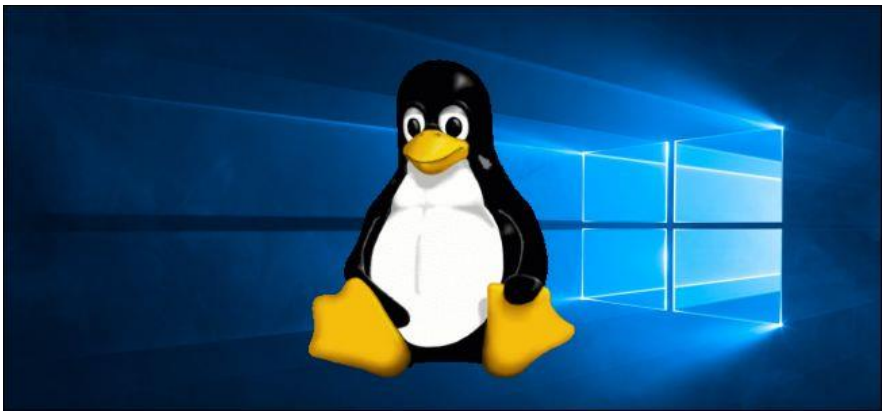
UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT



Operating Systems

Experiment 4

Introduction to VI Editor and Shell Programming –I

CLO 2. ***Use modern tools and languages.***

CLO 3. ***Demonstrate an original solution of problem under discussion.***

CLO 4. ***Work individually as well as in teams***

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT

Implementing Linux Commands

Introduction to Vi Editor

Text related commands

1) find

The find command is used to locate files in a directory.
The –name option: This lists out the specific files in directory. Wild cards can be used.

Examples:

```
$find
```

Displays all the files present in a directory

```
$find –name '*.txt'
```

Displays all the files (.txt) type present in a directory

2) history

List all commands typed so far.

3) more FILE

Display the contents of FILE, pausing after each screen.
There are several keys which control the output once a screen has been printed.
<enter> Will advance the output one line at a time.
<space bar> Will advance the output by another full screen.
"q" Will quit and return you to the Linux prompt.

Examples:

```
more +3 myfile.txt
```

Display the contents of file **myfile.txt**, beginning at line 3.

```
more +/"hope" myfile.txt
```

Display the contents of file **myfile.txt**, beginning at the first line containing the string "**hope**".

4) less FILE

"less" is a program like "more", but which allows backward movement in the file as well as forward movement.

Absolute path and relative path

Generally, if a command is given it will affect only the current working directory. For example the following command will create a directory named curr in the current working directory.

```
$ mkdir curr
```

The directory can also be created elsewhere in the file system using the absolute and relative path. If the path is given with respect to the root directory, then it is called **full path or absolute path.**

```
$ mkdir /home/it2006/it2k601/newdir
```

The full path always starts with the **/,** which represents the **root** directory.

If the path is given with respect to the current working directory or parent directory then it is called **relative path**.

```
$ mkdir ../newdir
```

The above command will create a directory named newdir in the **parent directory**.

```
$ mkdir  ./newdir
```

What about the above command?

```
$ mkdir ./first/newdir
```

The above command will create a directory named "newdir" inside "first directory", where the directory first is in the current working directory.

