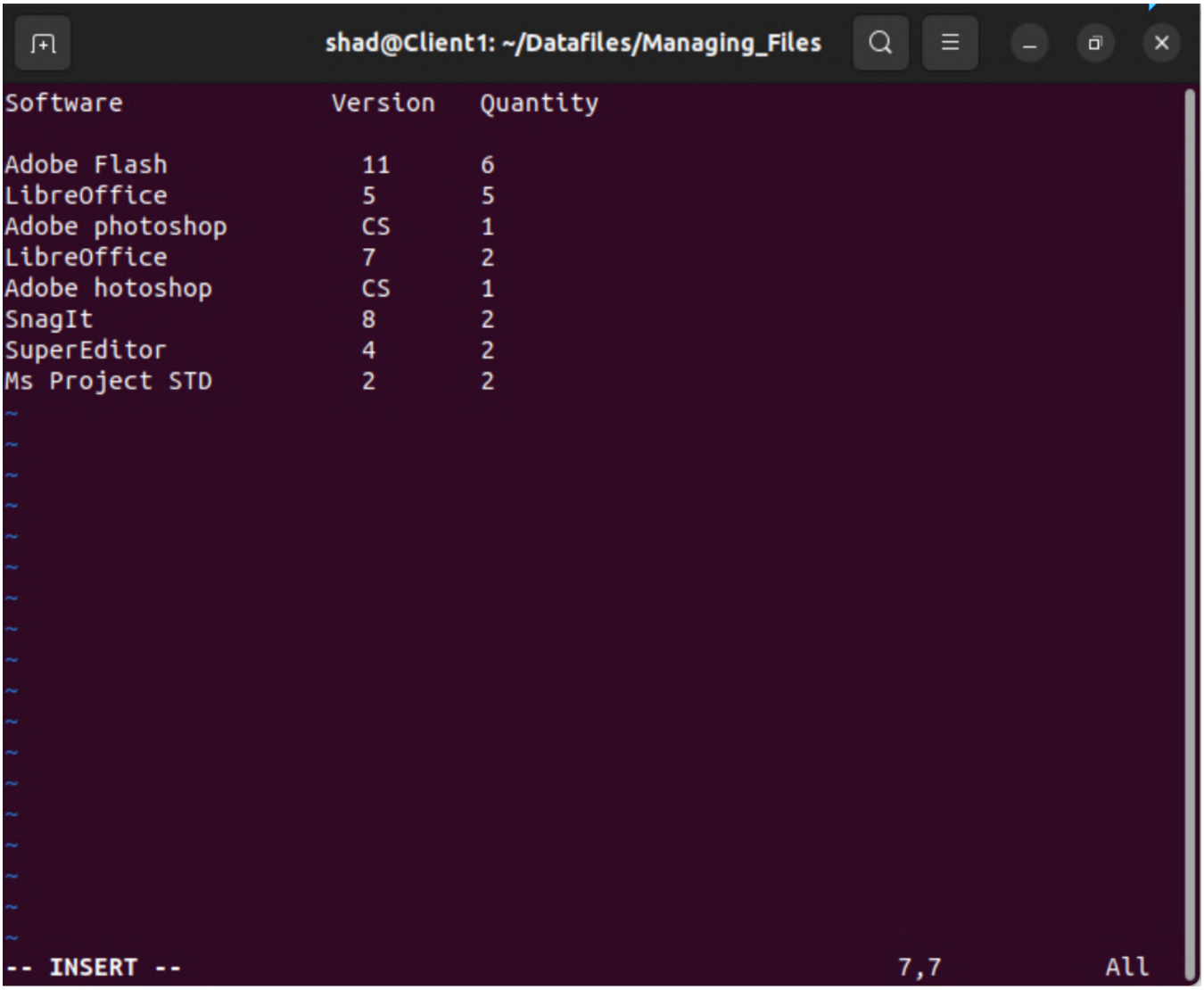


7. Change the text “**Liber**” to “**Libre**”.

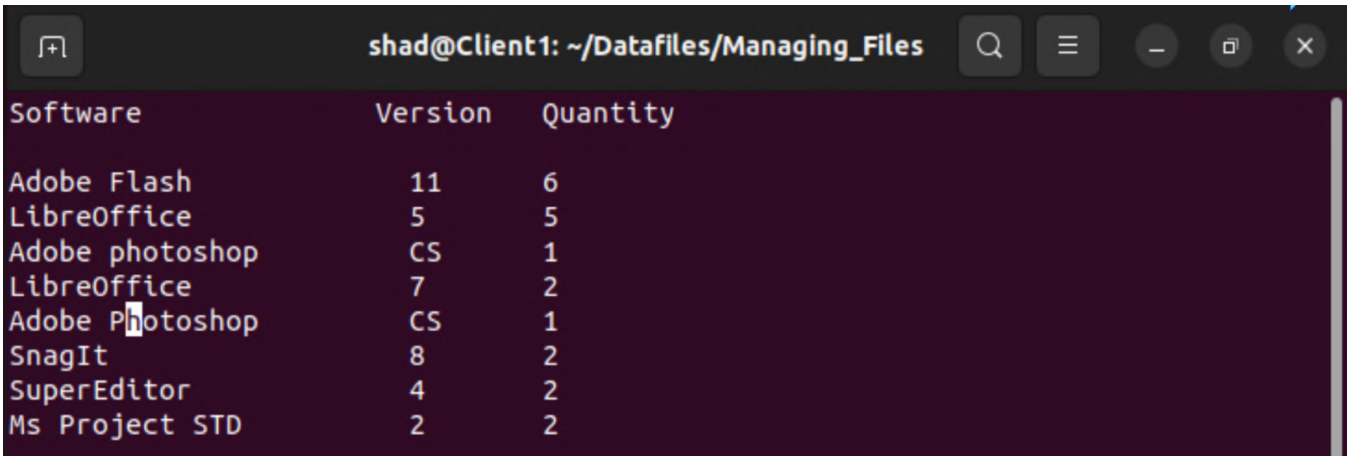


8. Press **j**. This command moves the cursor to the next line. The next error you will correct is to capitalize **photoshop** which is the name of a software application. Place the cursor below the

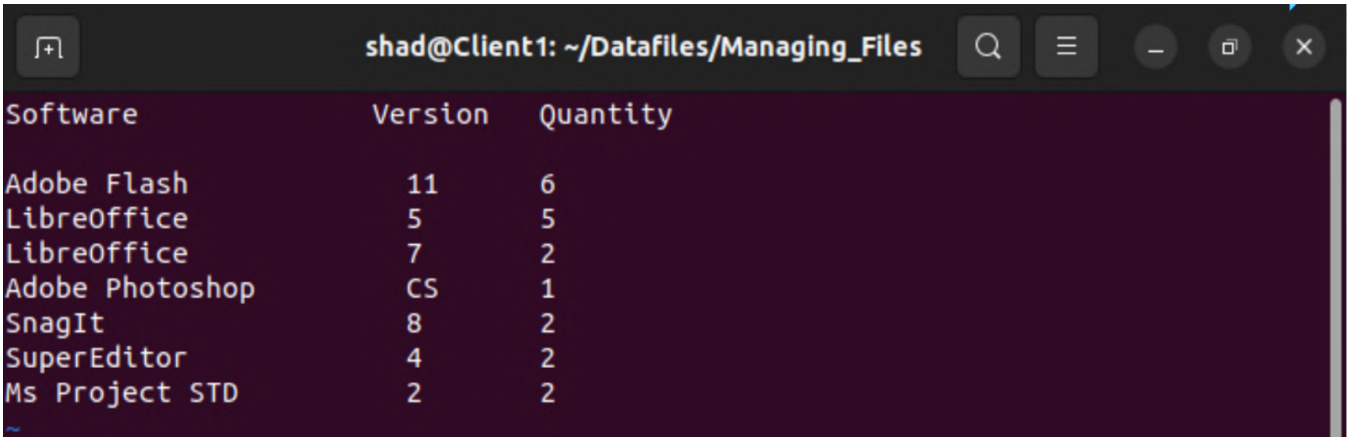
first occurrence of the letter “h” in the word “photoshop.” Press **c** and then press **b**. This deletes the first letter.



9. Type **P**. This changes the word to “Photoshop.” Press **Esc** to return to command mode. Now that the software is spelled correctly, you need to remove the duplicate line.



10. Press the **up arrow** key twice to move up to the duplicate line. Press **d** twice to delete the duplicate line.



11. Type **:wq** and then press **Enter**. This command saves and then closes the file.

Searching Text Files

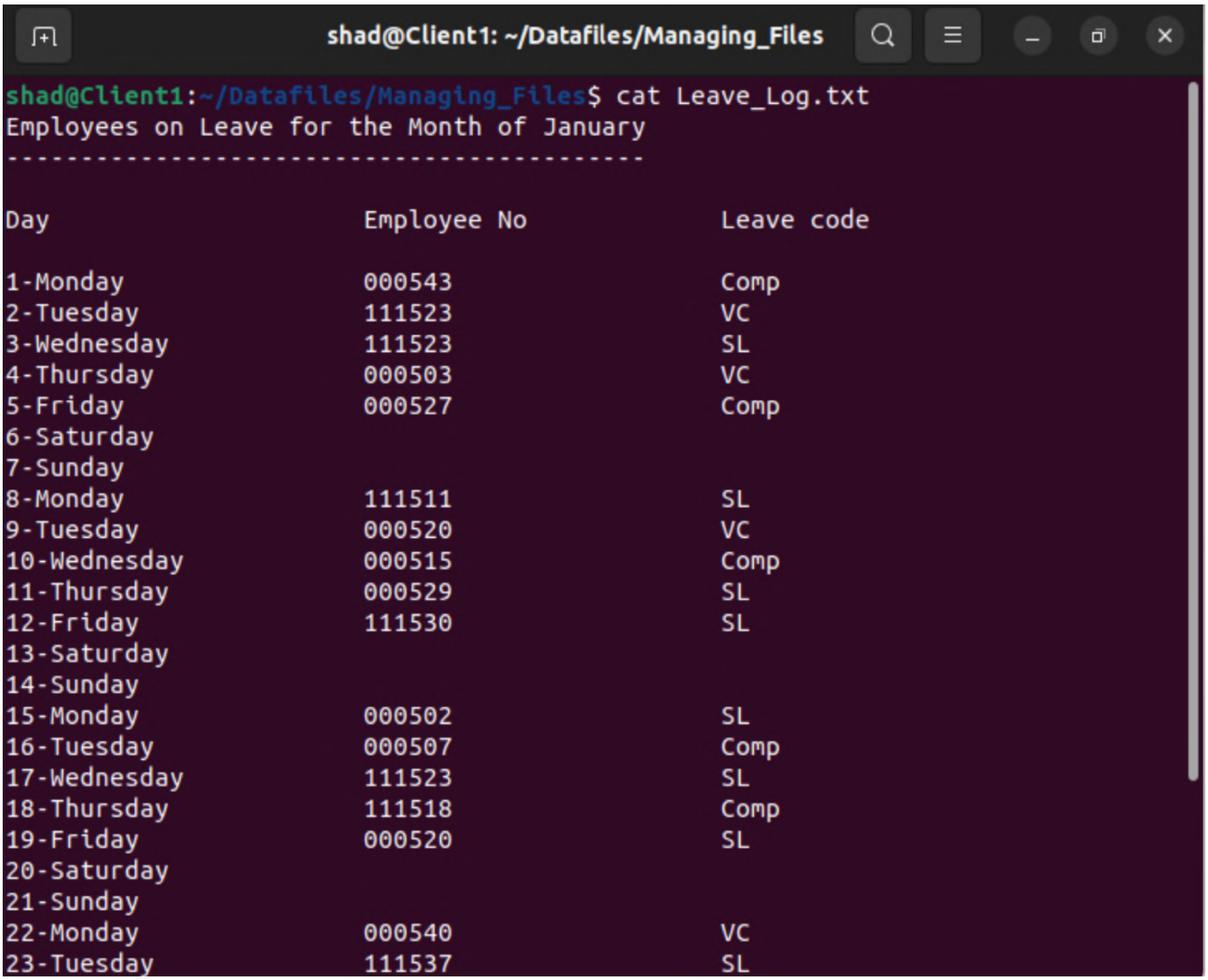
In this task you will be searching the text in a file.
An HR employee in your organization wants to collect the following leave details, for the month of January, for all employees from the software and network departments:

- Employees from the network department who have taken sick leave.
- Employees from the software department who were awarded compensatory leave.
- Employees who have gone on vacation from both the departments.

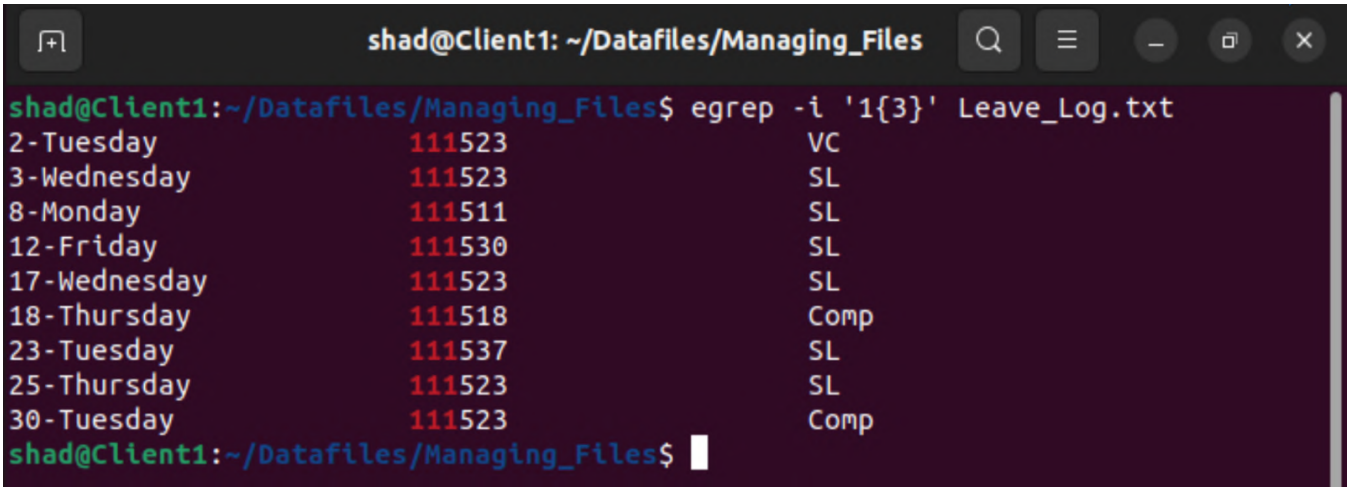
Because the Leave_Log.txt file, located in the Managing_Files directory, has many details, the HR employee finds it difficult to manually go through the file and collect the details.
You need to get the specified details using the department and leave identification codes listed in the following.

- Employee code for the software department: 000
- Employee code for the network department: 111
- Sick leave: SL
- Compensatory leave: Comp
- Vacation: VC

Before you start this exercise, make sure that the Client1 virtual machine is running. Open a terminal and change the working directory to ~/Datafiles/Managing_Files.
1. Type **cat Leave_Log.txt** and then press **Enter**. This command prints the contents of the file on the screen. Notice all the Employee No start with either **111** for the network department or **000** for the software department.



2. Type **egrep -i '1{3}' Leave_Log.txt** and press **Enter**. This command lists all the network department employees who have taken leave.The search will list employees in department **111**. Be sure you enter the number one in the command.



3. Type **egrep -i '1{3}.*SL' Leave_Log.txt** and press **Enter**. This command lists all the network department employees who have taken sick leave.

```
shad@Client1: ~/Datafiles/Managing_Files
shad@Client1:~/Datafiles/Managing_Files$ egrep -i '1{3}.*SL' Leave_Log.txt
3-Wednesday             111523                  SL
8-Monday                111511                  SL
12-Friday               111530                  SL
17-Wednesday            111523                  SL
23-Tuesday              111537                  SL
25-Thursday             111523                  SL
shad@Client1:~/Datafiles/Managing_Files$
```

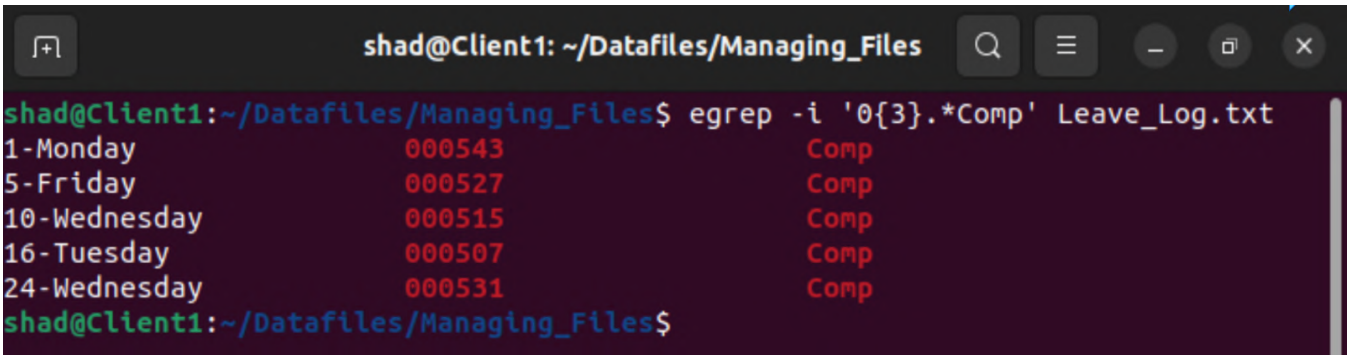
4. Type **grep -i 'VC' Leave_Log.txt** and press **Enter**. This command lists all the employees gone on vacation.

```
shad@Client1: ~/Datafiles/Managing_Files
shad@Client1:~/Datafiles/Managing_Files$ grep -i 'VC' Leave_Log.txt
2-Tuesday               111523                  VC
4-Thursday              000503                  VC
9-Tuesday               000520                  VC
22-Monday               000540                  VC
29-Monday               000512                  VC
shad@Client1:~/Datafiles/Managing_Files$
```

5. Type **egrep -i '0{3}' Leave_Log.txt** and press **Enter**. This command lists all the software department employees who have taken leave.

```
shad@Client1: ~/Datafiles/Managing_Files
shad@Client1:~/Datafiles/Managing_Files$ egrep -i '0{3}' Leave_Log.txt
1-Monday                000543                  Comp
4-Thursday              000503                  VC
5-Friday                000527                  Comp
9-Tuesday               000520                  VC
10-Wednesday            000515                  Comp
11-Thursday             000529                  SL
15-Monday               000502                  SL
16-Tuesday              000507                  Comp
19-Friday               000520                  SL
22-Monday               000540                  VC
24-Wednesday            000531                  Comp
26-Friday               000533                  SL
29-Monday               000512                  VC
shad@Client1:~/Datafiles/Managing_Files$
```

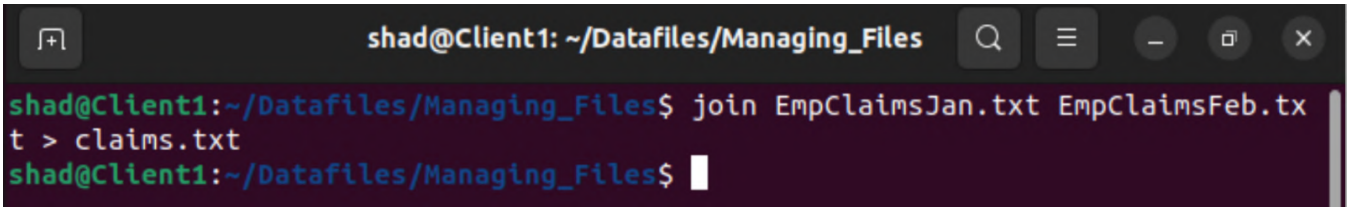
6. Type **egrep -i '0{3}.*Comp' Leave_Log.txt** and press **Enter**. This command lists all the software department employees who have been awarded compensatory leave.



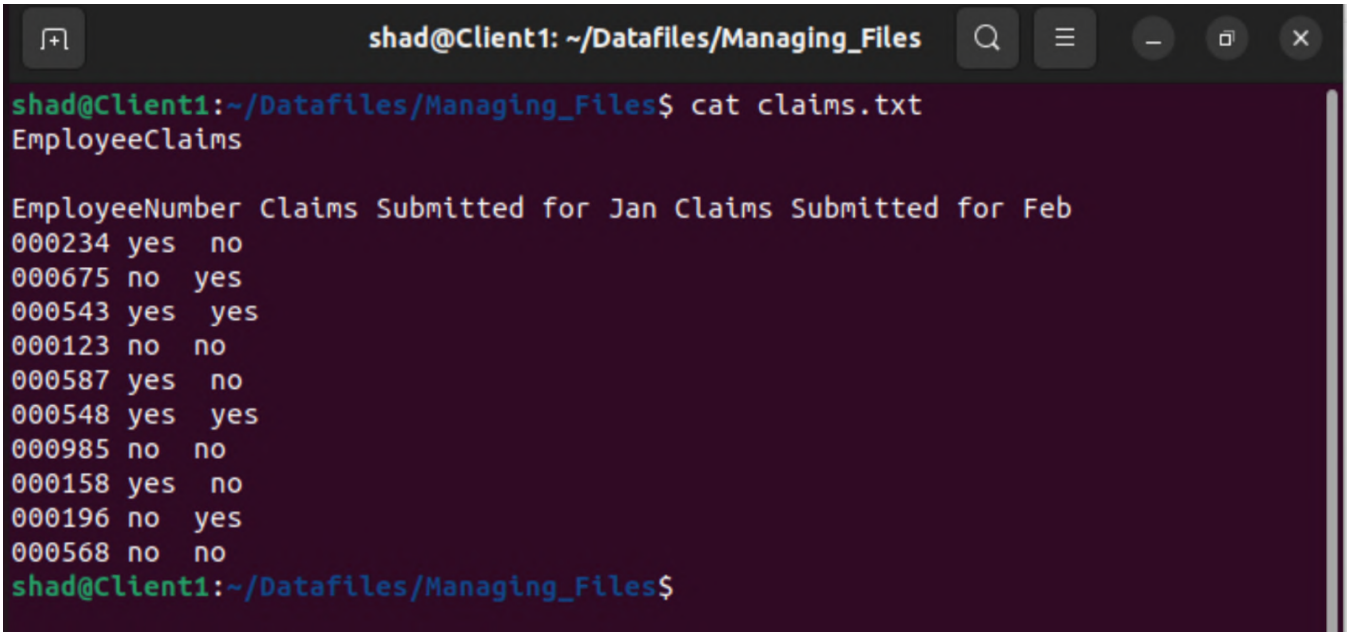
Applying textutil Commands to Modify the Output

A colleague in the finance department wants to create a consolidated bimonthly report for claims submitted by employees. This report would be for the months of January and February, sorted by employee number. You will help create the consolidated report which should be named EmpClaims.txt.The raw data is being kept in two files: EmpClaimsJan.txt and EmpClaimsFeb.txt. Before you start this exercise, make sure that the Client1 virtual machine is running. Open a terminal and change the working directory to ~/Datafiles/Managing_Files.

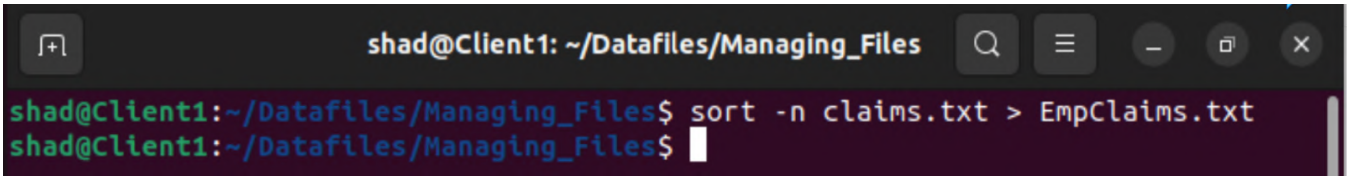
1. Enter **join EmpClaimsJan.txt EmpClaimsFeb.txt > claims.txt** and press **Enter**. This creates a file named **claims.txt** that contains all the claims from both months.



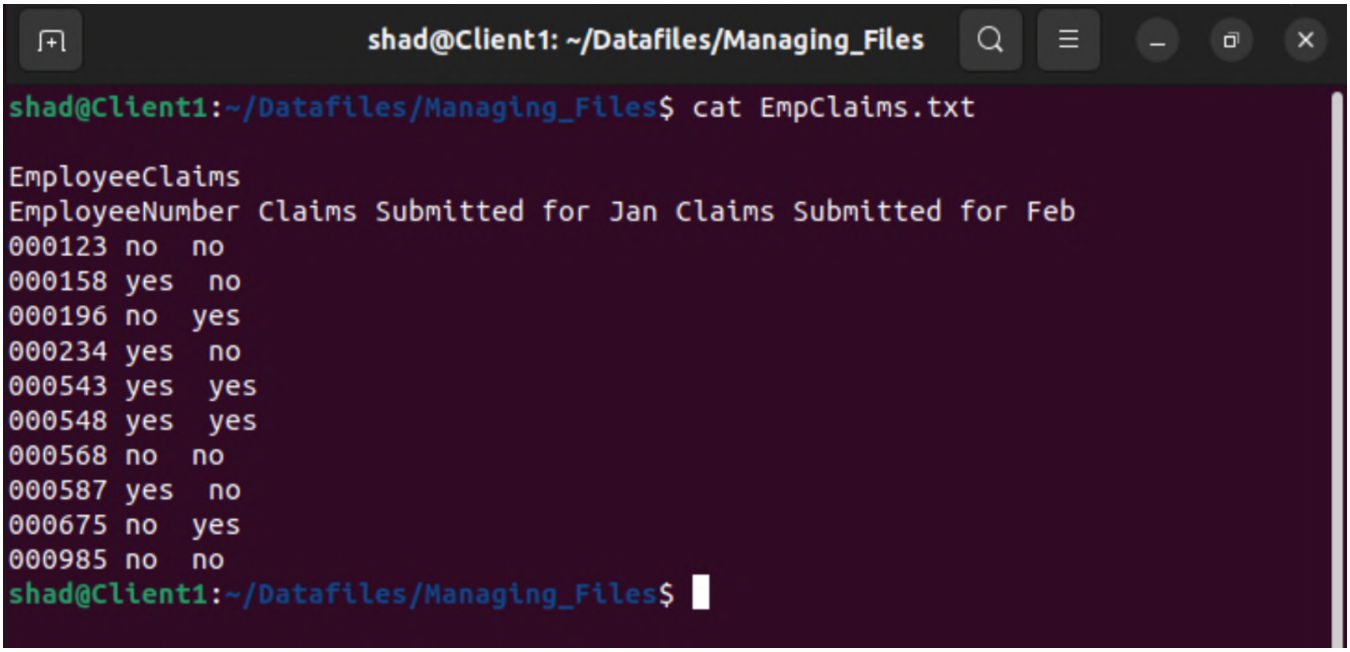
2. Type **cat claims.txt** and press **Enter**. This command prints all the text from **claims.txt** to the screen.



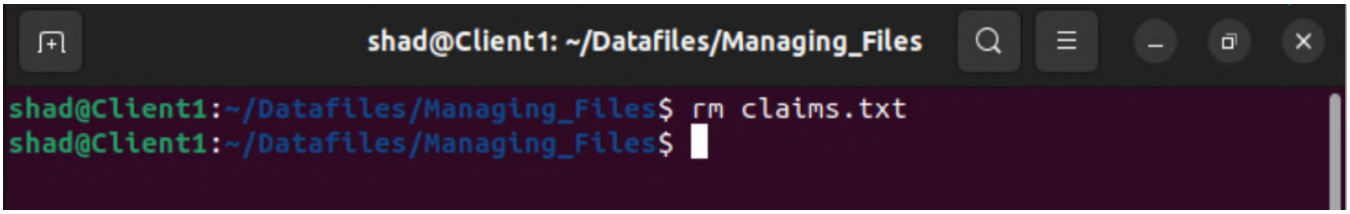
3. Type **sort -n claims.txt > EmpClaims.txt** and press **Enter**. This command sorts the **claims.txt** file in ascending order of the **EmployeeNumber** field and redirects the sorted output to the **Empclaim.txt** file.



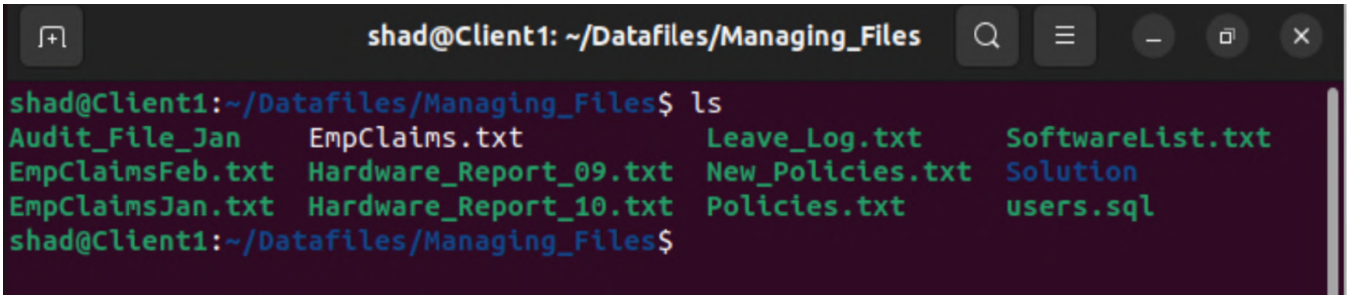
4. Type **cat EmpClaims.txt** and press **Enter**. This prints the contents of the **EmpClaims.txt** to the screen.Verify that the contents of the file are sorted in ascending order of the **EmployeeNumber** field.



5. Now that you have created the desired file, you need to get rid of the **claims.txt** file. Type **rm claims.txt** and press **Enter**.



6. To verify that the file has been deleted, type **ls** and then press **Enter**. The file should not be listed.



User Accounts

Creating User Accounts

A user account is a collection of information that defines a user on a system. It is the representation of the user on a computer. User account information includes the username, a password for the user to log in to the system, groups to which the user belongs, and rights and permissions that the user has to access the system and its resources. When an account is created, it is assigned a unique number that is called a User ID (UID). Usernames and UIDs should be unique. That is, there should never be two users with the same name or UID on one computer.

The useraddand adduser Commands

The **useradd** command is used to add a new user to the operating system. You need to specify the username along with the command to create a new user account.
The syntax of the **useradd** command is **useradd [options]{username}**.
When you create a user using the **useradd** command, it creates the user with the username specified. It also creates a user private group with the same name as the username specified. It does not set a password for the user, and it does not create a home directory for the user.
If you want to create a user, the user private group, and the user's home directory and set a password for the user, you should use the **adduser**command.
The syntax for the **adduser**command is **adduser [options]{username}**.

Special User Accounts

Special user accounts are required to run processes associated with certain services. For example, daemon is a user account that is used to run the daemon service.In special user accounts, the UID value for the users will be less than the default UID value, which is 500. Such special users will not have a home directory. You can create a special user account using the **useradd -r {special user name}** command.

Adding User Accounts by Editing the Password File

Linux allows you to add user accounts by directly editing the **/etc/passwd** file which contains a list of all user accounts. However, this is not recommended because you may damage your system if you accidentally leave something out or alter existing user accounts. If the system is damaged, nobody will be able to log in — not even the root user. In such a case, you will have to reinstall your system and redefine the user accounts.

The id Command

The **id** command is used to display UID and group ID (GID) information. Entering the command with no options displays information about the user who is currently logged in. You can also specify a username as an option to display ID information about a specific user.

The finger Command

The **finger** command is used to display information about users, including login name, real name, terminal name, write status, idle time, login time, office location, and office phone number. Some of these fields may be empty if no information was included when the user account was created. You can also view information

about a specific user by entering **finger [user name]**. Many distributions of Linux do not have the **finger** command installed by default. To add support for **finger** use the **sudo apt install finger** command.

The chage Command

The **chage** command is used to change a number of settings that relate to the password and status of the user account.
The syntax of the **chage** command is **chage [options] {username}**.
When you run the **chage** command with no options, it will prompt you to set:

Setting	Purpose
Minimum Password Age	The minimum number of days that must pass after the user has changed their password before they are allowed to change it again. The default is 0 which means Minimum Password Age is disabled.
Maximum Password Age	The maximum number of days the user may use a password before they must change it. The default is 99999 which is over 200 years. Effectively, this means the Maximum Password Age is disabled.
Last Password Change	The date on which the user last changed their password.
Password Expiration Warning	The number of days before the Maximum Password Age is reached when the user should be warned that their password is going to expire. This reminds the user to change the password a week before the Maximum Password Age is reached.
Password Inactive	This setting indicates whether the password is inactive. The default is -1 which means the password is not inactive. If the value is set to 1, the password is inactive and the account is disabled.
Account Expiration Date	The last date the account can be used. After the account expiration date has passed, the account is disabled.

The **chage** command has some options:

Option	Allows You To
chage -d *{yyyy-mm-dd} {user name}*	Sets the date of the last password change to the date specified.

chage -E *{yyyy-mm-dd} {user name}*	Sets the date when the account should expire.
chage -m {number} {user name}	Sets the Minimum Password Age to the number of days specified. Specifying zero will allow the user to change their password immediately.
chage -M {number} {user name}	Sets the Maximum Password Age to the number of days specified. Specifying 99999 will set the password to never expire.
chage -I {number} {user name}	Sets the number of days the account will be inactive after the password expires. During the inactive period, the user can use the expired password to log in to change the password. After the inactive period expires, the user will not be able to change their own password to reactivate the account.
chage -W {number} {user name}	Sets the warning period to the number of days specified. The user will be warned this number of days in advance of the password expiration date that they should change their password.
chage -l {user name}	Lists the password aging information for the specified user.

The userdel Command

The **userdel** command allows you to modify the system account files, deleting all entries that refer to the login of an existing user. However, it will not allow you to remove an account if the user is currently logged in. You must kill any running processes that belong to an account before deleting the account.
The syntax of the **userdel** command is **userdel [options] {username}**.

The -r Option

The **-r** option will delete the files in the user's home directory, along with the home directory itself. Files owned by this user and located in other locations will have to be searched for and deleted manually.

The usermod Command

The **usermod** command has options that enable you to modify various user account parameters. You can change a user's name, default groups, UID, or passwords.
The syntax of the **usermod** command is **usermod [options] {username}**.
Some of the common **usermod** command options and their descriptions are given in the following table.

Option	Allows You To
usermod -l*{new_username} {oldusername}*	Modify the login name of the user.
usermod -c *“First Last” username*	Modify the user’s full name.
usermod -f *{number of days} {login}*	Modify the number of days for a password to expire and to disable the account permanently.
usermod -u *{new unique user ID} {login}*	Modify the numerical value of a user’s ID, which has to be unique.
usermod -d *{new login directory} login*	Modify the user’s default home directory.
usermod -L *{user name}*	Lock the password and suspend the user account temporarily. However, if the user has some other authentication method configured, they will still be able to log in.
usermod -U *{user name}*	Unlock the password.
usermod -e *{yyyy-mm-dd} {user name}*	Change the expiration date for the user account. After the expiration date, the account will be disabled. Expired accounts can be re-enabled by setting a new expiration date. If you use “” as the expiration date, the account will be enabled indefinitely with no expiration date. Example: usermod -e “” user If you use 1 as the expiration date, the account will immediately expire and be disabled. Example: usermod -e 1 user
Usermod -G {group name} {user name}	Adds the user to the specified group.

Lock User Login

In Linux, you can lock a user's login to temporarily prevent a user from logging in to a system. This is done by disabling the user's password using the **passwd -l** or **usermod -L** command. The user's login is usually locked as a security measure, to prevent unauthorized usage when the user is unavailable.

Default User Accounts

Numerous user accounts are created by default upon system installation. Some of the main user accounts include the following:

- root
- bin
- daemon
- ftp
- sshd
- nfsnobody
- apache
- rpc
- gnome

The root User

Every Linux system has at least one system administrator whose job is to maintain the system and make it available to users. This user is the **root** user. The **root** user can perform any task on the Linux system without restrictions. System administrators are also responsible for adding new users to the system and for setting up their initial environment.

Password & Groups

Groups

A group is a collection of system users having the same access rights. Every user must be a member of a group. Users can also be members of more than one group. Group membership is used to limit access to files and system resources. The **groupadd** command allows you to add a group without creating a new user.
The syntax of the **groupadd** command is **groupadd {group name}**.

Standard Groups

The table lists the standard groups set up by the installation process.

Group	GID	Default Member
root	0	root

bin	1	root, bin, and daemon
daemon	2	root, bin, and daemon
sys	3	root, bin, and adm
adm	4	root, adm, and daemon
tty	5	None
disk	6	root
lp	7	daemon and lp
mem	8	None
kmem	9	None
wheel	10	root
mail	12	mail
man	15	None
games	20	None
gopher	30	None
dip	40	None
ftp	50	None
nobody	99	None

users	100	None

User Private Groups

A User Private Group (UPG) is a unique group that is created by default whenever a new user account is created. This is the primary group of the new user account. Only the new user is a member of this group.

The /etc/group File

The **/etc/group** file contains a list of groups, each on a separate line. Each line consists of four fields for attribute definition, separated by colons. The **/etc/group** file is also called the group database.
Note: The **/etc/gpasswd** file stores the encrypted passwords for groups.

Field	Description
Group name	Stores the name of the group.
Group password	Stores the password of the group in an encrypted form.
GID	Stores the group identifier; similar to a UID for groups. The default GID value is 500.
Members	Stores the names of the members of the group separated by commas.

Group Management

Groups, like users, are identified by a system with a unique number known as GID. In Linux, users can be members of one primary group and multiple supplemental groups. The **groupdel** and **groupmod** commands are useful in managing groups.

Command	Allows You To
groupdel	Delete a group from the system.
groupmod	Change the group's name and the numerical value of the group's ID by modifying the system account files.

The syntax of the **groupdel** command is **groupdel {group name}**.
The syntax of the **groupmod** command is **groupmod -g{GID}**.

Group Account with GID

To add a new group to the system with a name of print_users and a GID of 700, enter **groupadd - g 700 print_users** at the command line.

