National University
of computer and emerging sciences

CL 2006 OPERATING SYSTEM

BS Software Engineering

Fall-2024

Course Instructors: Syed Daniyal Hussain Shah, Hifza Umar

Course Coordinator: Syed Daniyal Hussain Shah

National University
of computer and emerging sciences

Lab Policy and Rules:

1. **100% attendance is mandatory.** In case of an emergency, you can avail yourself of up to 3 absences. So, don't waste your absences , save them for emergencies. If you are debarred from the course due to low attendance, do not come to me to correct it.

2**. Disturbing the class environment during lab sessions.** such as by talking, using mobile phones, eating, etc. will result in penalties (e.g., deduction of marks).

3. **Lab tasks will not be accepted if you are absent.** If you miss a lab, you cannot submit that lab task.

4**. Lab demos for each lab task will be conducted at the end of each session.** If you miss the demo for any reason, no retake will be allowed, and that lab will be marked as 0.

5. **All quizzes will be unannounced**. Be prepared for surprise quizzes. Quizzes will cover content from previous labs as well as theory lectures from the corresponding theory course.

6. **You can take help from the internet for lab tasks,** but simply copying and pasting without understanding will be marked as 0. You should be able to explain the syntax or material used in your tasks.

7**. Do not ask for personal favors.** If you have concerns, such as short attendance, please speak with the relevant authority (e.g., academics).

8**. Students on warning:** Now is the time to study and earn a good grade. Do not ask for extra marks at the end if you are unable to clear your warning.

National University of computer and emerging sciences

Lab 1: Introduction to shell and some basic linux commands

1. Shell:

The shell is a command-line interpreter that provides a user interface for interacting with the operating system. It allows users to execute commands, run scripts, and manage files and processes. The shell interprets the commands you type and translates them into actions by the operating system.

2. Terminal:

The terminal is a software application that provides a text-based interface to interact with the shell. It's where you type your commands and see the output. Essentially, the terminal acts as a container for the shell, allowing you to interact with it.

3. Organization of linux file system:

- ### Root directory:

  The root directory in Linux, represented by a single forward slash /, is the top-level directory in the file system hierarchy. All other files and directories are nested within this directory, forming a tree-like structure.

- ### Directory:

  A folder which holds other files and folders.

- ### Subdirectory:

  A directory residing within another directory.

- ### Current working directory:

  The directly currently being used.

- ### Home directory:

  The directory selected by the linux as the working directory when a user logs in. Its name usually matches with your login name.

  #### Example

  /home/ahmed (path of ahmed home directory).

- ### Path/Pathname:

  A path or pathname is the specific location of a file or directory in the file system. It tells the system where to find or place a file or directory. There are two types of paths: absolute paths and relative paths.

  Example "C:\Users\HP\Desktop\BootCamp"

a. Absolute pathname:

An absolute path specifies a file or directory's location in the file system starting from the root directory (/). It provides the full path from the root to the file or directory, regardless of the current working directory.

Example:

/home/user/documents/report.txt

b. Relative pathname:

A relative path specifies a file or directory's location relative to the current working directory. It does not start from the root directory but rather from wherever you are currently located in the file system.

Example:

If your current working directory is /home/user:

documents/report.txt refers to the report.txt file inside the documents directory within the current directory (/home/user).

../config/settings.conf would refer to the settings.conf file located in the config directory, which is one level up from the current directory.

c. Special Symbols in Paths:

(/): Root directory or directory separator.

(..) : Refers to the parent directory (one level up).

4. cd command:

The cd (change directory) command in Linux is used to navigate the file system by changing the current working directory to a different directory. This command is one of the most basic and frequently used in the terminal.

Examples:

To navigate to a directory named documents:

```
cd documents
```

To move to a directory using an absolute path:

```
cd /home/user/documents
```

Change to a directory using a relative path:

```
cd ../documents
```

To move to the parent directory of your current location:

```
cd ..
```

Simply typing cd without any arguments will take you to your home directory:

```
cd
```

You can also use the tilde (~) symbol:

```
cd ~
```

To navigate directly to the root directory:

```
cd /
```

To switch back to the previous directory you were in:

```
cd -
```

5. pwd command:

The pwd (print working directory) command in Linux is used to display the absolute path of the current working directory. It shows you where you are in the file system at any given moment. pwd always shows the absolute path, starting from the root (/).

Example:

```
pwd
```

Output:

```
/home/user/documents
```

6. mkdir command:

The mkdir (make directory) command in Linux is used to create one or more new directories. It allows you to specify the name and location of the directories you want to create.

