COMP111: Software Tools

Midterm Exam – Qiang Yang

Fall 2001 (Wed Oct 10, 2001 12:00-14:00)

Student Name: ______________________________

Student Number: ______________________________

Email: ______________________________

Lab Section: ______________________________

Instructions:

1. There are 12 problems worth 100 points total.

2. Check that you have all 7 pages.

3. Close book, close notes, work on your own, and you cannot use any computer.

4. Answer all questions in the space provided. Rough work can be done only on the back pages.

5. Leave all pages stapled together.

6. The examination period will last for **120 minutes.**

7. Stop writing immediately when the time is up.

For Grading Purposes Only:

```
Problems 1________________ / 20
Problems 2   ________________ / 5
Problems 3   ________________ / 5
Problems 4   ________________ / 5
Problems 5   ________________ / 5
Problem  6    ________________ / 5
Problems 7     ________________ / 5
Problems 8   ________________ / 5
Problem  9    ________________ / 8
Problems 10     ________________ / 7
Problems 11   ________________ / 15
Problems 12   ________________ / 15

Total:   ________________ /100
```

1) (20 points) Use a ***single*** command to satisfy your requirement. The command must work ***whatever*** your current working directory and home directory are

1.1. List the last 5 lines of the file `letter1` in the current working directory.
```
    tail  –5 letter1
```
1.2. Append a file `f1` to another file `f2` (assume both files are in the current working directory)
```
    cat f1 >> f2
```
1.3. List the names of all the files in the current directory that begin with a letter and ends with extension .txt
```
    ls [a-zA-Z]*.txt
```
1.4. Delete the file tmp.$$$ in the parent directory of your current working directory
```
    rm ../tmp.\$\$\$
    or, rm '../tmp.$$$'
```
1.5. Copy the whole subdirectory tree of your home directory to a new one called `backup` (assuming backup exists, and both in the current directory).
```
    cp –R ~ backup   (or –r)
```
1.6. Delete the whole subdirectory tree `test` in your home directory
```
    rm –R ~/test (or –r)
```
1.7. Add a user account called `peter` if you are the administrator (root)
```
    useradd peter
```
1.8. Change your own password
```
    passwd
```
1.9. Delete a group called `comp111` if you are the administrator (root).
```
    groupdel comp111
```
1.10. Add `jane` to the group `comp111` if you are the administrator.
```
    usermod -G comp111 jane
```
1.11. List only the information of the group `comp111`
```
    ypcat group | grep comp111
```
1.12. find all lines in the file `names` that do not contain ‘a’, or ‘b’, or ‘c’.
```
   grep –v “[a,b,c]” names
```
1.13. List all the mounted file systems in your UNIX computer
```
    df
```
1.14. List the disk usage statistics in number of blocks for all subdirectories in the root directory
```
    du /
```
1.15 Count the number of unique names in file names.text, assuming the names are listed one per line
```
    Sort names.text | uniq | wc –l
```
1.16. Archive and compress the current directory and put the result in a file called backup.tar.zip
```
    tar cf - . | gzip  > backup.tar.zip
```
1.17. find and list all files in the current directory and all subdirectories whose names start with “COMP”
```
          find . –name “COMP*” –print
```
1.18. copy a directory mydir to another directory yourdir
```
          cp –R mydir yourdir
```
1.19. create a symbolic link snames to a file names
```
          ln –s snames names
```
1.20. include a current working directory in the path variable, and then show its value
```
          set path = (. $path); echo $path
```

2) (5 points) What is wrong with the following command? Show the correct command
$ cat file1 >file2 | sort >file3

```
Answer: cat has two outputs! (>file2 and the pipe) llegal,
can only have one output.
Correct: cat file1 | tee file2 | sort > file3
```

3) (5 points) What is the working directory after executing the following commands? (Assume all the specified directories exist.)

```
$ cd /tmp/tmp/tmp; (cd ..; cd local/bin); \
cd ..; (cd ../tmp); cd .././..;
```

/

4) (5 points) A file called `names` contains the following text:

```
Mr. Bill
Bill Gates
Jackie Chan
President Woo
Bill Gates
Dr. Andrew Horner
President Bill Clinton
```

What is displayed on the screen after executing the following pipeline?

```
$ sort names | grep Bill | uniq | grep G
```

Answer: `Bill Gates`

5) (5 points) Give two ways that you can set the permission of the file pgm so that the owner, group-mates and all other users has only read and execute permission? (Assume that pgm is in the current directory, and your result should not depend on the previous permission settings of pgm).