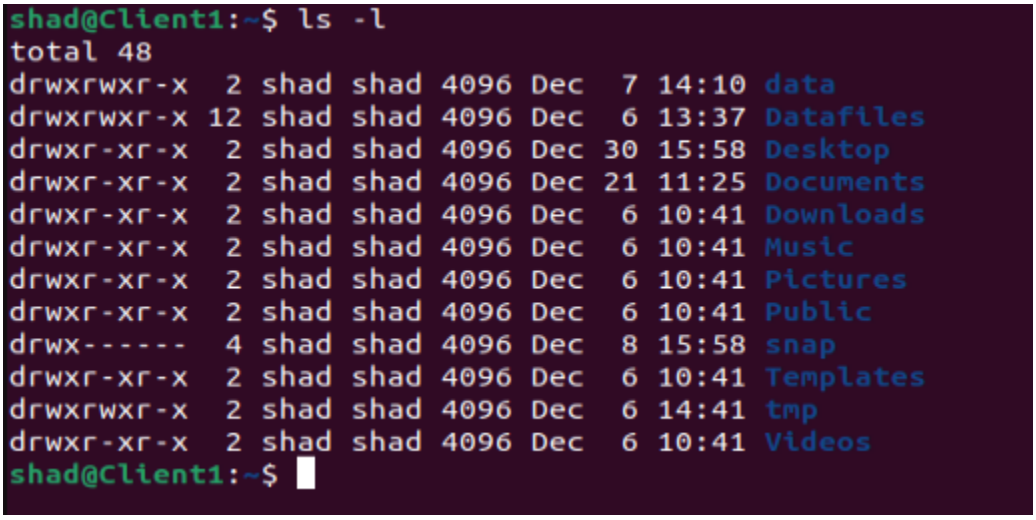
Adding Users to a Group

As with users, the group file can be directly edited to add groups. You can also use the **groupmod-G {group}** command to add users to the group instead of editing the group file. Using the command is a better option as it avoids unnecessary errors that may occur by directly editing the file.

Managing Linux Permissions and Ownerships

Modify File and Directory Permissions

Permissions are access rights assigned to users, which enable them to access or modify files and directories. Permissions can be set at different levels and for different access categories. The **ls -l** command can be used to view the permissions of a file.
The **ls -l** command gives you a long list of the files and directories in your current working directory. Each item in the list contains seven columns. The contents of the columns are described in the following table.



Column Number	Description
1	Permission string. This identifies if the item is a file or directory, the user, group, and other permission assignment, and the access method.

2	Number of links. Files generally have a link count of 1. For directories, the link count is the number of directories under it plus 2; 1 for the directory itself and 1 for the parent. Links are similar to Windows shortcuts; they point to the location where the file exists and allow you to access and view the file.
3	Displays the owner of the file or directory.
4	Displays the group to which the owner of the file belongs. All members of this group have the group permission listed in the permission string. The administrator adds users to a group so that permissions can be assigned to the group instead of to each user.
5	Lists the size (in bytes) of the file or directory.
6	Displays the date and time the file was created or last modified.
7	Displays the file or directory name.

Use the **ls -ld [directory name]** command to list directory entries of the specified directory. The contents of the directory will not be displayed.

The output of the **ls -l** command shows the permission string for a file or directory. The permission string contains 11 characters.

```
shad@Client1:~$ ls -l
total 48
drwxrwxr-x  2 shad shad 4096 Dec  7 14:10 data
drwxrwxr-x 12 shad shad 4096 Dec  6 13:37 Datafiles
drwxr-xr-x  2 shad shad 4096 Dec 30 15:58 Desktop
drwxr-xr-x  2 shad shad 4096 Dec 21 11:25 Documents
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Downloads
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Music
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Pictures
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Public
drwx------  4 shad shad 4096 Dec  8 15:58 snap
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Templates
drwxrwxr-x  2 shad shad 4096 Dec  6 14:41 tmp
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Videos
shad@Client1:~$
```

- The first character indicates the type of file; d for directory and hyphen (-) for file.
- Characters at the second, third, and fourth positions denote permissions of the owner or user of the file or directory.

- Characters at the fifth, sixth, and seventh positions denote group permissions.
- Characters at the eight, ninth, and tenth positions denote permissions for others.
- The final character indicates the access method for the file; period (.) for SELinux security context and plus (+) for any other combination of alternate access methods.

Effective Permissions

Effective permissions are the permissions the subject actually has to the object in a particular situation.
In Linux, effective permissions are calculated as follows:

1. If the user is the user or owner listed on the file or directory, the user has the effective permissions of the permissions listed in the user portion of the permission string.
2. If the user is **not** the owner or user listed on the file but **is** a member of the group listed on the file, the user has the effective permissions of the permissions listed in the group portion of the permission string.
3. If the user is **not** the owner or user listed on the file, and is **not** a member of the group listed on the file, the user has the effective permissions of the permissions listed in the other portion of the permission string.

Permission Levels

Permissions are granted or denied by the owner of the file. The following table lists the levels of various permissions and their description.

Level of Permission	Description
User level r/w/x permission	Only the owner can read, write, and execute the file.
Group level r/w/x permission	Only the members of groups to which the file belongs to can read, write, and execute the file.
Other level r/w/x permission	All users can read, write, and execute the file.

Access Categories

Access categories in Linux permissions decide how Linux interprets the permissions of a file. If a user's UID matches the permissions of the file, the user level permissions are applied. If the GID of the user matches the permissions, group permissions are granted. If neither of the permissions match, the general permissions for others are applied. The symbols for the access categories are listed in the following table.

Access Category	Description
u	Modifies permissions at user level.

g	Modifies permissions at group level.
o	Modifies permissions for other users.
a	Modifies permissions for all users globally.

Chmod command

Chmod - character mode
The **chmod** command enables you to modify default permissions of a file or directory. Only the owner of the file or the system administrator can change the permissions of the file or directory.
The syntax of the **chmod** command is **chmod [options] {mode} {file name}**.

Example-
chmod u+w permissions.sh -> gives the user write permission to the file permissions.sh
+ - grants the permission, - revokes the permission

chmod u+w, g-w,o-x permissions.sh
The above grants write permission to user, revokes write permission of group and revokes execute permission for other on the permissions.sh file

Another way is-
chmod u=rwx,g=rx,o=wx permissions.sh
This Command is like, Hey no matter what the permissions are currently, set user to read, write and execute and set group permissions to read and execute and set all others as write and execute on the file permissions.sh

We can also do this using numeric mode
So, 4- read, 2- write and 1- execute

chmod 654 permissions.sh

So first place is for user permissions, second for group and third for others
Now, add up the permissions you want for users, group and others and just put the number

In the above command 6 for user means 4(read)+2(write) and 5 for group means 4(read)+1(execute) and 7 for others means 4(read) + 2(write)+ 1(execute) on the file permissions.sh

Default File and Directory Permissions

In Linux, default permissions are assigned to newly created files and directories based on user privileges. For files created by the root user, the default permission is 644, which means that the root user has read and write permissions,

while group users and others will have only read permission. For directories created by the root user, the default permission is 755, which means that the root user has read, write, and execute permissions, while group users and others will have only read and execute permissions.

In the case of users with limited access rights, Linux assigns a permission of 664 for newly created files and 775 for newly created directories.

These default permissions are determined by the user file creation mask, or **umask**. However, the default permissions may be altered by the root user.

The **umask** command automatically alters the default permissions on newly created files and directories. The default permissions on newly created files and directories can be changed for security reasons. The syntax of the **umask** command is **umask {number}**.

If you enter **umask** without specifying any numbers, the command shows the current umask value.

Chown command

The **chown** command can be used to change the owner, the group, or both for a file or directory. The following table describes how to use this command.

Command Syntax	Description
chown {user name} {file name}	Changes the owner but not the group. Example: chown user file1.txt
chown {user name}:{group name} {file name}	Changes the owner and the group. Example: chown user:group1 file1.txt
chown {user name}: {file name}	Changes the owner and the group. The group will be changed to the specified user's login group. Example: chown user: file1.txt (Note: there must be a space after the colon before the file name.)
chown :{group name} {file name}	Changes the group but not the owner. This is the same as using the chgrp command. Example: chown :group1 file1.txt

Recursively Changing Ownership

You can combine the **chown** command with the **-R** option to recursively change ownership through a directory structure. You can also use metacharacters to change ownership of groups of files at the same time. By default, the root account has permission to change ownership. If logged in with a user account, you may need to use the **sudo** command with the **chown** command.

```
shad@Client1:~/Desktop$ ls -l
total 12
-rwxrw--wx 1 shad shad 345 Dec 27 18:52 ActiveEmp.txt
-rw-rw-r-- 1 shad shad 336 Dec 27 23:58 commands.txt
-rw-rw-r-- 1 shad shad  39 Dec 27 17:45 MyTest.txt
shad@Client1:~/Desktop$ sudo chown root ActiveEmp.txt
[sudo] password for shad:
shad@Client1:~/Desktop$ ls -l
total 12
-rwxrw--wx 1 root shad 345 Dec 27 18:52 ActiveEmp.txt
-rw-rw-r-- 1 shad shad 336 Dec 27 23:58 commands.txt
-rw-rw-r-- 1 shad shad  39 Dec 27 17:45 MyTest.txt
shad@Client1:~/Desktop$
```

In this example, the user **shad** is the owner of the **ActiveEmp.txt** file. After executing the **sudo chown root ActiveEmp.txt** command, the owner is changed to the **root** user.

You can also use to change the group as **sudo chown naveed:nav ActiveEmp.txt**
The above will change the owner to naveed and also the group to nav

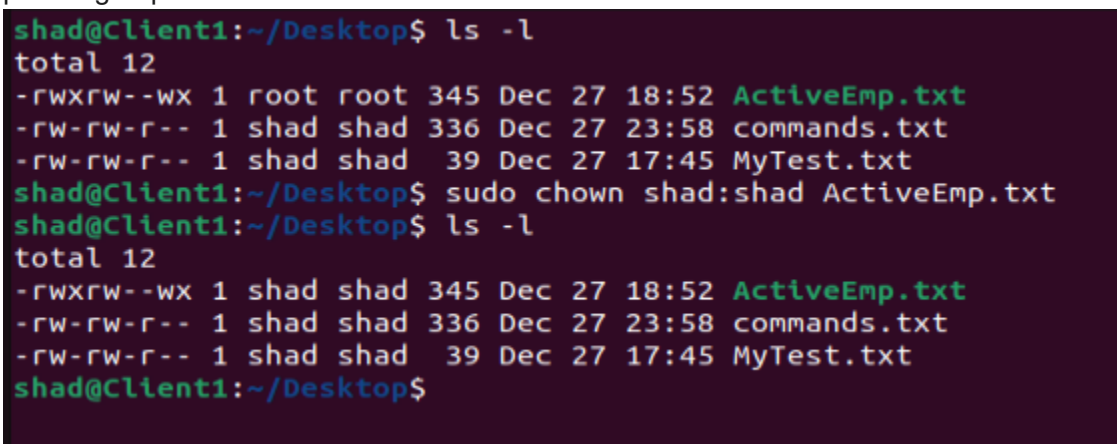
You can also use **sudo chown :nav ActiveEmp.txt**
This will keep the owner intact but just change the group

Changing Group Ownership

The **chgrp** command is used to change the group ownership of a file or directory. The syntax of the command is **chgrp {group name} {filename}**.

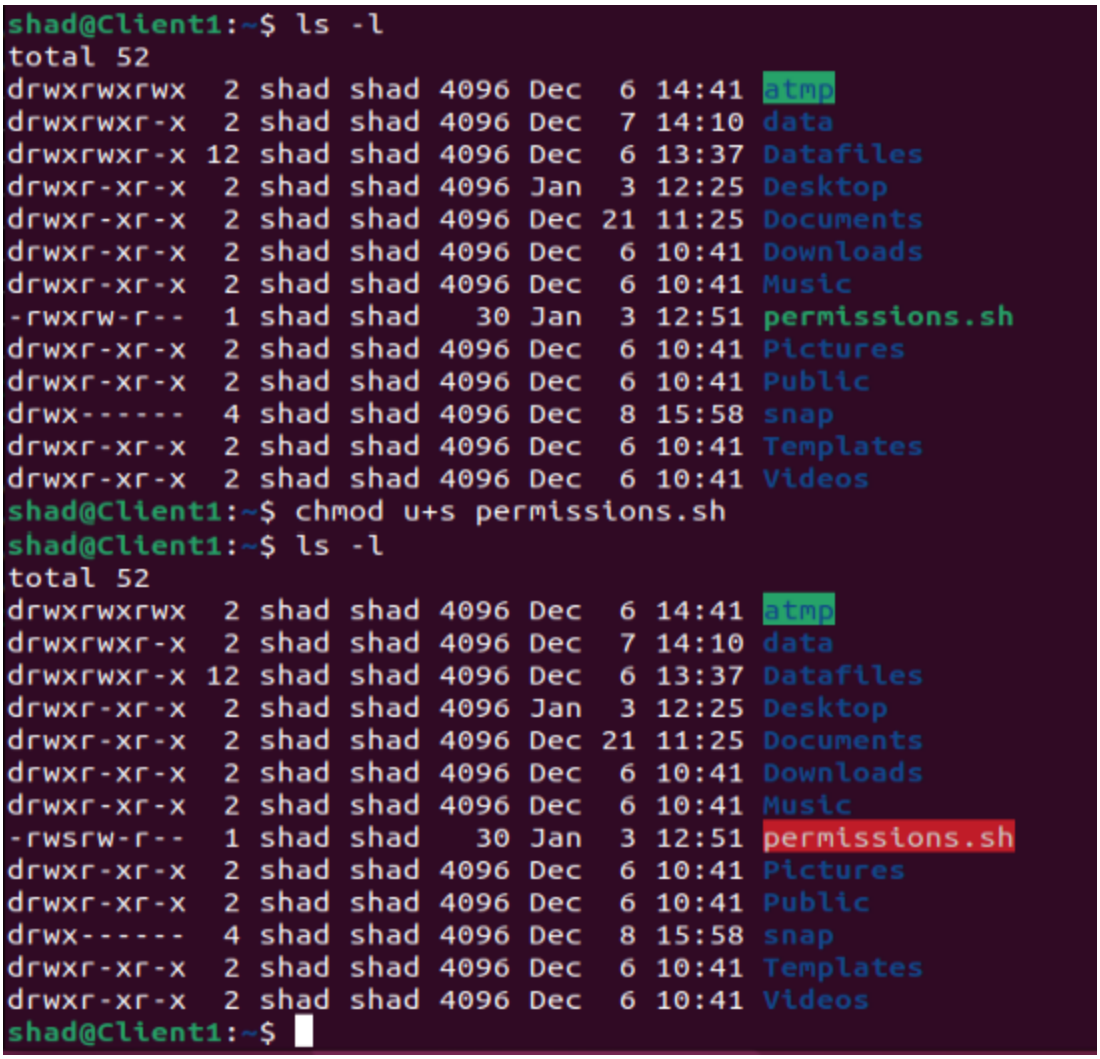
```
shad@Client1:~/Desktop$ ls -l
total 12
-rwxrw--wx 1 root shad 345 Dec 27 18:52 ActiveEmp.txt
-rw-rw-r-- 1 shad shad 336 Dec 27 23:58 commands.txt
-rw-rw-r-- 1 shad shad  39 Dec 27 17:45 MyTest.txt
shad@Client1:~/Desktop$ sudo chgrp root ActiveEmp.txt
shad@Client1:~/Desktop$ ls -l
total 12
-rwxrw--wx 1 root root 345 Dec 27 18:52 ActiveEmp.txt
-rw-rw-r-- 1 shad shad 336 Dec 27 23:58 commands.txt
-rw-rw-r-- 1 shad shad  39 Dec 27 17:45 MyTest.txt
shad@Client1:~/Desktop$
```

In this example, the user private group **shad** is the group assigned to **ActiveEmp.txt**. After the **sudo chgrp root ActiveEmp.txt** command, the group has been changed to the user private group **root**. In the example below, the **chown** command is being used to reset the owner and group to the user **shad** and the user private group **shad**.



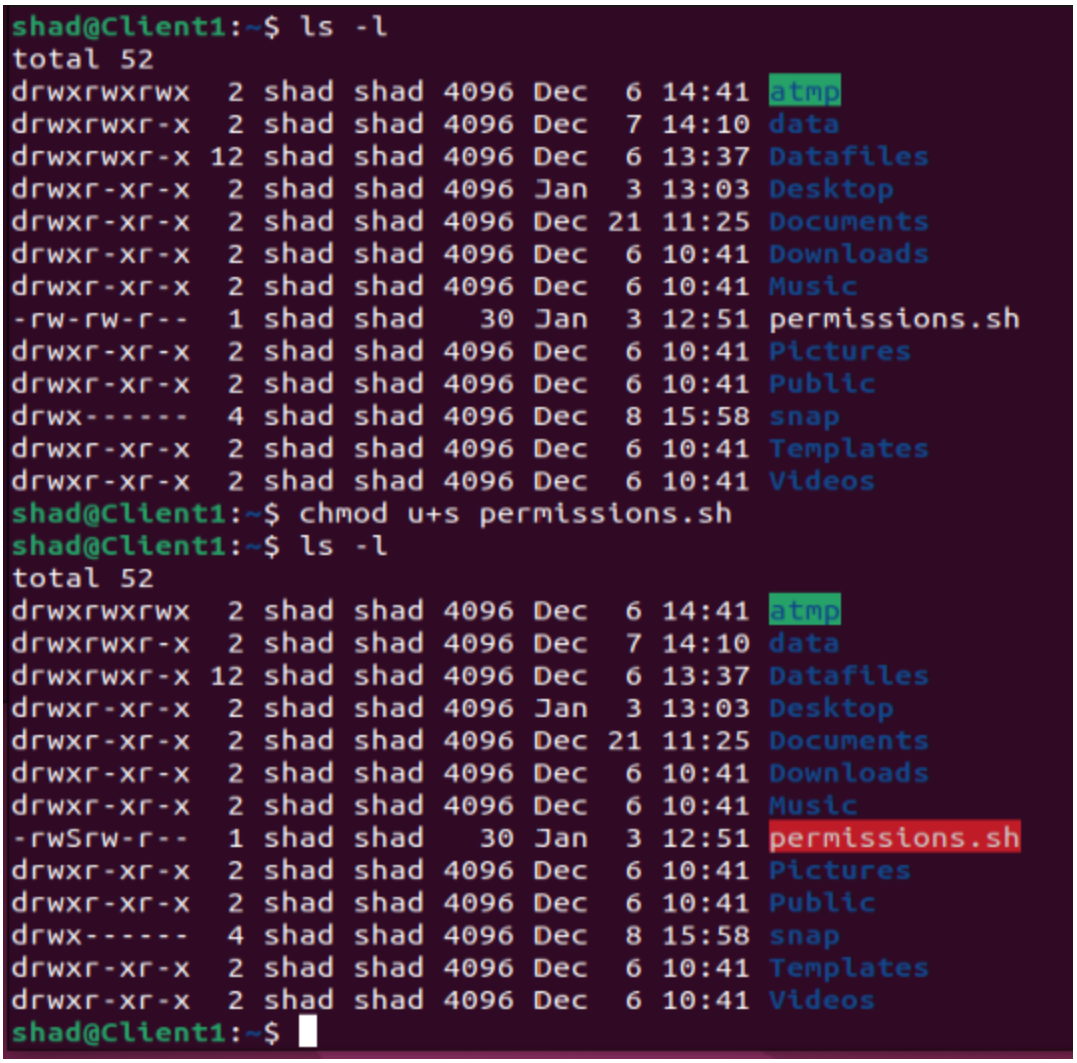
Special Permissions

Special permissions are used in place of the execute permission. They allow users or groups to execute a file they would normally not be allowed to execute.
When the Set User ID (SUID), or setuid, is set, the file is always executed with the permissions of the owner or user listed on the file regardless of which user attempts to execute the file. A lower case “s” in the user permission string indicates that the SUID has been set and the user has the execute permission. An upper case “S” indicates that the SUID has been set but the user **does not** have the execute permission.

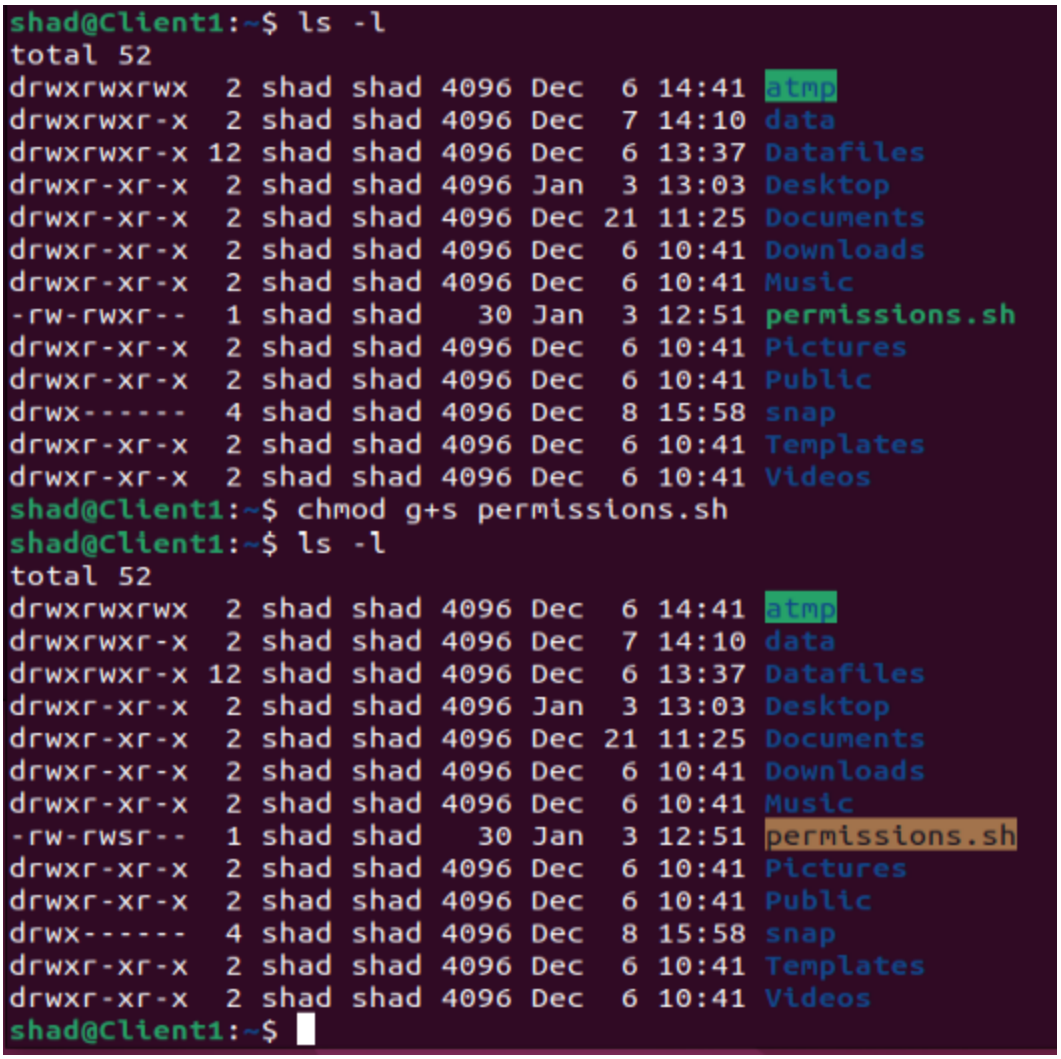


In the above example, you can see from the first **ls -l** that the user has the execute permission because there is an “x” in the fourth position in the permission string. After the **chmod u+s** command, there is an “s” in the fourth position which indicates that the SUID has been set. No matter who executes the file, it will be executed with the shad user’s permission.

In the below example, you can see from the first **ls -l** that the user does not have the execute permission because there is a “-“ in the fourth position. After the **chmod u+s** command, there is an “S” in the fourth position which indicates that the SUID has been set. No matter which user tries to execute the file, it will be executed with the shad user’s permissions which do not include the permission to execute the file. Essentially, the file is unexecutable unless the execute permission comes from group or other.

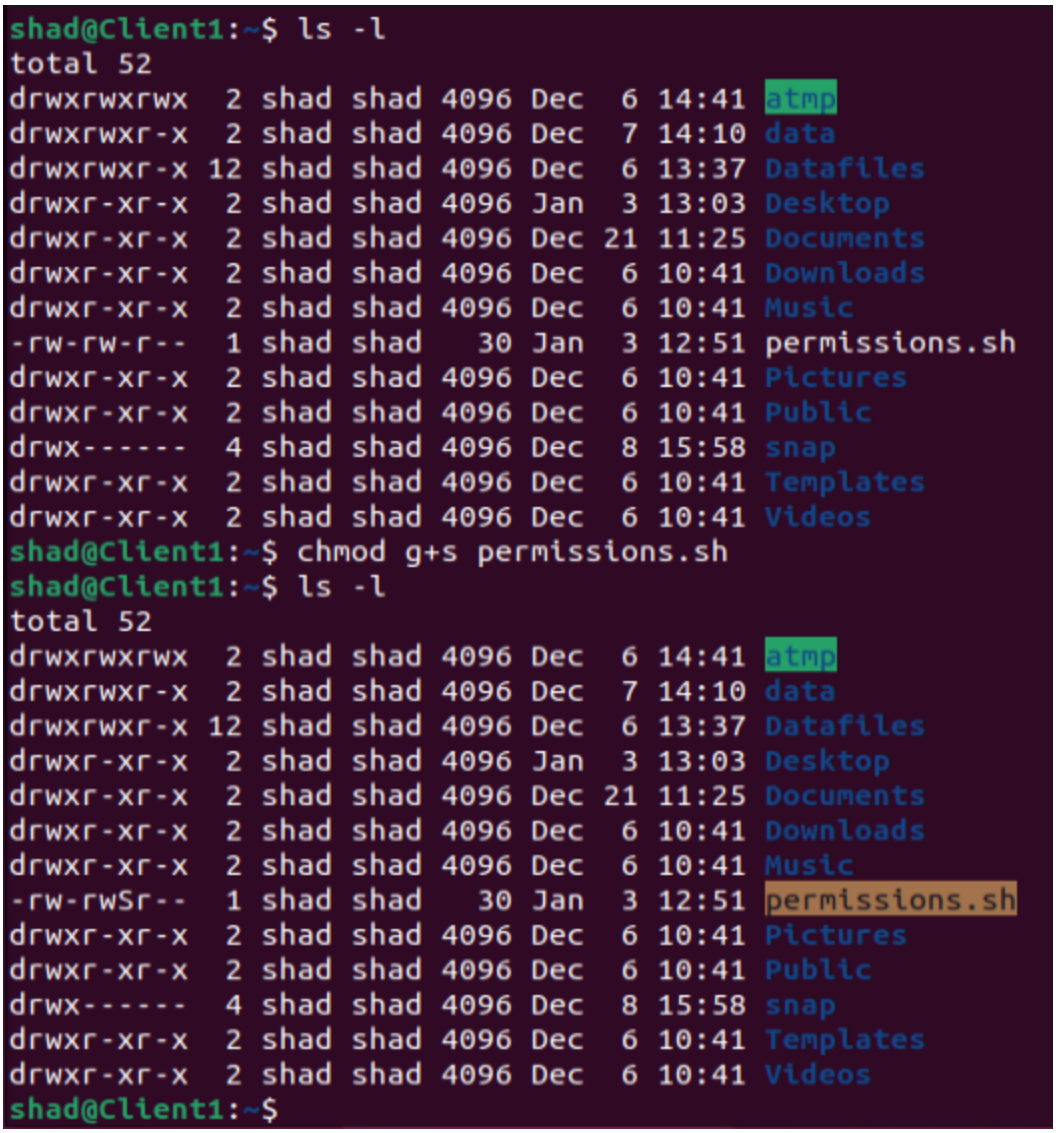


When the Set Group ID (SGID), or setgid, is set on a file, the file is always executed with the permissions of the group listed on the file regardless of which group attempts to execute the file. If the SGID is set on a directory, any files created in the directory will have their ownership set to the ownership of the directory.



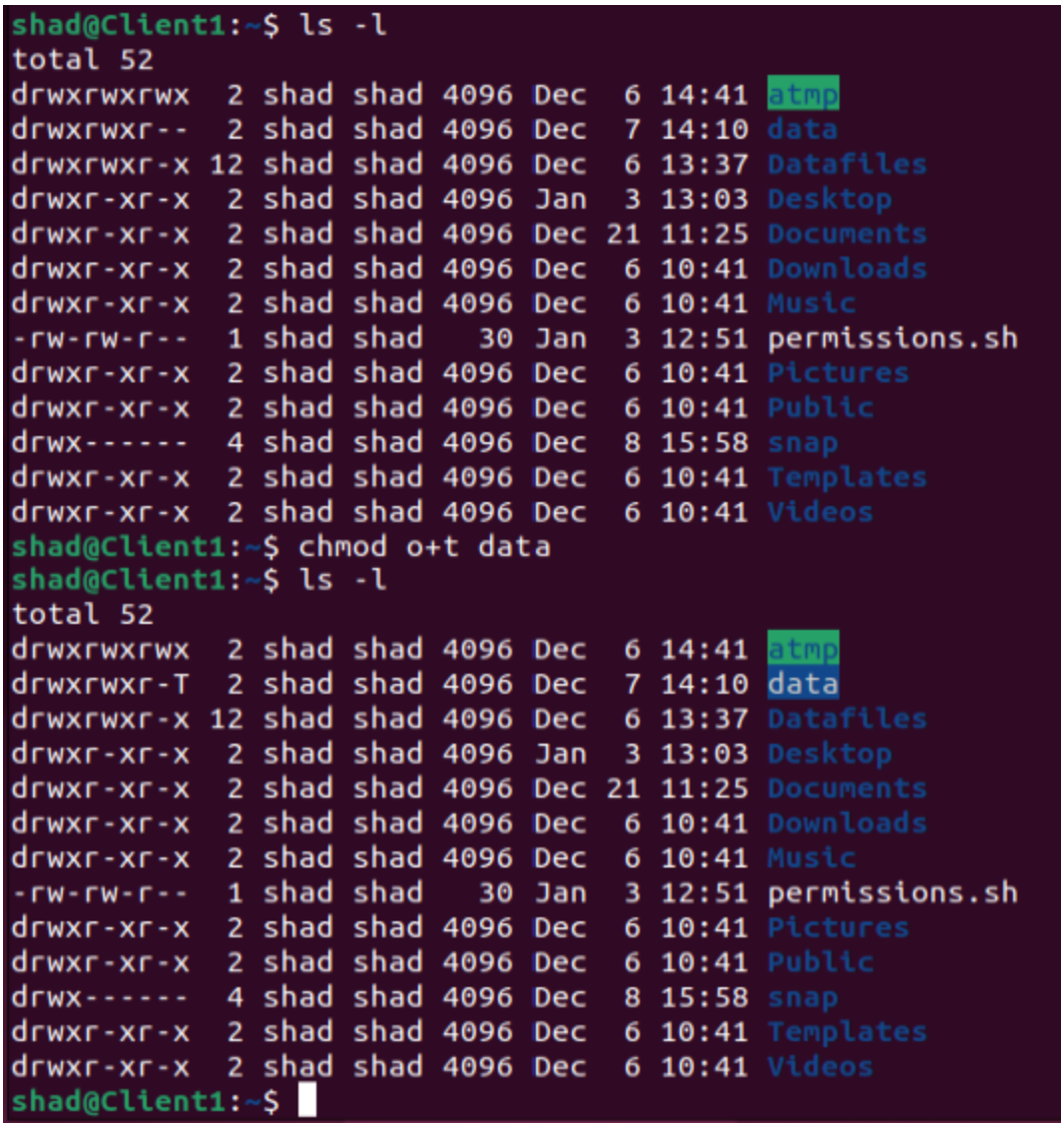
In the above example, you can see from the first **ls -l** that the group has the execute permission because there is an “x” in the seventh position in the permission string. After the **chmod g+s** command, there is an “s” in the seventh position which indicates that the SGID has been set. No matter which group executes the file, it will be executed with the shad group’s permission.

In the below example, you can see from the first **ls -l** that the group does not have the execute permission because there is a “-“ in the seventh position. After the **chmod g+s** command, there is an “S” in the seventh position which indicates that the SGID has been set. No matter which group tries to execute the file, it will be executed with the shad group’s permissions which do not include the permission to execute the file. Essentially, the file is unexecutable by any group. Permission to execute the file would need to come from the user or other permissions.



The last special permission is called the “sticky bit.” This permission does not affect files. It only affects directories. At the directory level, it restricts file deletion. Only the owner of a file (or the root user) can delete files in directories with the sticky bit set. The sticky bit appears as a lower case “t” in the place where normally the execute permission for others would appear in the permission string, at the tenth character, if the directory had the execute permission set for others. It appears as an upper case “T” in the tenth character if the directory did not have the execute permission set for others.

```
shad@Client1:~$ ls -l
total 52
drwxrwxrwx  2 shad shad 4096 Dec  6 14:41 atmp
drwxrwxr-x  2 shad shad 4096 Dec  7 14:10 data
drwxrwxr-x 12 shad shad 4096 Dec  6 13:37 Datafiles
drwxr-xr-x  2 shad shad 4096 Jan  3 13:03 Desktop
drwxr-xr-x  2 shad shad 4096 Dec 21 11:25 Documents
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Downloads
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Music
-rw-rw-r--  1 shad shad   30 Jan  3 12:51 permissions.sh
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Pictures
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Public
drwx------  4 shad shad 4096 Dec  8 15:58 snap
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Templates
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Videos
shad@Client1:~$ chmod o+t data
shad@Client1:~$ ls -l
total 52
drwxrwxrwx  2 shad shad 4096 Dec  6 14:41 atmp
drwxrwxr-t  2 shad shad 4096 Dec  7 14:10 data
drwxrwxr-x 12 shad shad 4096 Dec  6 13:37 Datafiles
drwxr-xr-x  2 shad shad 4096 Jan  3 13:03 Desktop
drwxr-xr-x  2 shad shad 4096 Dec 21 11:25 Documents
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Downloads
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Music
-rw-rw-r--  1 shad shad   30 Jan  3 12:51 permissions.sh
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Pictures
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Public
drwx------  4 shad shad 4096 Dec  8 15:58 snap
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Templates
drwxr-xr-x  2 shad shad 4096 Dec  6 10:41 Videos
shad@Client1:~$
```



The SUID and SGID Permissions

The SUID and SGID commands are powerful tools that enable users to perform tasks without problems that could arise with users having the actual permissions of that user or group. However, these can be dangerous tools too.

While changing the permissions of a file to be either SUID or SGID, the following points should be considered:

- Use the lowest permissions needed to accomplish a task. It is recommended not to give a file the same SUID or SGID as the root user. A user with fewer privileges often can be configured to perform the task.

Lsattr command

The **lsattr** command is used to list the attributes of a file on a Linux filesystem.
The syntax of the **lsattr** command is **lsattr [-RVadv] [file names]**.
The following table describes the options used in the syntax of the lsattr command.

Command Option	Used To
-R	Recursively list the attributes of directories and their contents.
-V	Display the program version.
-a	List all files in directories.
-d	List directories like files, instead of listing their contents.
-v	List the version number of the file.

Chattr command

chattr +i test.txt
Or
chattr -V +i test.txt
This adds the immutable attribute to the file
So this text.txt cannot be removed from here
Now,

The **chattr** command is used to change the attributes of a file on a Linux filesystem.
The syntax of the **chattr** command is **chattr [-RV] [-v version] {[mode]{file names}}**.
The following table lists the description for the options used in the syntax of the chattr command.

Command Option	Used To
-R	Recursively change the attributes of directories and their contents.

-V	Display the output of the chattr command and print the program version.
-v {version}	Set the version number of a file.

Manage jobs and Background Processes

Processes

A process is an instance of a running program that performs a data processing task. A process consists of a sequence of steps stored on a system; these steps convert input data to output data. Processes can be subdivided into threads. Every process is assigned a uniqueProcess ID (PID) and includes time limits, shared memory, or child processes. Processes may run in the foreground or background of the system.

The Process ID

Whenever a process is started, the system allocates a unique PID to identify the process. Also, every process inherits the User ID (UID) and Group ID (GID) of the user who starts the process. This is similar to the ownership of files and directories on the Linux filesystem.

The init Process

The first process, called **init** in Linux, is started by the kernel at boot time and never terminates. The PID of the **init** process is always 1. In modern Linux systems, the **init** is often **systemd**.

Foreground Processes

A foreground process is a program with which a user interacts at a particular time. Only one foreground process can be run at a time. As the user switches between programs, whatever program the user is interacting with becomes the foreground process. A foreground process is initiated by entering a command at the prompt or by clicking a shortcut in the Graphical User Interface (GUI).

Background Processes

A background process is a program that is not running in the foreground. Background processes enable Linux to run multiple processes at the same time. While the user is interacting with the foreground process, any number of programs can run as background processes. The shell does not have to wait for one process to end before it can run more. A process can be run in the background by adding an ampersand (&) separated by a space to the end of the command.

Daemons

Daemons always run as background processes that never require user input. Other processes remain in the background temporarily, while the user is busy with the current foreground process.

The Program and Process Relationship

A program is a set of instructions describing how to carry out a task. A command that resides on your system is a program. When you enter a command at the prompt, a set of instructions perform a task. A process is a program that executes instructions. The operating system creates a process to carry out that task. Processes have unique identities and exist until their tasks are completed. When the task is completed, the process is terminated.

Multitasking

Multitasking is a method of allowing the operating system to run concurrent programs simultaneously without degrading system performance. Multitasking enables several programs to share the same system resources. Processes spawned by multitasking are all active at the same time. They are not in a sequence or a suspended state waiting to be run. Processes placed in a multitasking state remain active until completed, unless terminated or suspended by the user. One or more users may run multiple tasks on a system.

The Jobs Table

You can display the jobs table using the **jobs** command. The jobs table is a table containing information about jobs running in the background. It contains entries only for those jobs that are running in the current shell. The jobs table contains a numeric label for each job indicating the order in which the jobs were started. In addition, the jobs table includes a plus sign (+) to designate the current or the most recently started job and a minus sign (-) to designate the job that was started just prior to the most recent job. It also includes the status and name of each job.
Note: The job name listed in the jobs table is the command that initiated the job. The plus (+) and minus (-) signs indicate only the order in which jobs are started. All jobs, however, are run simultaneously.

Job Status

There are four possibilities for the status of a job.

Status	**Description**
Running	An active job.
Stopped	A job that is suspended.
Terminated	A job that is killed.

Done	A completed job.

Jobs in a New Shell

Any job that a user placed in the background will appear in that user's jobs table, but other users' jobs will not appear. If you were to start a new shell, the jobs table for the new shell will be empty. However, the jobs started in the previous shell will continue to run.