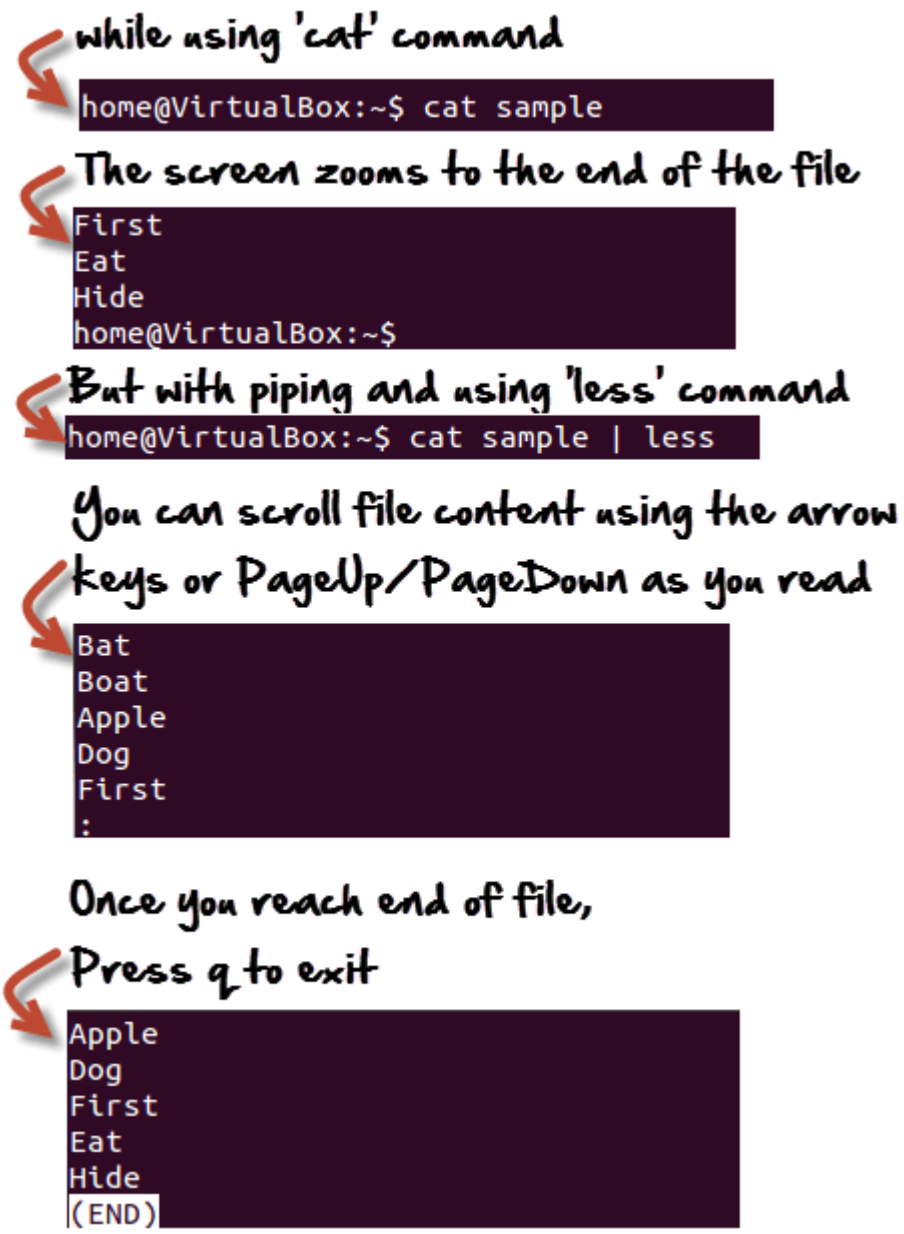
Note **"."** Represents current directory and **".."** represents parent directory.

PIPES AND FILTERS

In Linux, commands were created to perform single tasks only. If we want to perform multiple tasks we can go for pipes and filters.

PIPES

The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next. In short, the output of each process directly as input to the next one like a pipeline. The symbol **'|'** denotes a pipe.



Pipes help you mash-up two or more commands at the same time and run them consecutively.

The 'grep' command

It will scan the document for the desired information and present the result in a format you want.

Syntax:

```
grep search_string
```

The contents of the 'sample' file

```
home@VirtualBox:~$ cat sample
Bat
Goat
Apple
Dog
First
Eat
Hide
```

Using 'grep' for searching Apple

```
home@VirtualBox:~$ cat sample | grep Apple
Apple
```

Using 'grep' for searching Eat

```
home@VirtualBox:~$ cat sample | grep Eat
Eat
```

Following options can be used with this command.