i) chmod 555 pgm ii) chmod a=rx pgm

6) (5 points) After the following commands are executed, what are the contents of file1, file2, file3, and file4?

```
$ sh
$ ps
PID   TTY      TIME     CMD
12357 pts/1    00:00:00 bash
12395 pts/1    00:00:00 sh
12409 pts/1    00:00:00 ps
$ ps | wc -l > file2
$ cp file2 file1
$ ln file1 file4
$ ln -s file4 file3
$ rm file1
$ echo $$ >> file1
$ echo $/$\$ >> file2
```

```
File 1    File 2    File3     File4

12395    4        4         4
         $/$$
```

7) (5 points) The contents of the following files are:

file1	file2:
a	c
b	d

What is output to the screen, and what are the contents of file1 and file2 after the following command?

```
$(cat file1; head -1 file2) | tee file1; tail -1 <file1 |tee file2
```

Answer:

```
screen:   file1:    file2:
a         a         c
b         b
c         c
c
```

8) (5 points) Write a shell program line that is equivalent to the expression

```
n = (2n+10) * 2;
```

Clearly indicate spaces with the "." character.

Answer: `n=‘expr.\(.2.\*.$n.+.10.\).\*.2‘`

9) (8 points) Write a shell script that displays **all except the last line** of the file /tmp/log.

#!/bin/sh [1]

num_line=` grep -c '.' /tmp/log ` [1] (Use wc –l < /tmp/log also OK, wc –l /tmp/log – 0.5 pt)
num_line=`expr $num_line - 1 ` [1]

head "-$num_line" /tmp/log [1]

10) (7 points) Write a shell program that checks who the user is and how many users are logged in, and prints out the result. Clearly indicate spaces with the "˜." character. Make your output look like the following:

Hi qyang! There are 3 users logged in.

Answer:
```
#!/bin/sh
user=‘whoami‘
numuser=‘who.|.wc.-l‘
echo."Hi $user! There are $numuser users logged in."
```

11) (15 points) Mail merge. You are given a data file called score.txt

cs_abc@stu.ust.hk Lab1: 10 #Lab2: 10 #Lab3: -1 #Lab4: -1
cs_def@stu.ust.hk Lab1: 8 #Lab2: 3 #Lab3: 0 #Lab4: 7

Each line consists of the email address of a student & the fields remained are the marks for the student's assignments (-1 means the student haven't submit the assignment)> each score field is separated by a "#".

You are required to send an email to notify each student the results for all assignments, if there is any assignment not submitted, you should urge the student to submit the assignment. You can use any UNIX commands in this question, but your program is not allowed to create any temporary files during execution (i.e., commands such as echo xxx >> letter is not allowed). (Reminder: you should send email to the full email address)

The following depicts the format required for the output:

Dear cs_abc:

Here are the scores of each assignment
=====================================
Lab1: 10
Lab2: 10
Lab3: -1
Lab4: -1

You have missed 2 assignments, please submit it ASAP

Regards,
Eric

Dear cs_def:

Here are the scores of each assignment
===
Lab1: 8
Lab2: 3
Lab3: 0
Lab4: 7

Regards,
Eric

Please complete the program on the next page to answer this question.
Hints: To extract from the 2nd field to the last field separated by ' ', use " cut -f2- -d' ' "

Question 11 – The skeleton: fill in the blanks

```
#!/bin/sh

# Create a loop that reads the students’ scores line by line
(while read field1 do
  # Separate the user's email from his marks
  email=`echo "$field1" | cut -f1 -d' '`
  marks=`echo "$field1" | cut –f2- -d‘ ‘`

  # Separate the marks for each assignment into lines

  each_marks=`echo "$marks" | tr '#' '\012' `

  # Check how many assignments are not submitted
  # If so, prepare the warning message, if there is no missed
  # assignment, the warning message is " "
  # miss_work = 2 if there are 2 missed assignments

  miss_work=`echo "$each_marks" | grep -c '\-1' | wc -l `

  # Construct warning message

      if [ $miss_work -gt 0 ]
      then
         warning="You Have missed $miss_work assignments, please\
submit it ASAP"
      else
         warning=" "
        fi


  # obtain the name of the student from email
  receiver=` echo "$email" | cut -f1 -d'@' `

  # Send email to the student
  mail $email << EOF

Dear $receiver:

Here are the scores of each assignment
======================================
$each_marks
$warning

Regards,
Eric
EOF
done) < score.txt
```

12) (15 marks) Complete, by filling in the blanks, the following shell script (called "sumnum") that returns the sum of all the numbers specified as command line parameters. (You need to check for the validity of the parameters as numbers.) Your script must support unlimited number of parameters. If the user does not supply any parameter, display the usage summary of your script.

```
#!/bin/sh

if [ $# = 0 ]
then
        echo usage: sumnum number1 number2 ... ;
         exit
fi
sum=0
while [ $1 ]
do
  sum= `expr $sum + $1`
  shift
done
echo $sum
```