Examples:

To create a directory named new_folder in the current directory.

```
mkdir new_folder
```

To create several directories at once:

```
mkdir dir1 dir2 dir3
```

To create a directory within another directory (if the parent directory exists):

```
mkdir /home/user/new_folder
```

If you want to create a directory and any necessary parent directories that don't already exist, use the -p option:

```
mkdir -p /home/user/projects/project1
```

7. rmdir:

The rmdir (remove directory) command is specifically used to delete empty directories. It will only remove a directory if it is empty; if the directory contains any files or subdirectories, rmdir will not remove it and will return an error.

Examples:

Remove an empty directory:

```
rmdir empty_folder
```

Remove multiple empty directories:

```
rmdir dir1 dir2
```

8. rm:

The rm (remove) command is used to delete files and directories. Unlike rmdir, rm can delete both files and directories, and it does not require directories to be empty. The rm command is more powerful and versatile, but it also requires careful use because it can permanently delete files and directories.

Examples:

Remove a directory and its contents:

```
rm -r folder_name
```

Remove an empty directory:

```
rm -d empty_folder
```

9. mv:

The mv command in Linux is used for moving and renaming files and directories. It's a versatile command that can relocate files from one directory to another or change their names within the same directory.

Examples:

Rename a File:

```
mv old_filename.txt new_filename.txt
```

Rename a Directory:

```
mv old_directory_name new_directory_name
```

Move a File to a Different Directory:

```
mv filename.txt /path/to/destination/
```

Move Multiple Files to a Directory:

```
mv file1.txt file2.txt /path/to/destination/
```

Move a Directory to Another Location:

```
mv directory_name /path/to/destination/
```

Prevent Overwriting Existing Files:

If destination_file.txt exists, it will not be overwritten.

```
mv -n source_file.txt destination_file.txt
```

You can use wildcards (*) to move multiple files that match a certain pattern.

```
mv *.txt /home/user/text_files/
```

10. cp:

The cp command in Linux is used to copy files and directories from one location to another. It's a fundamental command for file management, allowing you to duplicate files or entire directory structures.

Examples:

Copy a Single File:

```
cp document.txt backup.txt
```

Copy a File to Another Directory:

```
cp photo.jpg /home/user/pictures/
```

Copy Multiple Files:

```
cp file1.txt file2.txt /home/user/backup/
```

Copy a Directory and Its Contents:

The -r (recursive) option is necessary to copy directories. This command copies the entire source directory and its contents to the destination.

```
cp -r project /home/user/projects_backup/
```

Do not overwrite an existing file.

```
cp -n source_file.txt destination_file.txt
```

11. ls:

The ls command in Linux is used to list the contents of a directory. It displays files, directories, and other types of items within the specified directory or the current directory if no directory is specified.

Examples:

Running `ls` without any options simply lists the names of files and directories in the current directory.

```
ls
```

Displays detailed information about each file or directory, including permissions, number of links, owner, group, size, and modification date.

```
ls -l
```

Lists all files, including hidden files (those starting with a dot (.).

```
ls -a
```

Lists all files and directories recursively, showing the contents of all subdirectories.

```
ls -R
```

The -t option sorts the files and directories by their last modification time. The most recently modified items are listed first.

```
ls -t
```

The -s option sorts files and directories by their size, displaying the largest files first.

```
ls -S
```

The -r option reverses the order of the listing.

```
ls -r
```

Single Commands

- **To clear the screen**
  Syntax: $ clear
- **To view commands history**
  Syntax: $ history
- **To display first 10 lines of a file**
  Syntax: $ head file.txt
- **To display first 6 lines of a file**
  Syntax: $ head —n 6 file.txt
- **To display first 5 lines from 2 files**
  Syntax: $ head —n 5 title .txt file2.txt
- **To display last 10 lines of a file**
  Syntax: $ tail file.txt
- **To display last 6 lines of a file**
  Syntax: $ tail —n 6 file.txt
- **The wc command without passing any parameter will display a basic result Of "file_txt' file. Result Of wc command will be (number of lines), (number or words) and (number of chars/bytes) of the file.**

Syntax: $ wc file.txt

- **To display the number or characters in a file**
  Syntax: $ wc —c file.txt
- **To display the number of lines**
  Syntax: $ wc—I file.txt
- **To display the number of words**
  Syntax: $ wc —w file.txt
- **TO display number Of lines with numbers**
  Syntax: $ nl file.txt
- **To sort the content of file**
  Syntax: $ sort file.txt
- **To sort the content Of file in reverse orderz**
  Syntax: $ sort -r file-txt
- **To display the calendar.**
  Syntax: $ cal
- **To display system date.**
  Syntax: $ date
- **To display the login user details**
- Syntax: $ whoami
- **To display the contents of a file, line by line and page by page. we can use less command.**
  Syntax: $ less file.txt
- Note: to move up or down line by line use Arrow keys. And to move up page by page use 'b' key and for move down use Space key. To quit form the less command use ‘q' key.
- **To copy the file to another file**
  $ cp file1 -txt file2.txt
- **To copy a file within the other directory**
  $ cp file2.txt "Documents/directory1
- **To move a file to another directory**
  $ rmv file1.txt directory2
  $ mv —Documents/file1.txt directory3
- **To rename a file:**

