

Study Guide

Question 1

awk

Awk is a scripting language used for processing and displaying text. Awk can work with a text file or from standard output awk + options + {awk command} + file + file to save is optional awk '{print \$1}'

~/Documents/Csv/cars.csv awk -F: '{print NR,\$1,\$3}' /etc/passwd awk 'NR > 3 { print }' /etc/passwd

cat

The cat command is used for displaying the content of a file. cat + option +files to display cat todo.lst, cat

~/Documents/todo.lst cat -s ~/Documents/todo.md

cp

cp copies files/directories from a source to a destination cp + files to copy + destination cp

Downloads/wallpapers.zip Pictures/ cp -r ~/Downloads/wallpapers ~/Pictures/ cp Downloads/wallpapers/* ~/Pictures/

cut

The cut command is used to extract a specific section of each line of a file and display it to the screen cut +

option + files cut -d ':' -f1 /etc/passwd cut -d ':' -f1,7 /etc/passwd cut -b 1-5 usernames.txt

grep

Grep is used to search text in given file. grep works line by line basis grep + option + search criteria + files

grep 'dracula' ~/Documents/dracula.txt grep -i 'dracula' ~/Documents/Books/dracula.txt grep -v 'war'

~/Documents/books/war-and-peace.txt

head

The head command displays the top N number of lines of a given file head + option + files head

~/Documents/Book/dracula.txt head -5 ~/Documents/Book/dracula.txt head 5

~/Documents/Book/dracula.txt

ls

ls is used for listing the content of a given directory or the file itself ls + option + directory to list ls -a ls -a

~/Pictures ls -lR ~/Pictures

man

man (manual) view the manual of a command man + command man ls man 5 passwd man -k file

mkdir

mkdir is used for creating a single directory or multiple directories mkdir + the name of the directory mkdir wallpapers mkdir wallpapers/ocean mkdir ~/wallpapers/forest

mv

mv moves and renames directories mv + source + destination mv Downloads/homework.pdf Documents/sudo mv ~/Downloads/theme /usr/share/themes mv homework.docx cis106homework.docx

tac

The tac command is used for displaying the content of a file in reverse order tac + option + files to display tac todo.md tac ~/Documents/todo.md tac lab7.md

tail

The tail command displays the last N number of lines of a given file tail + option + file tail ~/Documents/Book/dracula.txt tail -5 ~/Documents/Book/dracula.txt tail 5 ~/Documents/Book/dracula.txt

touch

Touch is used for creating files touch + option touch list touch ~/Downloads/games.txt touch "list of foods.txt"

tr

The tr command is used for translating or deleting characters from standard output. Standard output | tr + option + set + set cat file.txt | tr ' ' , ' ' cat program.py | tr "[:space:]" '/t' cat file.py | tr -s "[:space:]" ' '

tree

The tree command shows a list of files in a select folder tree + folder/installation tree -d ~ tree -a Directory/ tree -f Directory/

Question 2

How to work with multiple terminals open

You can work with multiple terminals by having them in different tabs or in separate panes in the same tab You can switch between them using the arrow keys in conjunction with the tab and control/shift keys

How to work with manual pages

Use the command man + command + name

How to search for specific words in the manual page

man -s or man -k

How to redirect output > and |

The | command is called a pipe. It is used to pipe, or transfer, the standard output from the command on its left into the standard input of the command on its right. "Hello Worlds" | wc -w The > symbol is used to redirect output by taking the output from the command on the left and passing as input to the file on the right. "Hello" > hello.txt

How to append the output of a command to a file

The >> shell command is used to redirect the standard output of the command on the left and append (add) it to the end of the file on the right. "Hello World!" >> greetings.txt

How to use wildcards

WILDCARDS/FILE GLOBBING CHEAT SHEET

The * Wildcard	The ? Wildcard
List all txt and python files	Copy all the files that have 2 characters between 2 letters.
ls -A *.txt *.py	cp Downloads/b??k.pdf Documents/
List all the files that have 'demo' in the name	List all the files with a 2 letter file extension
ls -A *demo*	ls -A Scripts/*.?? Programs/program.?? Downloads/setup*.??
Move all the files inside a directory	Remove all the hidden files in a given directory
mv Pictures/* ~/Backup/	rm Documents/.??*.doc
Delete all files that start with a given word	List all the hidden files that have a 4 letter file extension
rm Downloads/copy* Documents/new*.docx	ls -A .??*.????

The [] wildcard
List all the text files that start with an uppercase letter and all the python files that start with a number
ls -A [A-Z]*.txt [0-9]*.py
List all the ruby files that do not start with a number.
ls -A [!0-9]*.rb
List all the files that have one of the characters in a set before the extension
List *[xyz].*
List all files whose name begins with any 3 combination of numbers and the current user's username:
ls -A [0-9][0-9][0-9]\$USER*

Wildcard	Matches	Example
*	0 or multiple characters	ls *.pdf
?	1 character	ls program?.py
[]	1 character from a given set of characters	ls document[A-Z].doc
[!]	The opposite of the given set	ls new-doc[!0-9].docx

POSIX CHARACTER CLASSES		
POSIX class	Equivalent to	Matches
[:alnum:]	[A-Za-z0-9]	Digits, uppercase and lowercase letters
[:alpha:]	[A-Za-z]	Upper- and lowercase letters
[:ascii:]	[\x00-\x7F]	ASCII characters
[:blank:]	[\t]	Space and TAB characters only
[:cntrl:]	[\x00-\x1F\x7F]	Control characters
[:digit:]	[0-9]	Digits
[:graph:]	[*[:cntrl:]]	Characters which have graphic representation
[:lower:]	[a-z]	Lowercase letters
[:print:]	[[:graph:]]	Graphic characters and space
[:punct:]	[~!@#\$%^&*()-+=~.;:;<=>?[]{}~]	Punctuation characters except letters and digits
[:space:]	[\t\n\r\f\v]	All whitespace characters
[:upper:]	[A-Z]	Uppercase letters
[:word:]	[A-Za-z0-9_]	Word characters
[:xdigit:]	[0-9A-Fa-f]	Hexadecimal digits

How to use brace expansion

Brace expansions use commands to make folders shortcuts. To use them rm -r wallpapers/ mkdir -p wallpapers/cars/{1080p,2k,4k} tree wallpapers/