Suspend vs. Terminate a Process

The **Ctrl+Z** key combination suspends a job. The **Ctrl+C** key combination terminates or kills a job. If you display the jobs table after you press **Ctrl+Z** to suspend a job, you will see that the current job is in a suspended state (labeled in the jobs table as "Stopped"). Although the jobs table lists jobs running in the background, a foreground job that gets suspended appears in the jobs table to remind the user that there is a suspended job waiting to be restarted or terminated. Refer to the following table for a summary of job control commands.

Action	**Foreground**	**Background**
Suspend a job	**Ctrl+Z**	Bring to foreground, then press **Ctrl+Z**
Terminate a job	**Ctrl+C**	kill %#

Restarting a Suspended Job

The **bg** command, with the syntax **bg {%#}** (where # is the job number) can be used to restart a specified background job that has been suspended. If there is only one job running in the background, then you do not have to specify the number. You can type **bg %** to restart it.

Bringing a Job to the Foreground

If you need to bring a job from the background to the foreground, use the **fg** command, with the syntax **fg {%#}**(where # is the job number). You do not have to enter a number after the percent sign if there is only one job running in the background.

Tool	**Tool Enables You To**
jobs	View the status of the jobs running in the background.
Ctrl+Z	Halt a running process temporarily.

fg {%job number}	Bring the specified process to the foreground.
bg {%job number}	Send the specified process to the background.
kill {%job number}	Terminate the specified process.

The Process Table

The process table is a record that summarizes the current running processes on a system. It enables the administrator to keep track of all processes run by different users. Some of the details displayed in the process table include the PID, the size of the program in memory, the name of the user who owns the process, and time.

The Process Table vs. the Jobs Table

The process table has options that are different from the jobs table. The process table can display all processes running on the system irrespective of which user started it, including system processes started automatically at boot time. However, the jobs table shows only the processes started in a user's current shell. Also, the unique PIDs of processes are displayed in the process table, while the jobs table shows only their job number according to the order in which they were started. In the jobs table, only the original process is displayed as an entry, but in the process table, the original process and all subsequent processes that were started are displayed. So, a single entry in the jobs table may have more than one corresponding entry in the process table. Certain job control commands can be applied only by referring to processes by their job number.

The ps Command

The **ps** command invokes the process table. When the command is run without any option, it displays the processes run by the current shell with details such as the PID, the terminal associated with the process, the accumulated CPU time, and the command that started the process. However, different options may be used along with the command to filter the displayed fields or processes.
The syntax of the **ps** command is **ps [options]**.
The **ps** command supports several options. Some of the important options are listed here.

Option	Description
a	Lists all user-triggered processes.
-e	Lists all processes.
-l	A job that is killed. Lists processes using a long listing format.

u	Lists processes along with the user name and start time.
r	Excludes processes that are not running currently.
x	Includes processes without a terminal.
T	Excludes processes that were started by any terminal other than the current one.

Note: Unlike many commands in Linux, the **ps** command supports options with and without a hyphen before them. However, the function of the same options with or without a hyphen may differ greatly. Some common **ps** command options can be used to select a specific set of processes.

Option	**Used To**
-U {*user name*}	Display the processes based on the specified user.
-p {*PID*}	Display only the specified process associated with the PID.
-C {*command*}	Display all processes by command name.
--tty {*terminal number*}	Display all processes running on the specified terminal.

Fields Displayed by the ps Command

Various options display different fields. Several fields can be displayed using the ps command.

Field	**Description**
PRI	Process scheduling priority. Processes with low priority have higher numbers.
NI	Process nice value. Processes using less CPU time have higher numbers.
SIZE	Virtual image size.
RSS	Physical memory in KB.

WCHAN	Kernel function in which the process resides.
STAT	Status. Values include R (running), T (stopped), D (asleep and uninterruptible), S (asleep), Z (zombie), and N (positive nice value).
TT	The TTY or terminal associated with the process.
PAGEIN	The number of major page faults.
TRS	Resident text size
SWAP	Number of KB of swap used.
SHARE	Amount of shared memory.

Child Processes

A process created by a running process is called a child process. The process table contains both parent processes and child processes. There may be several levels of processes. The parent process can spawn a child process, the child process can spawn another child process, and so on. The parent process must be running for the child processes to run. Parent processes are assigned a unique Parent Process ID (PPID).

Identifying Child Processes

Identifying child processes is not an easy task, especially if there are multiple processes and child processes running at the same time. By examining the order of the PIDs, you may be able to determine the order in which the processes were created and infer which processes are related.

The pstree Command

The **pstree** command enables you to list the processes running on a Linux system in a tree-like format. This helps you track parent and child processes. All processes are listed as child processes to init and this is represented by the initial branching. The processes started within a shell will branch out of the shell's parent process.

Process Identification Commands

Process identification commands enable you to extract information about a process using its name or some other attribute associated with it.

Command	Description
pidof	Displays the PID of the process whose name is specified and can be used only when the name of the process is known. However, it is recommended that a full path name of the process be given because more than one process could run with the same name. The syntax of this command is pidof*[options] {string}.*
pgrep	Displays the PID of processes that match any given criteria such as the name or UID of the user who invoked it, the start time, the parent PID, and so on. The syntax of this command is pgrep*[options] {process name}.*

pidof Command Options

The pidof command supports only two options.

Option	Used To
-s	Instruct the program to display only one PID.
-c	Instruct the program to display the PIDs that are running from the same root directory.

pgrep Command Options

The pgrep command supports different options by which one or more conditions for search may be specified.

Option	Used To
-f	Specify the full path name of the process.
-l	Print the name of the process along with its PID.
-u {*userid*}	Specify the UID of the user who started it.
-G {*groupid*}	Specify the GID related to the process.

-n	Specify the most recent process.
-o	Specify the oldest process.

Process States

A process state enables you to identify the current stage of a process. It is indicated by a single letter notation in the process table.
The various process states are given in the following table.

State	Description
Uninterruptible sleep (D)	The process is permanently inactive.
Running (R)	The process may be running or ready to be run.
Interruptible sleep (S)	The process is waiting to be run after some specific trigger.
Stopped (T)	The process may be temporarily stopped by a job control tool or because it is being traced.
Dead (X)	The process has been killed. This state is never displayed.
Defunct (Z)	The process has ended, but only after its parent process. This implies that it has not been killed properly and it will remain as a “zombie.”

Signals

Signals are messages sent to a process to perform certain actions. They are used to suspend or terminate processes. Signals may affect only the process specified and its child processes. Signals may be executed, caught, blocked, or ignored by processes.

kill Commands

Different commands are used to send signals to processes to end or “kill” them.

Command	Description

kill	Sends any specified signal, or by default the termination signal, to one or more processes. The PID must be specified as the argument. The syntax of this command is kill*[options] {PID}.*
pkill	Signals processes based on the name and other identifiers as in the pgrep command. The syntax of this command is pkill*[options] {command}.*
killall	Kills all processes by the name specified. The syntax of this command is killall*[options] {command}.*

Note: The **kill** command accepts either the PID or the job number as an argument. So, this command can also be used as a job control tool.

Kill Signal Options

You can either use the kill signal option or its corresponding numerical value to send a signal to terminate a process. The following table lists the most frequently used kill signal options and their description.

Option	**Used To**
SIGKILL or 9	Send the kill signal to a process.
SIGTERM or 15	Send the termination signal to a process.
SIGSTOP or 19	Stop a process.

Using the PID Number to Terminate Processes

You can use the **kill** command with the process table to end processes. By entering kill followed by the PID, you can terminate specific processes.
When you use the **kill** command with the jobs table, you are working only with the jobs that you started. However, the process table may display processes that do not belong to you. As a user, you can use the **kill** command only with processes that you own. As root, you can kill anyone's processes.
There are many options available with the **kill** command. These options are referred to as kill signals. Some processes cannot be eliminated by the **kill** command. To terminate these processes, use the **kill** command with the -9 signal. This terminates the processes immediately.

The top Command

The **top** command lists all tasks running on a Linux system. It acts as a process management tool by allowing users to prioritize, sort, or terminate processes interactively. It displays a dynamic process

status, reflecting real-time changes. Different keystrokes within this tool execute process management actions.

Useful Keys to Manage Processes

The **top** command provides an interactive tool to manage processes by using some simple shortcuts. Some of the frequently used shortcuts are listed here.

Key	Function
Enter	Refreshes the status of all processes.
Shift+n	Sorts tasks in the decreasing order of their PID.
u	Displays processes belonging to the user specified at the prompt.
k	Terminates the process for which you specify the PID.
-n	Renices the process for which you specify the PID.
h	Displays a help screen.
q	Exits the task list.

The nice Command

The **nice** command allows you to assign a priority level to a process. The nice value of a process indicates how “nice” the process is to others in sharing system resources. You can run a command at a priority higher or lower than the command’s normal priority. You must have the root user authority to run a command at a higher priority. The priority of a process is often called its nice value.
The syntax of the command is **nice -n {priority} {command}**, where the priority is specified by a number.

The nice Values of Processes

The niceness of a process may range from -20 to 19, where -20 indicates the highest priority and 19 the lowest. In the absence of an increment value, the nice command assumes an increment of 10 by default. Once lowered, the priority for any process cannot be increased by normal users, even if they own the process. By default, all processes in Linux have a nice value of zero.

The renice Command

The **renice** command enables you to alter the scheduling priority of a running process. When you renice a process group, it causes all processes in the process group to have their scheduling priority altered. When you renice a user, it alters the scheduling priority of all processes owned by the user. By default, the processes affected are specified by their PIDs.
The syntax of the **renice** command is: **renice {priority} {PID} [[-g] [groupid]] [[-u] [userid]]**.

Delayed and Detached Jobs

Delayed and detached jobs are job processes that enable users to put off the start of a job.

Delayed Jobs

A delayed job is one that can be run at some specified time after you issue the command. For example, a CPU-intensive job that can slow down the system is one that you may want to delay for off-peak work hours.

Detached Jobs

A detached job is a job that can be set to run after you log out of the system. For example, a task that will not be completed until after you leave can be set to continue running after you log out of the system.
To delay the start of a job, use the **sleep** command followed by the delay in seconds and the command name. The sleep command suspends any action upon the specified command for the specified number of seconds and then the command specified is executed. The delay can be up to 2,147,483,647 seconds. This is roughly 596,523 hours; 24,855 days; or 68 years so that the amount of time can easily be customized. You may also use the at command to run a command at a specified date and time.

The nohup Command

The **nohup** (no hangup) command tells a program to ignore the hangup signal that was sent while disconnecting. The nohup.out file stores the output of the nohup command, which will normally be displayed on the terminal.
If you have a task that cannot be completed until after you leave work, or if you have a task that is CPU-intensive and may slow the system, you can start the task before you leave and specify that it continues even after you log out of the system. You can do this by using the nohup (no hangup) command. The nohup command should run in the background so that it does not tie up your terminal. To enable a command to run in the background after you have logged out, use the syntax nohup [command] &.

The screen Command

The GNU screen command is a full-screen window manager that multiplexes a physical terminal between several processes, typically interactive shells. The **screen** command is another way that you

can leave work running after you leave the system, which can then be resumed at a later point by reconnecting to your active screen session.
If you have a task that cannot be completed until after you leave work, or if you have a task that is CPU-intensive and may slow the system, you can start the task before you leave and specify that it continues even after you log out of the system. You can do this by using the **screen** command. The screen command will continue to keep the interactive shell open and run your program in the background so that it does not tie up your terminal. When you next connect to your server, you can restore the active screen session via the **screen -r** command.
Note: The screen command is not installed by default. To install it, use the **sudo apt-get install screen** command.

Cron

Cron is a daemon that runs in the background on a Linux system and executes specified tasks at a designated time or date. Cron is normally used to schedule periodically executed tasks defined in the crontab file.
The syntax of the cron daemon is **cron [option] {mail command}**.

Significance of the /etc/cron Directories

Under the /etc directory, you will find directories such as cron.d, cron.hourly, cron.daily, cron.weekly, and cron.monthly. Depending on the frequency of the execution of bash script, you need to place your script file in the cron.hourly, cron.daily, cron.weekly, or cron.monthly directory. If you want to run a shell script for a frequency other than hourly, daily, weekly, or monthly, you need to place the script in the cron.d directory.

Cron Jobs

A task scheduled via cron is called a cron job. These jobs will run either at system level or at user level. The cron jobs that you create for users are stored in the /var/spool/cron/[user name] file. System default cron jobs are stored in the /etc/crontab file. Only a root user can add system level jobs
Scheduling a cron job is accomplished by adding the job to the system-wide /etc/crontab file. The crontab file may also contain environment variables that will be passed to the commands at the time of execution. Jobs in the crontab file are called entries, and they include a time description, the user name to run the command, and the command. The format of a crontab entry is: **{minute} {hour} {day of month} {month} {day of week} {user command}**.
The time fields in the crontab entry are listed here.

Field	Allowed Value
Minute	0-59
Hour	0-23

Day of the month	1-31
Month	1-12 or Jan-Dec
Day of the week	0-7 (0 or 7 is Sunday) or Sun-Sat

In addition to specifying a particular time and day, a pattern can be described by using asterisks (*) to specify all of a particular field. For example, an asterisk in the minute field indicates that the command should be carried out every minute. In addition to asterisks, time ranges are permitted by separating values with a dash (-) and lists of values are specified by separating values with a comma (,).

Common Daily Jobs

The tmpreaper Command

The **tmpreaper** command can be run as a daily cron job that is used to delete files, such as the files in the /tmp directory, which have not been accessed for some time and are utilizing disk space.
The syntax of the **tmpreaper** command is **tmpreaper [options] {hours}**.
The **tmpreaper** command has the following options.

Option	Enables You To
-c	Delete files according to the time they were created or the permissions were changed.
-m	Delete files according to the time they were modified.
-a	Remove all file types, including directories.
-f	Remove files forcefully, overriding all access regulations.
--showdeleted	Show what files and directories are deleted.

The logrotate Command

The **logrotate** command is run as a daily cron job that is used to compress, delete, or mail log files. It may be configured to run on a weekly or monthly basis depending on the log size. The configuration file for **logrotate** is **/etc/logrotate.conf**.
The **logrotate** command has the following options.

Option	Enables You To
-d	Turn on debug mode to disable any change from being made to the logs.
-f	Force log rotation by deleting old files irrespective of their importance and create new ones.
-m {*subject*} {*recipient*}	Mail the logs to the recipient. The default syntax is /bin/mail -s.

Most Linux applications and commands store their log files in the /var/log directory. This is frequently where log files and their archives are managed, rotated, and archived.

The logwatch Utility

The **logwatch** utility is run as a daily cron job that is used to monitor logs. It is fully customizable via the **/etc/logwatch/conf/logwatch.conf** file. The utility searches logs and reports suspicious messages, and enables you to set detail levels for reports. 10, 5, and 0, correspond to high, medium, and low level details, respectively.
The **logwatch** command may not be installed by default. In that case, you can use the command **sudo apt install logwatch** to install it.
The **logwatch** utility has the following options.

Option	Enables You To
--detail {*level*}	Set the detail level of the log report.
--print	Print the report generated by the command.
--range {*range*}	*Set the range for analysis. It can accept any value among*
--mailto {*address*}	Mail the results to the recipient's mail ID.
save {*file name*}	Save the output to a file instead of displaying it.

Crontab files

System crontab Files

System crontab files are the configuration files for the cron utility. They are stored in the **/etc/crontab** file. The name of the user running the command is indicated in the sixth field of the file. When you create a crontab entry for a specific user, the sixth field contains the command that needs to be run at the specified time. System crontab files can be edited by the root user.

User crontab Files

In addition to system-level cron jobs, individual users can schedule cron jobs. Unlike the system- level crontab, users have their own crontab files. The format of entries in this file is the same as that of the system-level crontab, with the exception of the user field. Because the entire crontab file is dedicated to a single user, the user field is not included. While the **/etc/crontab** file can be edited directly, user crontab files are best edited via the crontab utility.

The at Command

The **at** command executes a given set of commands at a specified time. This command is useful for executing a set of commands only once. Using either the **-f** option or input redirection, the **at** command reads the list of commands from a file. This file needs to be an executable shell script.
The syntax of the **at** command is **at [options] {time}**.
The following table lists some frequently used at command options and their descriptions.

Option	**Enables You To**
atq	Display the job queue of all users except the superuser.
atq -V	Display the version number.
at -q [a-z]	Display the jobs in the specified queue.
at -m	Send mail to the user when the job is complete.
at -f {file name}	Read the job from the file rather than the standard input.

at -l	Print all the jobs queued for the user.
at -v	Display the time that the job will be executed before reading the job.

If the **at** command is not installed, you can use the **sudo apt install at** command to install it.

Specifying Time Using the at Command

There are a number of common time formats. Some of the common time formats in which you can schedule a job are given in the following table.

Time Format	Description
HH:MM A.M. or HH:MM P.M.	Specifies the hour and minute.
MMDDYY or MM/DD/YY or DD.MM.YY	Specifies the day, month, and year.
JAN or FEB or MAR	Specifies the month.
SUN or MON or TUE	Specifies the day of the week.

Anacron

Anacron is a daemon that executes jobs at intervals, which are specified in days, without requiring the system to be running continuously. **Anacron** is used to control the execution of daily, weekly, or monthly jobs.
The **/etc/anacrontab** file is the configuration file for the anacron utility. This file has four fields. The first field displays the number of days the job has not been run, the second field displays the time after which the job has to be run (after reboot), the third field displays the job identifier, and the fourth field displays the job to be run by the anacron utility.

Maintaining System Time

System Time

System time is the time maintained by a computer's internal clock. It is coordinated universal time with a resolution in milliseconds. The internal clock circuitry is backed up by a battery that keeps the clock running even when the computer is switched off. System time is used to date-stamp files with the time of their creation or revision. It can also be changed with difference in time zones.

Clock Drift

Clock drift is the gradual variation in time between the hardware clock and the system clock. The hardware clock is also known as the Real Time Clock (RTC). It keeps track of the time when the system is turned off and not when the system is on. The system clock, however, functions only when the system is running and needs to be initialized at boot time. The hardware and system clocks will drift at different rates, apart from each other and also away from the real time. To synchronize both clocks, their drift rates need to be measured and corrected.

UTC

Coordinated Universal Time (UTC) is a time scale that forms the official measure of time in the world. UTC is independent of time zones. It was previously referred to as Greenwich Mean Time (GMT). It is the time at the prime meridian at Greenwich, England. Unlike GMT, leap seconds are included in UTC.

Leap Seconds

A leap second is the adjustment made to UTC, to account for the irregularity in the earth's rotation. The standard second is stable, while the motion of the earth is not. Therefore, occasionally, the standard minute is adjusted by adding a leap second. As a result, some minutes have 61 seconds. Standard hours are always 60 minutes, though one of the minutes may be a second longer than usual. Standard days are always 24 hours.

Linux and Time Zones

In Linux, you can use the **tzselect** command to access a menu driven utility that will allow you to select the time zone for your system according to your geographic location. You need to define an environment variable, TZ, in the /etc/profile file to set the time zone for your system.

The Date/Time Format

The International Organization for Standardization (ISO) specifies numeric representation of date and time. The standard format for date is **YYYY-MM-DD**, where **YYYY** represents the year in the Gregorian calendar, **MM** represents the month in the year, and **DD** represents the day in the month. The American format of date is **MM-DD-YYYY**. However, Europeans write the day before the month. The separators used with numbers also vary among countries. The common format for time is **hh-mm-ss**, where **hh** represents hours, **mm** represents minutes, and **ss** represents seconds.

The /etc/timezone File

The **/etc/timezone** file is available with the Debian® and Ubuntu® distributions of Linux and is used to store the time zone information of the system. This file typically consists of a single line entry based on the **continent/time zone** format, such as **America/New_York**.

The /usr/share/zoneinfo/ Directory

The **/usr/share/zoneinfo/** directory contains time zone details relating to different countries. When you export a time zone, details of that time zone are obtained from this directory.

The /etc/localtime Directory

The current time details of the system are stored in the **/etc/localtime** directory. If you make any change to your system time, the **/etc/localtime** directory gets updated.

Epoch Time Format

Epoch Time, also known as Unix Time, is a system for describing a point in time based on the number of seconds that have passed since the Unix epoch, or 00:00:00 UTC on 1 January 1970. Leap seconds are ignored, and every day is treated as if it contains exactly 86400 seconds.

If You Need To	Use This hwclock Command Option
Set the BIOS clock to the time given by --date.	--set
Specify the time that will be set for the BIOS clock.	--date=[YYYY-MM-DD hh:mm:ss]’
Set the system time from the BIOS clock.	--hctosys
Set the BIOS clock from the system time.	--systohc
Set the BIOS clock to the UTC.	--utc

The Network Time Protocol (NTP)

Network Time Protocol (NTP) is a standard Internet protocol for synchronizing the internal system clock with the true time or the average time on a number of high accuracy clocks around the world. NTP is used for transmitting and receiving time on Transmission Control Protocol/Internet Protocol (TCP/IP) networks. NTP is also used to set the clock of one computer to match that of another and synchronize it with the network clock.

The pool.ntp.org Service

The **pool.ntp.org** is a collection of servers on the Internet that provides accurate time to the Linux systems using NTP.

Drift Files

A drift file is a file found in the **/etc/ntp** directory. The NTP drift file is used by the ntpd daemon to reset the time when the system is restarted. The drift file synchronizes the system clock and the clock drift to display the time from the NTP server.

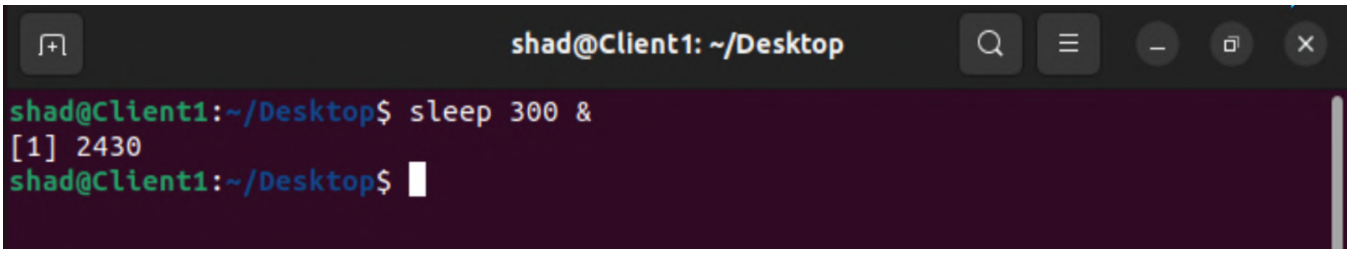
The ntp.conf File

The **ntp.conf** file found in the **/etc** directory contains configuration options for the NTP server. The file contains settings for all hosts on local and public servers. The ntpd daemon reads the **ntp.conf** file for synchronization settings and then connects to the NTP server.

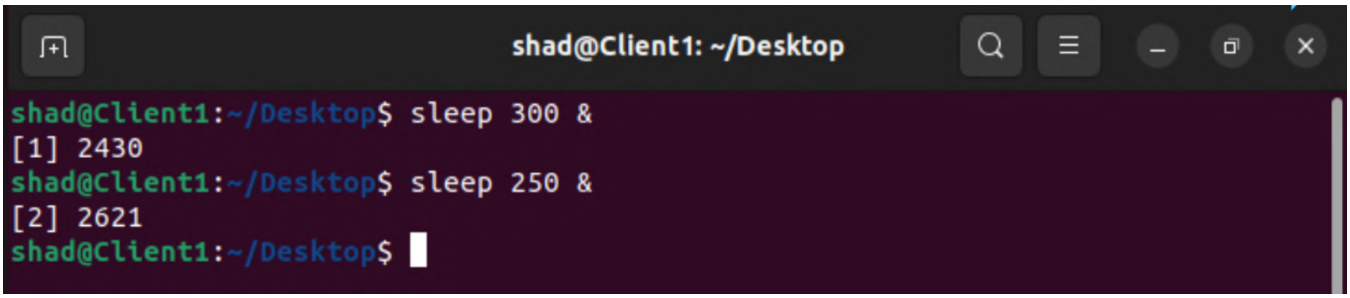
Managing Jobs and Processes

TASK A

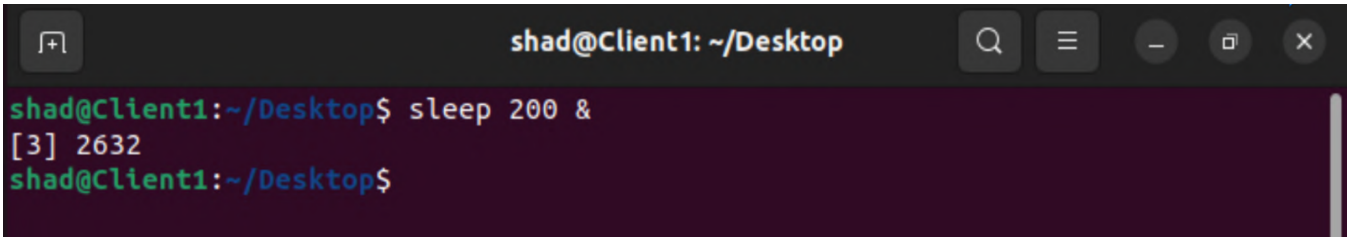
1. You need to start a process that will do nothing (wait) for 300 seconds and then exit. Type **sleep 300 &** and then press **Enter**.



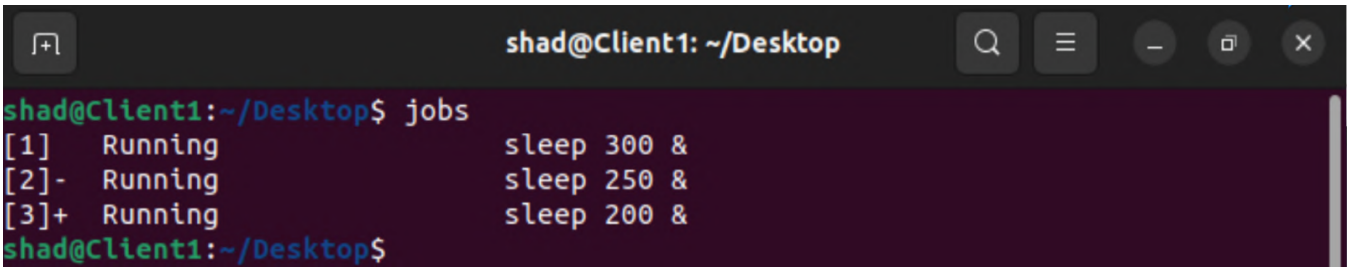
2. You need to start a process that will do nothing (wait) for 250 seconds and then exit. Type **sleep 250 &** and then press **Enter**.



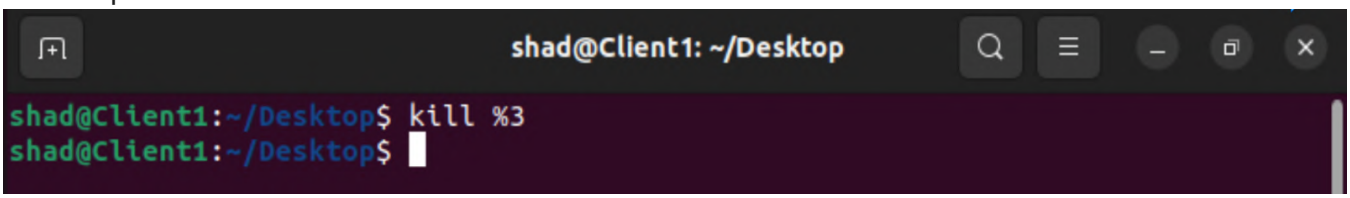
3. You need to start a process that will do nothing (wait) for 200 seconds and then exit. Type **sleep 200 &** and then press **Enter**.



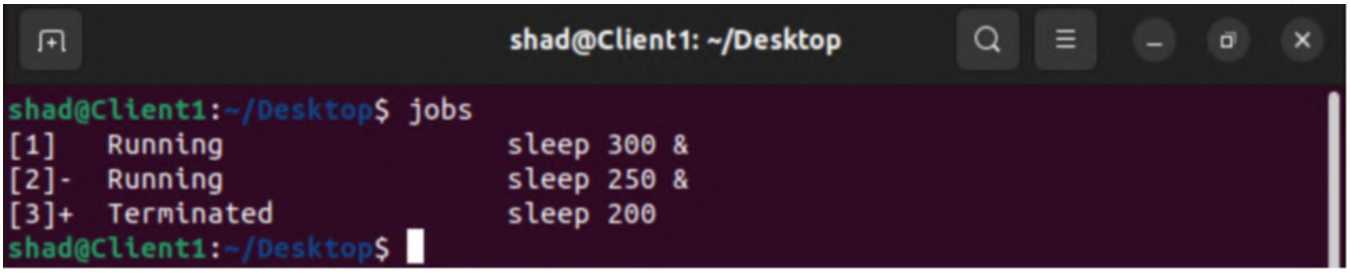
4. To view the list of processes running in the background, type **jobs** and press **Enter**. Verify that the three jobs are listed.



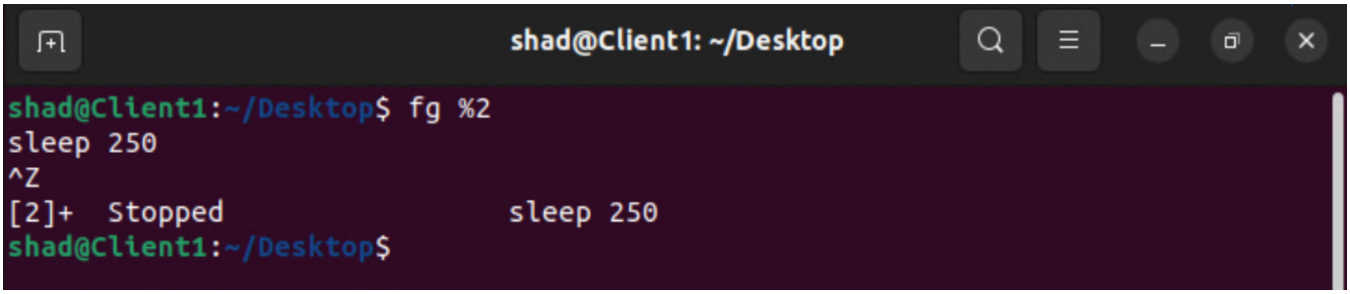
5. You want to kill the last job. To kill the process that is executing the **sleep 200 &** command, type **kill %3** and press **Enter**.



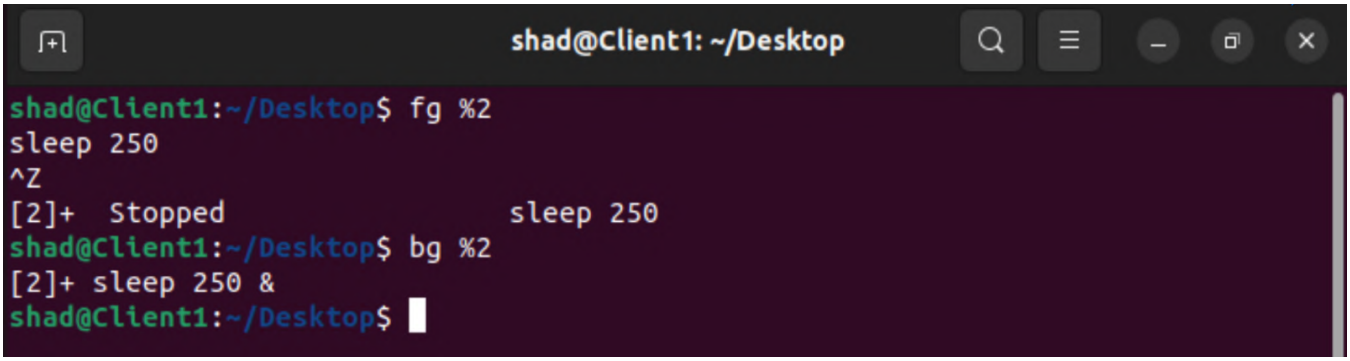
6. To view the list of processes running in the background, type **jobs** and press **Enter**.Verify that the status of the third job displays “Terminated,” indicating that the job is terminated.



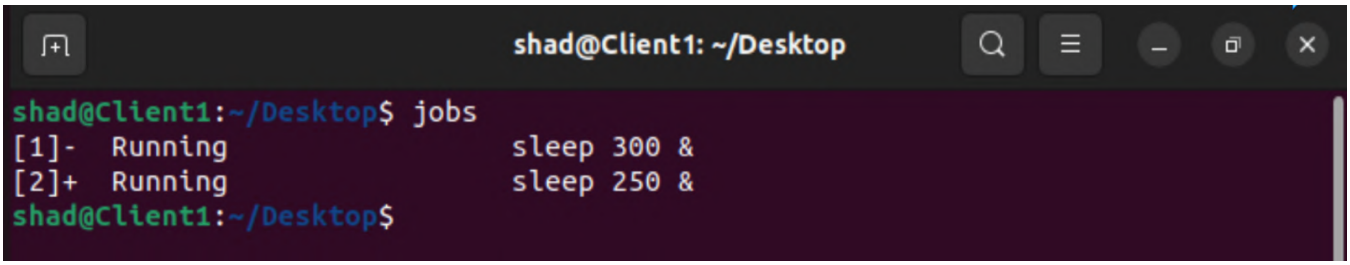
7. You want to move the job to sleep for 250 seconds to the foreground. Type **fg %2** and press E**nter**.
8. You want to suspend the foreground job. Press **Ctrl+Z**. Verify that the status of the job is displayed as “**Stopped**,” indicating that the job is stopped.



9. You want to restart the suspended job. Type **bg %2** and press **Enter**.



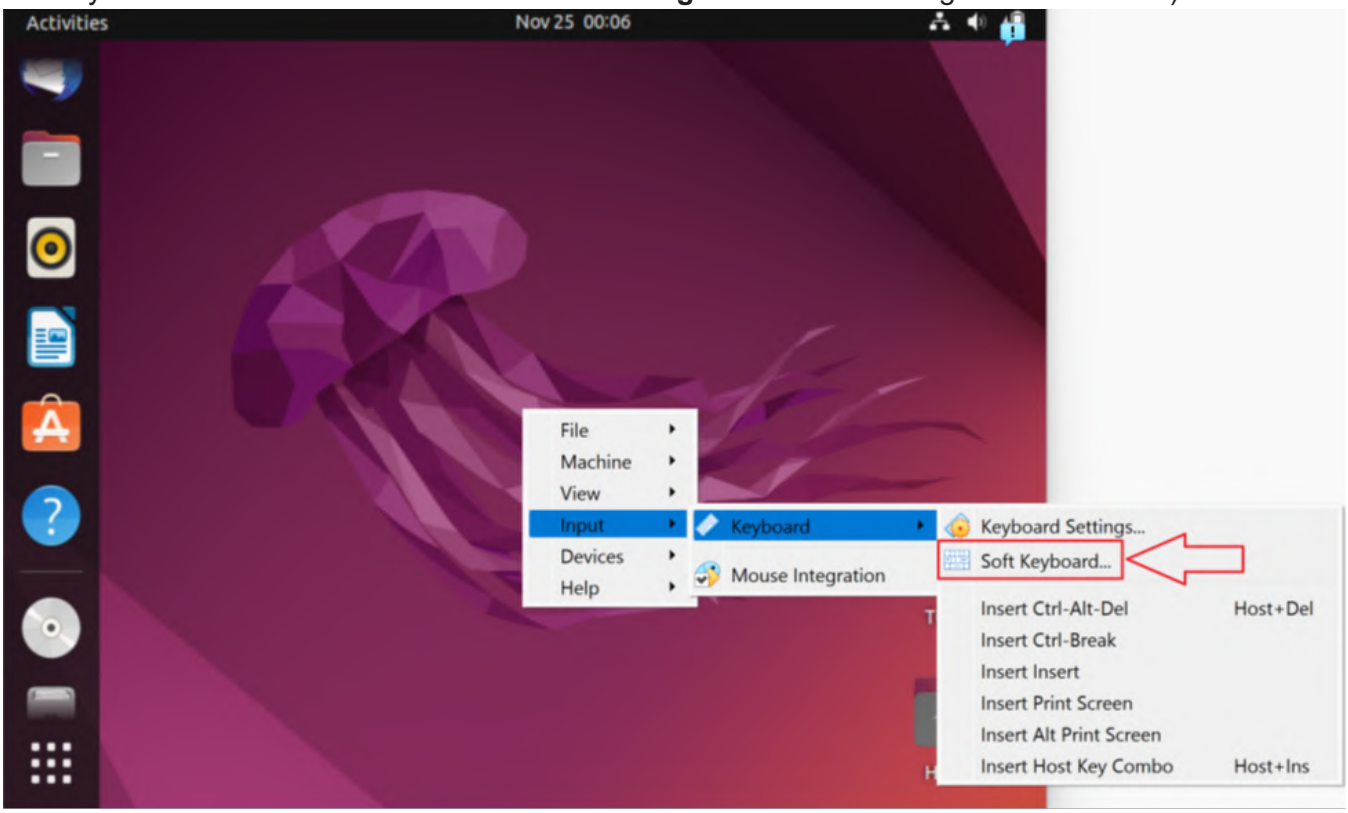
10. To view the list of processes running in the background, type **jobs** and press **Enter**. Verify that the job to sleep for 250 seconds is **running**.



TASK B

Some users complained of processes on the Linux server taking longer than normal to complete. You discover several processes that are not needed are still running and were never successfully terminated. You need to manage the system processes and the processes issued by other users. In this task, you will explore how to kill processes.

1. From the **Input** menu, click **Keyboard** and then click **Soft Keyboard**. (Note: If you are in **Scaled Mode** you will not see the menu bar. You can use **rightCTRL+Home** to get the menu bar.)