- $ mv file1.txt file2.txt
- $ mv —Documents/file1.txt —/Doeuments/file2.txt
- **To delete a file**
  $ rm file1.txt
- $ rm Dir1/file2.txt

Task:

1. Create a new directory named `new_dir`.
2. Create a nested directory structure `parent/child/grandchild`.
3. Create multiple directories `dir1`, `dir2`, and `dir3` in one command.
4. Remove an empty directory named `new_dir`.
5. Try to remove a directory named `non_empty_dir`(which is parent dir).
6. Remove a file named `file1.txt`.
7. Remove a directory named `dir1` and all of its contents.
8. Remove an empty directory named `dir2`.
9. List all files in the current directory, including hidden files (those starting with a dot).
10. List files sorted by modification time, with the most recently modified files first.
11. List files sorted by size, with the largest files first.
12. Copy a file named `file1.txt` to a new file named `file2.txt`.
13. Copy a directory named `source_dir` and its contents to a new directory named `destination_dir`.
14. Copy a file named `file1.txt` to `file2.txt`, but only if `file2.txt` does not already exist.
15. Copy only the contents of `source_dir` to `destination_dir`, without copying the `source_dir` itself.
16. Move a directory named `dir1` to a new location or rename it to `dir2`.
17. Change to an absolute path directory, e.g., `/home/user/Documents`.
18. Change to the root directory of the filesystem.
19. Change to your home directory using its absolute path.
20. If there's a directory named `My Projects` under `/home/user`, navigate to it.
21. From your current directory, navigate to a subdirectory named `docs`.
22. Move up one directory level from your current location.
23. Move up two directory levels from your current location.
24. If you are in `/home/user/Projects/2024`, navigate up to `Projects` and then to a subdirectory `old_projects`.
25. From any directory, change to your home directory without using the name of the home directory.

Single Commands

- To clear the screen
  - Syntax: `$ clear`

- To view command history
  - Syntax: `$ history`

- To display the first 10 lines of a file
  - Syntax: `$ head file.txt`

- To display the first 6 lines of a file
  - Syntax: `$ head -n 6 file.txt`

- To display the first 5 lines from 2 files
  - Syntax: `$ head -n 5 file1.txt file2.txt`

- To display the last 10 lines of a file
  - Syntax: `$ tail file.txt`

- To display the last 6 lines of a file
  - Syntax: `$ tail -n 6 file.txt`

- The `wc` command without parameters will display the number of lines, words, and characters/bytes of the file.
  - Syntax: `$ wc file.txt`

- To display the number of characters in a file
  - Syntax: `$ wc -c file.txt`

- To display the number of lines
  - Syntax: `$ wc -l file.txt`

- To display the number of words
  - Syntax: `$ wc -w file.txt`

- To display number of lines with line numbers
  - Syntax: `$ nl file.txt`

- To sort the content of a file
  - Syntax: `$ sort file.txt`

- To sort the content of a file in reverse order
  - Syntax: `$ sort -r file.txt`

- To display the calendar
  - Syntax: `$ cal`

- To display the system date
  - Syntax: `$ date`

- To display the login user details
  - Syntax: `$ whoami`

- To display the contents of a file, line by line and page by page (using the `less` command)
  - Syntax: `$ less file.txt`
  - Note: Use Arrow keys to move up or down line by line. Use 'b' to move up page by page, and Space to move down. Quit using 'q'.

- To copy a file to another file
  - Syntax: `$ cp file1.txt file2.txt`

- To copy a file to another directory
  - Syntax: `$ cp file2.txt Documents/directory1`

- To move a file to another directory
  - Syntax: `$ mv file1.txt directory2`
  - Syntax: `$ mv Documents/file1.txt directory3`

- To rename a file
  - Syntax: `$ mv file1.txt file2.txt`
  - Syntax: `$ mv Documents/file1.txt Documents/file2.txt`
- To delete a file
  - Syntax: `$ rm file1.txt`
  - Syntax: `$ rm Dir1/file2.txt`

Task Submission Guidelines:

1. Make a word document and paste all solutions there and save it as pdf or odt.
2. Include your name and roll no. at the front page.
3. Files other than pdf or odt will not be accepted and will be marked as 0.