

Final Assignment

Question 1

awk

- Description:
 - awk is a command-line utility that allows you to manipulate text files and extract data using patterns.
- Syntax:
 - awk + options + {awk command} + file
- Examples:
 - Print first field of /etc/passwd file
 - `awk -F: '{print $1}' /etc/passwd`
 - Print the last field of the /etc/passwd file
 - `awk -F: '{print $1}' /etc/passwd`
 - Print the first and last field of the /etc/passwd
 - `awk -F: '{print $1," = ",$NF}' /etc/passwd`

cat

- Description:
 - cat is a command-line utility that allows you to view the contents of a file.
- Syntax:
 - cat + options + file
- Examples:
 - Display the contents of a file in the working directory
 - `cat filename.txt`
 - Display the contents of a file using absolute path
 - `cat /home/jdpaz/filename.txt`
 - Display the content of a file with line numbers
 - `cat -n filename.txt`

cp

- Description:
 - cat is a command-line utility that allows you to copy files or directories
- Syntax:

- `cp sourceFile destinationLocation`

- Examples:

- Copy a file to a new location
 - `cp file.txt /home/jdpaz/`
- Copy a directory and its contents to a new location
 - `cp -r /old/location /new/location/`
- Rename a file
 - `cp oldfile.txt newfile.txt`

cut

- Description:

- `cut` is a command-line utility that allows you to extract specific columns or fields from a file

- Syntax:

- `cut + option + file(s)`

- Examples:

- Print the first column of a file
 - `cut -d ' ' -f 1 file.txt`
- Print the second and third columns of a file
 - `cut -d ',' -f 2,3 file.txt`
- Print the first three characters of each line in a file
 - `cut -c 1-3 file.txt`

grep

- Description:

- A command-line utility that allows you to search for patterns in a file

- Syntax:

- `grep + option + search criteria + file`

- Examples:

- Search any line that contains the word "dracula" in the given file
 - `grep 'dracula' dracula.txt`
- Search any line that contains the word "dracula" in the given file and display any time the pattern is matched
 - `grep -c 'dracula' dracula.txt`
- Search files for any line that contains the word "dracula" in given directory
 - `grep -r 'dracula' draculaFolder`

head

- Description:
 - Displays the top N number of lines of a given file.
- Syntax:
 - head + option + file(s)
- Examples:
 - Display the first 10 lines of a file
 - head example.txt
 - Display the first 7 lines of a file
 - head -7 example.txt
 - Display the first 20 bytes of a file
 - head -c 20 example.txt

ls

- Description:
 - Lists all files and folders in a directory
- Syntax:
 - ls + option + file(s)
- Examples:
 - List contents of the directory "example"
 - ls example
 - List contents of Documents folder using absolute path
 - ls ~/Documents
 - Long list contents of a directory
 - ls -l example

man

- Description:
 - Displays user manual of any Linux command
- Syntax:
 - man + option + command name
- Examples:
 - Read the manual of the ls command
 - man ls
 - List all available manual pages on the system and provide a short description
 - man -k .
 - Read a short description of a manual

- `man -f ls`

mkdir

- Description:
 - Creates a directory
- Syntax:
 - `mkdir + option + folder name`
- Examples:
 - Create a folder
 - `mkdir example`
 - Create a folder recursively
 - `mkdir -r example/test/class`
 - Create multiple folders
 - `mkdir example test class`

mv

- Description:
 - Moves and renames directories
- Syntax:
 - `mv + source + destination`
- Examples:
 - Move file from a directory to another using relative path
 - `mv Downloads/homework.pdf Documents/`
 - Move file from a directory to another using absolute path
 - `sudo mv ~/Downloads/theme /usr/share/themes`
 - Move multiple directories to a different directory
 - `mv games/ wallpapers/ /media/student/flashdrive/`

tac

- Description:
 - Displays contents of a file in reverse order
- Syntax:
 - `tac + option + file`
- Examples:
 - Display content of a file using relative path

- `tac example.txt`
- Display content of a file using absolute path
 - `tac ~/Downloads/example.txt`
- Display the version information and exit
 - `tac --version`

tail

- Description:
 - Displays the last N number of lines of a given file.
- Syntax:
 - `tail + option + file`
- Examples:
 - Display the last 10 lines of the file
 - `tail example.txt`
 - Display the last 5 lines of a file
 - `tail -5 example.txt`
 - Display the last 5 lines of a file using absolute path
 - `tail -5 ~/Downloads/example.txt`

touch

- Description:
 - Used to create files
- Syntax:
 - `touch + option + file`
- Examples:
 - Create a file called list.txt
 - `touch list.txt`
 - Create several files
 - `touch list.txt example.csv`
 - Create a file with a space in the name
 - `touch "This is a test.txt"`

tr

- Description:
 - Translates or deletes characters from standard output
- Syntax:
 - `standard output | tr + option + set + set`

- Examples:
 - Translate a period into a comma
 - `cat example.csv | tr ',' ','`
 - Translate white space into tabs
 - `cat list.txt | tr "[:space:]" '\t'`
 - Translate tabs into space
 - `cat list.txt | tr -s "[:space:]" ' '`

tree

- Description:
 - Displays directory paths and files in each subdirectory in a tree shape
- Syntax:
 - `tree + option + folder`
- Examples:
 - List files and directories
 - `tree example/`
 - List files, directories, and their permissions
 - `tree -p example/`
 - List directory from absolute path
 - `tree ~/Documents`

Question 2

How to work with multiple terminals open?

If you are using a GUI, you can use Tilix to open multiple command line interfaces. If you are not using a GUI, you can use `ctrl + alt + F1` through `ctrl + alt + F6`. These will switch you between virtual console terminals.

How to work with manual pages?

To open a manual page, simply type in `man + command`. Once open, you can scroll with the arrow keys. To exit, press the `q` key.

How to parse for specific words in the manual page?

To search for a specific phrase, use `/<keyword>`.

How to redirect output (> and |)

The `>` symbol is used to redirect the output of a command to a file. The `|` command is used to redirect the output of a command to another command.

How to append the output of a command to a file

To append the output of a command to a file, you would use `>>`. This will add the output to the end of the file without overwriting it.

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

Wildcards can represent any number of any characters.

Example:

To move all files with the name `example` to the `exampleFolder`, regardless of file type.

```
mv example.* exampleFolder
```

How to use brace expansion for creating entire directory structures in one command

Brace expansion allows you to make multiple folders with a single command.

Example:

Create a directory structure with the main directory `example` and the subdirectories `test`, `sample`, and `practice`.

```
mkdir -p example/{test,sample,practice}
```