2. Using the **Soft Keyboard**, send the **Ctrl+Alt+F5** command. Log in with your user account.

```
Ubuntu 22.04.1 LTS Client1 tty5

Client1 login: shad
Password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

132 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Mon Nov 21 13:20:31 EST 2022 on tty4
shad@Client1:~$
```

3. You want to list only the processes running on the current terminal. Type **ps** and then press **Enter**. Verify that only some processes are listed.

```
shad@Client1:~$ ps
    PID TTY          TIME CMD
   2747 tty5     00:00:00 bash
   2757 tty5     00:00:00 ps
shad@Client1:~$ _
```

4. You want to list all the processes running on the system, type **ps -e** and press **Enter**. Verify that more processes are listed as compared to the output of the **ps** command.

```
shad@Client1:~$ ps
    PID TTY          TIME CMD
   2747 tty5     00:00:00 bash
   2760 tty5     00:00:00 ps
shad@Client1:~$ ps -e
```

5. To pause the system for 300 seconds, type **sleep 300 &** and then press **Enter**. Make a note of the PID of the **sleep 300 &** command.

```
shad@Client1:~$ sleep 300 &
[1] 2793
shad@Client1:~$ _
```

6. To list only the processes running on the current terminal, type **ps** and then press **Enter**. Verify that the sleep command is in the list of running processes.

```
shad@Client1:~$ ps
    PID TTY          TIME CMD
   2747 tty5     00:00:00 bash
   2793 tty5     00:00:00 sleep
   2795 tty5     00:00:00 ps
shad@Client1:~$
```

7. To terminate the process running the sleep command, type **kill [PID]** where [PID] is the PID noted earlier. Then type **ps** and press **Enter** to list the processes. Verify that the sleep command is not in the list of running processes and its status is displayed as “**Terminated**.”

```
shad@Client1:~$ kill 2793
shad@Client1:~$ ps
    PID TTY          TIME CMD
   2747 tty5     00:00:00 bash
   2797 tty5     00:00:00 ps
[1]+  Terminated              sleep 300
shad@Client1:~$ _
```

8. You should leave the Client1 virtual machine logged in on TTY5 and proceed to the next lab.

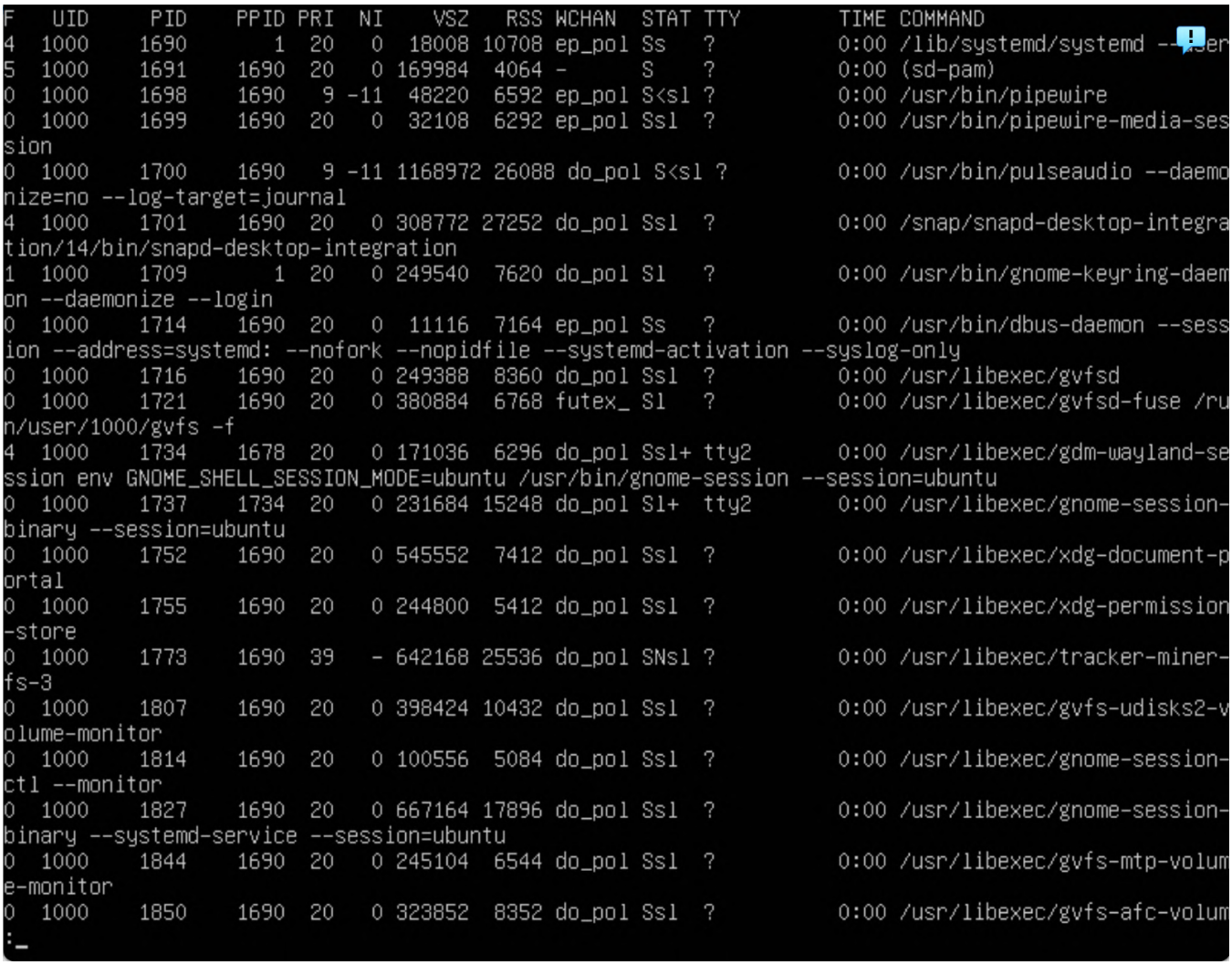
Managing Timing of Jobs and Processes

TASK A

You want to back up the local copy of your /etc configuration directory. You expect the copying process to be time-consuming and to continue after you log out of your system. You decide to increase the priority of the process to ensure that it is completed on time and to allow the process to continue even after you log out.

1. To view all processes run by users, type **ps xl | less** and then press **Enter**. Examine the processes that have the highest nice value. To view the next page of the list, press **Page Down**.

```
shad@Client1:~$ ps xl | less
```



```
F   UID     PID    PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
4  1000    1690       1  20   0  18008 10708 ep_pol Ss   ?          0:00 /lib/systemd/systemd --user
5  1000    1691    1690  20   0 169984  4064 -      S    ?          0:00 (sd-pam)
0  1000    1698    1690   9 -11  48220  6592 ep_pol S<sl ?          0:00 /usr/bin/pipewire
0  1000    1699    1690  20   0  32108  6292 ep_pol Ssl  ?          0:00 /usr/bin/pipewire-media-ses
sion
0  1000    1700    1690   9 -11 1168972 26088 do_pol S<sl ?         0:00 /usr/bin/pulseaudio --daemo
nize=no --log-target=journal
4  1000    1701    1690  20   0 308772 27252 do_pol Ssl  ?          0:00 /snap/snapd-desktop-integra
tion/14/bin/snapd-desktop-integration
1  1000    1709       1  20   0 249540  7620 do_pol Sl   ?          0:00 /usr/bin/gnome-keyring-daem
on --daemonize --login
0  1000    1714    1690  20   0  11116  7164 ep_pol Ss   ?          0:00 /usr/bin/dbus-daemon --sess
ion --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
0  1000    1716    1690  20   0 249388  8360 do_pol Ssl  ?          0:00 /usr/libexec/gvfsd
0  1000    1721    1690  20   0 380884  6768 futex_ Sl   ?          0:00 /usr/libexec/gvfsd-fuse /ru
n/user/1000/gvfs -f
4  1000    1734    1678  20   0 171036  6296 do_pol Ssl+ tty2       0:00 /usr/libexec/gdm-wayland-se
ssion env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubuntu
0  1000    1737    1734  20   0 231684 15248 do_pol Sl+  tty2       0:00 /usr/libexec/gnome-session-
binary --session=ubuntu
0  1000    1752    1690  20   0 545552  7412 do_pol Ssl  ?          0:00 /usr/libexec/xdg-document-p
ortal
0  1000    1755    1690  20   0 244800  5412 do_pol Ssl  ?          0:00 /usr/libexec/xdg-permission
-store
0  1000    1773    1690  39   - 642168 25536 do_pol SNsl ?          0:00 /usr/libexec/tracker-miner-
fs-3
0  1000    1807    1690  20   0 398424 10432 do_pol Ssl  ?          0:00 /usr/libexec/gvfs-udisks2-v
olume-monitor
0  1000    1814    1690  20   0 100556  5084 do_pol Ssl  ?          0:00 /usr/libexec/gnome-session-
ctl --monitor
0  1000    1827    1690  20   0 667164 17896 do_pol Ssl  ?          0:00 /usr/libexec/gnome-session-
binary --systemd-service --session=ubuntu
0  1000    1844    1690  20   0 245104  6544 do_pol Ssl  ?          0:00 /usr/libexec/gvfs-mtp-volum
e-monitor
0  1000    1850    1690  20   0 323852  8352 do_pol Ssl  ?          0:00 /usr/libexec/gvfs-afc-volum
:_
```

2. Press **Page Down** until you reach the end of the entire list.

```
oard
0  1000    2214    1858  20   0 183372 53744 ep_pol S    ?          0:00 /usr/bin/Xwayland :0 -rootl
ess -noreset -accessx -core -auth /run/user/1000/.mutter-Xwaylandauth.PJIPW1 -listen 4 -listen 5 -di
splayfd 6 -initfd 7
0  1000    2215    2047  20   0 172128  7208 do_pol Sl   ?          0:00 /usr/libexec/ibus-engine-si
mple
0  1000    2220    1690  20   0 516024 69804 do_pol Ssl  ?          0:00 /usr/libexec/gsd-xsettings
0  1000    2223    1690  20   0 171672  6716 do_pol Ssl  ?          0:00 /usr/libexec/gvfsd-metadata
0  1000    2272    1690  20   0 2608152 27520 do_pol Sl  ?          0:00 /usr/bin/gjs /usr/share/gno
me-shell/org.gnome.ScreenSaver
0  1000    2299    1690  20   0 202868 25408 do_pol Sl   ?          0:00 /usr/libexec/ibus-x11
0  1000    2301    1690  20   0 352568 24764 do_pol Ssl  ?          0:00 /usr/libexec/xdg-desktop-po
rtal-gtk
1  1000    2320    1690  20   0  28700   348 do_wai S    ?          0:00 /usr/bin/VBoxClient --seaml
ess
1  1000    2321    2320  20   0 226988  2496 futex_ Sl   ?          0:00 /usr/bin/VBoxClient --seaml
ess
1  1000    2328    1690  20   0  28700   348 do_wai S    ?          0:00 /usr/bin/VBoxClient --draga
nddrop
1  1000    2329    2328  20   0 227504  2460 futex_ Sl   ?          0:03 /usr/bin/VBoxClient --draga
nddrop
1  1000    2333    1690  20   0  28700   348 do_wai S    ?          0:00 /usr/bin/VBoxClient --vmsvg
a-session
1  1000    2334    2333  20   0 160804  2348 futex_ Sl   ?          0:00 /usr/bin/VBoxClient --vmsvg
a-session
0  1000    2389    1690  20   0  44496 19260 do_wai S    ?          0:00 /usr/bin/python3 /usr/bin/g
nome-terminal --wait
0  1000    2390    2389  20   0 391880 27628 do_pol Sl   ?          0:00 /usr/bin/gnome-terminal.rea
l --wait
0  1000    2395    1690  20   0 561620 51656 do_pol Ssl  ?          0:00 /usr/libexec/gnome-terminal
-server
0  1000    2413    2395  20   0  19920  5464 do_sel Ss+  pts/0      0:00 bash
0  1000    2439    1827  20   0 502732 31520 do_pol Sl   ?          0:00 update-notifier
4  1000    2747    2671  20   0  19868  5380 do_wai S    tty5       0:00 -bash
0  1000    2805    2747  20   0  21324  1600 -      R+   tty5       0:00 ps x1
0  1000    2806    2747  20   0    352     4 wait_o D+   tty5       0:00 [less]
(END)
```

3. To exit the list, press **q**.

4. You want to create a process to work with. Type **sleep 400 &** and press **Enter**. Verify that the PID and the job number are displayed. Record the PID.

```
shad@Client1:~$ sleep 400 &
[1] 2808
shad@Client1:~$
```

5. To open the process management tool, type **sudo top** and then press **Enter**.

```
shad@Client1:~$ top _
```

```
top - 17:22:27 up 49 min,  2 users,  load average: 0.06, 0.05, 0.02
Tasks: 198 total,   1 running, 197 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1975.8 total,    174.0 free,    704.6 used,   1097.1 buff/cache
MiB Swap:   5360.0 total,   5359.7 free,      0.3 used.   1095.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1070 root      20   0  298324   1116    980 S   0.3   0.1   0:00.58 VBoxDRMClient
   2329 shad      20   0  227504   2460   2104 S   0.3   0.1   0:03.40 VBoxClient
      1 root      20   0  166572  11868   8320 S   0.0   0.6   0:00.91 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.29 kworker/0:1H-kblockd
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.05 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.61 rcu_sched
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     17 root      20   0       0      0      0 I   0.0   0.0   0:01.83 kworker/0:1-events
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     20 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     21 root      rt   0       0      0      0 S   0.0   0.0   0:00.17 migration/1
     22 root      20   0       0      0      0 S   0.0   0.0   0:00.10 ksoftirqd/1
     24 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
     25 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kdevtmpfs
     26 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 inet_frag_wq
     27 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
     28 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
     29 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
     30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
     31 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kcompactd0
```

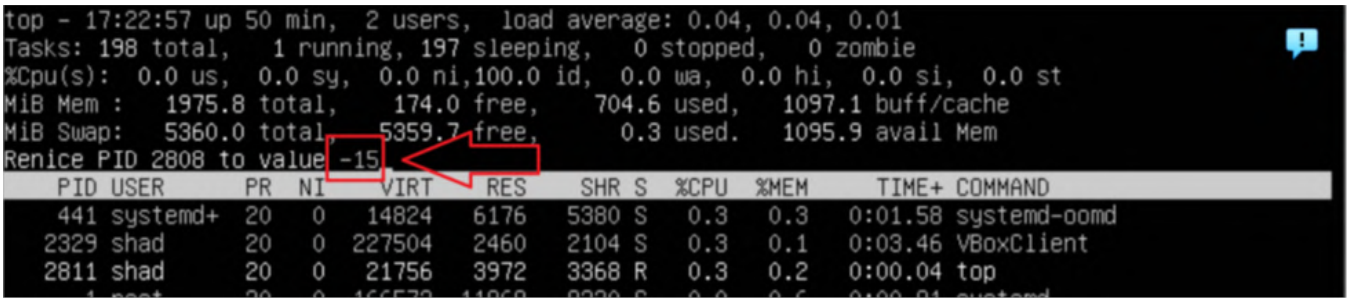
6. To renice the process, press **r**.

```
top - 17:22:57 up 50 min,  2 users,  load average: 0.04, 0.04, 0.01
Tasks: 198 total,   1 running, 197 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1975.8 total,    174.0 free,    704.6 used,   1097.1 buff/cache
MiB Swap:   5360.0 total,   5359.7 free,      0.3 used.   1095.9 avail Mem
PID to renice [default pid = 441]
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    441 systemd+  20   0   14824   6176   5380 S   0.3   0.3   0:01.58 systemd-oomd
```

7. To specify a process to renice, enter the PID you recorded and then press **Enter**.

```
top - 17:22:57 up 50 min,  2 users,  load average: 0.04, 0.04, 0.01
Tasks: 198 total,   1 running, 197 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1975.8 total,    174.0 free,    704.6 used,   1097.1 buff/cache
MiB Swap:   5360.0 total,   5359.7 free,      0.3 used.   1095.9 avail Mem
Renice PID 2808 to value _
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    441 systemd+  20   0   14824   6176   5380 S   0.3   0.3   0:01.58 systemd-oomd
   2329 shad      20   0  227504   2460   2104 S   0.3   0.1   0:03.46 VBoxClient
   2811 shad      20   0   21756   3972   3368 R   0.3   0.2   0:00.04 top
      1 root      20   0  166572  11868   8320 S   0.0   0.6   0:00.91 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
```

8. To specify the nice value, type**-15** and then press **Enter**.



9. To exit the process list, press **q**.

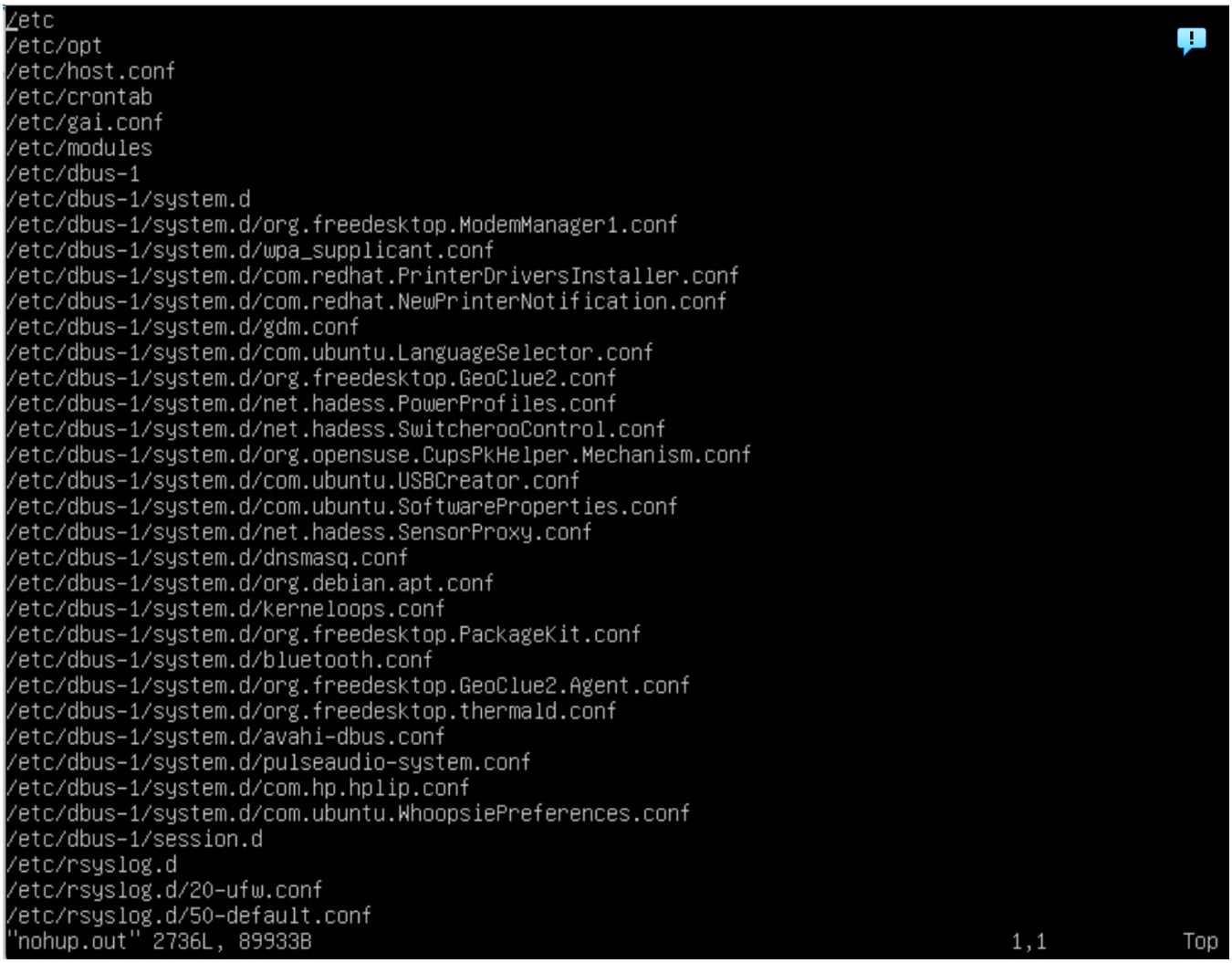
TASK B

You discovered that a large number of CPU-intensive jobs are being run during the normal business hours. Your manager gave you a list of jobs that are not time critical and can be rescheduled. You decide that the best time to run CPU-intensive applications is after you log out of the system.

1. To view the files that are available in the **/etc** directory, type **find /etc** and then press **Enter**. Note the last entry in the **/etc** directory.

2. To process a command in the background, type **nohup find /etc -print &** and then press **Enter**. Verify that a message is displayed indicating that the output of the command is added to the **nohup.out** file.

3. Type **logout** and then press **Enter** to log out.
4. Log in using your user account.
5. To open the **nohup.out** file, type **vim nohup.out** and then press **Enter**. Verify that the file contains a listing of files and directories in the **/etc** directory, which indicates that the job is complete.

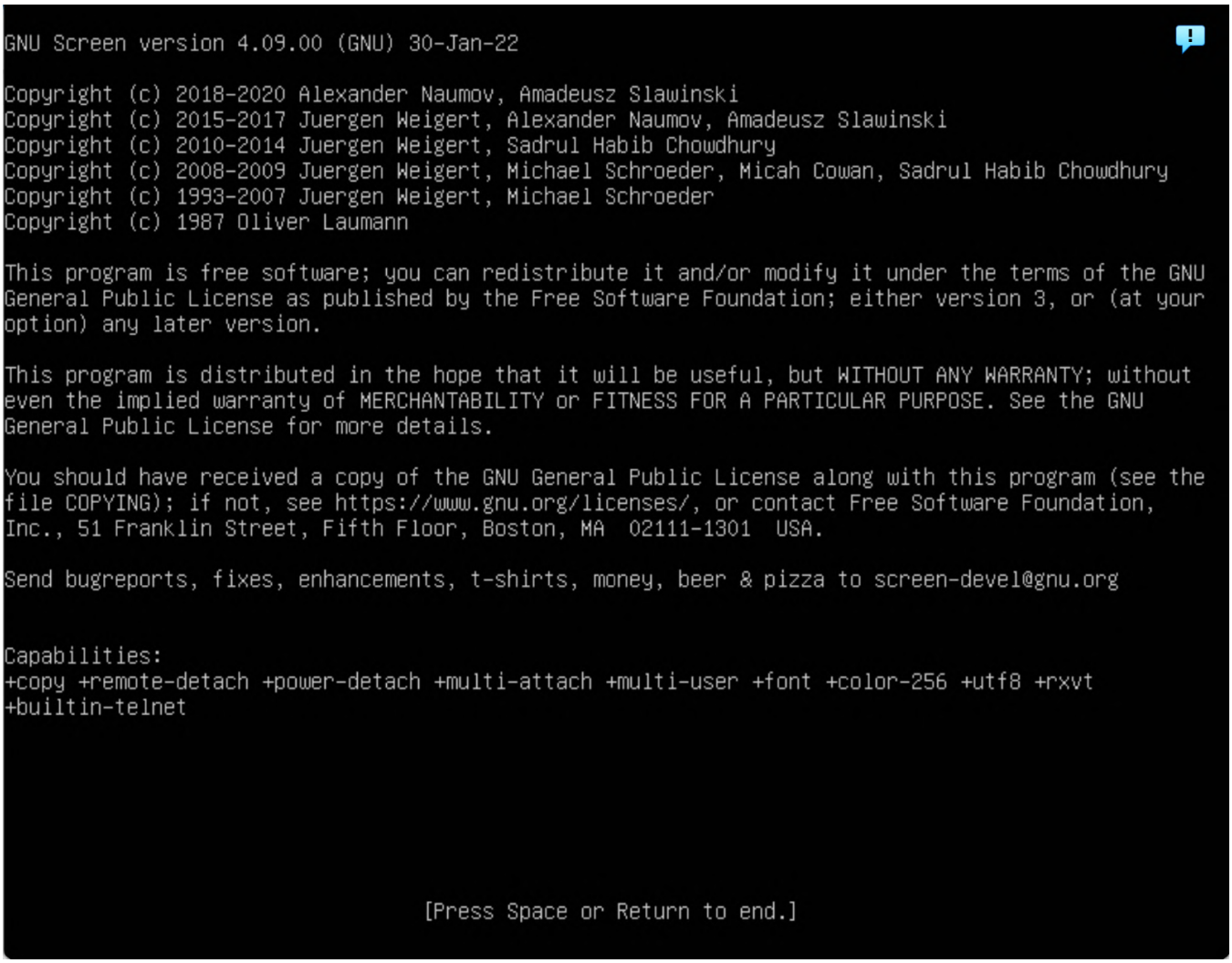


6. To move to the end of the file, press **Shift+G**. Verify that the same file as noted previously is the last entry listed.
7. Quit the file without saving by typing**:q** and then pressing **Enter**.
8. Type **sudo apt install screen** and press **Enter** to install the **screen** package. When prompted, enter your password. When prompted to continue, type **y** and then press **Enter**.

```
shad@Client1:~$ sudo apt install screen
[sudo] password for shad:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libutempter0
Suggested packages:
  byobu | screenie | iselect ncurses-term
The following NEW packages will be installed:
  libutempter0 screen
0 upgraded, 2 newly installed, 0 to remove and 135 not upgraded.
Need to get 680 kB of archives.
After this operation, 1,081 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libutempter0 amd64 1.2.1-2build2 [8,848 B
]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 screen amd64 4.9.0-1 [672 kB]
Fetched 680 kB in 0s (1,391 kB/s)
Selecting previously unselected package libutempter0:amd64.
(Reading database ... 197337 files and directories currently installed.)
Preparing to unpack .../libutempter0_1.2.1-2build2_amd64.deb ...
Unpacking libutempter0:amd64 (1.2.1-2build2) ...
Selecting previously unselected package screen.
Preparing to unpack .../screen_4.9.0-1_amd64.deb ...
Unpacking screen (4.9.0-1) ...
Setting up libutempter0:amd64 (1.2.1-2build2) ...
Setting up screen (4.9.0-1) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
shad@Client1:~$ _
```

9. To start a new screen session, type **screen** and then press **Enter**. Then press the **spacebar**.

```
shad@Client1:~$ screen
```

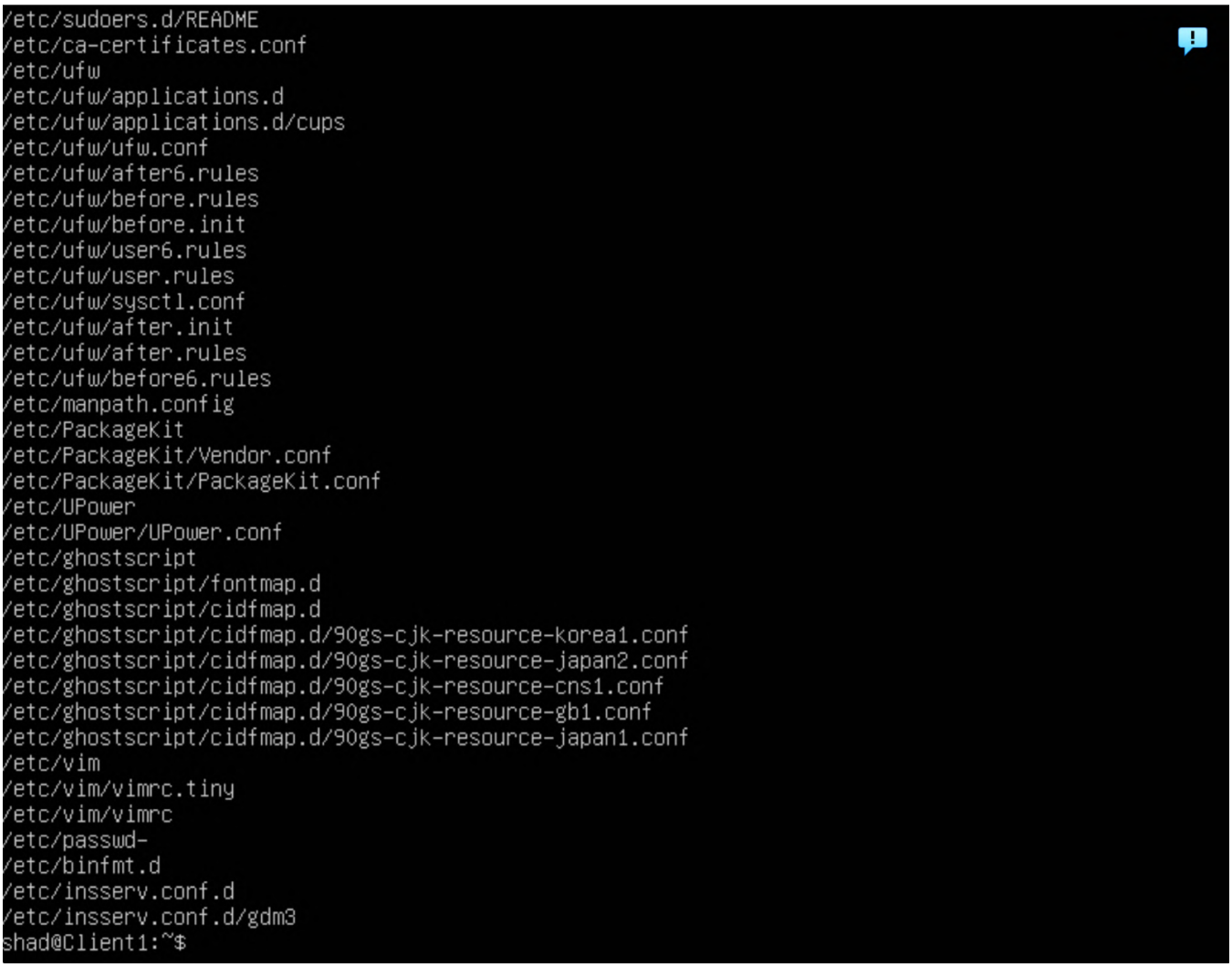


10. To view the files that are available in the **/** directory, on the terminal, type **find /** and then press **Enter**. While this command is running, detach from the current screen session by pressing **Ctrl+A+D**. NOTE: You may want to use the Soft Keyboard to issue this command.Verify that a message is displayed indicating that you have detached from the screen session.

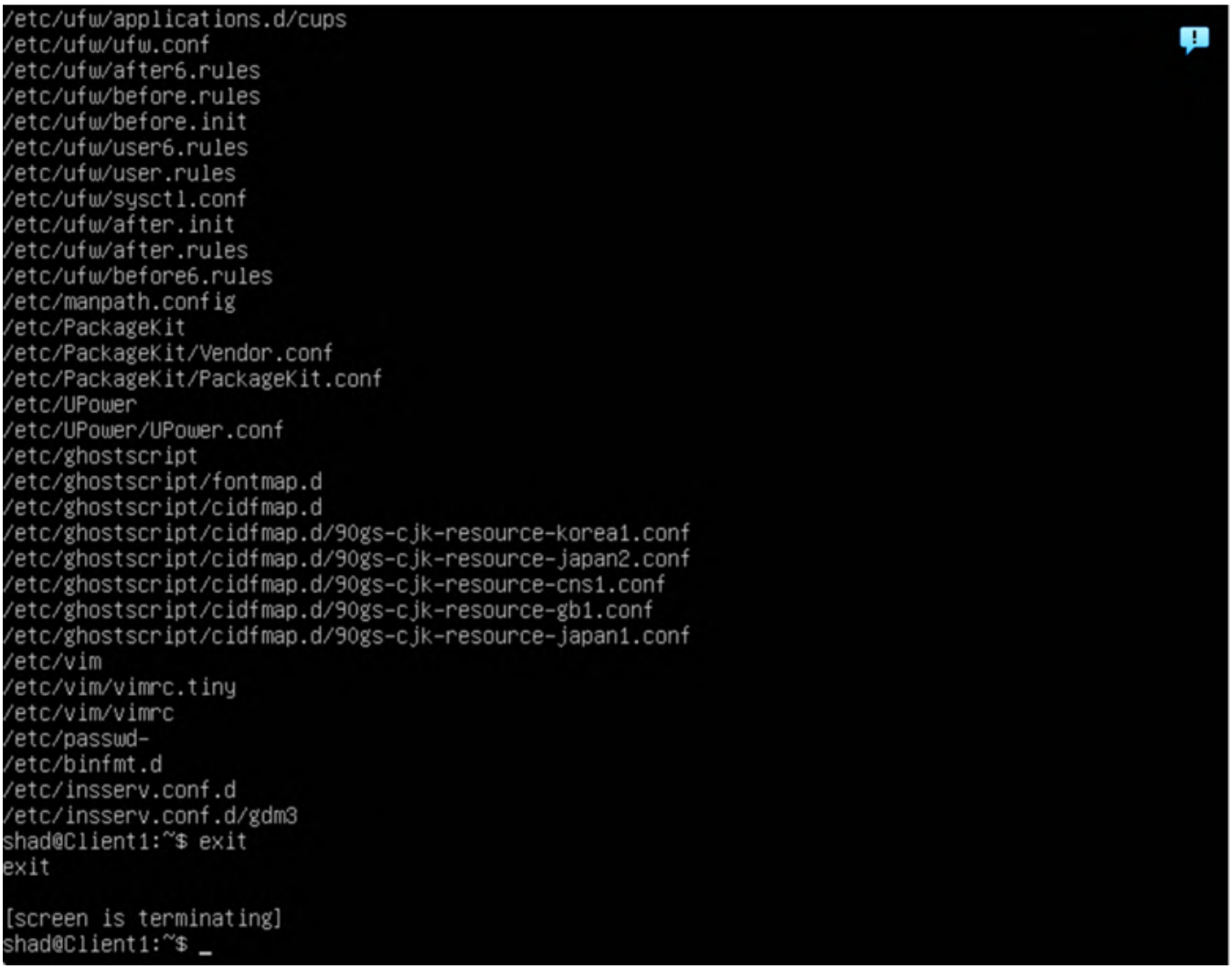
```
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/snd-seq-device.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/seq
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/seq/snd-seq-midi-emul.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/seq/snd-seq-dummy.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/seq/snd-seq-midi-event.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/seq/snd-seq.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/seq/snd-seq-midi.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/seq/snd-seq-virmidi.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/core/snd-hrtimer.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/synth
/usr/lib/modules/5.15.0-43-generic/kernel/sound/synth/emux
/usr/lib/modules/5.15.0-43-generic/kernel/sound/synth/emux/snd-emux-synth.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/synth/snd-util-mem.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soundcore.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/snd-sof-acpi.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/snd-sof.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-intel-ipc.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-pci-intel-tgl.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-pci-intel-tng.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-intel-hda.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-pci-intel-icl.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-intel-hda-common.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-intel-atom.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-pci-intel-cnl.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-acpi-intel-bdw.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-acpi-intel-byt.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/intel/snd-sof-pci-intel-apl.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/xtensa
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/xtensa/snd-sof-xtensa-dsp.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/sof/snd-sof-pci.ko
/usr/lib/modules/5.15.0-43-generic/kernel/sound/soc/amd

[detached from 3323.tty5.Client1]
shad@Client1:~$
```

11. Type **logout** and then press **Enter** to log out.
12. Log back in using your user account.
13. To restore your previous screen session, type **screen -r** and then press **Enter**. Verify that the find command for the earlier step is still running or has successfully completed in the screen session.

```
shad@Client1:~$ screen -r _
```



14. Type **exit** and then press **Enter** to close the screen session and exit screen. Verify that a message is displayed indicating that screen is terminating.



15. Use the **logout** command to log out. Using the **Soft Keyboard**, send the **Ctrl+Alt+F1** command to return to the GUI. If prompted, log in as your user account.
16. You may close all the windows and save the state of the virtual machine. This is the end of the labs for this module.

Manage the File System

Burning Discs

While managing your filesystem, you may have to back up some data on discs. Linux allows you to burn CDs and DVDs with GUI-based programs, as well as from the CLI. GUI-based programs guide you through the burning process.

ISO Images

An ISO image or disk image is an archive file format for files that are to be written to optical discs such as CDs and DVDs. It is a standard defined by the International Organization for Standardization (ISO) and has a file extension of **.iso**.
Note: You must have a CD or DVD writer installed on your system to be able to burn discs.

Mount Points

A mount point is an access point to information stored on a local or remote storage device. The mount point is typically an empty directory on which a filesystem is loaded, or mounted, to make the filesystem accessible to users. If the directory already has content, the content becomes invisible to the users until the mounted filesystem is unmounted.
Note: You can use the **/etc/fstab** file to list the filesystem to be mounted and unmounted when the Linux system boots and shuts down, respectively.

The mount Command

In Linux, a file system cannot be accessed directly. It has to be associated with a directory to make it accessible to users. This association is brought about by loading, or mounting, the filesystem in a directory by using the **mount** command. After using the file system, it needs to be disassociated from the directory by unloading, or unmounting, the file system using the **umount** command.
You can specify various mount options for a filesystem.

Option	Enables You To
auto	Specify that the device has to be mounted automatically.
noauto	Specify that the device need not be mounted automatically.
nouser	Specify that only the root user can mount a device or a filesystem.
user	Specify that all users can mount a device or a filesystem.
exec	Allow binaries in a filesystem to be executed.
noexec	Prevent binaries in a filesystem from being executed.

ro	Mount a filesystem as read-only.
rw	Mount a filesystem with read and write permissions.
sync	Specify that input and output operations in a filesystem should be done synchronously.
async	Specify that input and output operations in a filesystem should be done asynchronously.

Swap Space

Swap space is a partition on the hard disk that is used when the system runs out of physical memory. Linux pushes some of the unused files from the RAM to the swap space to free up memory. Usually, the swap space equals twice the RAM capacity.
Swap space can be one of three types.

Swap Type	Description
Device swap	Device swap space is configured when you partition the hard disk. It is used by the operating system to run large applications.
Filesystem swap	Filesystem swap space is configured primarily when you install Linux. It is utilized by the operating system as an emergency resource when the available swap space runs out.
Pseudo-swap	Pseudo-swap space allows large applications to run on computers with limited RAM.

Swap Files

Swap files are created for storing data that is to be transferred from a system’s memory to a disk. It is dynamic and changes in size when data is moved in and out of the memory. It is used as a medium to transfer data from RAM on to the hard disk.

Swap Partitions

A swap partition is an area of virtual memory on a hard disk to complement the physical RAM in the computer. Swap partitions are created by Linux because they perform better than swap filesystems.

The mkswap Command

The **mkswap** command is a system administration command that is used to create swap space on a disk partition.
The syntax of the **mkswap** command is **mkswap [options] device {size}**. The device argument of **mkswap** is generally a disk partition, such as /dev/hda2 or /dev/sdb3, but it can also be a file.
The **mkswap** command provides options to perform various tasks. Some are listed in the following table:

Option	Enables You To
-c	Verify that the device is free from bad sectors before mounting the swap space.
-f	Force a swap partition of an area larger than the permissible limit.
-p	Set the page size to be used by the mkswap command.
-L {label}	Activate the swap space using labels applied to partitions or filesystems.

Swap Partition Management Commands

A number of commands are used to manage swap partitions. The most important commands are **swapon** and **swapoff**.

Command	Description
swapon	Used to activate a swap partition on a specified device. It provides a number of options for specifying devices.
swapoff	Used to deactivate the swap space on devices.

Some of the frequently used **swapon** and **swapoff** command options are given in the following table.

