COMP111: Software Tools

Midterm Exam – Dickson Chiu

Summer 2001 (Monday July 16, 2001 12:00-14:00)

Student Name: ____

Student Number: ____

Email: ____

Lab Section: ____

Instructions:

1. There are 15 problems worth 100 points total.

2. Check that you have all 10 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:

Page 2: Problems 1-2	______________	/ 22
Page 3: Problems 3-6a	______________	/ 09
Page 4: Problems 6b-8	______________	/ 11
Page 5: Problems 9-10a	______________	/ 11
Page 6: Problems 10b-11	______________	/ 10
Page 7: Problem 12	______________	/ 10
Page 8: Problems 13	--	
Page 9: Problems 13	______________	/ 10
Page 10: Problems 14-15	______________	/ 17
Total:	______________	/100

1) (20 points) Command Identification. You must use a single command to satisfy your requirement and the command must work whatever your current working directory and home directory are.

Requirement	***Command***
List the last line of the file letter1 in the current working directory.	tail –1 letter1
Append a file f1 to another file f2 (assume both files are in the current working directory)	cat f1 >> f2
List the names of all the files in the current directory that begin with a letter and ends with extension .bak	ls [a-zA-Z]*.bak
Delete the file – in the current working directory (i.e., the file with dash as file name).	rm -- - or rm ./-
Delete the file tmp.$$$ in the parent directory of your current working directory	rm ../tmp.\$\$\$ rm '../tmp.$$$'
Copy the whole subdirectory tree of your home directory to a new one called backup (both in the current directory).	cp –R ~ backup (or –r)
Delete the whole subdirectory tree test in your home directory	rm –R ~/test backup (or –r)
Remove a user account called peter if you are the administrator (root)	useradd peter
Change your own password	passwd
Delete a group called comp111 if you are the administrator (root).	groupdel comp111
Add jane to the group comp111 if you are the administrator.	usermod -G comp111 jane
List the information of the group comp111	ypcat group | grep comp111
Eject the CDROM disk from device /dev/cdrom	eject /dev/cdrom
List all the mounted file systems in your UNIX computer	df
List the disk usage statistics in number of blocks for all subdirectories in the root directory	du /
(2 points) Archive the whole subdirectory tree /src/proj in tar format, compress it with gzip, and send it to peter@cs.ust.hk by email (You need to do this with one pipeline command)	gtar cvfz - /src/proj | mail peter@cs.ust.hk tar cvf - /src/proj | gzip –c | mail peter@cs.ust.hk
Extract all files with extension .c from the zip file /tmp/download.zip to your current working directory	unzip /tmp/download.zip “*.c” (without .zip / single quote ok)
(2 points) Run an executable program /etc/dailybatch as background job and collect all its output and error messages into a log file /etc/batchlog	/etc/dailybatch >/etc/batchlog 2>&1 &

2) (2 points) Give 2 ways to create a file /tmp/one with the size of **1 byte** *without using an editor*.
(i) echo > /tmp/one (ii) echo a | grep a > /tmp/one

3) (2 points) Give 2 reasons why you want to compress a directory of files into a single archive file

(i) To save disk space

(ii) To help organising the directories (or any other reasonable explanations)

4) (2 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 .././..;

    /
```

5) (3 points) What is the output of (a) (b) and (c) after running the following commands?

```
    $ sh
    $ cd /var/spool/news
    $ var=pwd

    (a)echo "$var"          _______pwd____________________

    (b)echo '$var'          _______$var___________________

    (c)echo `$var`          _______/var/spool/news_______
```

6) (6 points) File Permission

cs_abc executes the commands below and obtained the following

```
csl2su8:cs_abc:59> pwd
/homes/cs_abc/comp111

csl2su8:cs_abc:59> ls -l comp111
total 3
drwx------    2 cs_abc        512 Jul 10 13:19 test_d1
d-w-------    2 cs_abc        512 Jul 10 13:19 test_d2
d--x------    2 cs_abc        512 Jul 10 13:20 test_d3

csl2su8:cs_abc:93> grep '.' test_d1/test1
#!/bin/sh
ls -lR '..'

csl2su8:cs_abc:67> ls -l test_d3/test3
-rw-------   1 cs_abc           6 Jul 10 13:20 test_d3/test3
```

a) What is the error occurred when this command is executed? Explain your answer (2%)

