Final Assignment

Q1

awk

*mostly used for pattern scanning and processing *awk [options] 'selection_criteria {action}' input-file > output-file *EX. *$ awk '{print}' employee.txt *$ awk '/manager/ {print}' employee.txt *$ awk '{print $1,$4}' employee.txt

cat

*concatenates files together *cat (older file name) > (newer file name) *EX. *cat jtp.txt
*cat > javatpoint
*cat combo > combo2

cp

*copying files from one location to another *cp [Original] [Destination] *EX. *cp -v test.txt dest_directory/ *cp file1.txt file2.txt file3.txt dest_directory/ *cp -v -r source_dir/ dest_directory

cut

*allows you to cut out sections of a specified file or piped data and print the result to standard output *cut [option] [file] *EX. *cut -b 1 employees.txt *cut -c 10- employees.txt *cut -d [delimiter] [file]

grep

*searches for text and strings defined by users in a given file *grep [options] search-string [file] *EX. *grep -i "UNix" geekfile.txt *grep -c "unix" geekfile.txt *grep -l "unix" f1.txt f2.txt f3.xt f4.txt

head

*prints the first lines of one or more files to standard output *head [OPTION]... [FILE]... *EX. *$ head -n 5 state.txt *$ head -c 6 state.txt *$ head state.txt capital.txt

ls

*designed to list the names and features of files and directories *ls [Options] [File] *EX. *ls -a *ls -l *ls -F

man

*display the user manual of any command that we can run on the terminal *$man [OPTION]... [COMMAND NAME]... *EX. *man ls
*man 2 passwd
*man -a

mkdir

*allows users to create new directories *mkdir {dir} (replace {dir} with the desired name of your directory) *EX. *mkdir --version *mkdir -v [directories] *mkdir -p first/second/third

mv

*moves files and directories from one directory to another or renames a file or directory *mv [options(s)] [source_file_name(s)] [Destination_file_name] *EX. *mv intro manual/chap1 *mv chap3 manual *mv appendix apndx.a

tac

*concatenates and prints a file to the screen *tac [OPTION]... [FILE]... *EX. *tac -b concat.txt tacexample.txt *tac -r concat.txt tacexample.txt *tac -s concat.txt tacexample.txt

tail

*output the end of a file *tail [file_name] *EX. *tail error.log *tail -n 2 mynote.txt *tail -c 50 mynote.txt

touch

*modify a timestamp *touch *EX. *touch myfile1
*touch -a usr
*touch -m usr

tr

*translates or deletes characters from standard input and writes the result to standard output *tr [options] SET1 [SET2] *EX. *tr -s " " <<< "Welcome To GeeksforGeeks" *tr -d W <<< "Welcome to GeeksforGeeks" *$ tr -d [:digit:] <<< "my ID is 73535"

tree

- displays the directory structure of a file system in a tree-like format
- $ tree -a [DirectoryName/Path
- EX. *tree -a ./GFG
    - tree -P sample* .
    - tree --filelimit 3 ./GFG

Q2

How to work with multiple terminals open?

horizontal split: Ctrl + Shift + o

How to work with manual pages?

type man followed by the program you want to learn about

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

- / search string – find matches to “search string” in current man page”

- n – go to next match.
- shift + n – go to prior match.

How to redirect output (> and |)

using either the ">" (greater-than symbol), or using the "|" (pipe) operator

How to append the output of a command to a file

use the echo command with the >> operator

How to use wildcards

Wildcards can be used in almost any Linux/Unix command or utility that accepts multiple file parameters. From ls to pandoc

How to use brace expansion

You can use the brace expansion in the form of {x..y.. z} to generate values from x till y while incrementing by z