Option	Description
swapon -e	It is used to skip devices that do not exist.
swapon -a	It is used to activate all the swap space.
swapoff -a	It is used to deactivate all the swap space.

Maintain File System

Storage Devices

There are different types of storage devices in Linux. Each device has a particular use associated with it.

Device	Description
Hard disk	An internal device that can store large amounts of data. It can be accessed quickly.
Floppy drive	A removable medium that can store smaller amounts of data. It cannot be accessed as quickly as a hard disk.
Tape drive	A device that is used to store large amounts of data on a magnetic tape. Tape drives can be internal or external. External tape drives are portable, whereas in internal tape drives only the tape is removable. Data is accessed sequentially in a tape drive.
Flash drive	A small, portable, storage device that is used to store files that need to be carried around.

CD-R(W)	A removable optical disc that stores 650- 700 MB of data. It can be accessed faster than other removable storage media.
DVD-R(W)	A removable optical disc that stores 4.5 GB (or more) of data. It can be accessed faster than other removable storage media.

Mass Storage Devices

Mass storage devices are types of storage devices that provide fast access to large amounts of data in a small, reasonably reliable, physical package. Hard disks, tape drives, flash drives, CD-R(W), DVD-R(W), and zip drives are some of the common mass storage devices.

ATAPI

AT Attachment Packet Interface (ATAPI) is a protocol for controlling mass storage devices. ATAPI provides commands that are used for hard disks, CD-ROM drives, tape drives, and other devices.

Journaling File Systems

A journaling file system is a method that is used by an operating system to quickly recover after an unexpected interruption, such as a system crash. Journaling file systems can remove the need for a file system check when the system boots. By using journaling file systems, the system does not write modified files directly on the disk. Instead, a journal is maintained on the disk. The journaling file system process involves the following phases:

1. The journal describes all the changes that must be made to the disk.
2. A background process makes each change as and when it is entered in the journal.
3. If the system shuts down, pending changes are performed when it is rebooted.
4. Incomplete entries in the journal are discarded.

Performance Issues with Journaling

A journaled file system works well with small files and small drives. With the growth of file and drive sizes, performance will suffer. Some of the reasons for poor performance include:

- Filesystem recovery time after a power failure or improper shutdown.
- Bitmap method of tracking the filesystem
- Wasted space and fragmentation.

Journaling File Systems

A journaling file system is a method that is used by an operating system to quickly recover after an unexpected interruption, such as a system crash. Journaling file systems can remove the need for a file system check when the system boots. By using journaling file systems, the system does not write

modified files directly on the disk. Instead, a journal is maintained on the disk. The journaling file system process involves the following phases:

1. The journal describes all the changes that must be made to the disk.
2. A background process makes each change as and when it is entered in the journal.
3. If the system shuts down, pending changes are performed when it is rebooted.
4. Incomplete entries in the journal are discarded.

Performance Issues with Journaling

A journaled file system works well with small files and small drives. With the growth of file and drive sizes, performance will suffer. Some of the reasons for poor performance include:

- Filesystem recovery time after a power failure or improper shutdown.
- Bitmap method of tracking the filesystem
- Wasted space and fragmentation.

The fsck Command

The **fsck** command is used to check the integrity of a file system. File system integrity refers to the correctness and validity of a file system. Most systems automatically run the **fsck** command at boot time so that errors, if any, are detected and corrected before the system is used. File system errors are usually caused by power failures, hardware failures, or improper shutdown of the system.
Note: The **fsck** command is similar in concept to the **chkdsk** and **scandisk** commands you may be familiar with from DOS and Windows-based systems.
The syntax of the **fsck** command is **fsck -t {filesystem type} [options]**.

Repair File Systems

You can use **the fsck -r/dev/{filesystem}** command to repair a file system. The command will prompt you to confirm your actions. If you are simultaneously checking multiple file systems, you should not use this option because it allows you to repair only a single file system at a time.

The e2fsck Command

The **e2fsck** command allows you to check ext2, ext3, and ext4 file systems, and is identical to running the **fsck** command with ext2, ext3, or ext4 specified as the file system type. You need to unmount the file system before running the **e2fsck** command to prevent damage to the file system.
The syntax of the **e2fsck** command is **e2fsck /dev/{filesystem}**.

The xfs_repair Command

The **xfs_repair** command allows you to check an XFS file system. As with the **fsck** and **e2fsck** commands, you need to unmount the file system before running the **xfs_repair** command to prevent damage to the file system.
The syntax of the **xfs_repair** command is **xfs_repair [options]/dev/{filesystem}**.

The tune2fs Utility

The **tune2fs** utility helps tuning parameters associated with a Linux file system. Using this utility, a journal can be added to an existing ext2 or ext3 file system. If the file system is already mounted, the journal will be visible in the root directory of the file system. If the file system is not mounted, the journal will be hidden. The **tune2fs** utility is available with most Linux distributions.

Tunable Parameters

Using the **tune2fs** utility, you can adjust the parameters of the extended file systems, such as ext2, ext3, and ext4, that can be tuned on a Linux machine even after installation. Tunable parameters allow you to remove reserved blocks; alter reserved block count; and specify the number of mounts between checks, the time interval between checks, and the behavior of the kernel code, among others.
The syntax of the **tune2fs** utility is **tune2fs [options] {device name}**.
The **tune2fs** utility has various options.

Use This Option	To Do This
-j {partition}	Convert the existing file system to an ext3 file system.
-id|m|w	Specify the maximum time interval between file system checks in days, months, or weeks.
-c maximum mounts count	Specify the maximum number of mounts between file system checks.
-C mount count	Specify the number of times the file system can be mounted.
-r reserved blocks count	Specify the number of reserved file system blocks.
-e continue|remount-ro|panic	Specify the behavior of the kernel code, whether the file system should continue with normal execution, remount the file system in read-only mode, or cause a kernel panic, when errors are detected.
-l	List the contents within the superblock of the file system.
-U UUID	Set the specified Universally Unique Identifier (UUID) for the file system.

The xfs_admin Command

The **xfs_admin** command allows you to manage the parameters of an XFS file system. As with the **tune2fs** command, you need to unmount the file system before using the **xfs_admin** command to change parameters.
The syntax of the **xfs_admin** command is **xfs_admin [options] /dev/{filesystem}**.

The dumpe2fs Utility

The **dumpe2fs** utility is used for managing ext2, ext3, and ext4 (extended) file systems. It dumps the status of the extended file system onto the standard output device and prints the block group information for the selected device.
The syntax of the **dumpe2fs** command is **dumpe2fs [options] [block size] {device name}**.
The **dumpe2fs** utility has various options.

Option	Enables You To
-l	Print a detailed report about block numbers in the file system.
-b	Print the bad blocks in the file system.
-f	Force the utility to display the file system status irrespective of the file system flags.
-i	Display file system data from an image file created using the e2image utility.

The debugfs Utility

The **debugfs** utility allows you to examine and modify ext2, ext3, and ext4 file systems. When executed, the **debugfs** utility opens an interactive shell that can be used to examine and modify the extended file system.
The table provides some common commands supported by the **debugfs** utility in the interactive shell.

If You Need To	Use This Command
Open a file system	Convert the existing file system to an ext3 file system.
Close the file system	close
View the file system information	stats
Find a free block	ffb

xfs Tools

There are many **xfs** tools that allow you to work with the XFS file system.

xfs Tool	Enables You To
xfs_info	Display details about the XFS file system.
xfs_metadump	Copy the metadata information of the XFS file system to a file.
xfs_grow	Expand the XFS file system to fill the disk size.
xfs_repair	Repair and recover a corrupt XFS file system.
xfs_db	Debug the XFS file system.

Managing Partnerships and the Linux File System Lab

Creating Partitions - Lab

In this lab you will be adding a new hard drive to the virtual machine and configuring it for use.
Your organization has a support team that works in two shifts. One employee uses a system in the morning shift and the same system is used by another in the evening shift. Both need to have separate partitions mounted according to the details given here:

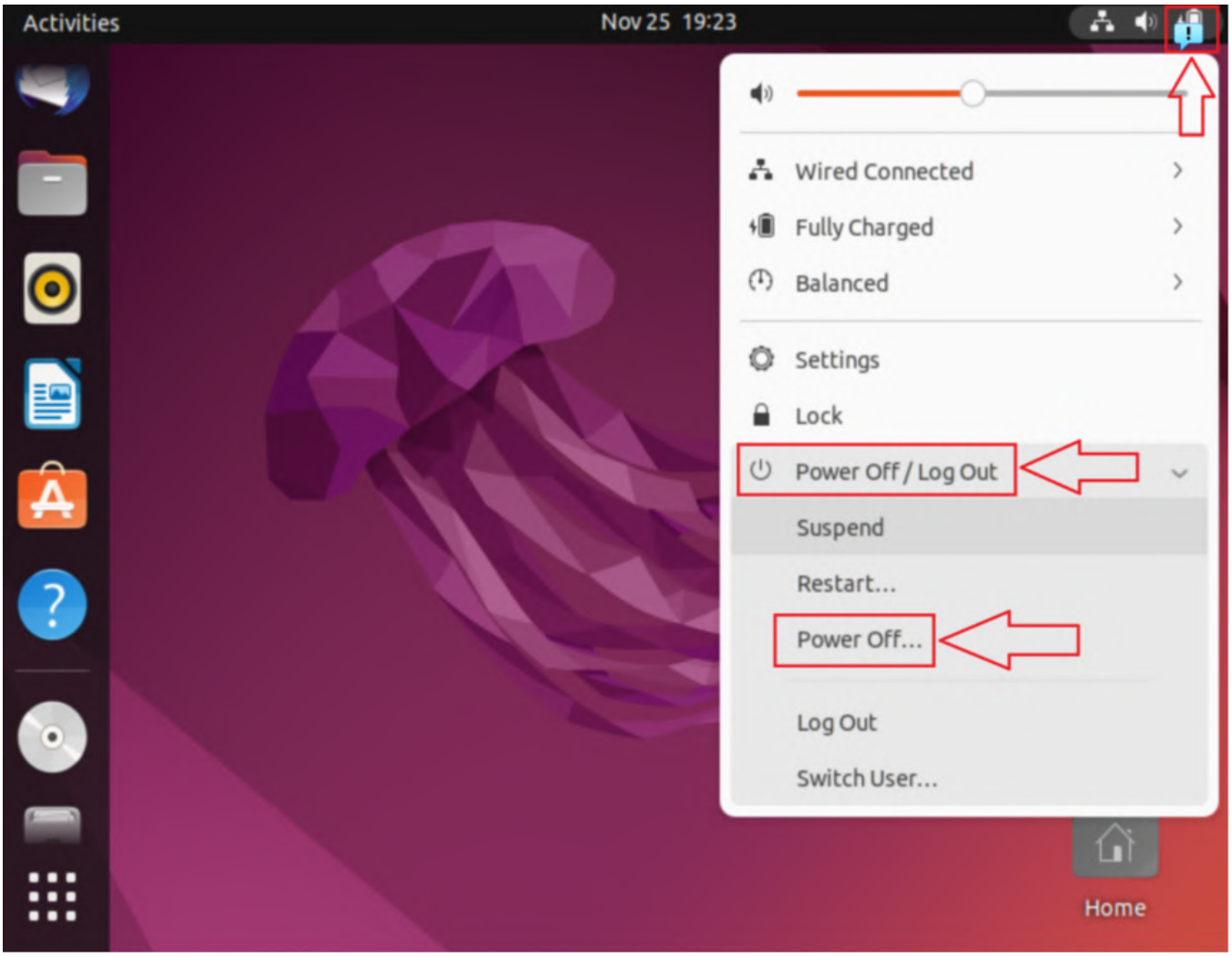
- The logical partitions, sdb5 and sdb6, need to be mounted in the /morning and /evening directories, respectively.
- Both partitions should be formatted with the ext4 file system.

You also need to ensure that these partitions are easily identified for maintenance. The labels that need to be applied to the partitions and used for mounting them are:

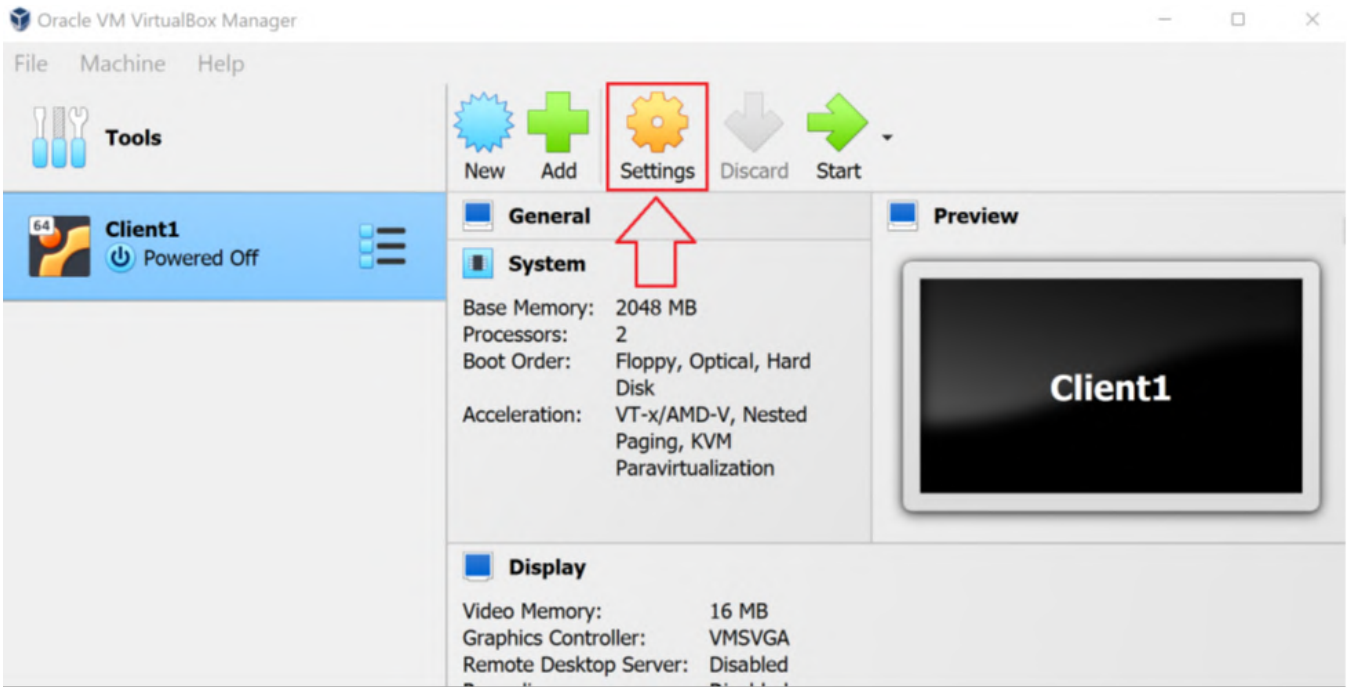
- For the morning shift: Mrng
- For the evening shift: Evng

TASK A

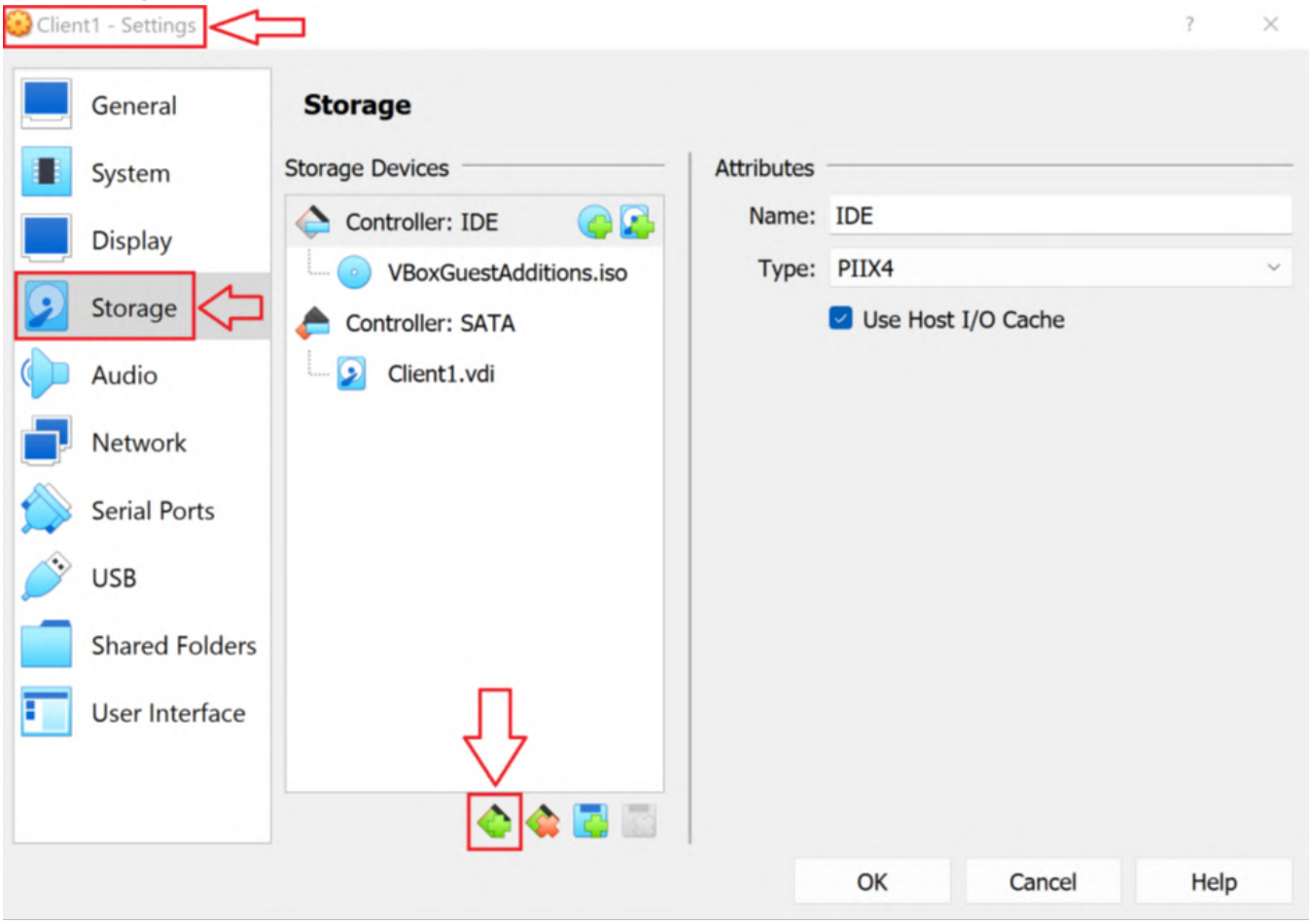
In this task you will add the virtual hardware to the virtual machine. For this task, the virtual machine must be in an off state. It cannot be in a saved state. If you have previously saved the virtual machine, you will need to launch it. Then using the battery icon, choose **Power Off/Log Out** and then click **Power Off**. When prompted, click the **Power Off** button.



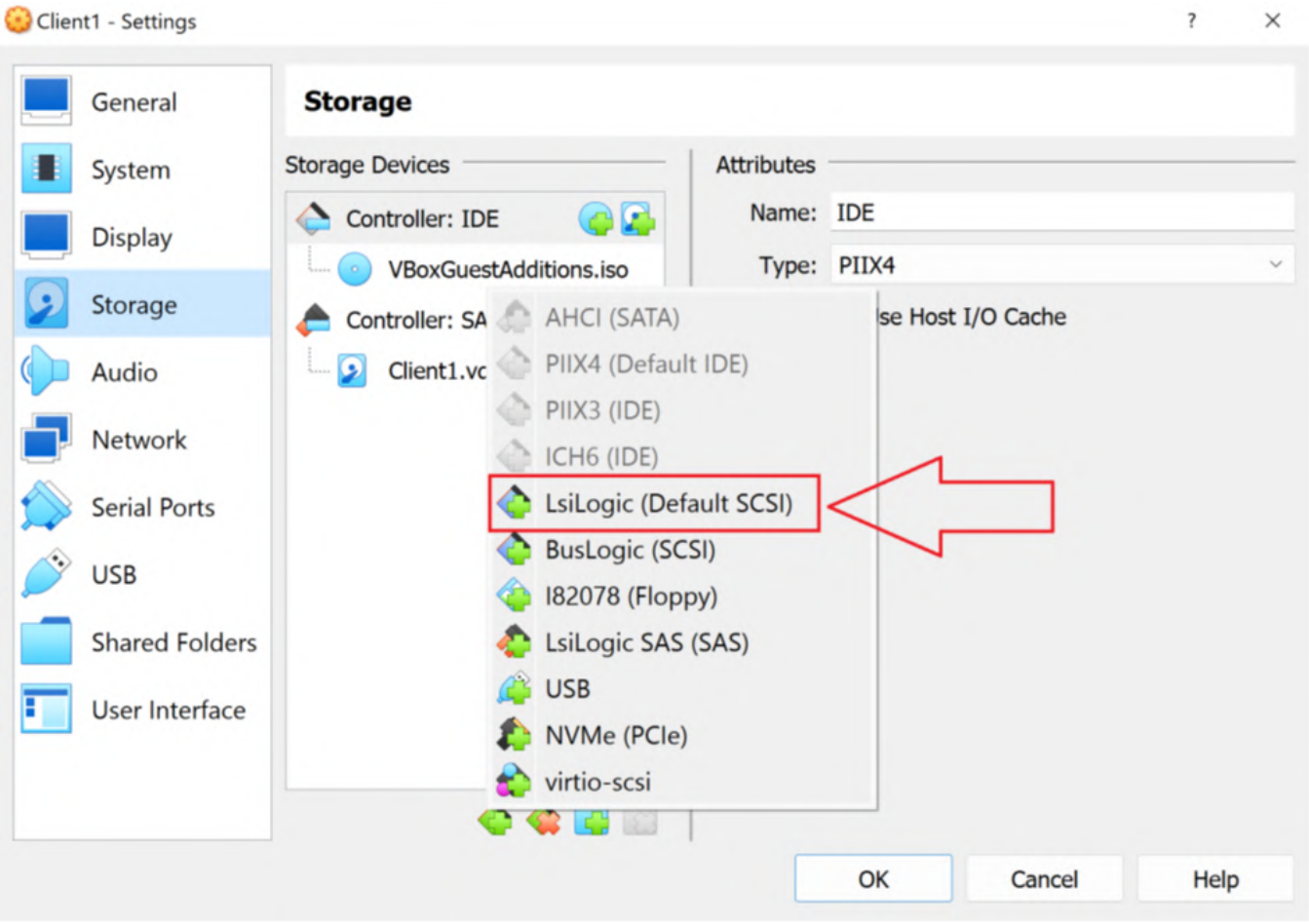
1. In the **Oracle VM VirtualBox Manager** window, with the **Client1** virtual machine selected, click **Settings**.



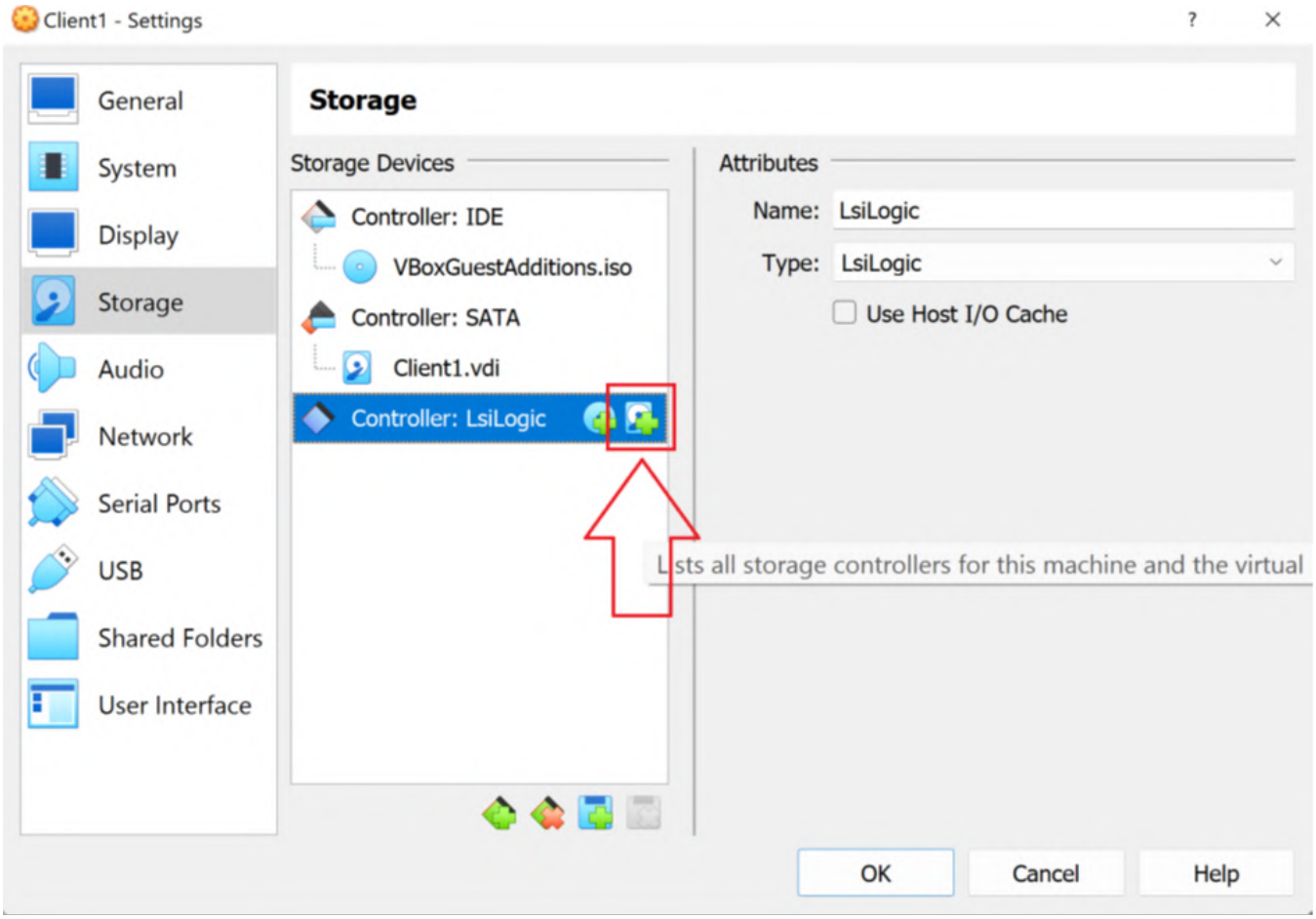
2. In the **Client1 - Settings** dialog box, select **Storage**. In the **Storage Devices** section, click the **Adds new storage controller** button.



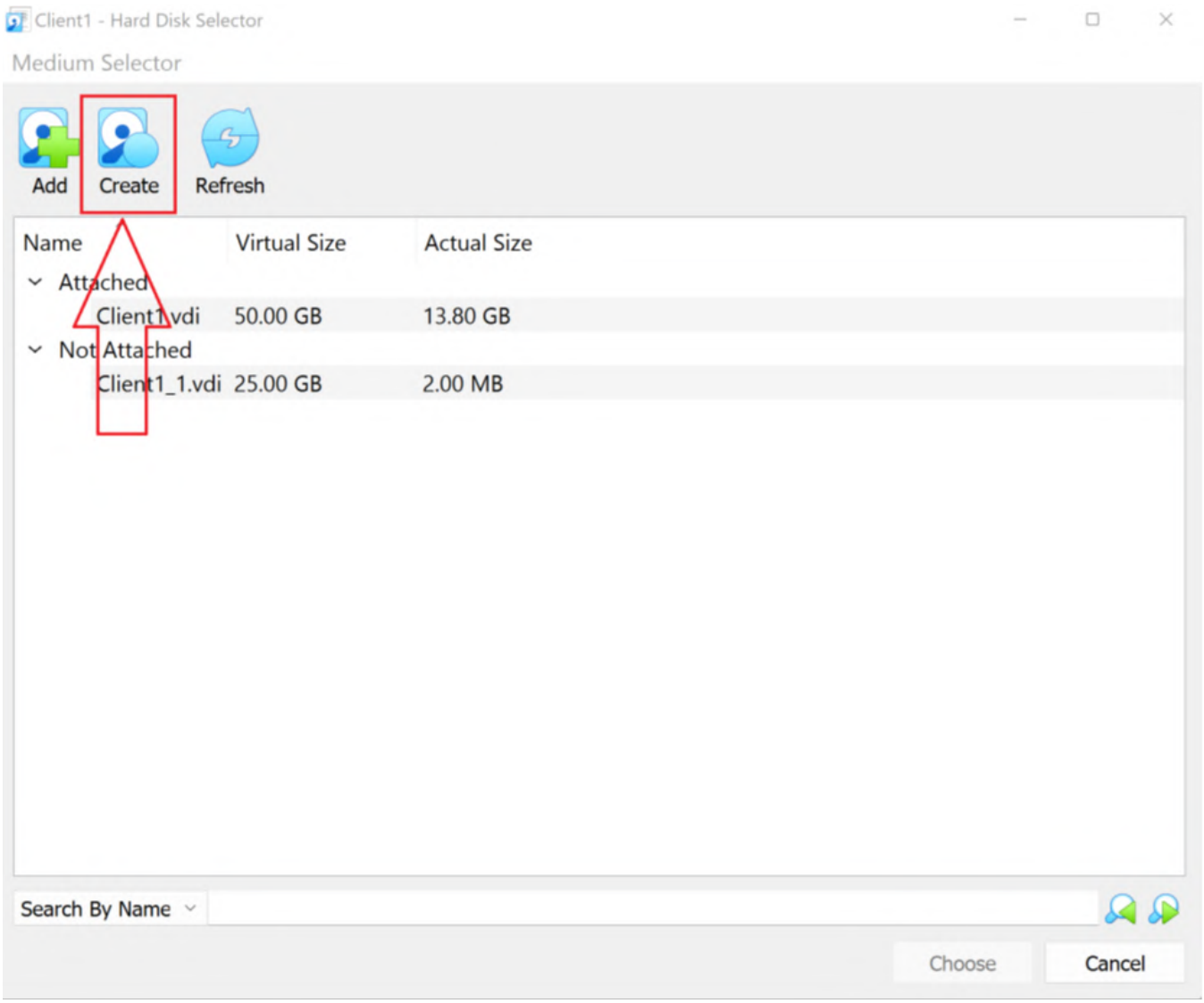
3. Select **LsiLogic (Default SCSI)** from the menu.



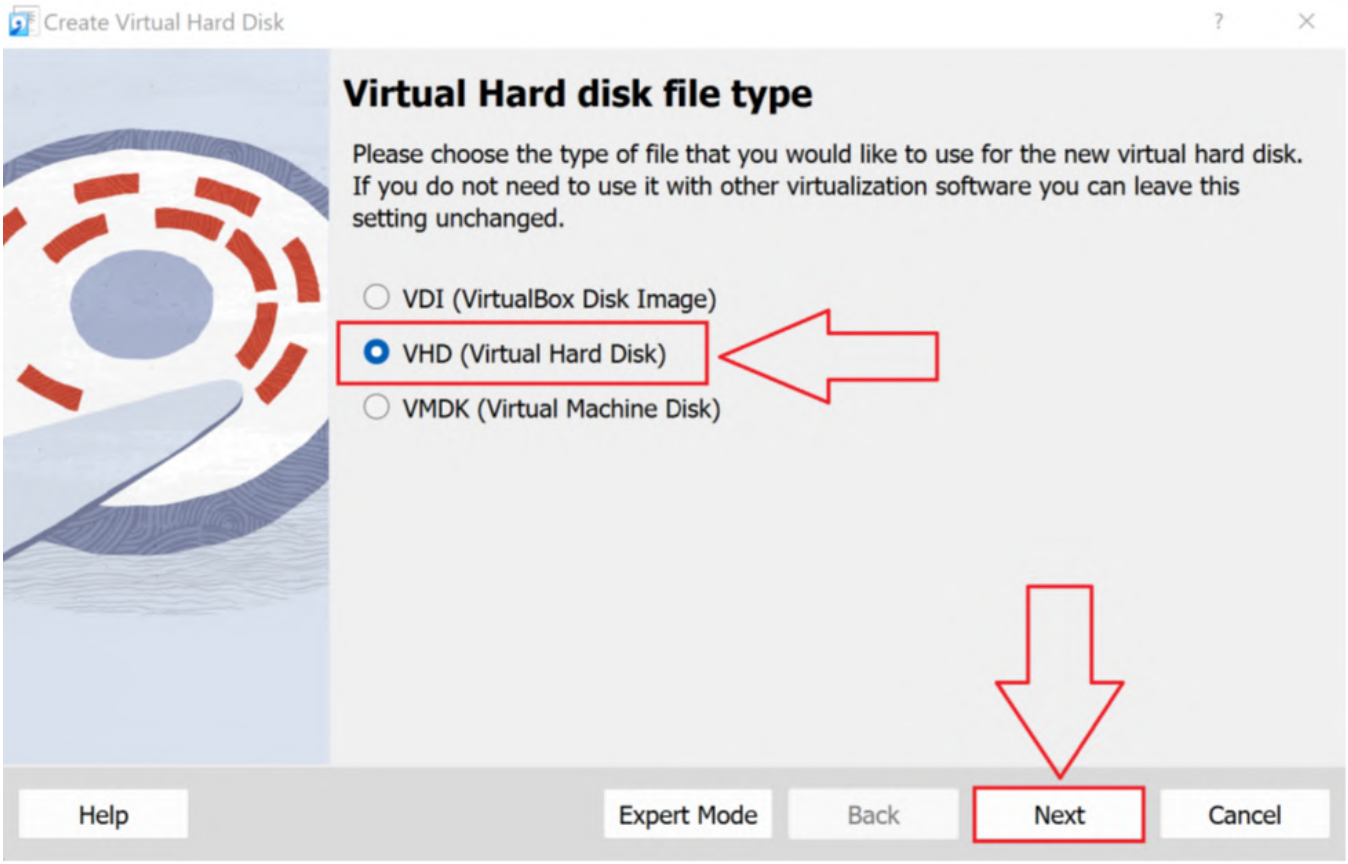
4. Next to **Controller: LsiLogic**, click the **Adds hard disk** button.



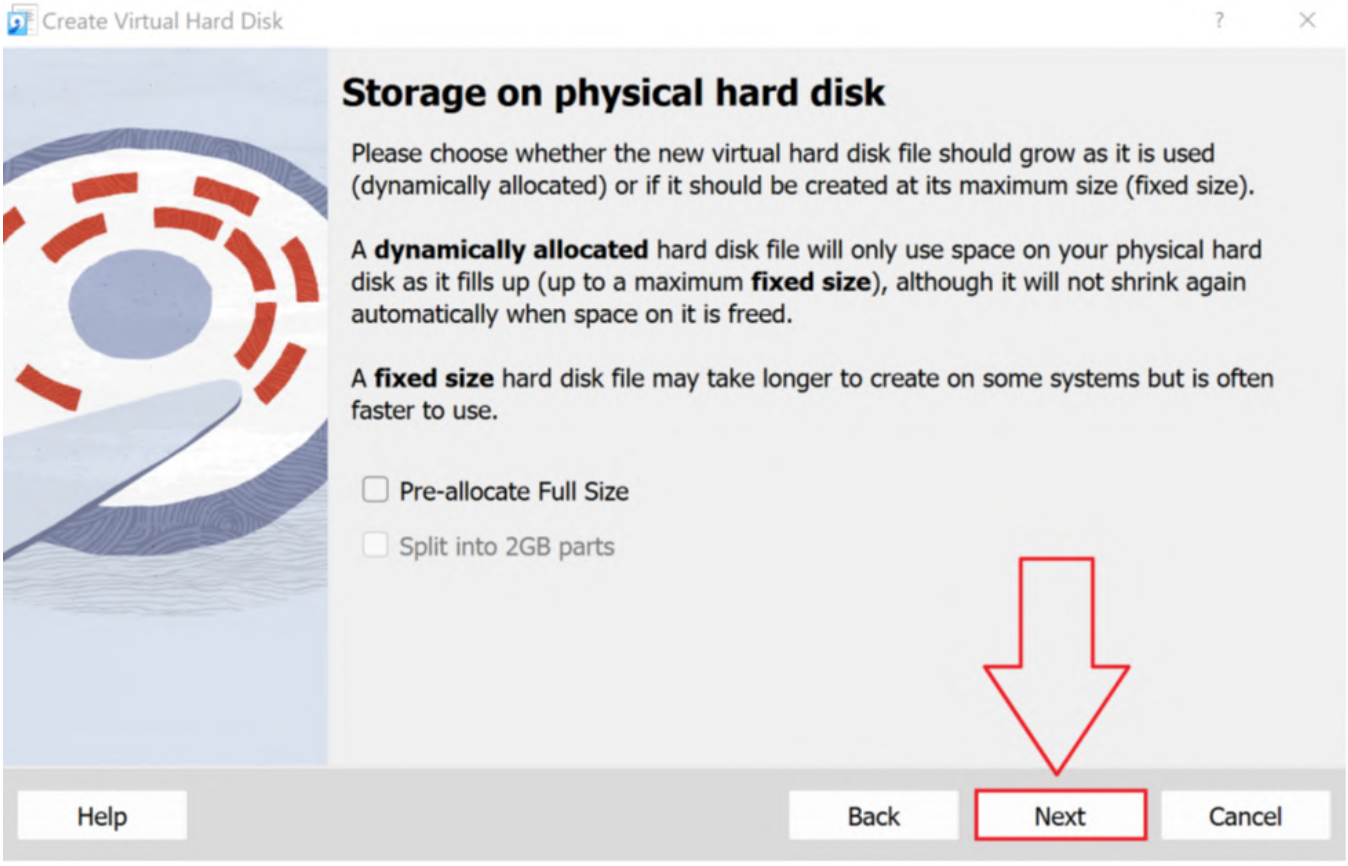
5. In the **Client1 - Hard Disk Selector** dialog box, click **Create**.