```
csl2su8:cs_abc:94>  cp test_d3/test3 test_d2/
```

Ans: Permission denied for directory test_d2 [1], as it is not readable [1]

b) Explain why the file ‘test_d3/test3’ cannot be listed using the command ‘ ls -l test_d3/test* ’ (1%)

No file will be listed because test_d3/test3 has no read permission so that the file cannot be search

c) What are the possible permission values (in numbers, e.g. 123) for the file "test_d1/test1" if the user can execute the file, while the group & other's permission are read only? (1%)

Permission values: 744 (-rwxr—r--) or 544 (-r-xr—r--) [Must be rx-able]

d) Change the permission under the directory "Comp111" with one command so that when the user and its group runs the command "ls ./test*/*" under the directory "Comp111", the results can show the files "test_d1/test1" & "test_d3/test3" (2%)

(Marks will be deducted if necessary permission is set to any of the files and directories)

Ans: chmod g+rx test_d1 test_d3 (1 for g+rw, 1 for filenames)
(g+rw can be replaced by ug+rw or 550 or ug=rw, test_d1 test_d3 can be replaced by test*)

7) (2 points) Give 2 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 does not depend on the previous permission settings of `pgm`).

i) chmod 555 pgm

ii) chmod a=rx pgm

8) (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	12395 $/$$	5	5

9) (5 points) The contents of the following files are:

file1	file2
f1	g1
f2	g2

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

```
$ (cat file2; head -1 file1) | tee /dev/stderr ; tail +1 <file2  | tee file1 >> file2
```

screen	file1	file2
g1	g1	g1
g2	g2	g2
g1		g1
g2		g2
f1		
f1		

10) (12 points) Consider the following pipe of commands showing who have logged in a machine:
‘who | cut -d' ' -f1 ‘

Sample Output of `who`	*After Cut*
`anurag  console Oct 18 14:10`	`anurag`
`chiyung ttyp0   Oct 19 09:08  (csz001.cs.ust.hk)`	`chiyung`
`mandy   ttyp1   Oct 8 10:38   (csb004.cs.ust.hk)`	`mandy`
`mandy  ttyp2 Oct 8 10:38  (csb004.cs.ust.hk)`	`mandy`
`kit ttypd Oct 8 10:40 (cssu141.cs.ust.hk)`	`kit`
`kit  ttype Oct 8 10:40 (cssu141.cs.ust.hk)`	`kit`
`cpegkit ttyq0 Oct 19 11:32 (csa050.cs.ust.hk)`	`cpegkit`
`anurag ttyq2 Oct 18 14:10`	`anurag`
`anurag ttyq5 Oct 18 14:10`	`anurag`
`kit   ttyq6 Oct 18 18:53 (csa014.cs.ust.hk)`	`kit`
`csccf  ttyq7 Oct 19 11:33 (csc267.cs.ust.hk)`	`csccf`
`mhyuen  ttyq8 Oct 14 12:17 (csa012.cs.ust.hk)`	`mhyuen`
`mhyuen  ttyq9 Oct 14 12:17 (csa012.cs.ust.hk)`	`mhyuen`

a) (6%) By adding pipes after ‘who | cut -d' ' -f1 | ‘ , accomplish the following task?

Username requirement	*Argument(s)*
Shows the usernames with 5 characters word beginning with "c" or "w".	`grep '^c...w$'`
Shows the usernames which start with "t" and contains no vowel	`grep '^t' | grep -v '[aeiou]'`
Shows the usernames which their first 3 characters are not vowel	`grep '^[^aeiou][^aeiou][^aeiou]'`

Shows the username contains an "l" but not appear as the first or last character	`grep  'l'  |  grep  '^[^l]*[^l]$'`
	`grep '^[^l]*.l.*[^l]$'`

Display how many (distinct) users have logged in the machine, report the number only	`sort | uniq | grep -c '.'`
	`sort | uniq | wc –l`

10b) (6%) The command 'ls -l' can displays the detailed information for the files and directories in the current directory. The format of the result is like this:

<Permission> <#of links> <User> <Group> <file size> <Last modification> <name>

```
-rwxrw-r-x   1  cslcw    cs            874     Jul  1    00:18   testfile.txt
-rwx------    1  cslcw    cs            11450   Jun 24   23:08   testfile.txt.bak
drwx------    4  cslcw    cs            512     Jul  9    19:09   testfolder
```

The number of spaces between each field may not be fixed, but the first '**:**'appear for each line is at the <Last modification> field. However, the filenames/directories may also contain '**:**' or spaces.

i) Report all files that has group read/write permission, report the results to a file named "my_res". The result should be ordered by decreasing file size (report the largest one first).

