CTIS166 Midterm 2 Solution 27/04/2015 - Akgül

Surname, Name: Section: Student No:

1. copy everything in LabM into Dir1 with rsync . term LabM will not be copied but everything inside it (recursively).

   ```
   rsync -av .  ../Answers/Dir1/
   ```

2. copy everything in the current directory into Dir3 using tar without using any explicit tar file, if possible. You can use an explicit tar file, but you will loose credit.

   ```
   tar cf - .  | tar xf - -C ../Answers/Dir3 or
   tar cf ../Answers/2.tar .  ; (cd ../Answers/Dir3; tar xf ../2.tar )  or
   tar cf ../Answers/2.tar .  ; tar xf - -C ../Answers/Dir3
   ```

3. copy everything in LabM into Dir5 with cpio with using an auxiliary cpio file. Keep the auxiliary file in Answers directory.

   ```
   find .  -depth | cpio -o > ../Answers/3.cpio; (cd ../Answers/Dir5; cpio -idv <
   ../3.cpio
   ```

4. find all files/directories which are installed with python; and find all program's related with with program python

   ```
   whereis python
   man -k python
   ```

5. Find list of all *.txt files which are older than aa.txt

   ```
   find .  -not -newer aa.txt -type f -name "*.txt" > ../Answers/5.txt
   ```

6. combine all *.txt files within LabM hierarchy

   ```
   find .  -type f -name "*.txt" | xargs cat > ../Answers/6.txt
   ```

7. find all files with suffix .gz, bz2 or .xz within LabM hierarchy and copy them into Dir4 while retaining directory structure

   ```
   find .  -type f −name”∗.gz”−o−name”∗.bz2”−o−name”∗.xz” | cpio -pd ../Answers/Dir4
   ```

8. In a single command create directories A B C E each containing F, H, J, K and each of which containing 1 2 3 4 , each of which containing X Y and Z in directory Answers (we need to see directory and corresponding command). Also in a single command create empty files f.txt in F's, y.txt in Y's and 4.txt in 4's.

   ```
   mkdir -p  ../Answers/{A,B,C,E}/{F,H,J,K}/{1,2,3,4}/{X,Y,Z}
   touch  ../Answers/{A,B,C,E}/F/f.txt
   touch   ../Answers/{A,B,C,E}/{F,H,J,K}/4/4.txt  touch      ../Answers/{A,B,C,E}/{F,H,J}/{
   ```

9. Consider all files with name *.txt in the current directory: including sub directories
   - determine all lines among all files which contains string ayse or elif, case insensitive

   ```
   find .  -type f -name "*.txt" | xargs cat | egrep -i "ayse|elif" > ../Answers/9-1.txt

   find .  -type f -name "*.txt" | xargs egrep -ih "ayse|elif" > ../Answers/9-2.txt
   find .  -type f -name "*.txt" -exec egrep -ih "ayse|elif"   '{}' \; >   ../Answers/9-3.tx
   ```

10. Consider all files in the current directory: including sub directories
    - determine all lines among all files which contains string ayse or elif, case insensitive

    ```
    egrep -ihr "elif|ayse"  .  > ../Answers/10.txt
    ```

11. Consider all *.txt files in LabM (on the surface) determine **lines** of these files which contains **elif** and **ayse** case insensitive on a line, (under 11.txt)

    ```
    grep -ih elif *txt | grep ayse > ../Answers/11.txt  or
    cat *.txt | grep -i elif | grep -i ayse > ../Answers/11x.txt
    ```

12. find all *.txt files within LabM hierarchy and put these files in TXT1.zip

    ```
    find .  -type f -name "*.txt" | xargs zip ../Answers/TXT1.zip
    ```

13. find all *.txt files within LabM hierarchy and put these files in TXT1.tar.gz

```
find .  -type f -name "*.txt" | xargs tar czf ../Answers/TXT1.tar.gz or
find .  -type f -name "*.txt" | xargs tar cf - | gzip > ../Answers/TXT2.tar.gz
```

14. find all *.txt files within LabM hierarchy and put these files in TXT1.cpio

```
find .  -type f -name "*.txt" | cpio -o > ../Answers/TXT1.cpio
```

15. Extract contents of AB.tar.xz into Dir6 without using gzip, bzip2, xz and their derivatives like zcat;, bzcat etc, i.e just tar with suitable options only You issue commands within LabM.

```
tar xJf AB.tar.xz -C ../Answers/Dir6
```