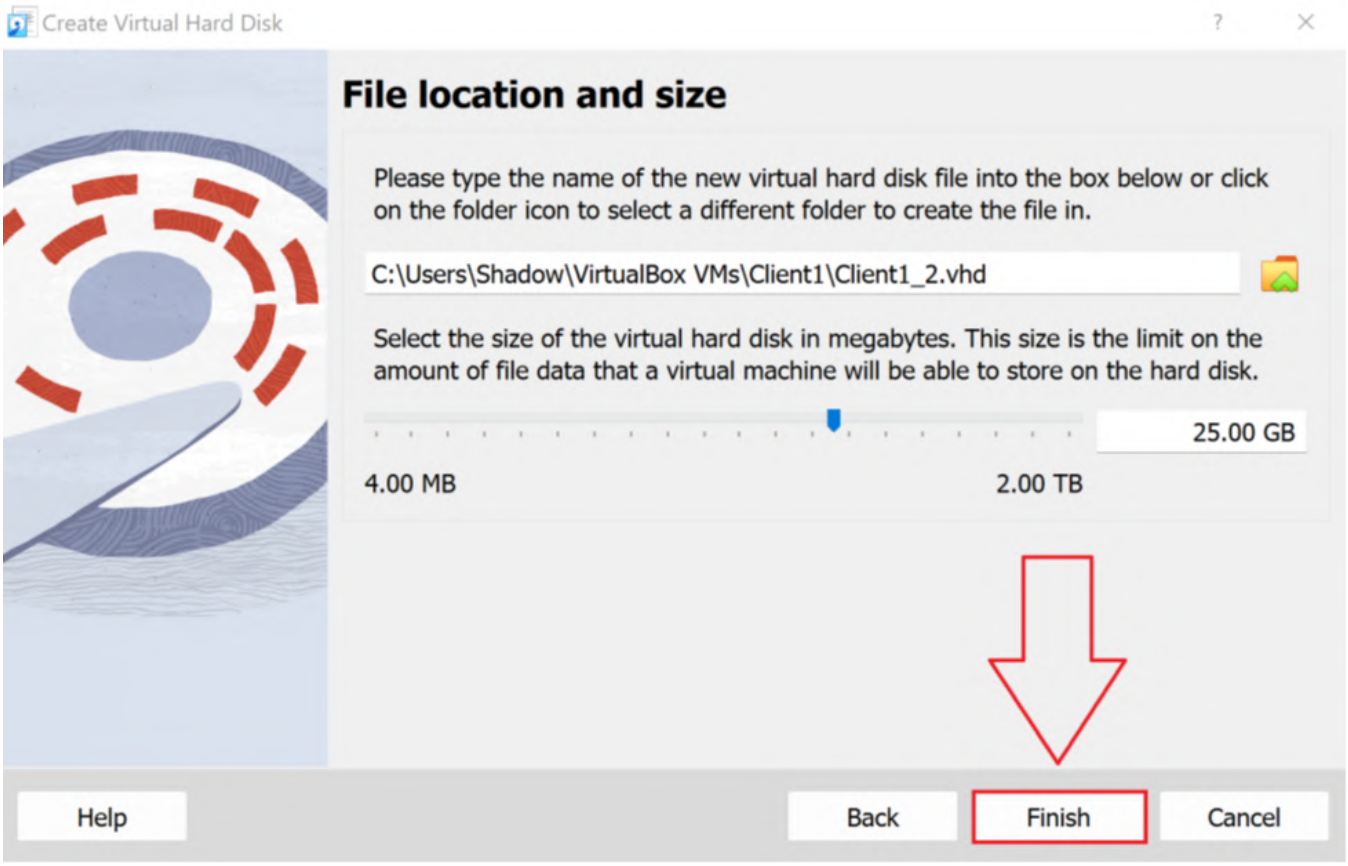
6. In the **Hard disk file type** dialog box, select **VHD (Virtual Hard Disk)**, and then click **Next**.



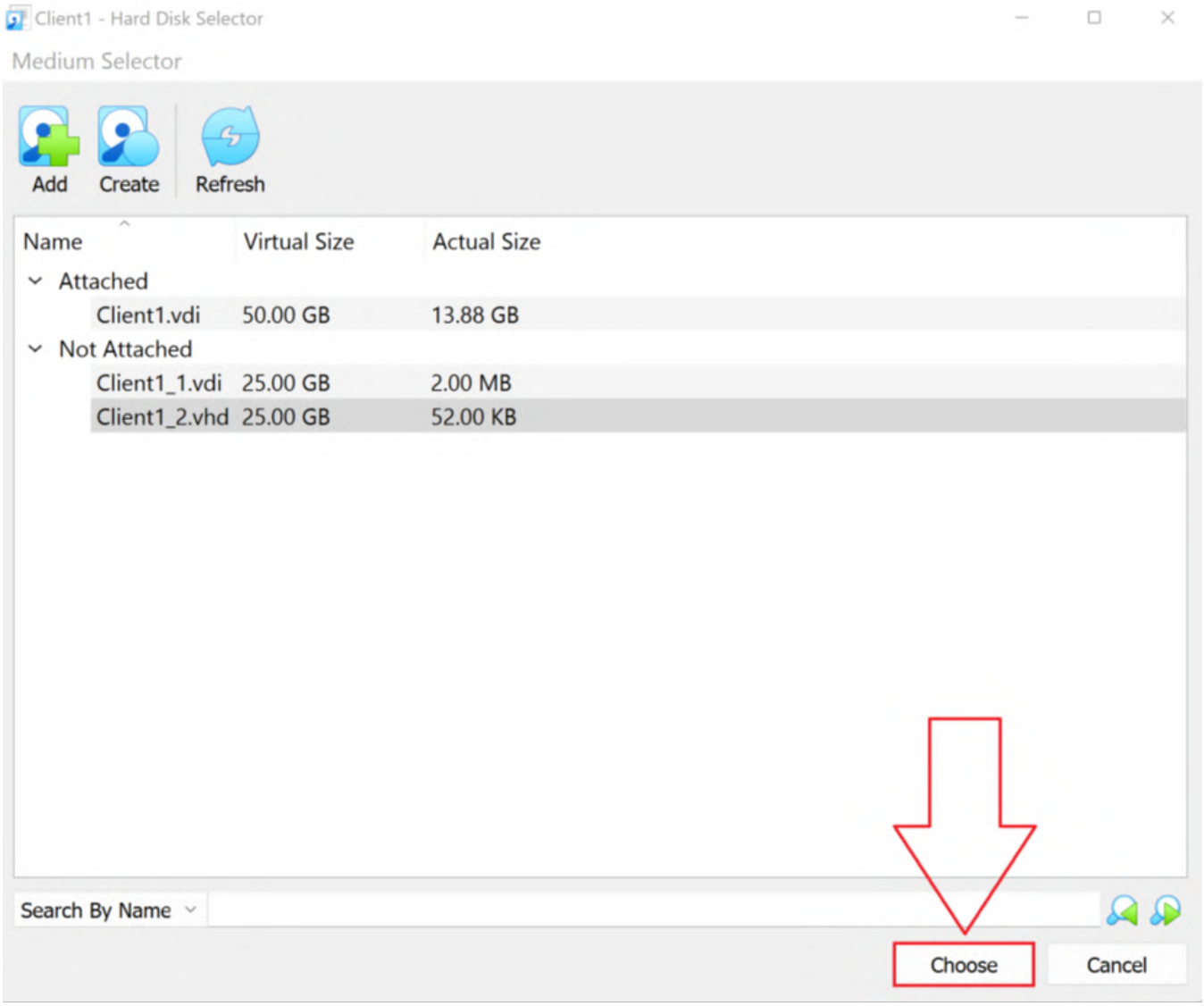
7. In the **Storage on physical hard disk** dialog box, click **Next**.



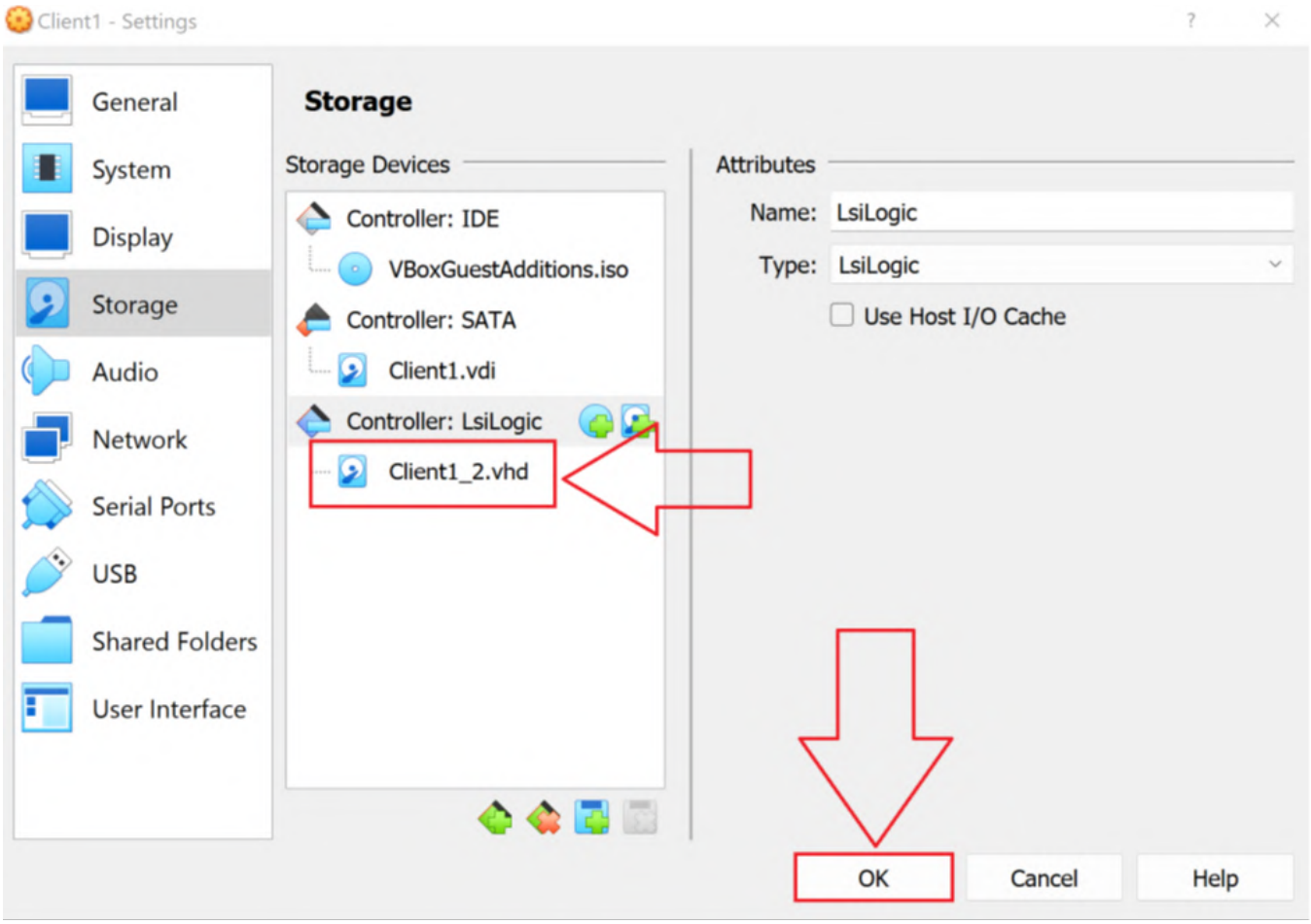
8. In the **File location and size** dialog box, click **Finish**.



9. In the **Client1 - Hard Disk Selector** dialog box, verify the new virtual hard disk is selected, and then click **Choose**.



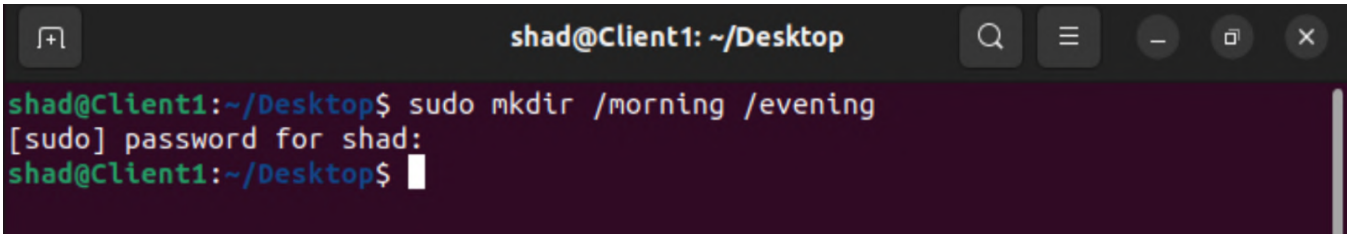
10. In the **Client1 - Settings** dialog box, verify the new hard disk appears, and then click **OK**.



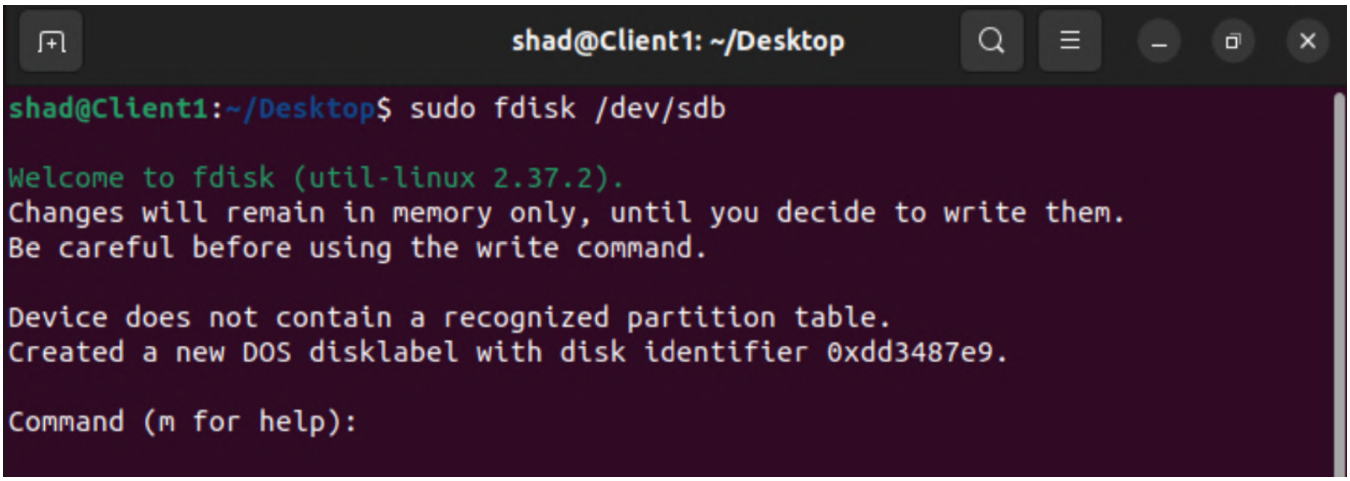
TASK B

In this task you will create the partitions and mount them.

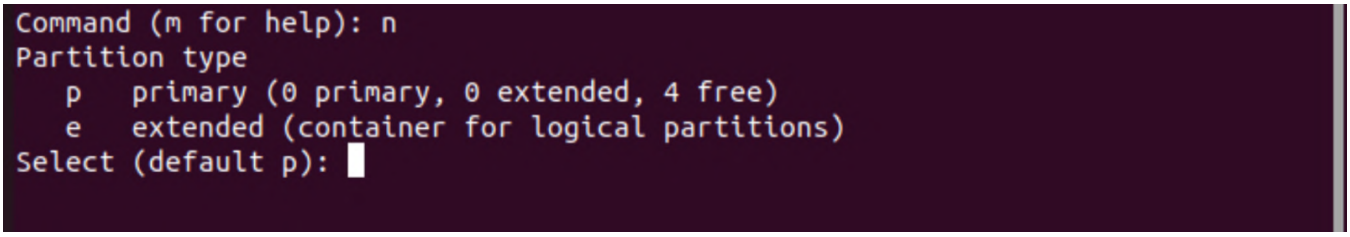
1. Start the **Client1** virtual machine.
2. Log in using your user account and then open a terminal.
3. To create two directories, at the command line, enter **sudo mkdir /morning /evening** and then press **Enter**.



4. To begin the disk partitioning process, type **sudofdisk /dev/sdb** and then press **Enter**.

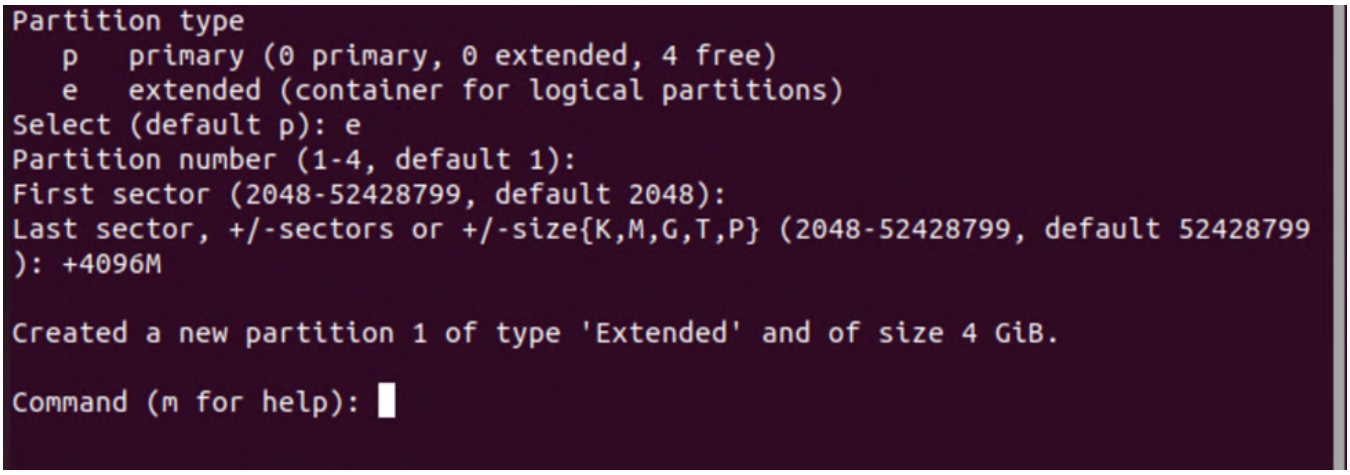


5. To create a new partition, type **n** and then press **Enter**.



6. To create an extended partition, type **e** and then press **Enter**.

7. To accept the default starting Partition number of 1, press **Enter**.To accept the default starting point of the First sector for the partition, press **Enter**.To specify the size of the partition, when prompted for the Last sector, enter **+4096M** and then press **Enter**.



8. To create a new partition, type **n** and then press **Enter**.To create a logical partition, enter **l** and then press **Enter**. Note: This character is a lowercase “L”.To accept the default starting point of the First sector for the partition, press **Enter**.To specify the size of the partition, type **+1024M** and then press **Enter**.

```
Command (m for help): n
Partition type
   p   primary (0 primary, 1 extended, 3 free)
   l   logical (numbered from 5)
Select (default p): l

Adding logical partition 5
First sector (4096-8390655, default 4096):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4096-8390655, default 8390655):
 +1024M

Created a new partition 5 of type 'Linux' and of size 1 GiB.

Command (m for help):
```

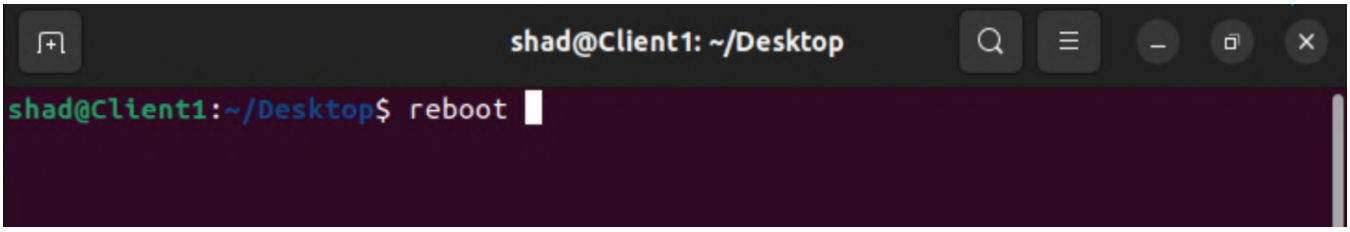
9. Repeat step 8 to create a second logical partition of 1024M.

```
Command (m for help): n
Partition type
   p   primary (0 primary, 1 extended, 3 free)
   l   logical (numbered from 5)
Select (default p): l

Adding logical partition 6
First sector (2103296-8390655, default 2103296):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2103296-8390655, default 839065
5): +1024M

Created a new partition 6 of type 'Linux' and of size 1 GiB.

Command (m for help):
```

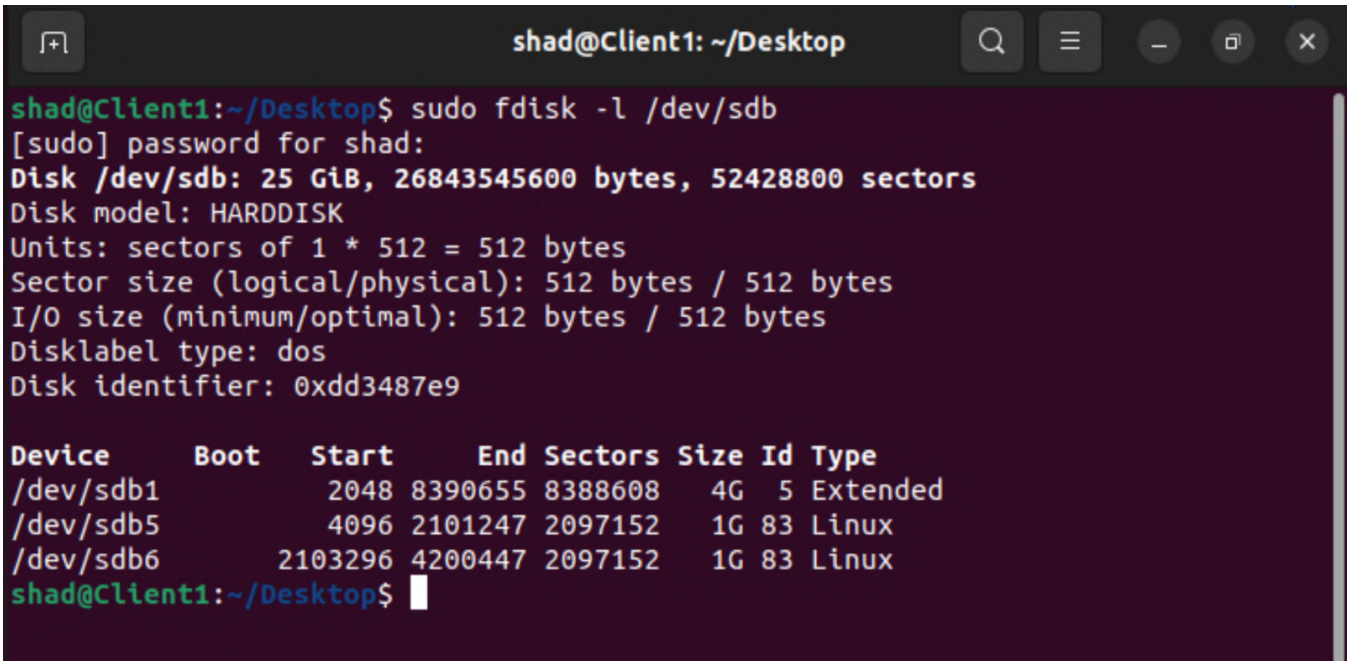
10. To write the partition table on the disk and exit the utility, type **w** and press **Enter**.

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

shad@Client1:~/Desktop$
```

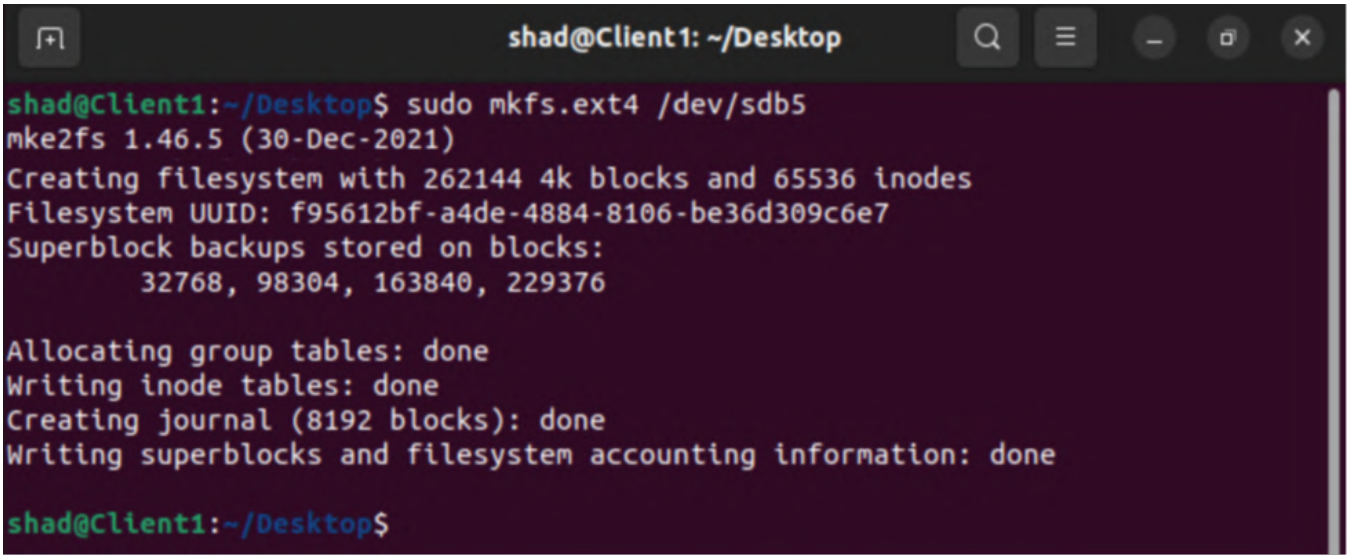
11. To restart the system, type **reboot** and press **Enter**.



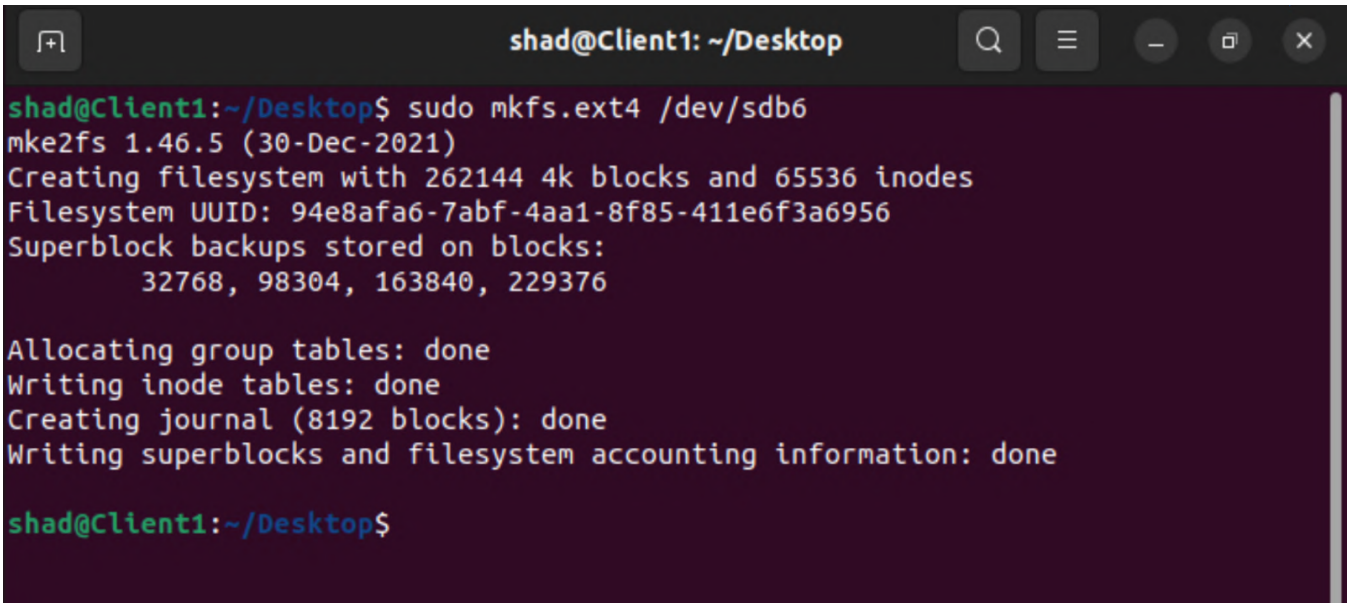
12. After the computer reboots, log in with your user account and then open a terminal.
13. To view the new disk partitioning scheme, type **sudo fdisk -l /dev/sdb** and press **Enter**.



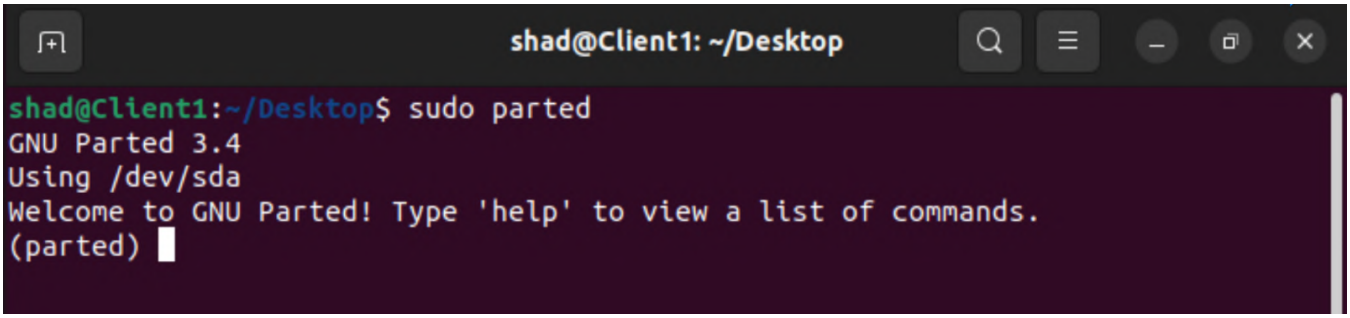
14. To create an ext4 filesystem on**/dev/sdb5**, type **sudo mkfs.ext4 /dev/sdb5** and then press **Enter**.



15. To create an ext4 filesystem on **/dev/sdb6**, type **sudo mkfs.ext4 /dev/sdb6** and then press **Enter**.

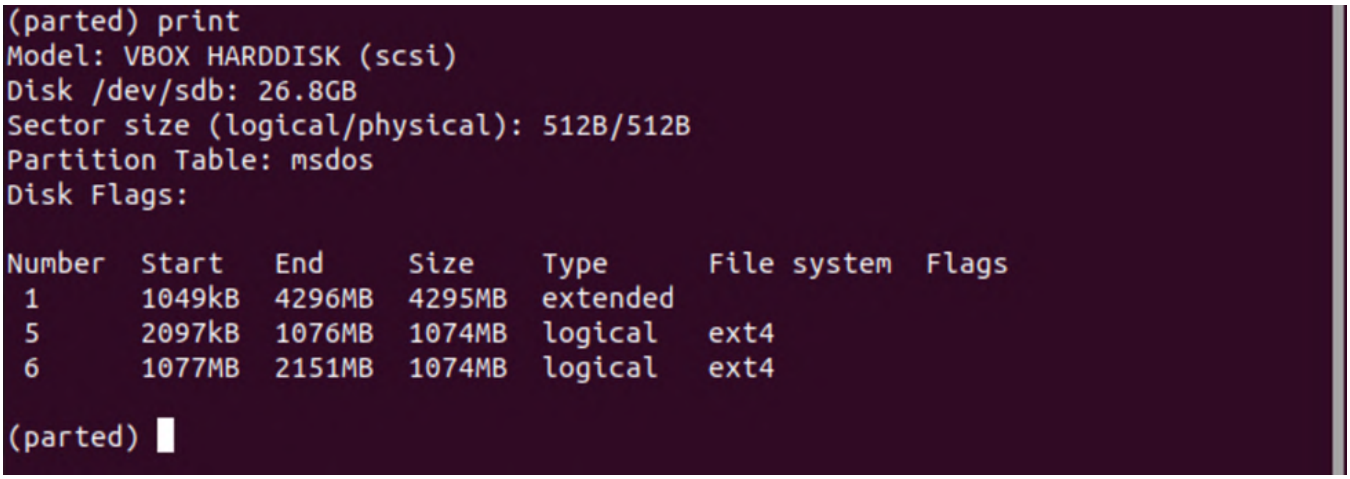


16. To run the **GNU Parted** utility, type **sudo parted** and then press **Enter**.



17. To select the **sdb** partition, type **select /dev/sdb** and then press **Enter**.

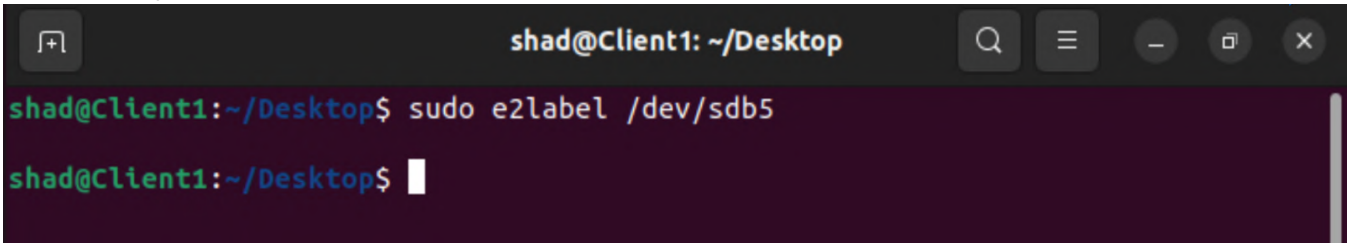
18. To view the list of existing partitions, at the **(parted)** prompt, type **print** and then press **Enter**. Verify that partitions five and six have the ext4.



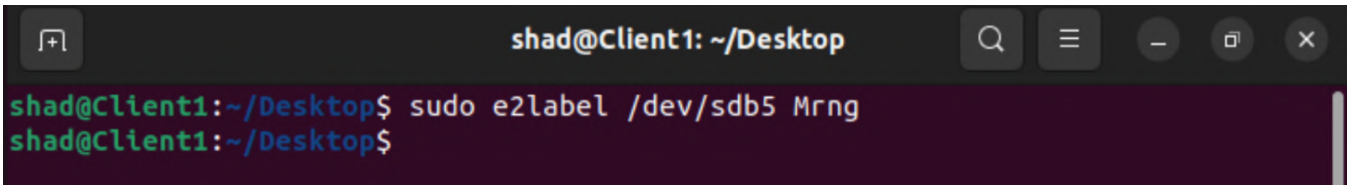
19. To quit from the parted utility, type **q** and then press **Enter**.

