CTIS156 Final 20/05/2011

Name: Section: Student No:

Closed Book, closed note exam. You may use calculators. No exchange of calculators. No Mobiles

Show your work! we must follow your reasoning.

You are required to write down commands with necessary arguments and options. Give the best result that you can give!

(**Hint: think external programs, not editors. Note also # means number . Each question/subquestion is 2 pts, unless otherwise stated!**

Be brief but to the point!)

I certify that this is my own work only.

Name: ....

Signature: ....

Time of submission: ....

1. Give brief answers! Describe

   - Discuss diffrence between open source software and free software. if possible give an example of software which is free and not open source, and vice verse?

   - Discuss difference between Microsoft License, BSD Licence and GNU GPL License.

   - list conditions for a software to be called free (be brief and precise)

2. explain briefly, each command is used as `command string` (**1 pts** each)

   - which

     tells location of the given program withing PATH of the users

   - whereis

     tells location of files related to the given program

   - whatis

     gives one line explanation of the program

   - apropos

     does a grep searcg on whatis database

   - man

     gives manual page of the program

3. Suppose you have directory structure (in your home) as `X Y Z` as sibling directories. Within X you have directory A, within Y directory B, and within Z directory C. You have files named alpha (with possibly different content) as a sibling of X and in directories A , B , C

   - how would you compare alpha in A with alpha in B ? (your are located in A) **use relative addressing !**

     ```
     diff alpha  ../../B/alpha
     ```

   - how would you append all 4 alpha files (in any order) into file Alpha in C, while giving the command at home directory ? (and using relative addressing!)

     ```
     cat alpha  X/A/alpha Y/B/alpha Z/C/alpha  >> Z/C/Alpha
     ```

4. Explain difference and limitations of i) `ln /dizin1/file1 /dizin2/file2` and ii) `ln -s /dizin1/file1 /dizin2/file2` . What happens after that `rm /dizin1/file1` ? (in both cases)

   ln -s is soft link, it can be given accross partitions, and to directories, when file1 removed file2 will not serve the contenet of file1, but link remains; so the object is copied to original it will work. A change in the new object will be valid in the original.

   ln , without -s, is called hard link; it can not apply across partitions, and for directories. When i) applied file1 and file2 are identical, their link count becomes 2, and file1 is removed file2 exits with the same content.

5. explain briefly each, tell whether it takes argument(s), and whether it is optional; give a simple example if necessary! (**1 point each**)

   - file

     tell type of the given file, such as txt, tar, zip, doc etc, by looking content of the file, nt the name of it

   - touch

     changes access time of the file, if it does not exist, it will create an empty one.

   - w

     sjows sysltem load, uptime (duration of the system since last reboot), and number of users currently on the system

   - who

     name of the users currently logged in

   - free

     amount of free memomry and swap (total, used, in cache and free)

   - uptime

     uptime (duration of the system since last reboot)

   - du

     disk usage, gives total disk usage of the given directories, cumulative, with no argument, PWD is default

   - df

6. Given the text file X

   - find lines in X containing **word** INTER and put these in new file x.1

     ```
     grep -w INTER X > x.1
     ```

   - Using `sed`, replace each word `INTER` with *inter* in file X, into file x.2 . You may assume there is no ”INTER” word on the boundary

     ```
       sed ’s/\<INTER\>/inter/g’ X > x.2
     ```

   - translate all upper case letters (A-Z) into lowercase (a-z) in X and write to x.4, using external programs

```
cat X | tr A-Z a-z > x.4
```

7. You want to have temp directory under /home/yourself, in which every user can create any number files, modify them, but can not write other persons files what permission should give to temp ? **4 pts**

   ```
     mkdir temp; chmod 01777 temp
   ```

8. You have a text file X which has 100 lines

   - you want to change the string `Internet` with `internet` between lines 30-50, in every occurrence. You have to use vi/vim! write into file Y (you need to use vi/vim, in command mode)

     ```
     vi X

     :30,50s:Internet:internet:g

     :w Y

     :q!
     ```

   - do it with `sed`

     ```
     sed ’30,50s/Internet/internet/g’ X > Y
     ```

9. You have a process `make` which is running in the background. Describe how

   - to determine its pid (process id)

     `ps ax | grep make | awk ’{print $1 } ’` OR

     first field of `ps ax | grep make` is PID

   - to bring into foreground

     `fg PID` or `fg make`

- to kill it if in background

  ```
  kill -9 PID
  ```

- to kill it if in foreground)

  `ˆC` in other words Control-C

10. **Monthly** We want to write script for backup named MyMountly. Recall that `date +%F` will yield date as in 2012-01-10. It will look for all odt files in /home will make a tar file in `/var/Yedek/` directory with name year-month-day.tar format. This script will run 1st day of each month at 1.05 am. Write script MyMountly. and specify correnponding cron/at entry. **5 pts**