Option	Function
-v	Shows all the lines that do not match the searched string
-c	Displays only the count of matching lines
-n	Shows the matching line and its number
-i	Match both (upper and lower) case
-l	Shows just the name of the file with the string

Let us try the first option '**-i**' on the same file use above -

Using the 'i' option grep has filtered the string 'a' (case-insensitive) from the all the lines.

```
home@VirtualBox:~$ cat sample | grep -i a
Bat
Goat
Apple
Eat
```

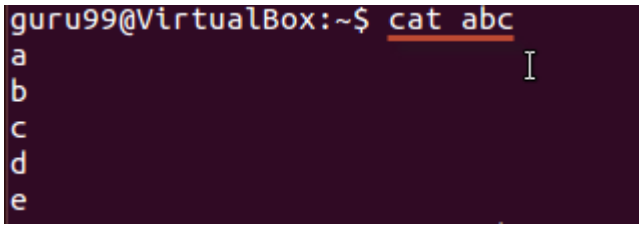
The 'sort' command

This command helps in **sorting out the contents of a file alphabetically.**

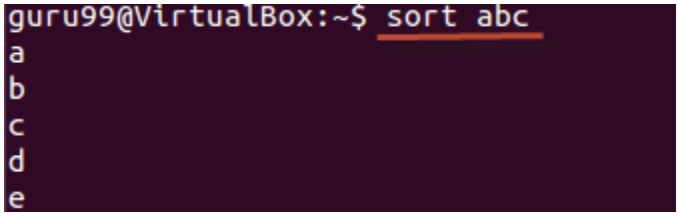
The syntax for this command is:

```
sort Filename
```

Consider the contents of a file.



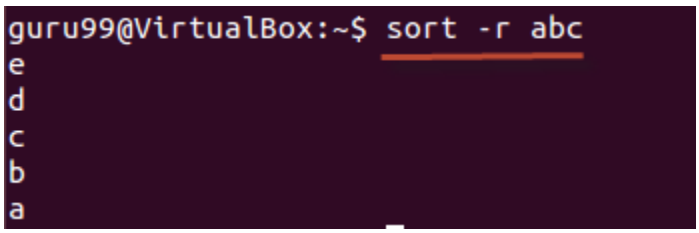
Using the sort command



There are **extensions** to this command as well, and they are listed below.

Option	Function
-r	Reverses sorting
-n	Sorts numerically
-f	Case insensitive sorting

The example below shows reverse sorting of the contents in file 'abc'.



UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT

FILTERS

Filters are used to extract the lines, which contain a specific pattern, to arrange the contents of a file in a sorted order, to replace existing characters with some other characters, etc. Linux has a lot of filter commands like awk, grep, sed, spell, and wc. A filter takes input from one command, does some processing, and gives output.

When you pipe two commands, the "filtered " output of the first command is given to the next.

1. Sort filter
The sort filter arranges the input taken from the standard input in alphabetical order.
The sort command when used with "-r" option will display the input taken from the keyboard in the reverse alphabetical order.
When used with "-n" option arranges the numbers, alphabets and special characters per their ASCII value.

2. Grep filter
This command is used to search for a particular pattern from a file or from standard input and display those lines on the standard output. Grep stands for "Global search for regular expression".

```
grep [option(s)] pattern [file(s)]
grep 'Zain' pstudent.txt
```

Where the given pattern is searched inside the lines contained in the named files. The objects listed inside the square brackets are optional. When the options and input file are not specified, then the grep command searches for standard input (which is by default the standard text typed on the keyboard) and each line that happens to constitute the given pattern is displayed.

3. Pg and more filter
These commands display the output of the command on the screen page by page. The difference between pg and more filter is that the viewing screen of the latter can be done by pressing space bar while that of the former is done by pressing enter.

What is the VI editor?

The VI editor is the most popular and classic text editor in the Linux family. Below, are some reasons which make it a widely used editor –

1) It is available in almost all Linux Distributions 2) It works the same across different platforms and Distributions 3) It is user-friendly. Hence, millions of Linux users love it and use it for their editing needs

Nowadays, there are advanced versions of the vi editor available, and the most popular one is **VIM** which is **V**i **Im**proved.