```
(parted) q
shad@Client1:~/Desktop$
```

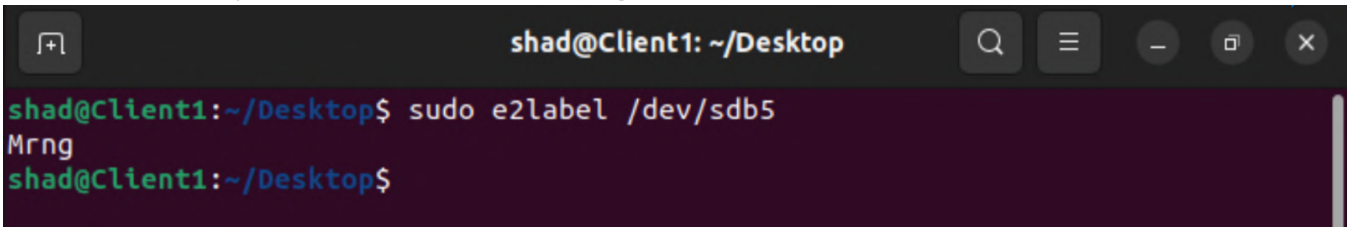
20. To view the existing label of the **/dev/sdb5** partition, type **sudoe2label /dev/sdb5** and then press **Enter**. Verify that there is no label set for the **/dev/sdb5** partition.



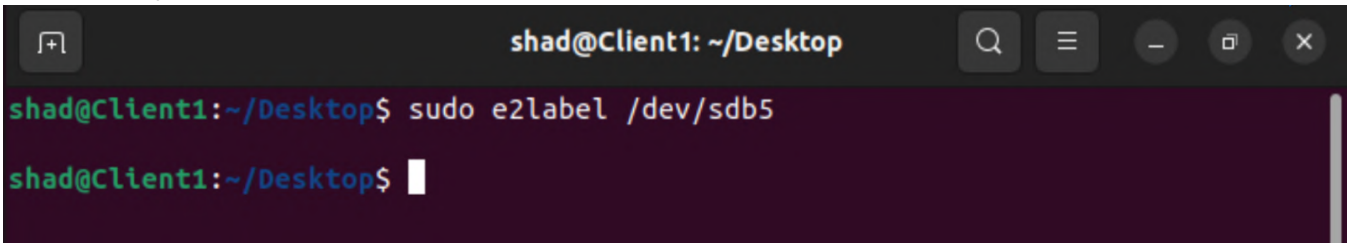
21. To apply a new label **Mrng** to the partition, type **sudoe2label /dev/sdb5 Mrng** and then press **Enter**.



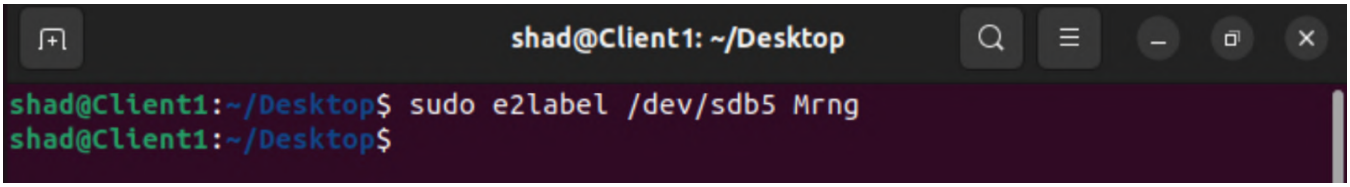
22. To verify that the partition label for **/dev/sdb5** has changed, type **sudo e2label /dev/sdb5** and then press **Enter**. Verify that the label is set as "**Mrng**" for the **/dev/sdb5** partition.



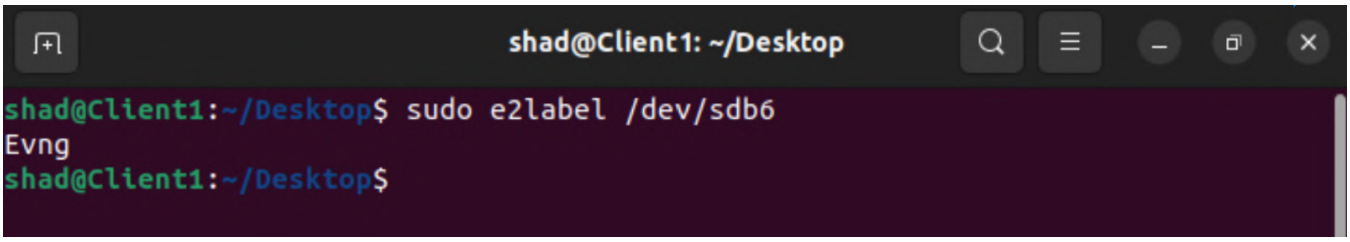
23. To view the existing label of the **/dev/sdb6** partition, type **sudo e2label /dev/sdb6** and then press **Enter**. Verify that there is no label set for the **/dev/sdb6** partition.



24. To apply a new label **Evng** to the partition, type **sudo e2label /dev/sdb6 Evng** and then press **Enter**.



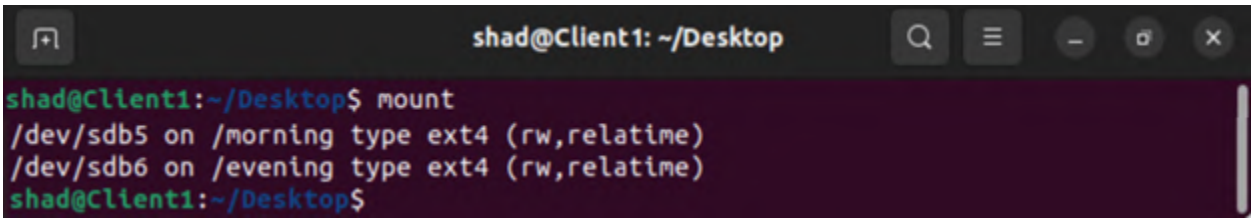
25. To verify that the partition label for **/dev/sdb6** has changed, type **sudo e2label /dev/sdb6** and then press **Enter**. Verify that the label is set as "**Evng**" for the **/dev/sdb6** partition.



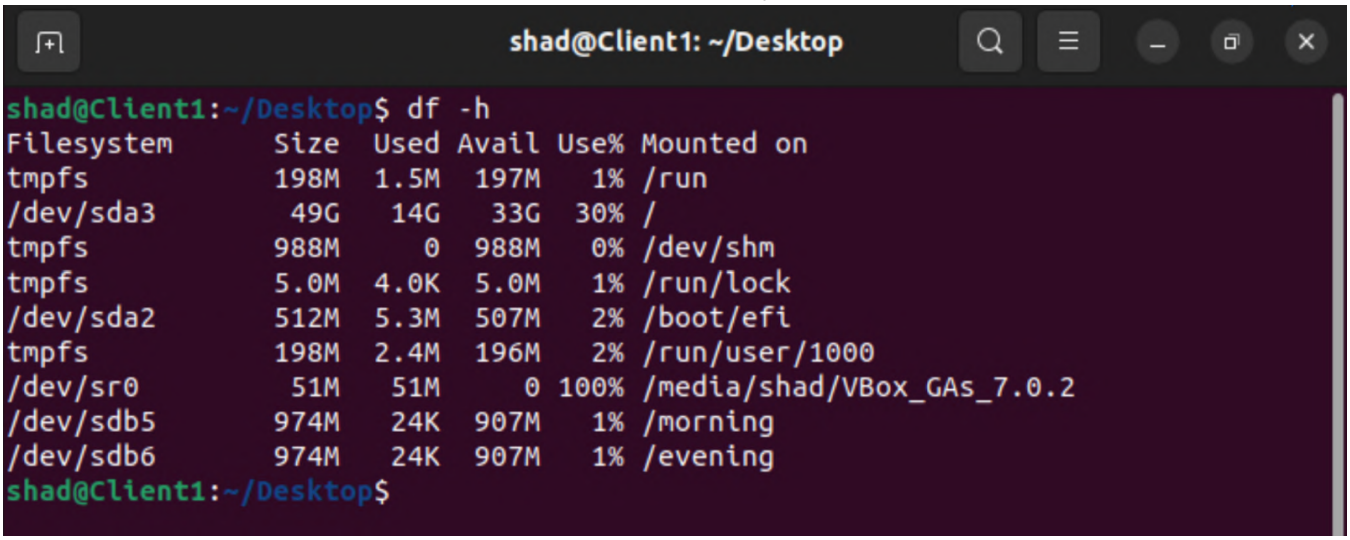
26. To mount the **/dev/sdb5** partition using its label, type **sudo mount LABEL=Mrng /morning** and then press **Enter**.

27. To mount the **/dev/sdb6** partition using its label, type **sudo mount LABEL=Evng /evening** and then press **Enter**.

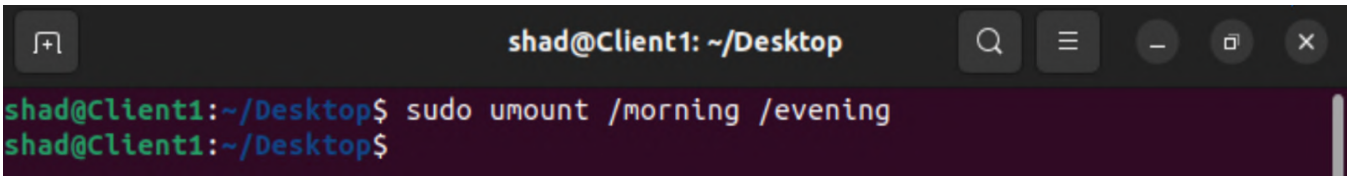
28. To verify that the partitions have been mounted using their labels, type **mount** and then press **Enter**.Verify that the partitions **/dev/sdb5** and **/dev/sdb6** are mounted on the**/morning** and **/evening** directories.



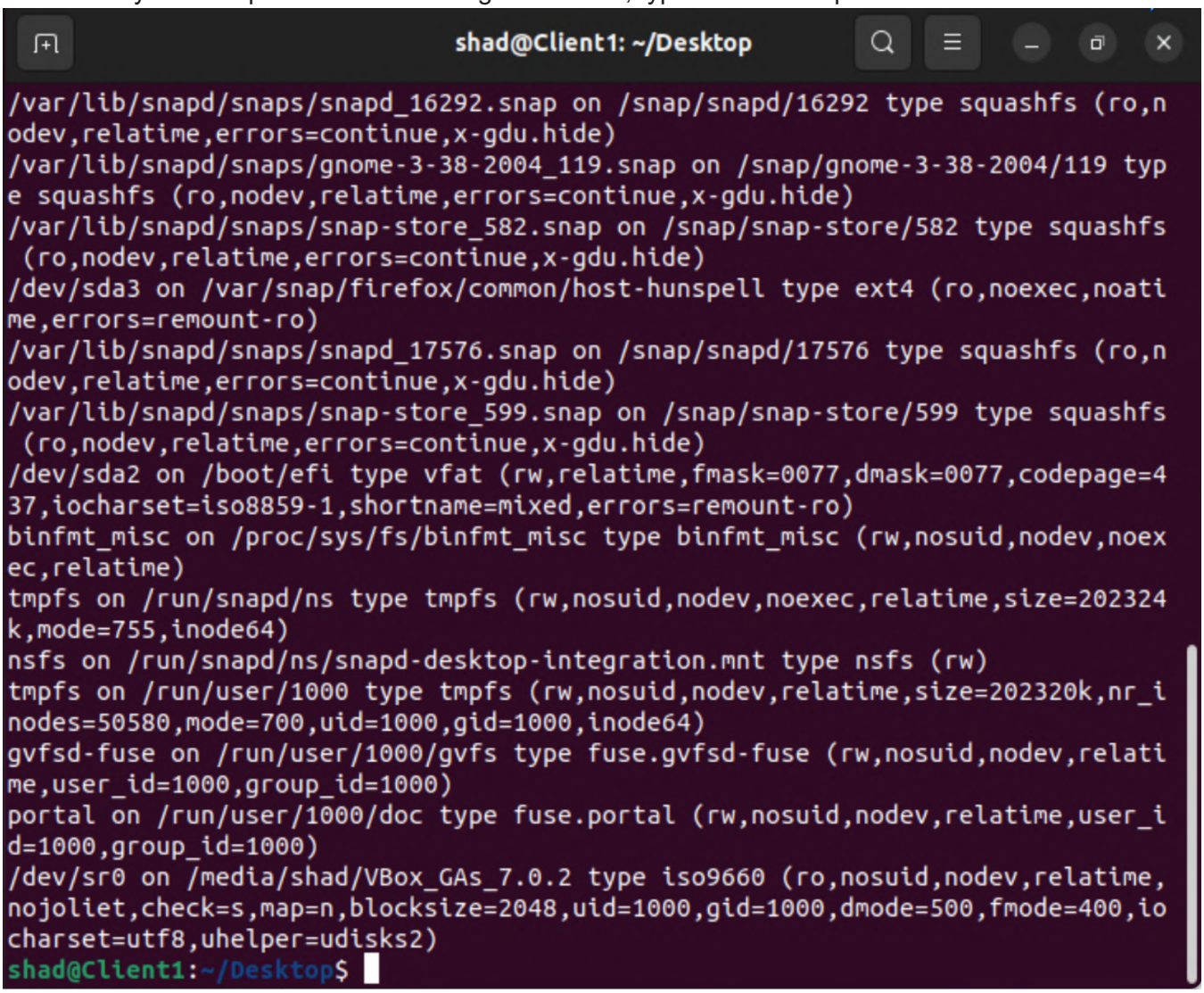
29. To show that the partitions have free space available, type **df -h** and press **Enter**.



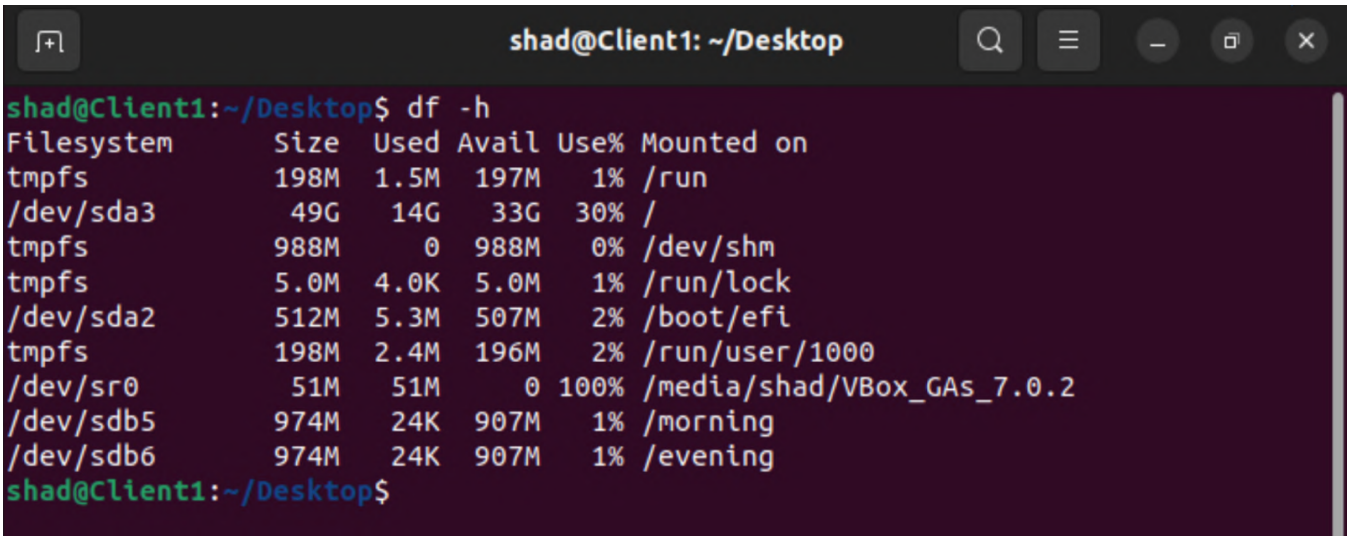
30. To unmount the partitions, type **sudo umount /morning /evening** and press **Enter**.



31. To verify that the partitions are no longer mounted, type **mount** and press **Enter**.



32. To remove the prior mount points, type **sudo rmdir /morning /evening** and press **Enter**.



Mounting File Systems - Lab

There is a meeting at your office. A couple of users want to have their systems moved to the conference room so that they can access their files during the conference. You find that there are multiple systems to be moved, and this will take a lot of time. Therefore, you decide to take the required files from the users and load them on the system in the conference room in separate partitions, so that the users can access their files. Use the following user and partition details to mount the filesystems:

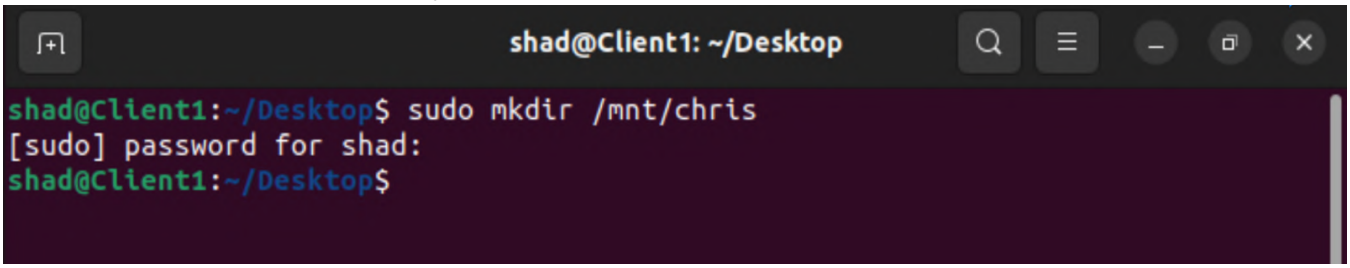
• User name: chris, Mount point: /mnt/chris, User and group owner of /mnt/chris: chris
• User name: pat, Mount point: /mnt/pat, User and group owner of /mnt/pat: pat

Before you start this lab, launch the Client1 virtual machine. Log on with your user account and open a terminal.

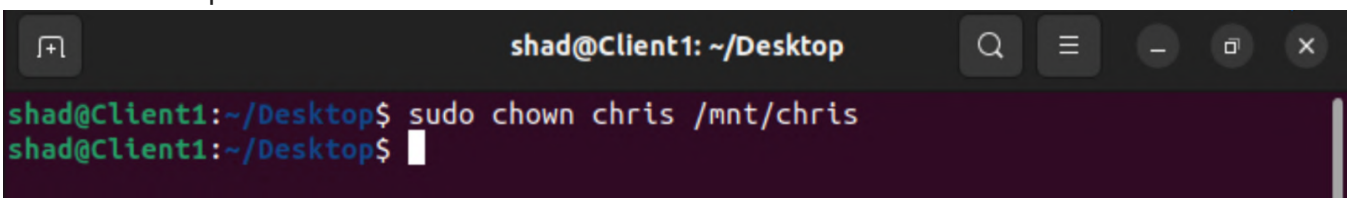
TASK A

In this task you will be creating the mount points and setting them to mount automatically.

1. To create a new mount point, type **sudo mkdir /mnt/chris** and press **Enter**.

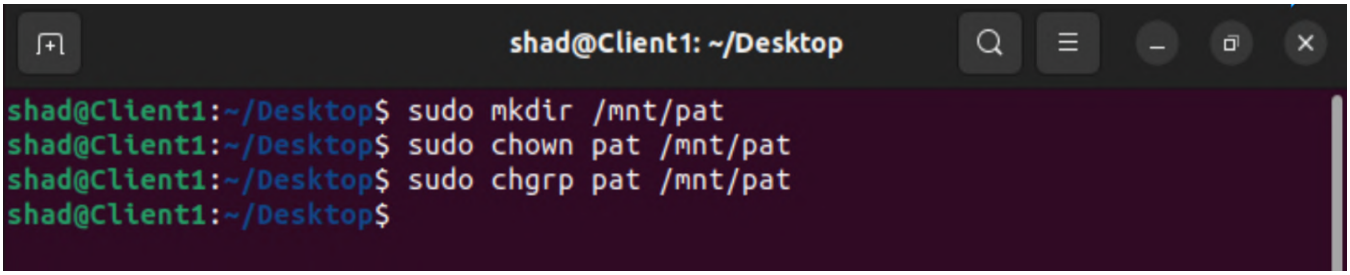


2. To set the **chris** user as the owner of the **/mnt/chris** mount point, type **sudo chown chris /mnt/chris** and press **Enter**.

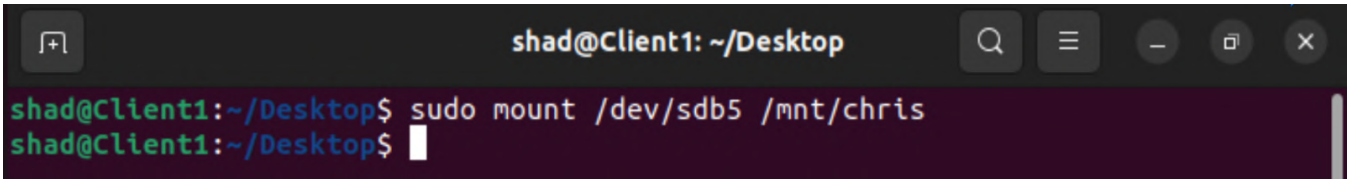


3. To set the **chris** group as the owner of **/mnt/chris** mount point, type **sudo chgrp chris /mnt/chris** and then press **Enter**.

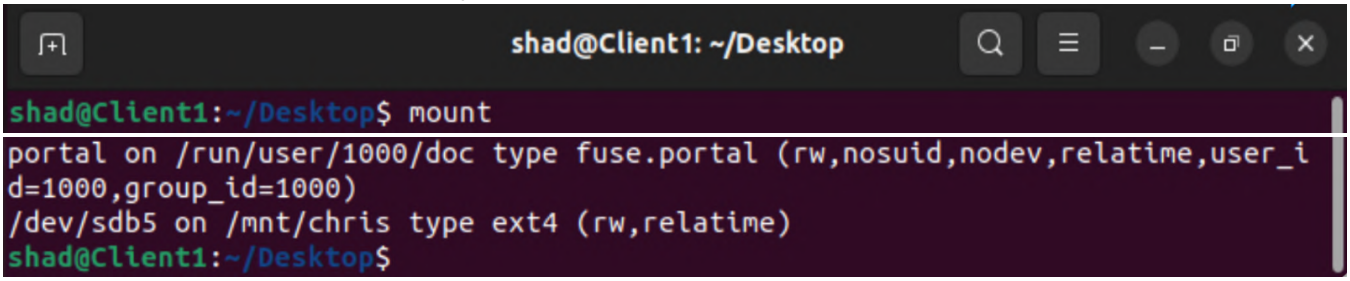
4. Create a mount point, **/mnt/pat**, and assign user ownership and group ownership to **pat** by repeating the steps from 1-3.



5. To mount the filesystem for **chris**, type **sudo mount /dev/sdb5 /mnt/chris** and press **Enter**.

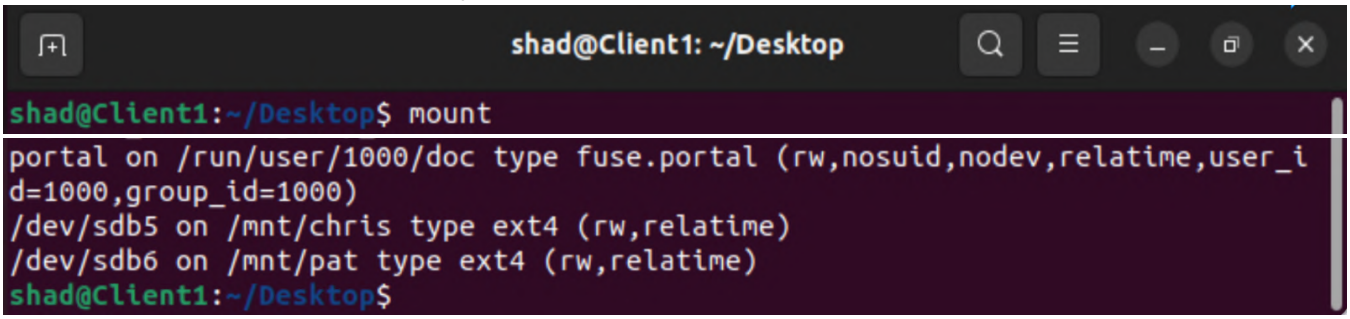


6. To view the mounted partitions, type **mount** and press **Enter**.

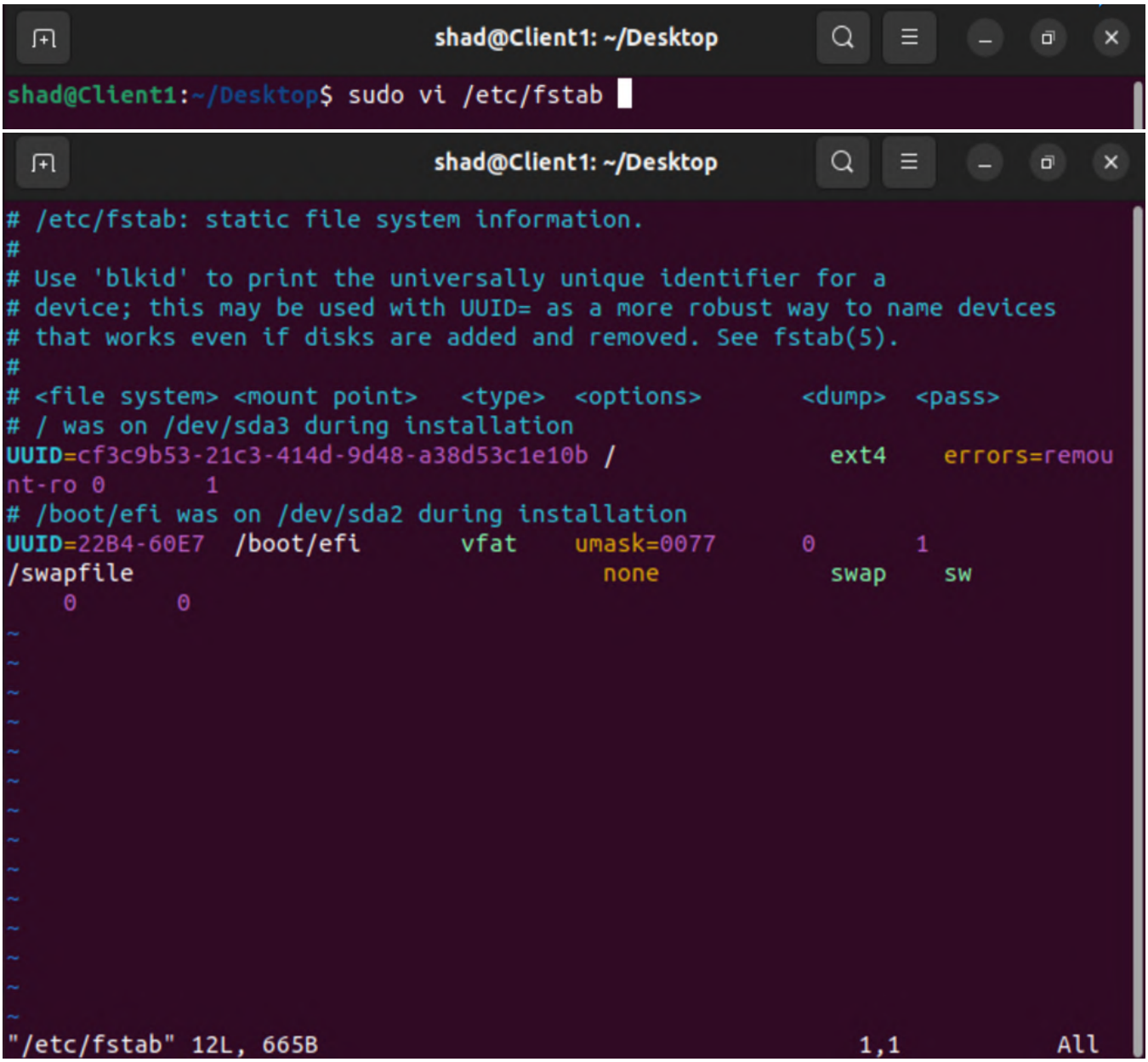


7. To mount the filesystem for **pat**, enter **sudo mount /dev/sdb6 /mnt/pat** and press **Enter**.

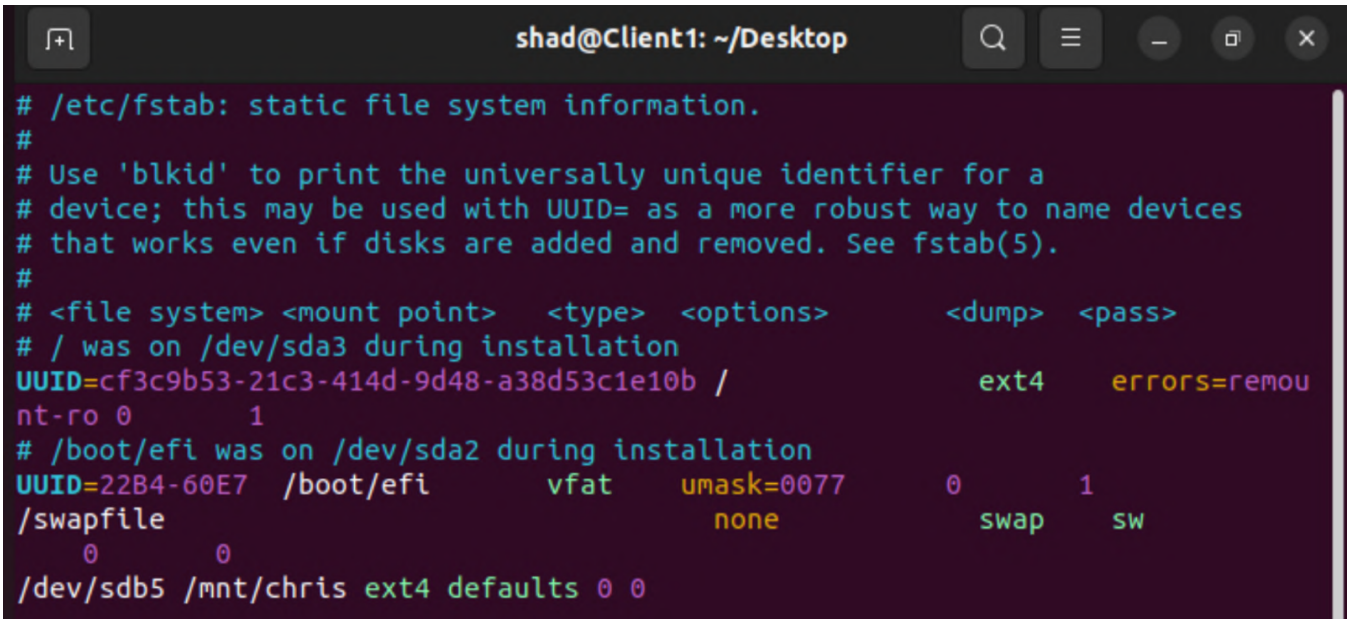
8. To view the mounted partitions, type **mount** and press **Enter**.



9. To open the **fstab** file, enter **sudo vi /etc/fstab** and press **Enter**.

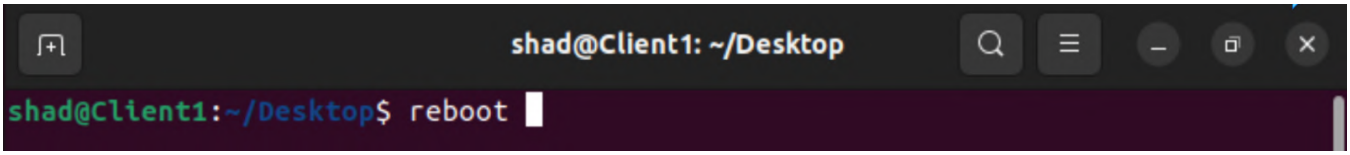


10. To go to the last line, press **Shift+G**.
11. To switch a new line in insert mode, press **o**.
12. To mount the **/dev/sdb5** filesystem when the system boots, on a new line after the last line in the file, type **/dev/sdb5 /mnt/chris ext4 defaults 0 0** and press **Enter**.

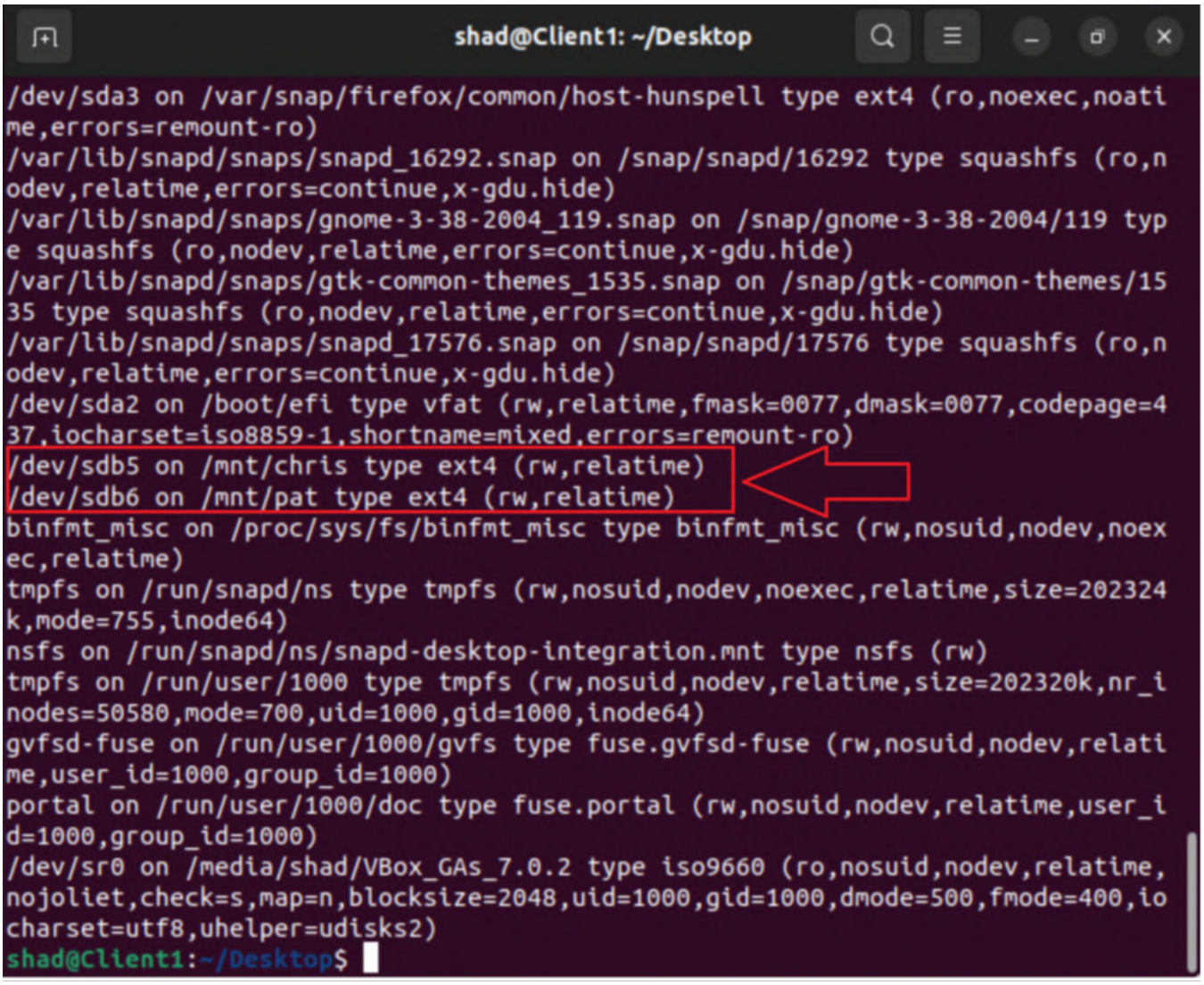


13. To mount the **/dev/sdb6** filesystem when the system boots, type **/dev/sdb6 /mnt/pat ext4 defaults 0 0** and press **Enter**.

14. Save and close the file by pressing **Esc** then typing **:wq** and pressing **Enter**.
15. To reboot the system, type **reboot** and press **Enter**.



16. After the virtual machine reboots, log on with your user account and open a terminal.
17. To verify that the filesystems are mounted at the specified mount points on boot, type **mount** and press **Enter**. Verify that the partitions **/dev/sdb5** and **/dev/sdb6** are mounted into the **/mnt/chris** and **/mnt/pat** directories, respectively.

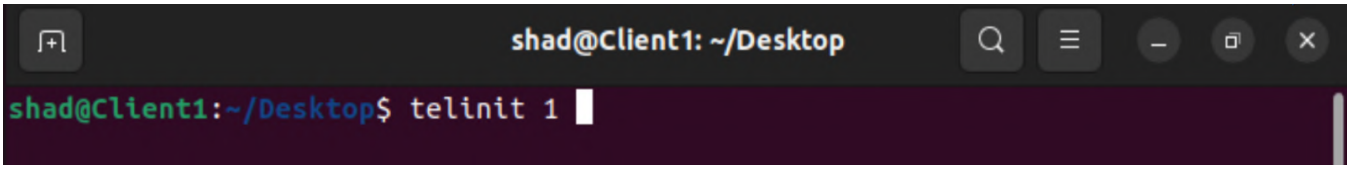


Maintaining Linux File System - Lab

Due to a thunderstorm, there was a brief power outage overnight and the Linux systems were not shutdown properly. The systems need to be checked for consistency. Using the fsck command, verify the drive and data integrity of the hard disk.
Before you start this lab, launch the Client1 virtual machine. Log on with your user account and open a terminal.

TASK A

1. To switch to single-user mode, type **telinit 1** and press **Enter**.



2. When prompted, type your user password. At the **Press Enter for maintenance** prompt, press **Enter**.

```
You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
Press Enter for maintenance
(or press Control-D to continue): _
```

3. Type **umount /dev/sdb5** and press **Enter**.

```
root@Client1:~# umount /dev/sdb5
root@Client1:~#
```

4. Type **fsck /dev/sdb5** and press **Enter**. Verify that the message displays "**clean**", which indicates that there is no error in the filesystem.

