

Obaida Kamil

250982002

Okmail2

Cs2211

Assignment 1

February 4th, 2019

Question 1: ssh and Basic Commands (12 Marks)

(a) Log in to the course server (cs2211b.gaul.csd.uwo.ca) using SSH as you would in the Lab.

Give the full command you gave to log in (no output is required for this question).

- cs2211b.gaul.csd.uwo.ca -> username okamil2 -> password: *****

(b) Issue a command to print the hostname of the server.

- Input: hostname
- Output: cs2211b.gaul.csd.uwo.ca

(c) Print a list of users currently logged into the server.

- Input: who
- Output:
 - dservos5 pts/0 2019-01-17 13:00 (pce-05.rndr.csd.uwo.ca)
 - ssafdar8 pts/1 2019-02-04 22:02 (dyn172-30-159-218.wireless.uwo.pri)
 - jwan949 pts/2 2019-02-04 18:50 (135.0.36.3)
 - ekuperma pts/3 2019-02-04 20:44 (cpebc4dfb4ccdf3-cmbc4dfb4ccdf0.cpe.net.cable.rogers.com)
 - ahamed4 pts/4 2019-02-01 19:33 (cpe64777d65e993-cm64777d65e990.cpe.net.cable.rogers.com)
 - dservos5 pts/5 2019-02-04 11:35 (207.189.31.7)
 - swang927 pts/6 2019-02-04 15:30 (dyn172-30-16-68.wireless.uwo.pri)
 - glasala pts/7 2019-02-04 20:24 (cpef81d0fa6d5b3-cmf81d0fa6d5b0.cpe.net.cable.rogers.com)

(d) Print the current date formatted as year-month-day (e.g. 2019-01-05).

- Input: date +%F
- Output: 2019-02-04

(e) Navigate to your home directory (if you are not already there) and print your current working directory.

- Input: `pwd`
- Output: `/gaul/s1/student/2019/okamil2`

(f) Using your favourite editor, create a file named `hello.txt` in your home directory containing the text “Hello World! My name is [Your Name].” where “[Your Name]” is your real name (only give the command to open the editor/file and not the output for this question).

➤ Input: `nano hello.txt`

(g) Without using an editor, give a command to display the contents of `hello.txt` assuming your working directory is your home directory. Use relative paths (give both the command and output).

- Input: `cat /gaul/s1/student/2019/okamil2/hello.txt`
- Output: `Hello World! My name is Obaida Kamil.`

(h) Without using an editor, give a command to display the contents of `hello.txt` that will work for any current working directory and any user (give both the command and output).

- Input: `cat ~/hello.txt`

(i) Make a directory called `text files` in your home directory.

- Input: `mkdir textfiles`

(j) Copy `hello.txt` into `text files` and name it `hello2.txt` (in one command).

- Input: `mv hello.txt textfiles/hello2.txt`

(k) With your home directory as your current directory, rename `hello2.txt` (in the `text files` directory) to `newhello.txt`. Use only one command.

- Input: `mv /gaul/s1/student/2019/okamil2/textfiles/{hello2,newhello}.txt`

(l) Change your current working directory to `text files` and delete `newhello.txt` (give both commands).

- Input: `cd textfiles ; rm newhellow.txt`

Question 2: List Command and Wild Cards (13 Marks)

On the `cs2211b` server change your working directory to `/usr/bin`. This location stores the

executable programs available to users on the server. Using only the ls command and wild cards, display the following files names in /usr/bin. Make sure you show both the command and output for each of the following parts.

(a) Files whose names are exactly 6 characters long.

- `ls ??????`
 - Output: spell colcrt funzip gtroff links2 msgfmt pkexec recode splint whatis
 - aulast column gcc-ar gunzip locale newgrp precat rename ssltap whoami
 - auvirt cowsay gcc-nm hostid logger nf-log prezip renice stdbuf x86_64
 - base64 csplit gctags import lsattr numfmt printf rpcgen stream xzdiff
 - busctl dracut gencat ionice lz4cat openvt ps2pdf runcon teamnl xzgrep
 - bzdifff dvipdf getent ipcalc makedb passwd ps2ps2 script telnet xzless
 - bzgrep expand getopt ispell md5sum pdf2ps pstack setuid umount xzmore
 - bzless factor gnroff kadmin mkfifo peekfd pstree shcomp unlink zegrep
 - bzmore fc-cat grotty keyctl mktemp pf2afm pwmake slogin uptime zfgrep
 - catman finger groups ktutil msgcat pflags python soelim usleep zforce
 - chatr font2c gstack ld.bfd msgcmp piconv ranlib splain vmstat

(b) Files that have t as the third letter and are at least 3 characters long.