16. Find lines in Dene1.txt.xz containing string elif and ayse case insensitive without using xz and xzcat directly

```
xzgrep -i ayse Dene1.txt.xz | grep -i elif > ../Answers/16.txt
```

17. Copy ABC and Data directories into Dir11 and compress all ordinary files in Dir11 with xz

```
cp -R ABC Data ../Answers/Dir11; find ../Answers/Dir11 -type f -exec xz   ’{}’ \;   cp
-R ABC Data ../Answers/Dir12; find ../Answers/Dir12 -type f | xargs xz
```

18. Add execute permissions for file **exec.SH** to all, remove write from others, add to suid property to owner; all in a single statement, and use symbolic method (Do these after copying into Answers)

```
cp exec.SH ../Answers; chmod a+x,o-w,u+s ../Answers/exec.SH
```

19. Consider Dene.txt, remove first 30 lines and last 20 lines obtaining DENE.TXT. Do not use information about size of Dene.txt, just assume it has more than 60 lines in it. We need to see original line numbers. Just use filters. you may use tac .

```
cat -n Dene.txt | tail -n +31 | head -n -20 > ../Answers/19.txt
```

20. Given a.txt, we want to determine the lines containing strings Net or Fox and put them in files 20-1.txt, 20-2.txt and 20-3.txt . How would you do it using: (each **2**)

grep: `egrep "Net|Fox" a.txt > ../Answers/20-1.txt`

sed:
```
sed -n ’/Net/p’ a.txt > ../Answers/20-2.txt
sed ’/Net/d’ a.txt | sed -n ’/Fox/p’ >> ../Answers/20-2.txt
```

awk:
```
awk ’/Net/’ a.txt > ../Answers/20-3.txt
awk ’!/Net/’ a.txt | awk ’/Fox/’ >> ../Answers/20-3.txt
```

21. Given a.txt, we want to determine lines containing neither Net nor Fox and put them in a files 21-1.txt, 21-2.txt, and 21-3.txt. How would you do it using: (each **2**)

grep: `egrep -v "Net|Fox" a.txt > ../Answers/21-1.txt`

sed: `sed ’/Net/d’ a.txt | sed ’/Fox/d’ > ../Answers/21-2.txt`

awk: `awk ’!/Net/’ a.txt | awk ’!/Fox/’ > ../Answers/21-3.txt`

22. Given a.txt, we want to determine the lines containing **word** Net and put them in files 22-1.txt, 22-2.txt and 22-3.txt How would you do it using:(each **2**)

grep: `grep -w Net a.txt > ../Answers/22-1.txt`

sed: `sed -n  ’/\<Net\>/p’ a.txt >  ../Answers/22-2.txt`

awk: `awk   ’/\<Net\>/’ a.txt >    ../Answers/22-3.txt`

23. **Sed**. Given A.txt we want to: (**2 pts each**)
- replace all strings Net with Internet between first line and line containing string FILIZ. Give command and write new file as Answers/B1.txt

```
sed ’1,/FILIZ/s/Net/Internet/g’ A.txt > ../Answers/B1.txt
```

- Consider lines with strings FILIZ and FATMA. On lines between them insert to the beginning of the line string ”BASLA ”. Give command and write file as Answers/B2.txt

```
sed  '/FILIZ/,/FATMA/s/ˆ/BASLA /'   A.txt > ../Answers/B2.txt
```

- Consider the line containing FATMA. Add lines ”LINE 1” and ”LINE 2” after it. Give commands and write file as Answers/B3.txt

```
sed -f ../Answers/SED A.txt > ../Answers/B3.txt
```

where file SED is:

```
 /FATMA/ a \
 LINE 1   \
 LINE 2
```

24. Run the following commands in succession: (**2 pts each**)
i) run uyu-1.sh in the background
ii) run uyu-2.sh in the foreground and then suspend it
iii) run uyu-3.sh in the background
iv) run uyu-4.sh in the foreground and then suspend it
v) kill uyu-2.sh
vi) put uyu-1.sh in the foreground and kill it
vii) run uyu-4.sh in the background

```
i) ./uyu-1.sh &
ii) ./uyu-2.sh (then)  ˆZ
iii) ./uyu-3.sh &
iv) ./uyu-4.sh (then)  ˆZ
v) kill %2
vi) fg %1 (then) ˆC
vii)  bg %4
```