```
ls -l | sort -nr 4 | grep '^[^d]...rw' > my_res  (sort :0.5pt , grep: 1pt, redirection: 0.5pt)
             -r -k 4 is also OK
```

ii) Using the file generated in (i), report the filename only on the console (you need 2 cuts)

```
cat my_res | cut -f2- -d':' | cut -f2- -f' ' (Using cut correctly 1.5, file input – 0.5)
or cut -f2- -d':' < my_res | cut -f2- -d' '  <== must put my_res at the 1st cut
or  ( cut -f2- -d':' | cut -f2- -d' ' ) < my_res
```

iii) Give one command (with pipes) that can perform b i) & b ii) at the same time. (N.B. using the commands "cp" and "cat" are not allowed)

```
ls -l | sort -nr 4 | grep '^[^d]...rw' | tee my_res | cut -f2- -d':' | cut -f2- -f' ' )
```

Key: (Not marks for not using tee). Use Tee 1 pt, tee at correct position 1pt

11) (4 points) Write a shell script that display **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]
```

12) (10 marks) Assume a vector $x=(x_1,x_2\ldots,x_n)$ is stored in file x in the format that line i contains the number x_i. Write a shell script (called "dotprod") that outputs the dot product of two vectors, whose filenames are specified by command line parameters. A dot product $x . y = x_1*y_1+x_2*y_2+\ldots+x_n*y_n$. You may assume the two files are in correct format. If the two vectors do not have the same number of elements (i.e., different number of lines), display an error message. If the user does not supply any parameter, display the usage summary of your script. (Hint: you may wish to use the command `paste f1 f2` to write lines consisting of the sequentially corresponding lines from `f1` and `f2` separated by TABs to standard output. Then you may pipe it through a while loop for processing…). Example:

`f1`	`f2`	Output of `paste f1 f2`	Output of `dotprod f1 f2`
2	3	2 3	49
3	5	3 5	
4	7	4 7	

```
#!/bin/sh

# Using $# to test the existence of the arguments
if [ "$#" != "2" ]
then
   echo "Usage : dotprod <file 1> <file 2>"
   exit        # need to exit
fi

if [ `wc -l < "$1"` != `wc -l < "$2"` ]
then
   echo "Error : 2 Vectors have different length"
   exit
fi

dot_sum=0

paste "$1" "$2" |
( while read xi yi
do
#       echo "$xi n $yi "
        dot_sum=`expr \( $xi \* $yi + $dot_sum \)`
done
        echo "Result " $dot_sum
)
```

13) (10 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 13 – The skeleton:

```
#!/bin/sh

# Create a loop that read the students’ score line by line
(while…… ……read…… field
…do…………
# Separate the user's email with his scores

email=`echo "……$field……………" | cut -f1 -d' '`
marks=`echo "……$field……………" | 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' `

      # Construct warning message

      if [ $miss_lab -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
```

14) (10 points) Write a shell script (called “`print-r`”) that prints every file with extension `.c`, `.bas`, or `.pas` and compress the file with gzip after printing, for a **whole sub-directory tree**. Make sure to check the item to be printed is really a file. The name of the sub-directory is specified by the first command line parameter of your script and the printer name is the second parameter. If the user does not specify **exactly** 2 parameters in the command line, display a usage summary of your script. Display an appropriate error message if the first parameter is not a valid directory.

```
#!/bin/sh
#  if [ $2 ] not sufficient because more than 2 parameters pass this test
if [ $# != 2 ]
then
   echo print-r <directory-name> <printer-name>
elif [ -d $1 ]
   echo $1 is not a valid directory
else
   for file in `find $1 –name '*.[bp]as' –print` `find $1 –name '*.c' –print`
   do
      if [ -f $file ]
      then
            lpr –P$2 $file
            gzip $file
      fi
   done
fi
```

15) (7 marks) Write a shell script (called “sumnum”) that returns the sum of all the numbers specified as command line parameters. (You need not 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
# for $i ; do ; ... ; done is also good
while [ $1 ]
do
    sum=`expr $sum + $1`
    shift
done
echo $sum
```