COMP111: Software Tools

Midterm Exam – Dickson Chiu

Spring 2001 (March 14, 2001 7pm-9pm)

Student	Name:	
Student	Number:	
Email: _		
Lab Sect	cion:	

Instructions:

- 1. There are 16 problems worth 100 points total.
- 2. Check that you have all 8 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:



1) (18 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 first line of the file letter1 in the	head –1 letter1
current directory.	
List all lines except the first line of the file	tail +2 letter1
letter1 in the current directory.	
List the names of all the files in the current	ls [0-9]*.c
directory that begin with a digit and ends with	
extension .c	
Delete all the files in the current directory with	rm *.bak
an extension . bak	
Copy the whole subdirectory free junk to a new	cp – R junk backup (or – r)
directory)	
Delate the whole subdirectory tree in the	rm Divulthealan (ar r)
current directory	rm – K Junk backup (or – r)
Create a new user peter if you are the	useradd neter
administrator	useradu peter
Change the password of the user peter if you are	nasswd neter
the administrator.	usermod -n newpassword neter
Create a new group students if you are the	orounadd students
administrator.	groupudd studonts
Add peter to the group students if you are	usermod -G students peter
the administrator.	
List the members of the group students	ypcat group grep students
List all the mounted file systems in your UNIX	df
computer	1
List the disk usage in number of blocks for the	du ~
List the content of a file called	aton stuf /two /dl ton as
List the content of a file caned /tmp/dl_tax_gg_in "gginned tar" format	gtar Ztv1/tmp/d1.tar.gz
sand a small to not an (as wet he with the file colled	gzip - d - c/tmp/di.taf.gz taf tvi -
i putit a transformation and its content in your home	mail peter(ω cs.ust.nk < ~/invitation
directory	
compress the subdirectory directory called send	zin_r/tmn/send zin~/send
in you home directory in zin format and store it as	zip -r /tmp/send -/send
/tmp/send_zip_(so that you can send it to	$2 \ln p - 1 / (\ln p / send / send)$
your PC and process with the Winzip utility)	() (
(2 points) Run an executable program	/utils/biob > /tmp/log 2 > & 1 &
/utils/bjob as background job and collect all	
its output and error messages into a log file	
/tmp/log	

- 2) (2 points) Give 2 ways that you can create a file /tmp/000 with a length of 0 bytes. touch /tmp/000 or cp /dev/null /tmp/000 or true > /tmp/000 or echo a | grep b > /tmp/000 or edit the file with any editor and save without typing (etc.) (note: echo > /tmp/000 will give 1 byte)
- 3) (2 points) Give the name of any two text editors under UNIX.
 vi / pico / edit / ed etc.

4) (2 points) What is the working directory after executing the following commands? (Assume all the specified directories exist.)

```
$ cd /course/comp111/tmp/tmp; (cd /usr; cd local/bin); \
  cd ..; (cd ../tmp); cd .././..;
       /course/
5) (4 points) What is the error for the following command?
   $ sort | uniq >file1 <file2
              There are 2 inputs to the uniq
   Rewrite the above in 3 different correct ways.
```

```
sort <file2 | uniq > file1
(sort | uniq ) >file1 <file2
sort file2 | uniq > file1
etc.
```

6) (6 points) File Permission

csdavid executes 'ls -ld /homes/csdavid' and obtained the following

```
drwx--xr-- 2 csdavid
                                512 Oct 19 23:49 /homes/csdavid
```

csdavid executes 'ls -l /homes/csdavid' and obtained the following:

-rwxrr	1	csdavid	512	Oct	20	12:22	data
-rwxr-x-wx	1	csdavid	6022	Oct	20	12:23	marks
-rwxx	1	csdavid	3222	Oct	20	12:23	running
-rwxrr-x	1	csdavid	3221	Oct	20	12:23	run

kwchiu executes the following commands:

a)	lg	-ld /homes/csdavid	
u)	тb		
C)	rm	/homes/csdavid/marks	
e)	/ho	omes/csdavid/run	
h)	ls	-1 /homes/csdavid/marks	
k)	ls	-1 /homes/csdavid/run	
j)	cat	/homes/csdavid/marks	

- b) ls -l /homes/csdavid d)
- /homes/csdavid/running
- ls -1 /homes/csdavid/data f)
- g) ls -l /homes/csdavid/running
- i) cat /homes/csdavid/data

If kwchiu and csdavid are in the same group, which of the above commands will not give error messages.

a, f, g, h, i, j, k

If kwchiu and csdavid are not in the same group, which of the above commands will not give error messages.