- `ls ??[t]*`
 - Output: batch getopts patch setleds
 - cat gettext pathchk setmetamode
 - catchsegv gettext.sh pftp setpriv
 - catman git pkttyagent setuid
 - cut git-receive-pack prtstat setterm
 - cvtsudoers git-shell pstack setup-nssysinit
 - date git-upload-archive pstree setup-nssysinit.sh
 - dotlockfile git-upload-pack pstree.x11 setvtrgb
 - fmt gstack pstruct sftp
 - gctags iptables-xml python sotruss
 - getcert mktemp python2 stty
 - getconf netaddr python2.7 uptime
 - getent netstat setarch watch
 - getfacl nettle-hash setfacl watchgnupg
 - getkeycodes nettle-lfib-stream setfont wftopfa
 - getopt opt setkeycodes

(c) Files that do not end with a letter (can be any other character or symbol).

- `ls *[^a-z]`
 - Output: [gdk-pixbuf-query-loaders-64 perl5.16.3
 - base64 geopipllookup6 ping6
 - bashbug-64 gio-querymodules-64 ps2pdf12
 - bunzip2 gpg2 ps2pdf13

```
➤ bzip2      gpgv2      ps2pdf14
➤ c89        grub2-mkpasswd-pbkdf2  ps2ps2
➤ c99        gsdj500     pstree.x11
➤ clang++    gtk-query-immodules-3.0-64  python2
➤ csslint-0.6  i386      python2.7
➤ db_dump185  ksh93     sqlite3
➤ diff3      links2     tracepath6
➤ easy_install-2.7  linux32  unlz4
➤ emacs-24.3  linux64    x86_64
➤ fc-cache-64  lz4
➤ gconftool-2  pango-querymodules-64
```

(d) Files that start with cat or pi.

- `ls pi* ; ls cat*`
 - Output: `pic pico piconv pinentry pinentry-curses ping ping6 pinky`
 - `cat catchsegv catman`

(e) Files that contain the word “cat” anywhere in their name or start with the prefix “pre” and have at least one more character in their name. For example, the following would be valid: cat, zcat, catman, fallocate, precat, prezip and pre-grohtml. If a file with just the name “pre” existed, it would not be matched.

- `ls *cat* ; ls pre*`
 - `bdftruncate catchsegv db_replicate gapplication msgcat systemd-cat xzcat`
 - `bzcat catman fallocate gencat ncat truncate zcat`
 - `cat chcat fc-cat lz4cat precat xmlcatalog`
 - `precat preconv pre-grohtml preunzip prezip prezip-bin`

(f) Explain the difference between `ls -d *` and `echo *`. Your explanation should not be about the format of the output.

2 of ??

- The `ls -d` is to list directories and `echo` is to print all files and folders.

CS2211 - Software Tools and Systems Programming Assignment #1

Question 3: Redirection (13 Marks)

Go to your home directory and create a text file called `nums.txt`. Write 15 lines in this file, where each line will have just a number from the list 01, 02, 03, ..., 13, 14, 15 in this order. You do not need to show how you made this file.

Use the commands cat, tail or head for part B-D:

(a) Display the content of nums.txt.

- Input: cat nums.txt

➤ Output: 01

➤ 02

➤ 03

➤ 04

➤ 05

➤ 06

➤ 07

➤ 08

➤ 09

➤ 10

➤ 11

➤ 12

➤ 13

➤ 14

➤ 15

(b) Display the last 5 lines of nums.txt

- Input: tail -5 nums.txt

➤ Output: 11

➤ 12

➤ 13

➤ 14

➤ 15

(c) Store the first 4 lines of nums.txt in nums2.txt. Provide the command and the contents of nums2.txt.

- Input: head -4 nums.txt > nums2.txt

- Input: cat nums2.txt

➤ Output:

➤ 01

➤ 02

➤ 03

➤ 04

(d) Use the sort command to reverse the contents of nums.txt and both store the result in nums3.txt and display it to the screen (Hint: Check the manual page for sort and tee). The command should not alter the contents of nums.txt in anyway.

- Input: Sort -r nums.txt > nums3.txt

- Input: `cat nums3.txt` (to check what's in it)
- Input: `cat nums.txt` (to make sure it is not altered)

(e) Use the `tr` command to replace the following letters (both upper or lowercase) in `text.txt` with the numbers in the corresponding row in the table and store the result in `leet.txt`. Use only one command and do not edit the contents of `text.txt` in any way. You may provide your own content for the `text.txt` file.

