Final Assignment

Question 1

awk

Descripton: Awk is a powerful text processing tool that can be used for pattern scanning and processing.

Formula/Syntax: awk 'pattern { action }' file

Examples:

```
awk -F',' '{print $2}' data.csv
```

```
awk '/error/ {print}' logfile.txt
```

```
awk '{sum+=$3} END {print sum}' data.txt
```

cat

Descripton: Concatenate and display the content of files.

Formula/Syntax: cat [file1] [file2] ...

Examples:

```
cat file.txt
```

```
cat file1.txt file2.txt
```

```
cat -n file.txt
```

cp

Descripton: Copy files or directories.

Formula/Syntax: cp [options] source destination

Examples:

```
cp file.txt /path/to/destination/
```

```
cp -r source_directory/ destination_directory/
```

```
cp *.txt /path/to/destination/
```

cut

Descripton: Remove sections from each line of a file.

Formula/Syntax: cut [options] filename

Examples:

`cut -c 1-3 filename.txt`

`cut -f 2,4 -d$'\t' data.tsv`

`cut -c 5-10 file.txt`

grep

Descripton: Search for patterns in a file.

Formula/Syntax: grep [options] pattern [file]

Examples:

`grep 'example' file.txt`

`grep -r 'pattern' /path/to/directory/`

`grep -v 'error' logfile.txt`

head

Descripton: Display the first part of a file.

Formula/Syntax: head [options] [filename]

Examples:

`head file.txt`

`head -n 5 file1.txt file2.txt`

`head -n 20 file.txt | tail -n +20`

ls

Descripton: List directory contents.

Formula/Syntax: ls [options] [file/directory]

Examples:

`ls`

`ls -la`

`ls *.txt`

man

Descripton: Display the manual for a command.

Formula/Syntax: man [command]

Examples:

`man ls`

`man grep`

`man awk`

mkdir

Descripton: Create a new directory.

Formula/Syntax: mkdir [options] directory_name

Examples:

`mkdir new_directory`

`mkdir -p path/to/new_directory`

`mkdir dir1 dir2 dir3`

mv

Descripton: Move or rename files or directories.

Formula/Syntax: mv [options] source destination

Examples:

`mv old_filename.txt new_filename.txt`

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

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

tac

Descripton: Concatenate and display the content of files in reverse.

Formula/Syntax: tac [file1] [file2] ...

Examples:

`tac file.txt`

`tac file1.txt file2.txt`

`tac -n file.txt`

tail

Descripton: Display the last part of a file.

Formula/Syntax: tail [options] [filename]

Examples:

`tail file.txt`

`tail -n 5 file1.txt file2.txt`

`tail -n +20 file.txt`

touch

Description: Create an empty file or update the access and modification timestamps of a file.

Formula/Syntax: touch [options] filename

Examples:

`touch newfile.txt`

`touch -c existingfile.txt`

`touch -t 202301011200.00 newfile.txt`

Question 2

How to work with multiple terminals open?

Either by manually opening terminals by right clicking also you can use `ctrl + alt + T`. The alternative is using a split screen terminal emulator like **Tilux**

How to work with manual pages?

To access manual pages in the terminal, use the `man` command followed by the name of the command or topic you want information about, like `man ls`. Navigate within the manual using arrow keys, and exit by pressing `q`.

How to parse (search) for specific words in the manual page

To search for specific words in a manual page, use the `grep` and `man` commands. Example: `man ls | grep "option"` shows the manual page for ls and searches for the word "option."

How to redirect output (> and |)

Use `>` to redirect the output of a command to a file, like `ls > filelist.txt`, this lists files and saves the output to filelist.txt. Use `|` to pipe the output of one command as an input to another, like `ls | grep "keyword"` to list files and filter lines that have keyword in it.

How to append the output of a command to a file

Use `>>`. For example, `echo "New content" >> myfile.txt` This adds the line "New Content" at the end of "myfile.txt".

How to use wildcards; for copying and moving multiple files at the same time

`*` can be used for copying and moving multiple files. For example, `cp *.txt /path/to/destination/` copies all text files to the destination directory, and `mv *.jpg /path/to/destination/` moves all .jpg files to the destination directory.

How to use brace expansion; for creating entire directory structures in a single command

Brace expansion gives you the ability to make an entire directory in a single command. For example, `mkdir -p project/{images,docs,src}` creates subdirectories "images," "docs," and "src" under the "project" directory. You can also add `-p` to make parent folders if they don't already exist.