To work on VI editor, you need to understand **its operation modes**. They can be divided into two main parts.

Command mode:

- The vi editor opens in this mode, and it only **understands commands**
- In this mode, you can, **move the cursor and cut, copy, paste the text**
- This mode also saves the changes you have made to the file
- **Commands are case sensitive.** You should use the right letter case.

Insert mode:

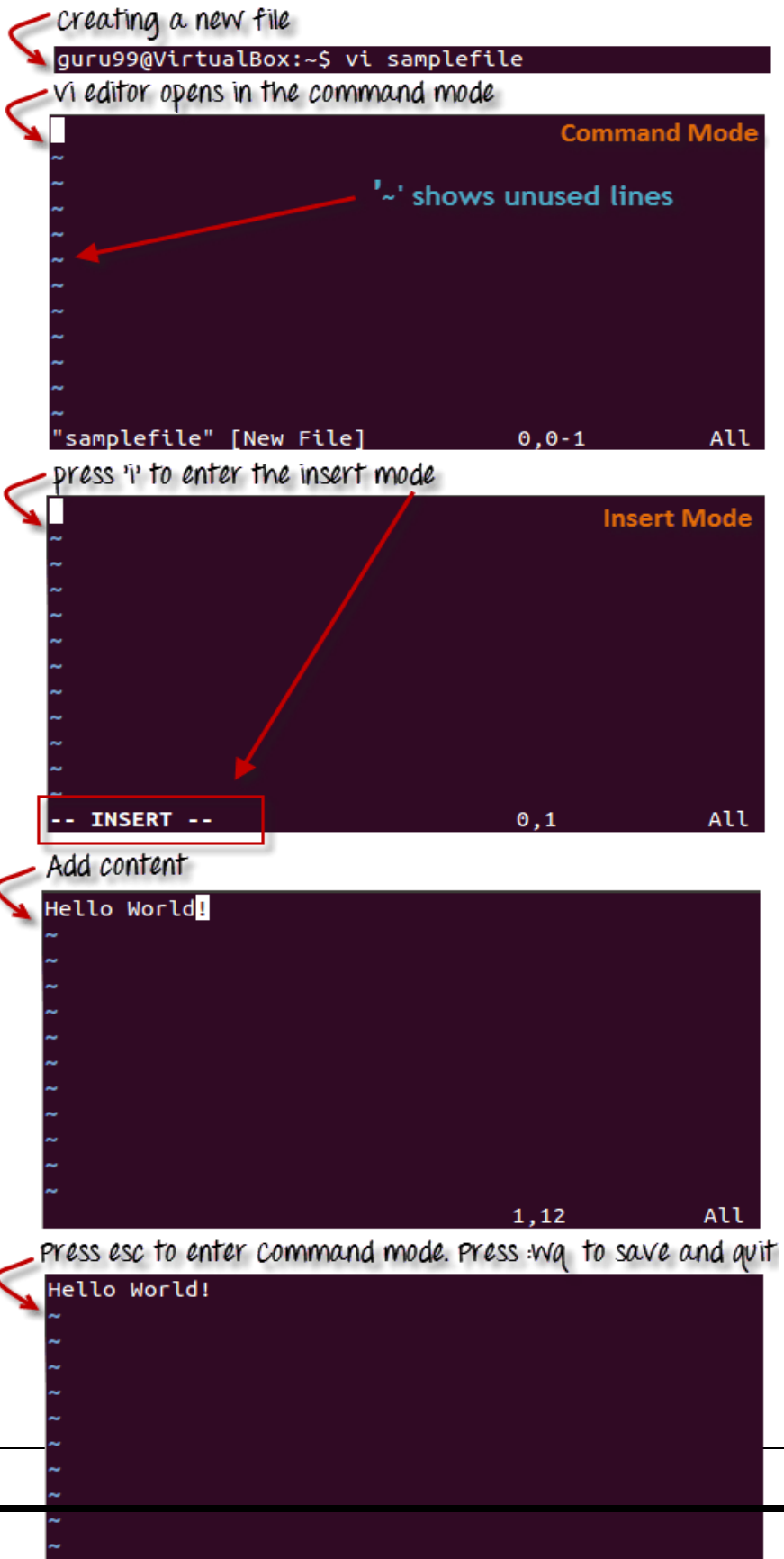
- This mode is for inserting text in the file.
- You can switch to the Insert mode from the command mode **by pressing 'i' on the keyboard**
- Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.
- To return to the command mode and save the changes you have made you need to press the Esc key