    Monthly.sh:

    ```
    #!/bin/sh
    dosya=$( date +%F )
    find /home -type f -name "*.odt" -exec tar rf /var/Yedek/$dosya.tar \
       '{}' \;
    
    crontab -e
    5  1 1 * *   Monthy.sh
    ```

11. `<GroupA>`

    Dictionary. Suppose you have a large file, say, dene.txt. We want to find words in this file. Assume we have punctuation characters**. , ; @** only. It has also space and tabs as white space (You can represent them as ␣ and `\t` in your solution). Thus, in this file words are separated by punctuation characters ; and white spaces; or both. We want each line contains a single word , and no white space and no punctuation characters. Just **use sed, and sort**

    - Replace all white space and all punctuation characters with ”,” (obtain dene1.txt)

      ```
      sed 's/[.;@ \t]/,/g' dene.txt > dene1.txt
      
      
      tr -s , dene1.txt  > dene2.txt
      ```

    - Cut lines along , (obtain dene3.txt)

let file SED be:

```
s/,/\
/g
```

```
sed -f SED dene2.txt > dene3.txt
```

- Last remove empty lines, and remove repetitions if any. Write the result into deney.txt

  ```
  sed '/^$/d' dene3.txt | sort -u > deney.txt
  ```

How would you do it ? Be precise

```
</GroupA>
```

12. `<GroupB>` Assume you have test files dene.txt, denex.txt . On lines containing Internet at the end of the line, replace every occurrence of "Firefox" with "Mozilla Firefox" (**using sed**)

    ```
    sed '/Internet$/s/Firefox/Mozilla Firefox/g' dene.txt
    ```

13. On lines containing Internet at the beginning of the line, replace every occurrence of "Firefox" with "Godzilla Firefox" (**using vi**)

    ```
    :g/^Internet/s/Firefox/Godzilla Firefox/g
    ```

    ```
    </GroupB>
    ```

14. **txt-html Bash** Assume you have lots of `*.txt` files. We want to serve them as html files. from a.txt we should obtain a.html which will look like

    ```
    <html><head><title> </title></head><body><pre>
    ---contents  of  a.txt ----
    </pre></body></html>
    ```

    Let file `HEAD` be `<html><head><title></title></head><body><pre>`

    and `TAIL` be `</pre></body></html>`

    You could use a.txt.html or a.html in the following:

    - Write a sed script say SED such that `sed -f SED  a.txt > a.html` will work!

      SED:`1i<html><head><title></title></head><body><pre>`

      ```
      $s</pre></body></html>
      ```

- How would you obtain a.html from a.txt with above info without using sed

```
cat HEAD a.txt TAIL > a.html
```

- write a bash script which will transform all .txt files into html files using HEAD and TAIL files but not using sed.

4 pts

```
#!/bin/sh
for file in *.txt
do
new=${file/txt/html}
cat HEAD $file TAIL > $new
done
```

15. Count # of lines containing string ctis, case insensitive, among all *.txt files in the current directory

```
 grep -i ctis *.txt | wc -l  OR
cat *.txt | grep -i ctis | wc -l
```

16. find lines containing strings ctis and mat, case insensitive, in file a.txt:

```
grep -i ctis a.txt | grep -i mat
```

17. find lines which contains only one of Ctis and Case words case sensitive : and output file will be d3.txt

```
grep -w Ctis a.txt | grep -vw Case > d3.txt
grep -w Case a.txt | grep -vw Ctis >> d3.txt
```

18. find all ordinary files under directory Lab, put names in /tmp/Files.TXT; and among these find those entries with name *.txt , place them in /tmp/files.txt .

```
 find Lab -type f -print > /tmp/Files.TXT
grep txt$ /tmp/Files.TXT > /tmp/files.txt  </GroupD>
```

19. `<GroupE>`

Suppose you have in /home/pardus/bin programs MyProgA, and MyProgB which takes long hours to run. Input to these programs in InA.txt and InB.txt. We want outputs should go to OutA.txt and OutB.txt. We want to collect error messages in MyError.txt. We want to run these program in the order A, B, (that is B starts after A finishes).