```
root@Client1:~# fsck /dev/sdb5
fsck from util-linux 2.37.2
e2fsck 1.46.5 (30-Dec-2021)
Mrng: clean, 11/65536 files, 12955/262144 blocks
root@Client1:~# _
```

5. Reboot the computer by typing **reboot** and pressing **Enter**.
6. You may close all the windows and save the state of the virtual machine. This is the end of the labs for this module.

Configuring System Services and Manage Logs

System Initialization

System initialization begins when a system is booted. It involves the loading of the operating system and its various components, including the boot process. System initialization is carried out by the **init** program in Linux. The **init** program refers to the configuration file and initiates the processes listed in it. This prepares the system to run the required software. Programs on the system will not run without system initialization. We will cover both SysVinit and Systemd initialization in this topic.

The /etc/init.d Directory

The **init.d** directory found in the **/etc** directory stores initialization scripts for services. These scripts, called system V scripts, control the initiation of services in a particular runlevel. These runlevels are called system V runlevels. The scripts are invoked from the **/etc/inittab** file when the system initialization begins, using the symbolic links found in the file. System V scripts are highly flexible and can be configured according to the needs of a user. Some of the services listed in the init.d directory are anacron, cups, and bluetooth.
The syntax for running scripts of the services in the /**etc/init.d** directory is **/{service name} {start|stop|status|restart}**.

The systemclt command

The purpose of an **init** system is to initialize the components that must be started after the Linux kernel is booted (traditionally known as “userland” components). The **init** system is also used to manage services and daemons for the server at any point while the system is running.
In **systemd**, a suite of basic building blocks for a Linux system. It provides a system and service manager that runs as PID 1 and starts the rest of the system, the target of most actions are “units”, which are resources that **systemd** knows how to manage. Units are categorized by the type of resource they represent and they are defined with files known as unit files. The type of each unit can be inferred from the suffix on the end of the file.
For service management tasks, the target unit will be service units, which have unit files with a suffix of **.service**. However, for most service management commands, you can actually leave off the **.service** suffix, as **systemd** is smart enough to know that you probably want to operate on a service when using service management commands.

systemctl Option	**Description**
enable *service*	Enable a service to be started on boot.
list-unit-files	List configured system services and their boot configuration.
disable *service*	Disable a service so that it is no longer started on boot.
start *service*	Start (activate) a service immediately.
stop *service*	Stop (deactivate) a service immediately.
restart *service*	Restart a service immediately.
enable *service*	Enable a service to be started on boot.

System Logs

System logs are records of system activities that are tracked and maintained by the **syslogd** utility. The **syslogd** utility runs as a daemon. System logs are usually started at boot. System log messages include the date, the process that delivered the message, and the actual message.

Logging Services

A logging service is a daemon that is used to track logs or errors that are generated in a system. Log messages are stored in a separate file called the log file, which is stored in the **/var/log** directory. The main log file is **/var/log/messages**. In addition to this log file, some services create their own log files.

Automating Log Analysis

During maintenance sessions, instead of manually parsing large log files, you can automate the log analysis by writing Perl or Bash scripts. For example, you can write a Perl script to automatically parse a mail log file and inform you about the rejected email messages. Ensure that you make a crontab entry for the script.

Automatic Rotation

Automatic rotation is a system of regular rotation of logs to maintain a minimum log file size. The **logrotate** utility is used to perform automatic rotation. When executed, **logrotate** adds a .1 to the end of the file name of the current version of the log files. Previously rotated files are suffixed with .2, .3, and so on. Older logs have larger numbers at the end of their file names. Using automatic rotation, all copies of a file, with dates from when they were created, will be stored. Log files can be rotated on a daily, weekly, or monthly basis. Automatic rotation saves disk space because older log files are pushed out when a size limit is reached. This is important because a computer that runs out of free space on the hard drive will crash. Automatic rotation prevents logs from using up all the available free space.

Centralized Logging

The Central Network Log Server

The central network log server is a server that is used to implement centralized logging services. This server receives all syslog messages from Linux or Windows® servers and from network devices such as routers, switches, firewalls, and workstations, across a network. The server logs data mining and online alerts, performs log analysis, and generates reports. Centralized logging allows computers to send their logs to a central location. This means the logs will be available even if the disk or computer where they were created crashes.

The syslogd Utility

The **syslogd** utility tracks remote and local system logs. Logs are characterized by their hostname and program field. The settings for **syslogd** are configured using the **/etc/syslog.conf** file.
The syntax of the **syslogd** utility is **syslogd [options]**.
The **syslogd** utility provides a number of options to manage specific functions. Some of the frequently used options are listed in the following table.

Option	Used To

-d	Turn on debug mode.
-f {file name}	Specify a new configuration file instead of */etc/syslog.conf.*
-m {interval}	Specify a time interval between two mark timestamp lines in the log.
-r	Enable the syslogd utility to receive messages from a network.

logger

The **logger** is the command interface to the system log module. The logger has options that allow you to customize the content that needs to be logged.
Note: The **syslogd** may not be installed by default. You can use the **sudo apt install inetutils-syslogd** to install it.

The /etc/syslog.conf File

The **/etc/syslog.conf** file controls the location where the **syslogd** information is recorded. This file consists of two columns. The first column lists the facilities and severities of the messages. The second column lists the files the messages should be logged to. By default, most messages are stored in the **/var/log/messages** file.
Some applications maintain their own log files and directories independent of the syslog.conf file. Each service has its own log storage file. Some of the frequently used log files are listed in the following table.

Log File	Description
/var/log/syslog	Stores the system log file, which contains information about the system.
/var/log/maillog	Stores mail messages.
/var/log/samba	Stores Samba messages.
/var/log/mrtg	Stores Multi Router Traffic Grapher (MRTG) messages.
/var/log/httpd	Stores Apache web server messages.

MRTG

Multi Router Traffic Grapher (MRTG) is free software, licensed under GNU General Public License (GPL), that is used to monitor and measure the traffic load on network links. The traffic load on a network is represented in graphical form.

The rsyslog Utility

The **rsyslog** utility tracks, forwards, and stores messages via the syslog protocol and local system logs, and is a more modern alternative to the older **syslogd** utility. Logs are characterized by their hostname and program field. The settings for **rsyslog** are configured using the **/etc/rsyslog.conf** file as well as multiple configuration files in the **/etc/rsyslogd** configuration directory.
The syntax of the **rsyslog** utility **is rsyslog [options]**.
The **rsyslog** utility providesa number of options to manage specific functions. Some of the frequently used options are listed in the table.

Option	Used To
-d	Turn on debug mode.
-f {file name}	Specify a new configuration file instead of the default */etc/ rsyslog.conf.*
-N {level}	Check configuration files to confirm they are correct and valid. Use a level of 1 or higher to control verbosity.

Note: The **rsyslog** may not be installed by default. You can use the **sudo apt install rsyslog** to install it.

The journalctl Utility

The **journalctl** utility is a component of **systemd** that manages and views log files created by the journal component of **systemd**. It may be used on its own but is often used in conjunction with a traditional syslog daemon such as **syslogd** or **rsyslog**. Log information is collected and stored via the **systemdjournald**service, and may be viewed with the **journalctl** utility. The settings for **journald** are configured in the **/etc/systemd/journald.conf** file.
The syntax of the **journalctl** utility is **journalctl [options]**.
The **journalctl** utility provides a number of options to manage specific functions. Some of the frequently used options are listed in the following table.

Option	Used To
-n {number of lines}	Specify the number of lines of journal logs to display.

-o {output format}	*Specify the format of the output, for example: short, verbose, or export.*
-f	Specify that syslog-ng should be run as a foreground process (do not go into the background after initialization).
-p	Filter journal log output by priority (alert, err, warning, notice, info, etc.).
-b	Show log message from the current boot only (although previous boots may also be specified).

The /var/log/journal/ directory

In its default configuration, the **Systemd** Journal only stores logs in memory, and logs are cleared on each system reboot. The **Systemd** Journal logs may be persisted after a reboot by creating the directory **/var/log/journal**. **Systemd** is configured to automatically persist logs into this directory if it exists.

Log File Analysis

The process of examining messages generated by logging daemons in log files is referred to as log file analysis. Log messages are created in a format that is specific to an application or a vendor and are arranged in chronological order. During analysis, the format of log messages from different logging sources, such as operating systems, networks, and databases, is compared with a preset format. Also, log messages are categorized for each user with respect to the application, system, or system configuration accessed, to ensure user authentication.

The lastlog Command

The **lastlog** command utilizes data from the **/var/log/lastlog** file to display the latest login details of all users. In addition to the login name, date, and time, it displays the terminal from where a user last logged in. The **lastlog** command is used by administrators to view user accounts that have never been used.

The grep Command

The **grep** command searches a file or list of files for a string and prints the lines that match the search string. The **grep** command has various options that allow you to specify search criteria.
The syntax of the **grep** command is **grep [options] {keyword} {file name}**.
The following table lists the options of the grep command.

Option	Used To
-h	Print matching lines without file names.
-w	Restrict the search to whole words only.
-c	Display a count of the number of matching lines and not the lines themselves.
-i	Ignore case while searching.
-l	List the file names that contain matching lines.
-n	Precede each line with the line number where it was found.
-s	Suppress the display of any error message.
-e	Specify one or more patterns for searching.

fgrep and egrep Commands

The **fgrep** command searches for multiple text patterns; however, this command's search is not based on regular expressions.
The **egrep** command searches for multiple text patterns, which may include a larger set of regular expression elements than grep.

The tail Command

The **tail** command is used to retrieve data from a file. By default, it displays the last 10 lines of the file.
The syntax of the **tail** command is **tail [options] {file name}**.
The tail command has various options. Some of the frequently used options are listed in the following table.

Option	Enables You To
--retry	Force the tail command to open a file that cannot be opened.
-n {total no. of lines}	Print the specified number of lines from the end of a file.

-c {total no. of bytes}	Print the specified number of bytes from the end of a file.
-f	Update the output of the tail command if any change is made to a file.

Configure Security Enhanced Linux

Security-Enhanced Linux (SELinux) is the default security enhancement feature provided with CentOS and Red Hat Enterprise Linux, and is available on other distributions. It was developed by the U.S. National Security Agency while implementing various security policies on Linux operating systems. It provides additional filesystem and network security so that unauthorized processes cannot access or tamper with data, bypass security mechanisms, violate security policies, or execute untrustworthy programs. It enforces Mandatory Access Controls (MACs) on processes and resources and allows information to be classified and protected based on its confidentiality and integrity requirements. This confines the damage caused to information by malicious applications.
Note: You can use the **sudo apt install policycoreutilsselinux-utils selinux-basics** command to install SELinux on Ubuntu.

SELinux Modes

SELinux has three different modes.

Mode	Description
Disabled	In this mode, SELinux is turned off. So, MAC will not be implemented and the default DAC method will be prevalent.
Enforcing	In this mode, all the security policies are enforced. Therefore, processes cannot violate the security policies.
Permissive	In this mode, SELinux is enabled, but the security policies are not enforced. So, processes can bypass the security policies. However, when a security violation occurs, it is logged and a warning message is sent to the user.

Security Context

Security context is the collection of all security settings pertaining to processes, files, and directories. Security context consists of three elements: user, role, and type. Based on the security context attributes, SELinux decides how subjects access objects on the system.

Note: You activate SELinux using the **sudoselinux-activate** command.

Security Policies

Types of Access Controls

Access control is a method of restricting access to system resources. Only authorized programs will be allowed to access system resources. In Linux, there are two types of access controls.

Access Control Method	Description
Discretionary Access Control (DAC)	In DAC, the system checks the resources over which a user has access rights. The rights of the user are identified using the authentication information such as user identity and password. Under DAC, there are two types of permissions: the administrator permissions and the non-administrator permissions. For application programs to run, administrator access has to be provided. Administrator access provides full discretion over the filesystem and exposes it to security threats. For example, a malicious program or process started by a user having administrator access can damage data in a filesystem. DAC is the standard security strategy in Linux in which the User/ Group/Other file permissions are managed.
Mandatory Access Control (MAC)	In MAC, the system checks the resources over which a user does not have access rights. MAC is applied through SELinux. The rights of the user are identified using authentication such as the SELinux user identity, role, and type of access. MAC is the opposite of DAC, where permissions have to be defined for all processes (known as subjects) as to how they access resources (known as objects) such as files, directories, devices, memory resources, and other processes. An action is an operation, such as append, write, read, create, execute, and rename, that a subject can perform on an object. This is implemented using security policies that control the interaction between the processes and the objects. For example, when a subject tries to access an object, the security policy is checked to verify whether the subject is authorized to access the object before granting the access.

Security Policies

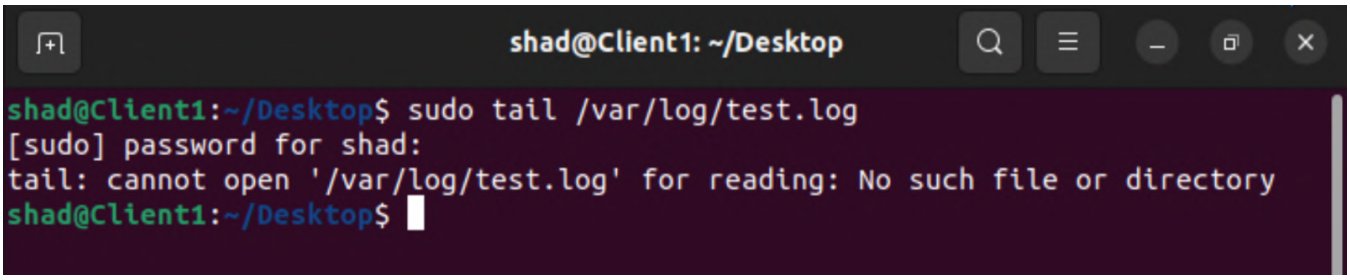
A security policy defines access parameters for every process and resource on the system. Configuration files and policy source files located in the **/etc/selinux** directory can be configured by the root user.

Security Policy Type	Description
Targeted	According to the targeted policy, except the targeted subjects and objects, all other subjects and objects will run in an unconfined environment. The untargeted subjects and objects will operate on the DAC method and the targeted ones will operate on the MAC method. A targeted policy is enabled by default.
Strict	A strict policy is the opposite of a targeted policy, where every subject and object of the system is enforced to operate on the MAC method.

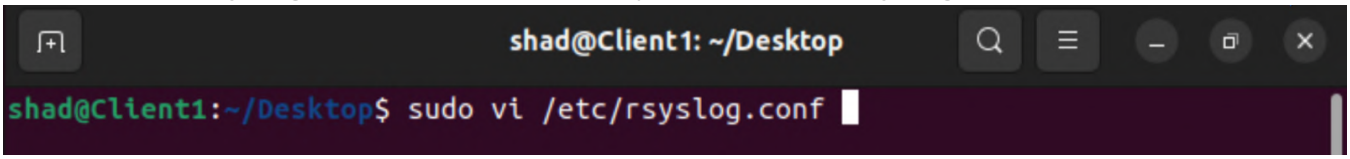
Managing System Services Lab

Configuring System Log Settings

You regularly perform routine maintenance checks on network systems. You find that the logs show a couple of errors because they were not entered properly. You want only the warnings and alerts to be shown in the logs. So, you decide to configure the settings for rsyslog. This lab walks through changing the logging settings for mailutils. Then you will install mailutils and test the logging.
Before you start this lab, launch the Client1 virtual machine. Log on with your user account and open a terminal.
TASK A
In this task you will configure the log settings.
1. To confirm that the log file you are about to configure does not already exist, type **sudo tail /var/log/test.log** and press **Enter**.

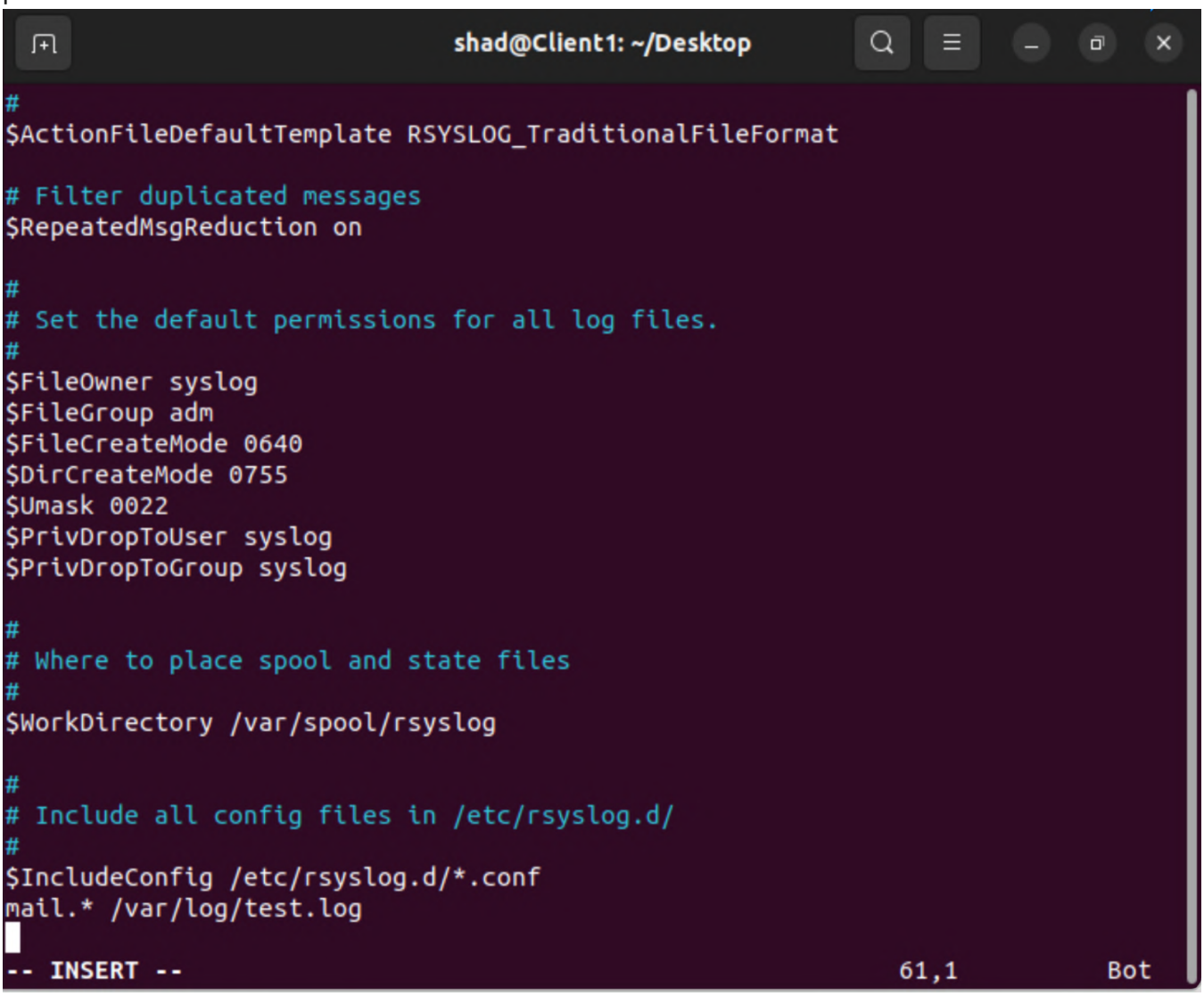


2. To open the **rsyslog.conf** file in the vi editor, type **sudovi /etc/rsyslog.conf** and press **Enter**.



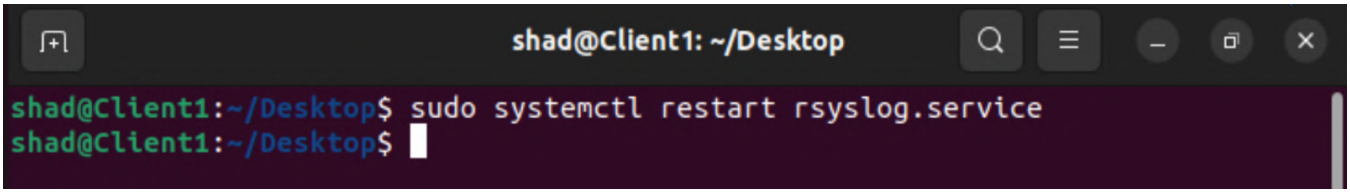
3. To move to the last line of the file, press **Shift+G**. To switch to insert mode, and then create a new blank line, type **o**.

4. To set the severity of the error to be logged for mail messages, type **mail.* /var/log/test.log** and press **Enter**.

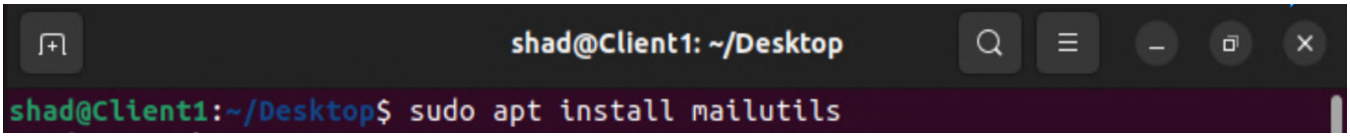


5. To save and close the file, press **Esc**. Type **:wq** and press **Enter**.

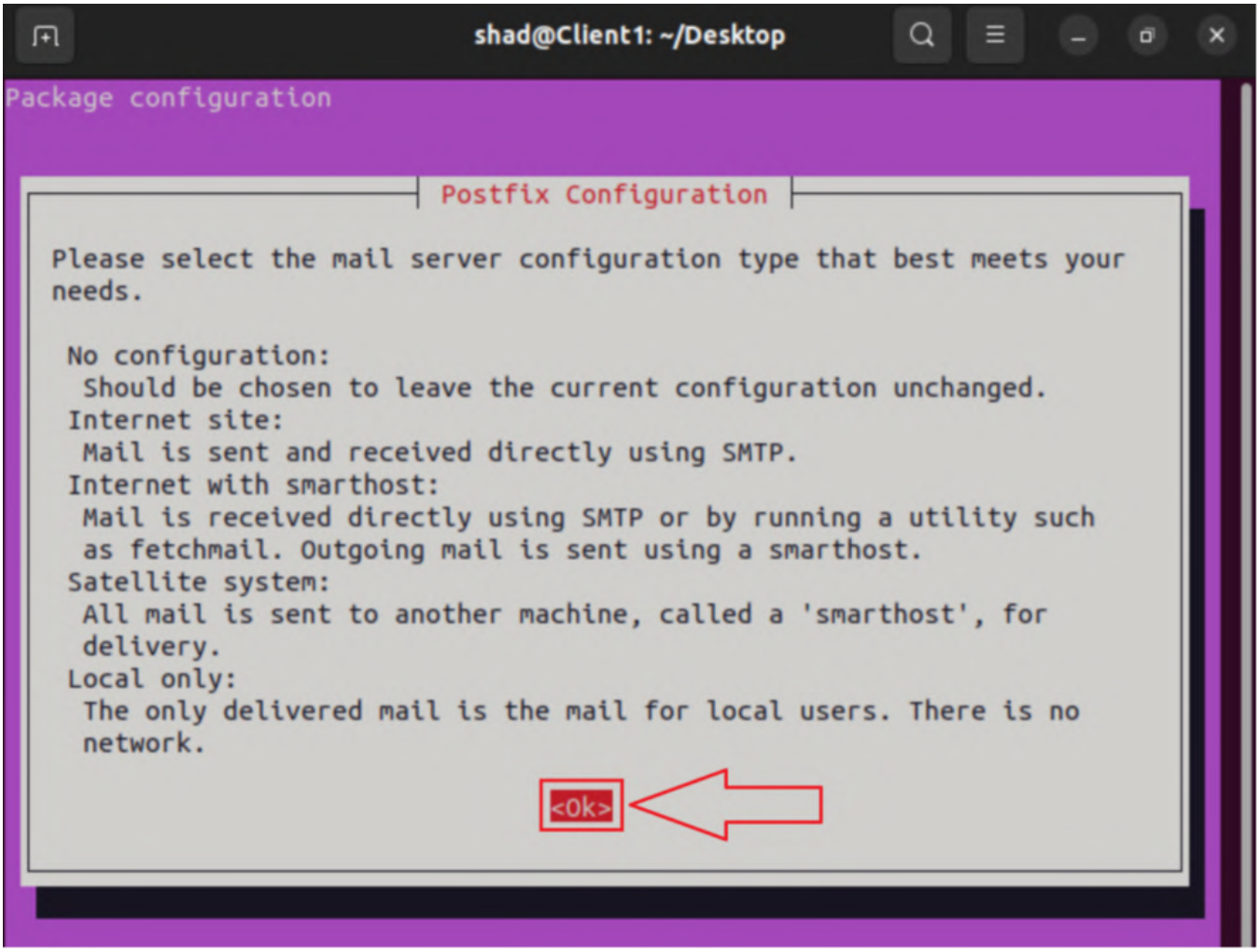
6. To restart the rsyslog service, in the terminal window, type **sudo systemctl restart rsyslog.service** and then press **Enter**.



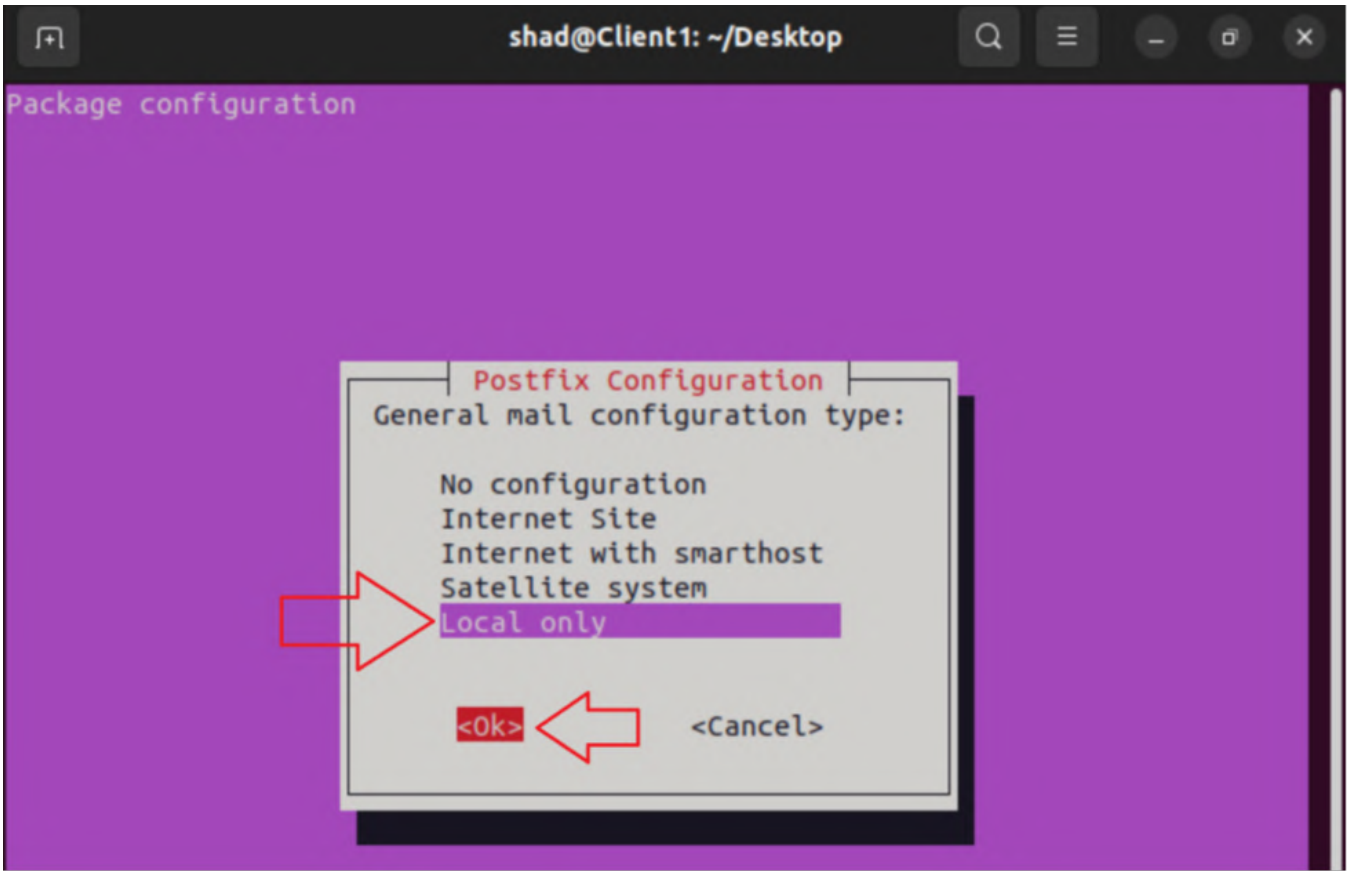
7. To install the mail service, type **sudo apt install mailutils** and then press **Enter**. When prompted to continue, type **Y** and then press **Enter**.



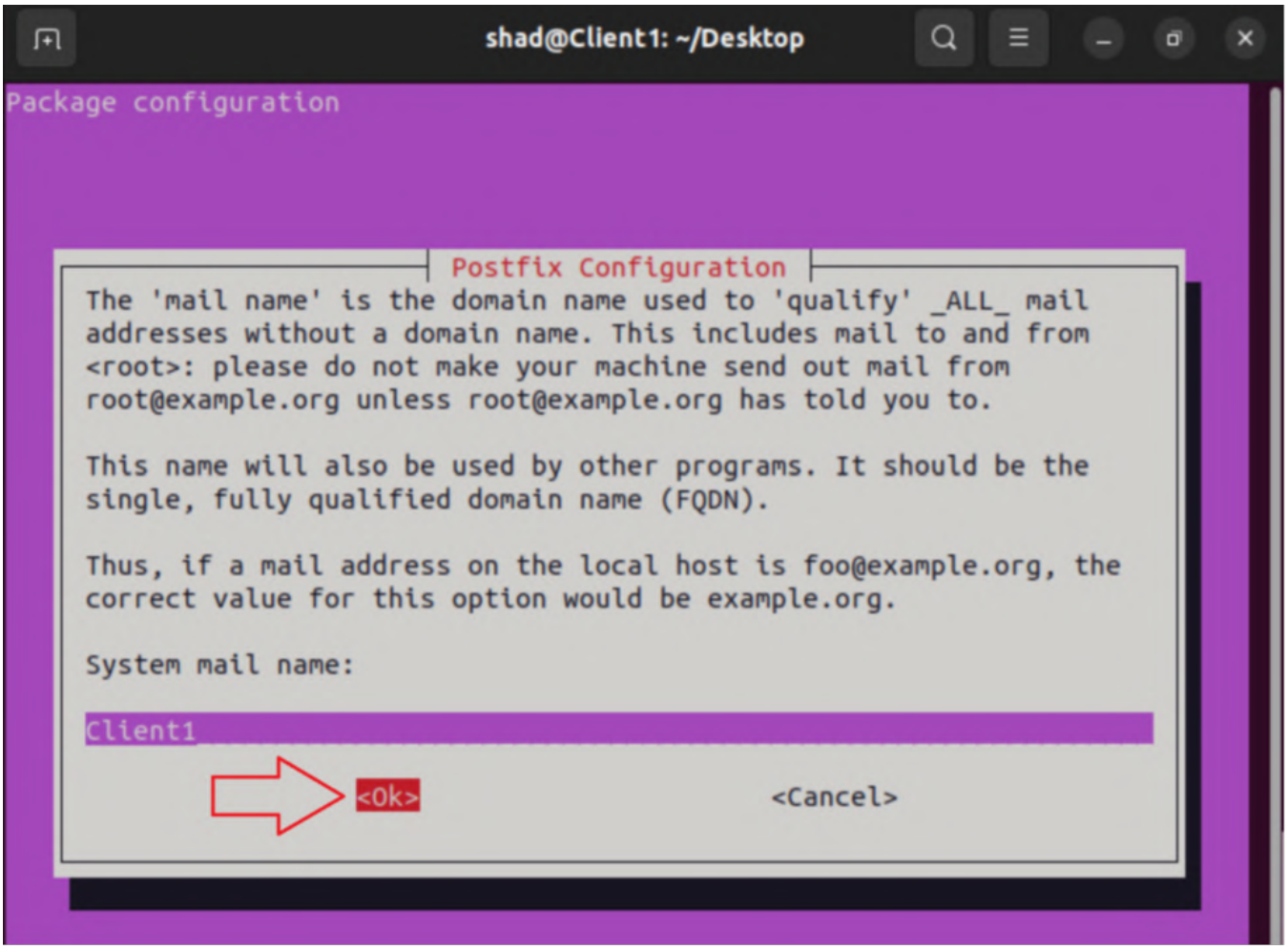
8. In the Postfix Configuration screen, select **Local only**, and then press **Enter**.
9. On the Profix Configuration screen asking you to the confirm the server name, select **OK** and then press **Enter**.



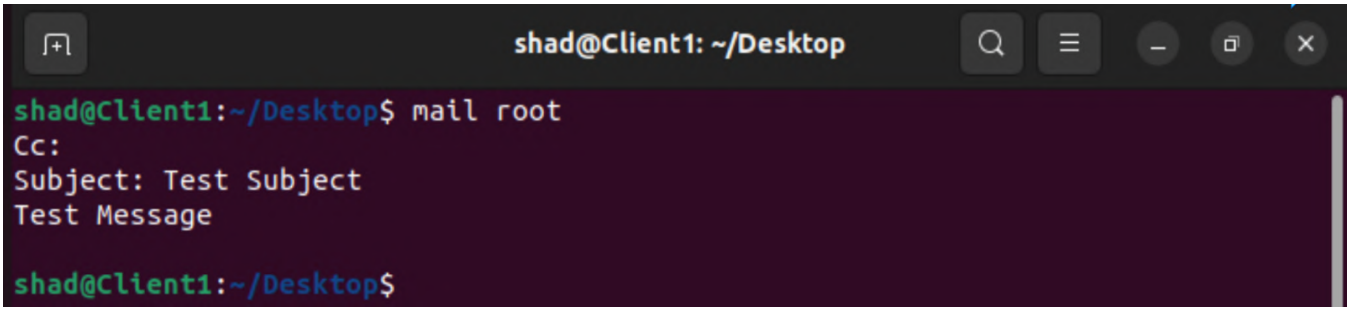
10. In the Postfix Configuration screen, select **Local only**, and then press **Tab**. When **Ok** is highlighted, press **Enter**.



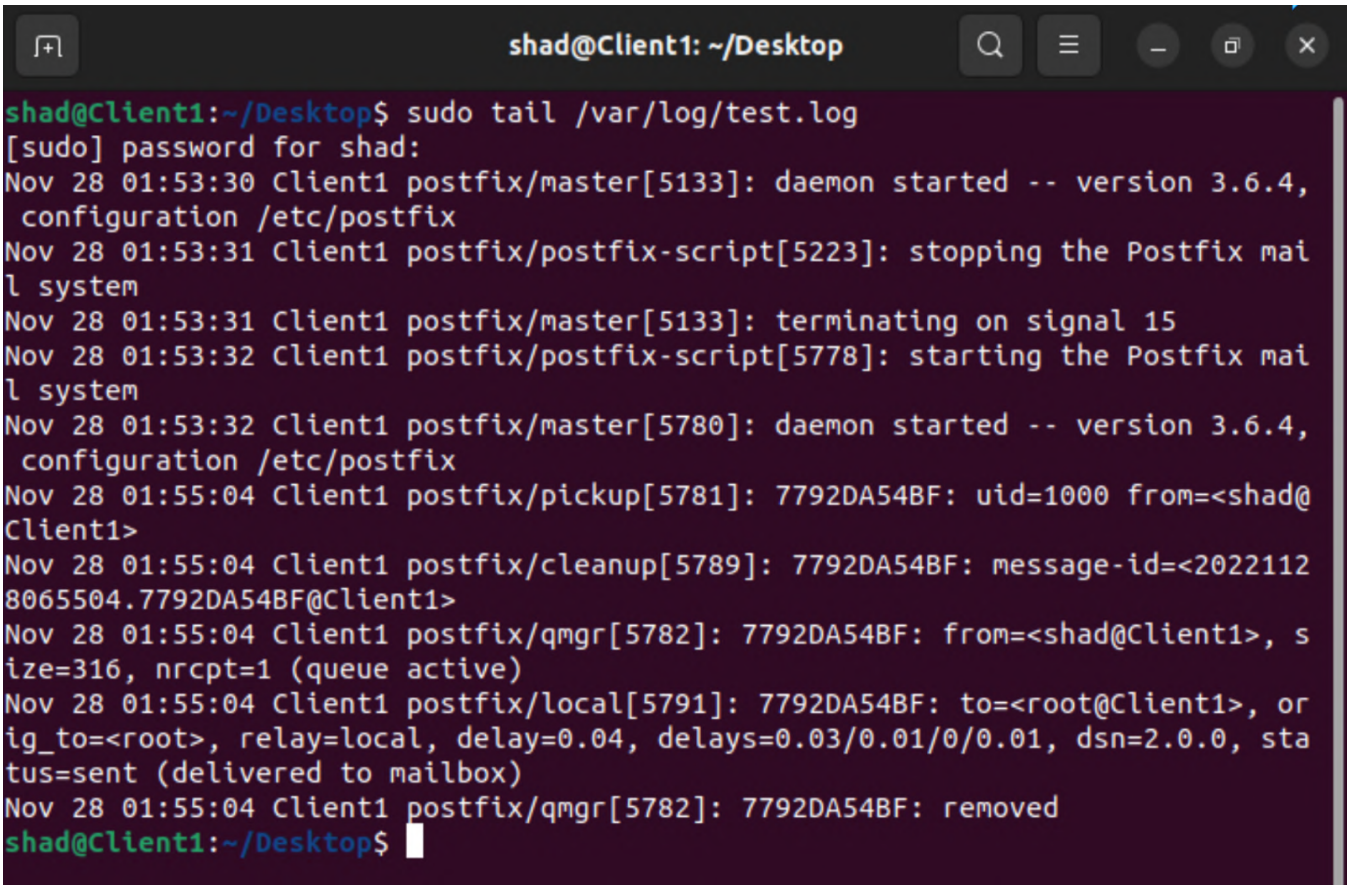
11. On the Profix Configuration screen asking you to the confirm the server name, press **Tab** to select **OK**, and then press **Enter**.



12. The installation will finish.
13. To send an email to test the logging service, type **mail root** and press **Enter**.At the Cc: prompt, press **Enter**.To provide a sample subject, type **Test Subject** and press **Enter**.To provide a sample message, enter **Test Message** on the next line.To complete and send the email, press **Ctrl+D**.



14. To confirm that the log file you have configured now contains log entries from the mail server that processed your message, type **sudo tail /var/log/test.log** and press **Enter**.


shad@Client1: ~/Desktop
shad@Client1:~/Desktop$ sudo tail /var/log/test.log
[sudo] password for shad:
Nov 28 01:53:30 Client1 postfix/master[5133]: daemon started -- version 3.6.4,
configuration /etc/postfix
Nov 28 01:53:31 Client1 postfix/postfix-script[5223]: stopping the Postfix mai
l system
Nov 28 01:53:31 Client1 postfix/master[5133]: terminating on signal 15
Nov 28 01:53:32 Client1 postfix/postfix-script[5778]: starting the Postfix mai
l system
Nov 28 01:53:32 Client1 postfix/master[5780]: daemon started -- version 3.6.4,
configuration /etc/postfix
Nov 28 01:55:04 Client1 postfix/pickup[5781]: 7792DA54BF: uid=1000 from=<shad@
Client1>
Nov 28 01:55:04 Client1 postfix/cleanup[5789]: 7792DA54BF: message-id=<2022112
8065504.7792DA54BF@Client1>
Nov 28 01:55:04 Client1 postfix/qmgr[5782]: 7792DA54BF: from=<shad@Client1>, s
ize=316, nrcpt=1 (queue active)
Nov 28 01:55:04 Client1 postfix/local[5791]: 7792DA54BF: to=<root@Client1>, or
ig_to=<root>, relay=local, delay=0.04, delays=0.03/0.01/0/0.01, dsn=2.0.0, sta
tus=sent (delivered to mailbox)
Nov 28 01:55:04 Client1 postfix/qmgr[5782]: 7792DA54BF: removed
shad@Client1:~/Desktop$

Configuring rsyslogd as a Central Log Server

In this lab you will be configuring the system log settings.

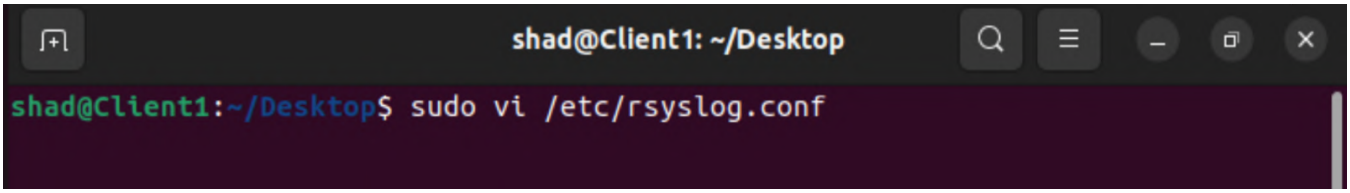
You regularly perform routine maintenance checks on network systems. You find that the logs show a couple of errors because they were not entered properly. You want only the warnings and alerts to be shown in the logs. So, you decide to configure the settings for rsyslog. This log walks through configuring the rsyslog service.

Before you start this lab, launch the Client1 virtual machine. Log on with your user account and open a terminal.

TASK A

In this task you will configure the log settings.

1. To open the **/etc/rsyslog.conf** file, type **sudo vi /etc/rsyslog.conf** and press **Enter**.

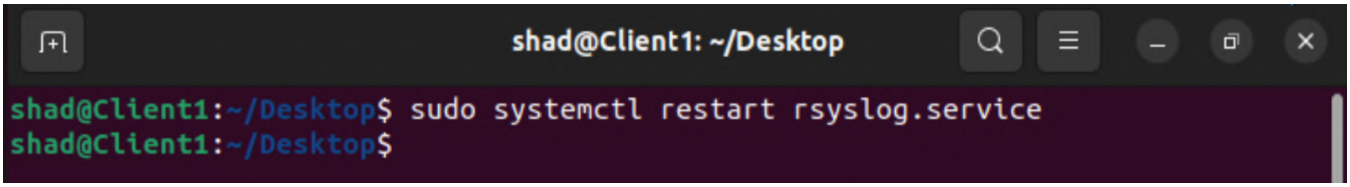


2. Navigate to the line containing **#module(load=imtcp)** and position the cursor at the beginning of the line.

3. Remove the # hash character on this line and the following line to set it as **module(load=”imtcp”)** and **input(type=”imtcp” port=”514)**.

4. To save and close the file, press **Esc**. Type **:wq** and press **Enter**.

5. To restart the rsyslogd service, type **sudo systemctl restart rsyslog.service** and press **Enter**.



6. You may close all the windows and save the state of the virtual machine. This is the end of the labs for this module.