25. Given data.txt, i) compute sum of the four grades for each student and write as student name sum, ii) compute class average of sum values;
write every thing into 25.txt put awk commands into file Awk.25 **5** pts

```
# Awk.25
{sum=$1+$2+$3+$4
total=total+sum
print $5, sum}
END{print "class average=", total/NR}
```

awk -f ../Answers/Awk.25 data.txt ¿ ../Answers/25.txt

26. Consider elma.txt. each line contains fruit-city-kg-TL information of a transaction. Find an Awk script to compute: **5** pts
i) total amount sold as kg for each fruit, script in Awk-26-1 and output in 26-1.txt
ii) total amount sold as TL for each city, script in Awk-26-2 and output in 26-2.txt

```
#Awk-26-1
{fruitKG[$1]+=$3}
END{  for ( i in fruitKG ) {
     print i, fruitKG[i]
}
}
```

```
#Awk-26-2
{cityTL[$2]+=$4}
 END{  for ( i in  cityTL ) {
    print i, cityTL[i]
  }
  }
```

```
awk -f ../Answers/Awk-26-1 elma.txt > ../Answers/26-1.txt
awk -f ../Answers/Awk-26-2 elma.txt > ../Answers/26-2.txt
```

27. Write down a shell script that takes an arbitrary number of integer arguments . Compute sum of its terms and print number of terms, each term on a separate line, and total sum; script in *sum.sh* and output in 27.txt $\boxed{\textbf{5}\text{ pts}}$

```
#!/bin/sh
sayi=$#
for x
do
sum=$(( sum + $x ))
echo $x
done
echo "Sum of $sayi  terms is:", $sum

../Answers/sum.sh 1 2 3 4 5 > ../Answers/27.txt
```

28. **If you remember your user-name and passwd on std machine, use /home/user/Project**. You have accounts on std with home as /home/ctisx; and lab machine that you are working. You have **Project** directory at /home/ctisx/Project on std and NAME/Answers/Project on the lab machine. The Project directories were identical, but you have added some files, changed some, and deleted some files on std machine. Assume you have passwd cTis1357xyZ on std and debian7ctis on lab. How can you make Project in lab identical with the one on std ? Give full command(s) with passwd, and apply them $\boxed{\textbf{5}\text{ pts}}$

```
rsync -av --delete ctisx@std.ctis.bilkent.edu.tr:/home/ctisx/Project/ ../Answers/Project
```

using passwd on std machine. Original Project directory kept under xproject.

29. File note.txt contains in each line name of the student, midterm1, midterm2 and final points. Weights are respectively, %30, %30 and %40. Students with weighted average greater than 80 will get A, and less than 40 will get F. Write an Awk script which will list students with A and F and at the end will print number of total students, with A and with F. ($\boxed{\textbf{5}\text{ pts}}$)

```
BEGIN{sayiA=0; sayiF=0}
{sum=0.3*$2+0.3*$3+0.4*$4 }
sum > 80 { grade=A; sayiA=sayiA+1; print $1  " got A " }
 sum < 40  {grade=F; sayiF=sayiF+1; print $1  " got F "  }
  END { print  "among NR students," sayiA "  got A, and " sayiF  " got F " }
```

30. i)You want to run Temizle.sh every Wednesday and Sunday, at 2:25. $\boxed{\textbf{2}}$
crontab: `25  2  * *  0,3  Temizle.sh`

ii) You want to run Backup.sh at 1:00 first day of every month $\boxed{\textbf{2}}$

crontab: `0   1  1 * * Backup.sh` Write down necessary commands and give configuration details

31. Given turing.txt determine unique words. Remove punctuation such as . : ; ? [ ] ( ) . is. No upper case. You can leave & as is. You can only use filters $\boxed{\textbf{5}}$

```
sed -f Sed.turing turing.txt | tr A-Z a-z | sort -u > ../Answers/31.txt
```
or
```
cat turing.txt | tr A-Z a-z | tr -d ’[:digit:]’  | tr ’[:punct:]’  ’  ’  | tr
’A-Z’ ’a-z’ | tr ’[:space:]’   ’\n’ |  sort -u > ../Answers/31x.txt
```

32. Consider all *.txt files in LabM (on the surface) determine **files** which has lines containing **elif** and **ayse** case insensitive on a line, (under 32.TXT) $\boxed{\textbf{5}}$

```
grep -i elif *.txt | grep -i ayse | awk -F: ’{print $1 }’ |sort -u > ../Answers/32.TXT
```