A 4

B 8

E 3

G 6

H 4

I 1

L 1

O 0

Q 9

S 5

T 7

Z 2

- Input: `tr 'ABEGHILOQSTZ' '483641109572' < text.txt > leet.txt`

(f) Give a command to create a hard link to the file `nums.txt` called `numbers.txt`. Redirect both the standard output and standard error to the same file named `output.txt`. Ensure that you use the verbose option (`-v`) such that the name of the linked file is printed. Try your commands with both `nums.txt` existing and not existing.

- `Ln -v nums.txt numbers.txt > output.txt ; cat output.txt`

(g) Explain the difference between `cat < nums.txt` and `cat nums.txt`.

- When we do `cat < nums.txt` the shell opens the file through `cat`. And when do `cat nums.txt`, the file opens with no visual difference.

Question 4: Pipes (12 Marks)

Give a command for each of the following and show the output you received on the `cs2211b` server.

Each command should use a pipe.

(a) Count the number of users currently on the server (it is ok to count users twice if they are logged in twice). (Hint: Check the manual page for the wc command).

3 of ??

- Input: `who | wc -l`

➤ Output: 37

(b) Use pipes to take the output of the fortune command, convert the text to all upper case and display it in a speech balloon above a cow and append the result to cowlog.txt. (Hint: Check the manual pages for the fortune, tr and cowsay commands)

- `fortune | tr "[a-z]" "[A-Z]" | cowsay > cowlog.txt ; cat cowlog.txt`

(c) Display the 6th to 10th lines (inclusively) of nums.txt from Question 3 using the head and tail commands.

- Input: `head -10 nums.txt | tail -6`

➤ Output: 05

➤ 06

➤ 07

➤ 08

➤ 09

➤ 10

(d) Display to the screen and store in last20.txt the last 20 lines of .bash history (a file in your home directory) with duplicate lines removed (Hint: Check the manual page for sort). Provide the command and both your output and the contents of last20.txt.

- `sort -u .bash_history | tail -20 .bash_history > last20.txt`

➤ output: `grep -i z*[^f]*ly /usr/share/dict/words`

➤ `grep -ic z*[^f]*ly /usr/share/dict/words`

➤ `grep -ic z**ly /usr/share/dict/words`

➤ `grep -ic z*^[^f]*ly /usr/share/dict/words`

➤ `grep -ic z* ^[^f]*ly /usr/share/dict/words`

➤ `grep -ic [^f]z*ly /usr/share/dict/words`

➤ `grep -i [^f]z*ly /usr/share/dict/words`

➤ wrongfully

➤ `grep -i [^f]z**ly /usr/share/dict/words`

➤ `grep -i ^z[^f]ly$ /usr/share/dict/words`

➤ `grep -ic ^z[^f]ly$ /usr/share/dict/words`

➤ `grep -ic ^z*[^f]ly$ /usr/share/dict/words`

➤ `grep -ic ^z*[^f]*ly$ /usr/share/dict/words`

- `grep -i ^z*[^tf]*ly$ /usr/share/dict/words`
- `grep -i ^z[^tf]*ly$ /usr/share/dict/words`
- `grep -ic ^z[^tf]*ly$ /usr/share/dict/words`
- `grep -i ^[a-z].[a-z]$ /usr/share/dict/words`
- `grep -i '^[a-z]\\1' /usr/share/dict/words`
- `grep -i '^[a-z]\\1' /usr/share/dict/words`
- `okamil`

Question 5: grep (18 Marks)

The file `/usr/share/dict/words` contains a list of dictionary words separated by new lines (one word per line). Use this file and the `grep` command find/do the following:

(a) Count the number of words that do not contain the word `mil` anywhere in them (do not use the `wc` command).

- Input: `grep -c -v mil /usr/share/dict/words`
- Output: 478521

(b) Count the number of words that end in `ing` (do not use the `wc` command).

- Input: `grep -c 'ing$' /usr/share/dict/words`
- Output: 23714

(c) Display any word that contains 5 or more vowels in a row. The match should be case insensitive (i.e. AAAAAA would match as would Aelou or eeeeeee).

- Input: `grep -i "[aeiou]{5}" /usr/share/dict/words`
- Output: AAAAAA
- Aeaea
- Aeaeae
- AIEEE
- cadiueio
- Chaouia
- cooeeing
- euouae
- Guauaenok
- miaoued
- miaouing
- Pauie
- queueing

(d) Display any word that starts with the letter z , ends with the letters ly and does not contain the letter f or t anywhere in the word.