а

- 7) (2 points) Give 2 ways that you can set the permission of the file letter1 so that the owner, "group" -mates and all other users has only read and write permission? (Assume that letter1 is in the current directory, and does not depend on the previous permission settings of letter1).
 - i) chmod 666 letter1 ii) chmod a=rw letter1

8) (6 points) After the following commands are executed, what are the contents of file1, file2, file3, and file4?

```
$ sh
$ echo $HOME
/home/dickson
$ echo $USER > file1
$ cp file1 file2
$ ln file1 file3
$ ln -s file3 file4
$ rm file1
$ echo \\ >> file2
$ echo `whoami` > file1
$ cat file2 >> file1
     <u>File 1</u>
                            <u>File 2</u>
                                               <u>File3</u>
                                                                          <u>File4</u>
     dickson
                            dickson
                                                 dickson
                                                                          dickson
     dickson
                            /
     /
```

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

<u>file1</u>	<u>file2</u>
bg1	f1
bg2	f2

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

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

screen	<u>file1</u>	<u>file2</u>
bg1	bg1	bg1
bg2	bg2	bg2
bg1	bg1	
bg2	bg2	
f2		
f2		

10) (4 marks) ls returns the following six files:

\$ ls | more
\$122\$1
aboyisaboy
aisa
bokb
try.grep

(i) Given the following script called getfile:

```
$ cat getfile
#!/bin/sh
ls | grep \^"[$1].*[$1]"\$
$
```

What does the command getfile aboy return?

```
aboyisaboy
aisa
bokb
```

- (ii) Give the command that can remove the file (i.e., the file with dash as file name).
 rm -- or rm ./- or rm * rm any name not in current directory (will give error message but still removes -)
- (iii) Give the command that can remove the file 122, rm 122 1 or rm 122
- 11) (8 marks) Consider the following pipe of commands which shows the user-names: ypcat passwd | cut -d":" -f1 | grep <arguments>

What argument would you use to accomplish the following?

Username requirement	Argument(s)
with "c" or "k" as the first character.	`^[ck]'
with "c" or "k" as the last character and "s" or	`^[ck].*[sz]\$'
"z" as the last character.	
with "e" as the fourth character	`e′
consist of only vowels (i.e., a/e/i/o/u)	`^[aeiou]*\$'
	-v `[^aeiou]'
does not contain any vowels (i.e., without	`^[^aeiou]*\$'
a,e,i,o,u)	-v `[aeiou]'
length 7 and "a" as the fourth character	`^a\$'
contain the char "j" and the substring does not	'^[^j].*j.*[^j]\$'
occupy the first or last character.	
contain "a", "c" and "e", where "a" will first	`a.*c.*e'
appear, followed by "c" and finally "e", e.g.	
janice, oracles, ma_ckyae	

12) (7 points) Write a shell program line that is equivalent to the following C++ line. Clearly indicate spaces with the " ∇ " character.

$$n = (n-4) / 2;$$

$$n = `expr \nabla (\nabla $n \nabla + \nabla 3 \nabla) \nabla / \nabla 2 `$$

If the value of n='7' before execution, what is the new value of n after executing your program line? 1

If the value of n='x' before execution, what is the error upon executing your program line? non-numeric argument

If the value of n='-' before, what is the error upon executing your program line? syntax error / non-numeric argument

13) (7 marks) The output of the command who is in the following format:

<username> <tty> <month> <day> <time> <location>

The separators between fields 1 to 5 are made up of **different** single spaces. Field 6 (location) can be empty, which means that the user is logon from console. Below is an instance of the information source and the corresponding output expected from your program:

a		C
Sampl	a Dutmut	oturha
Sampi	e Ouibui	

Sample Output of who	Your Output		
anurag console Oct 18 14:10	2 csa012.cs.ust.hk		
chiyung ttyp0 Oct 19 09:08 (csz001.cs.ust.hk)	2 csb004.cs.ust.hk		
mandy ttyp1 Oct 8 10:38 (csb004.cs.ust.hk)	2 cssul41.cs.ust.hk		
mandy ttyp2 Oct 8 10:38 (csb004.cs.ust.hk)	1 csa014.cs.ust.hk		
kit ttypd Oct 8 10:40 (cssu141.cs.ust.hk)	1 csa050.cs.ust.hk		
kit ttype Oct 8 10:40 (cssu141.cs.ust.hk)	1 csc267.cs.ust.hk		
cpegkit ttyq0 Oct 19 11:32 (csa050.cs.ust.hk)	1 csz001.cs.ust.hk		
anurag ttyq2 Oct 18 14:10			
anurag ttyq5 Oct 18 14:10			
kit ttyq6 Oct 18 18:53 (csa014.cs.ust.hk)			
csccf ttyq7 Oct 19 11:33 (csc267.cs.ust.hk)			
mhyuen ttyq8 Oct 14 12:17 (csa012.cs.ust.hk)			
mhyuen ttyq9 Oct 14 12:17 (csa012.cs.ust.hk)			

Give a single pipe of commands to count the number of users from each location. You can use any command except awk. (You must not show console logons.)

```
who | tr -s ' ()' ' ' | cut -f6 -d' ' | grep '.' | sort \
| uniq -c | sort -r -k1
who | grep '(' | cut -f2 -d'(' | cut -f1 -d')'| sort | uniq -c | sort -r -k1
```

14) (10 marks) Mail merge. You are provided with the following data file, called candidates

cpwong@cs.ust.hk 12000 leeks@hku.hk 0 piggy@ust.hk 10000 slee@hk.net 0

Each line consists of an email address and the result of the user's job application (a positive number is the amount of scholarship while zero means the application is unsuccessful). You are required to send an acceptance/rejection email to each candidate about the result. 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:

```
Dear cpwong,
I am glad to inform you that your application has been successful.
You will receive $12000 as scholarship.
Regards,
John Smith
Dear leeks,
I regret to inform you that your application has been rejected.
Regards,
John Smith
#!/bin/sh
(while read address amount
do
     name=`echo $address | cut -f1 -d'@' `
      if [ "$amount" != 0 ]
      then
           mail $address <<EOF</pre>
Dear $name,
I am glad to inform you that your application has been successful.
You will receive \$$amount as scholarship.
Regards,
John Smith
EOF
     else
           mail $address <<EOF</pre>
Dear $name,
I regret to inform you that your application has been rejected.
Regards,
John Smith
EOF
     fi
done) < candidates
```

15) (10 marks) Write a shell script (called "print-r") that prints every compressed file with extension .z or .Z or .gz in a whole sub-directory tree. Note that compressed files cannot be printed directly, but you are not allowed to decompress them on the disk or create extra temporary files because of limited hard disk space (so you need to use a pipe). 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 2 parameters, 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 '*.[zZ]' -print` `find $1 -name '*.gz' -print`
   do
      if [ -f $file ]
      then
            gzip -c -d $file | lpr -P$2
      fi
   done
fi
```

16) (7 marks) Write a shell script (called "maxnum") that returns the maximum number of all the numbers specified as command line parameters. You script must support unlimited number of parameters. Display an error message and stops processing if any parameter is not a positive integer. If the user does not supply any parameter, display the usage summary of your script.

```
#!/bin/sh
if [\$\# = 0]
then
        echo usage: maxnum number1 number2 ... ; exit
fi
max=$1
# for $i ; do ; ... ; done is also good
while [ $1 ]
do
# must check $1 > 0, non-numerics -le 0 is true!
# a better ans: `echo $1 | grep -c '^[0-9]*$'` -eq 0 -o $1 -le 0
   if [ $1 -le 0 ]
   then
        echo All arguments must be positive integers ;
                                                          exit
   fi
   if [ $1 -gt $max ]
   then
       max=$1
   fi
   shift
done
echo $max
```