- Write a script file, say MyRun, when executed, will do the job; and make it executable (Just put commands that should write on the command line into MyRun) File MyRun:

  ```
  #!/bin/sh
  cd /home/pardus
  bin/MyProgA < inA.txt > OutA.txt 2> MyError.txt
  bin/MyProgB < inB.txt > OutB.txt 2>> MyError.txt
  ```

  chmod +x MyRun

- We want to execute MyRun today at 22:00, and 5 days later at 1.00am. How would you do it ?

  ```
  At -f Myrun 10pm
    at -f MyRun 1am +5 days
  ```

- We want to execute MyRun, every Monday, Wednesday and Sunday at 1.45 in April, May and August. How would you do it ?

  using cron: after crontab -e;

  ```
  45 1 * 4,5,8 1,2,7 MyRun
  ```

</GroupE>

20. <GroupF>

- Given a text file note.txt containing lines of the format `name grade gpa` where name is lower case login names, grades A, B, C, D, F, P (for a particular course) and gpa are real numbers with 2 decimal points. Use `awk` to compute the following. Hint you can use `END{ ... }, BEGIN{ ... }` constructs.

  - print number of students with grade A, B, **2 pts**
  - print average gpa of of all students which got A, B, **2 pts** (single number)

  Output should look like:

  ```
  # of  A: ...
  # of  B: ...
  Average gpa: ....
  ```

```
</GroupF>

awk -f Awk.2 note.txt


Awk.2:


#!/bin/awk -f
$2 ˜ /A/ {sayiA=sayiA+1; Total=Total+$3  }
$2 ˜ /B/ { sayiB=sayiB+1, Total=Total+$3 }
END{  Sayi=SayiA+SayiB; Orta=Total/Sayi;
    print "# of A:", sayiA
    print " # of B:", sayiB
    print " Average  gpa : ", Orta
    }
```

21. `<GroupG>`

- How can you copy everything in /home/gnu/src into /usr/src as is? You are located in /home/gnu/src, and you have necessary permissions. **(you may use several commands separated by ; but choose a minimal one)** Each one worths **2 pts**

  a) using tar

  ```
   tar cf -  . | (cd /usr/src; tar xf - )
  ```

  b) using rsync

  ```
   rsync -av  . /usr/src
  ```

  c) using cpio

  ```
  find  . | cpip -pd /usr/src
  ```

  d) cp

  ```
  cp -R ../src   /usr  OR
   cp -R .  /usr/src
  ```

`</GroupG>`

22. a.txt.gz, a1.txt.gz a2.txt.gz a3.txt.gz b.txt.bz2, b1.txt.bz2, b2.txt.bz2 are very large files which are compressed. You also have A.txt B.txt, C.txt, D.txt list Name of files containing string Firefox at the beginning of line and Internet at the

end of the line, case is important. You may use 3 separate commands. You are asked also put name of the files into /tmp/Fox. **5 pts**

```
 grep -l "ˆFirefox.*Internet$ *.txt > /tmp/Fox

 zgrep -l "ˆFirefox.*Internet$ *.txt.gz >> /tmp/Fox

  bzgrep -l "ˆFirefox.*Internet$ *.txt.bz2 >> /tmp/Fox
```

23. Write a bash script which will take an argument; check whether it is an ordinary file; if not give a mesage and exit. If it is an ordinary file, check whether it is readable, writable and executable seperatly, and give the result as a message. **5 pts**

```
#!/bin/bash
if [ $# -ne 1 ]
then
    echo " number of arguments must be 1 and be name of the file"
    exit 1
fi


if  ! [ -f  $1 ]
then
    echo "$1  is an  not an ordinary file"
    exit 0
fi
if   [ -f  $1 ]
then
    echo "$1  is an  ordinary file "
fi
if [ -r $1 ]
then
    echo " $1 is readable "
fi
if [ -w $1 ]
then
```

```
    echo " $1 is writebale "
fi
if [ -x $1 ] ; then
    echo " $1 is executable "
fi
exit 0
```

24. Write Bash script which will take one argument, and evaluate `ls -ld` for the argument, and take out the first character of the result. Note that the first character could be `--` . **5 pts**

```
#!/bin/bash


if [ $# -ne 1 ]
then
 echo " numbe rof argumnets must be 1"
 exit 1
fi
nesne=$1
set -- $( ls -ld $nesne )
first=$( echo "$" | cut -c1 )
echo "First character is: $first "
```

25. Write down a shell script that *takes three arguments* as command line input ; last two are integers and first one is a single letter: s,S,m,M,c,C . If first character is s or S sum two integers and print sum of ... and ... is .... If the first character is m, M, c, C then multiply two numbers and prints a suitable number. Otherwise prints an error message and exits. Use *one case statement* and write as output as two integers, operator and result (sum of 3 and 5 is 8) You do not need to check the input. call shell script **1.sh** **5 pts**

```
#!/bin/bash
oper=$1


case $oper in
    s|S) echo "Sum of $1 and $3 is $(( $1 + $2 )) "
```

```
            ;;
        m|N|c|C)  echo " Product  of $1 and $2 is $(( $1 * $2 )) "
           ;;
        *)  echo " First argumnets should be one of s, S,m,M,c,C   and followed by 2 integers"
           ;;
    esac
```