Starting the vi editor

To launch the VI Editor -Open the Terminal (CLI) and type

```
vi <filename_NEW> or <filename_EXISTING>
```

And if you specify an existing file, then the editor would open it for you to edit. Else, you can create a new file.


creating a new file
guru99@VirtualBox:~$ vi samplefile
Vi editor opens in the command mode
Command Mode
'~' shows unused lines
"samplefile" [New File] 0,0-1 All
press 'i' to enter the insert mode
Insert Mode
-- INSERT -- 0,1 All
Add content
Hello World!
1,12 All
Press esc to enter Command mode. Press :wq to save and quit
Hello World!

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT

VI Editing commands

- i - Insert at cursor (goes into insert mode)
- a - Write after cursor (goes into insert mode)
- A - Write at the end of line (goes into insert mode)
- ESC - Terminate insert mode
- u - Undo last change
- U - Undo all changes to the entire line
- o - Open a new line (goes into insert mode)
- dd - Delete line
- 3dd - Delete 3 lines.
- D - Delete contents of line after the cursor
- C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
- dw - Delete word
- 4dw - Delete 4 words
- cw - Change word
- x - Delete character at the cursor
- r - Replace character
- R - Overwrite characters from cursor onward
- s - Substitute one character under cursor continue to insert
- S - Substitute entire line and begin to insert at the beginning of the line
- ~ - Change case of individual character

Note: You should be in the "**command mode" to execute these commands**. VI editor is **case-sensitive** so make sure you type the commands in the right letter-case.

Make sure you press the right command otherwise you will end up making undesirable changes to the file. You can also enter the insert mode by pressing a, A, o, as required.

Moving within a file

- k - Move cursor up
- j - Move cursor down
- h - Move cursor left
- l - Move cursor right

You need to be in the command mode to move within a file. The default keys for navigation are mentioned below else; You can **also use the arrow keys on the keyboard**.

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT

Saving and Closing the file

- **Shift+zz** - Save the file and quit
- **:w** - Save the file but keep it open
- **:q** - Quit without saving
- **:wq** - Save the file and quit

You should be in the **command mode to exit the editor and save changes** to the file.

Running Commands:

The vi has the capability to run commands from within the editor. To run a command, you only need to go to the command mode and type **:!** command.
For example, if you want to check whether a file exists before you try to save your file with that filename, you can type **:! ls** and you will see the output of **ls** on the screen.
You can press any key (or the command's escape sequence) to return to your vi session.

Touch Command:

The touch command is the easiest way to create new, empty files.

```
Touch <file name>
```

chmod command.

When we create a file, **by default it is created with read and write permission turned on** and **execute permission turned off**. A file can be made executable using chmod.

Make the file executable
$ chmod u+x ***script_file***

or $chmod u+x /home/user/show.sh

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT

Lab Tasks

Task 1:

Get into the vi editor and type some text then save the file with the name my_first_linux_file.

Task 2:

Open the given file in VI editor. Change the subject digital logic design into "Object oriented Programming". Also change programming fundamentals into Operating System. Also remove the lines "It is a bore subject. It is not helpful for us." from line 3.

```
we are learning digital logic design            in this semester
we are also learning programming fundamentals in this semester
we are also learning oop in this semester. It is a bore subject.It is not helful for us.
```

Task 3:

Make a Shell Script that make use of pipe command.
Firstly, list the word counts of present working directory.
Next make use of pipe command for the word count of already created file after sorting in single command.

Task 4:

Make a shell script that highlights only the lines in a text file that do not contain the character 'a', but the result should be in reverse order.