- Input: `grep -l ^z[^tf]*ly$ /usr/share/dict/words`

- Output: zanily
- zealously
- zigzaggedly
- zippingly
- Zolly
- zonally
- zoogeographically
- zoographically
- zoologically
- zoophily

(e) Display any word that starts and ends with the same two letters. For example, toronto, papa or eraser. You do not need to provide your output for this part.

- Input: `grep -i '^(\.)(\.)\2\1$$' /usr/share/dict/words`

Question 6: File Permissions (15 Marks)

On the cs2211b server, accomplish the following tasks and provide the commands you used and the output you received:

(a) Create a directory called Top1 in your home directory and setup its contents like so:

- Under Top1, create a sub-directories Dir1 and a regular file File1.
- Under Dir1, create directories Dir3 and Dir4.
- Under Dir3, create a regular file File3.
- Under Dir4, create three regular files File4, File5 and File6.
- Under Top1 make Dir2 a symbolic link to Dir1/Dir4.

- Mkdir Top1
- Cd Top1
- Mkdir Dir1 ; touch File1
- Cd Dir1
- Mkdir Dir3 ; mkdir4
- Cd Dir3
- Touch File3
- Cd Dir4
- Touch File4 ; Touch File5 ; Touch File6
- Cd
- Cd Top1

➤ In -s /gaul/s1/tudent/2019/okamil2/Top1/Dir2 Dir1/Dir4

(b) Set the permissions on the directory Top1 such that all permissions are granted for the owner, and none are granted for others and group.

- Input: `chmod u=rwx ~/Top1`
- Input: `chmod go -rwx Top1`

➤ Output: `drwx--S---`

(c) Set the permissions on Dir1 such that the owner has all permissions and only read and execute for others and group.

- Input: `chmod u=rwx Dir1`
- Input: `chmpd go+rx`
- Input `chmod go-w Dir1`

➤ Output: `drwxr-sr-x`

(d) Dir3 should have all permissions set for the owner, read permission alone for group, and none for others.

- Input: `chmod u=rwx Dir3`
- Input: `chmod g+w`
- Input: `chmod o-rwx`

➤ Output: `drwxrws---`

(e) File1 should have read permission alone set for all.

4 of ??

- Input: `chmod uog-wx File1`

➤ Output: `-r--r--r---`

(f) File5 should have execute permission alone set for others and group, and read and execute permission for the owner.

- Input `chmod u+rx File5`
- Input: `chmod og+x File5`
- Input: `chmod uog-w File5`

➤ Output: `-r-x--x--x`

(g) Display the permissions of Top1 and no other files.

- Input: `ls -dl Top1`

➤ Output: `drwx--S---`

(h) Display the permissions of File4, File5 and File6 with one command.

- input: `ls -i`

➤ output: Output: -rw-rw-r--. 1 okamil2 okamil2 0 Feb 4 17:00 File4

-r-x--x--x. 1 okamil2 okamil2 0 Feb 4 17:00 File5

-rw-rw-r--. 1 okamil2 okamil2 0 Feb 4 17:00 File6'

Question 7: Terminology and the Shell (17 Marks)

Using the proper terminology (e.g., command, option, option argument, and command argument), identify the constituent parts of the following UNIX commands (you do not need to explain what the command does, just identify the parts) and the number of each parts. For example, if the command was `ls -lad file1 file2` you should give the following answer:

One Command: `ls`

Two Command Arguments: `file1`, `file2`

Three Options: `l`, `a`, `d`

Zero Option Arguments

You should give the options and arguments as the command would see them, that is after the shell parses them for wild cards, escape sequences, etc.

UNIX Commands:

(a) `man ls`

➤ command is `man` and argument is `ls`

(b) `echo "There is no place like 127.0.0.1"`

➤ command is `echo`, command argument is `"there is no place like 127.0.0.1"`

(c) `wc -m -l myfile.txt nums.txt`

➤ command is `wc`, options are `-m -l` and command arguments are `myfile.txt` and `nums.txt`

(d) `ls --all -ld /usr/bin/cat*`

➤ command is `ls`, options are `--all` and `-ld` and command argument is `usr/bin/cat*`

(e) `nano -w -o /gaul/s1/student/1985/dservos5 myfile.txt`

Hint: It may be helpful to read the manual pages for these commands.

5 of ??

➤ command is `nano`, options are `-w -o` and command argument is `/gaul/s1/student/1985/dservos5` and `myfile.txt`