THE HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY

COMP2021: Unix and Script Programming

Midterm Exam Fall 2011

Tuesday 8 Nov 2011, 12:00-13:50

Student Name: _____*key*_____

Student ID: _____

Lab Section: LA1A / LA1B

Instructions:

This is a closed-book, closed-notes examination.

Write your name, student ID, and lab section on this page.

Answer all questions in the space provided.

For Grading Purposes Only:

Problem 1	_____/10
Problem 2	_____/10
Problem 3	_____/5
Problem 4	_____/10
Problem 5	_____/5
Problem 6	_____/6
Problem 7	_____/5
Problem 8	_____/5
Problem 9	_____/7
Problem 10	_____/10
Problem 11	_____/6
Problem 12	_____/7
Problem 13	_____/4
Problem 14	_____/5
Problem 15	_____/5

Total _____/100

1. Unix utilities *[10 marks]*

Circle the UNIX/Linux command(s) that can complete the task correctly. There may be *more than one* method to finish a task. *(1 mark per question, -0.5 mark for each mistake)*

a) To display who is logged on to the system: *Answer: AB*

A. finger
B. who
C. whois
D. logon

b) To delete the file **`textfile`**: *Answer: C*

A. del textfile
B. rem textfile
C. rm textfile
D. cat textfile | rm

c) To display the first 5 lines in **`textfile`**: *Answer: A*

A. head -n5 textfile
B. head 5 textfile
C. more +5 textfile
D. cat -5 textfile

d) To rename a file from **`text1`** to **`text2`**: *Answer: B*

A. ren text1 text2
B. mv text1 text2
C. cat text1 > text2
D. mv text1 > text2

e) To change the file content of **`textfile`** from view A (original) to view B (on screen): *Answer: BD*

View A:

```
bell
apple
1984
cisco
1072
1984
```

View B:

```
1072
1984
apple
bell
cisco
```

A. cat textfile > sort > uniq
B. cat textfile | sort | uniq
C. sort textfile; uniq
D. sort textfile | uniq

f) To output the current date on the screen and into a file **`date.txt`**: *Answer: C*

A. date > date.txt
B. date >> date.txt
C. date | tee date.txt
D. date > date.txt | tee

g) To list out the files in your home directory: *Answer: AD*

A. cd; ls
B. ls $MYHOME
C. cd ~ | ls
D. ls ~

h) To count the number of words in the man page of the command "**`wc`**": *Answer: C*

A. man wc > wc -w
B. wc -w < man wc
C. man wc | wc -w
D. wc -w wc.man

i) To find the appearances of the word **linux** in the file **linuxtext**: *Answer: ABC*

A. cat linuxtext | grep linux
B. grep linux linuxtext
C. grep "linux" linuxtext
D. grep linuxtext linux

j) To copy the file **textfile** from the parent directory "**cse**" to the current directory "**comp2021**": *Answer: BD*

A. cp ../textfile ~
B. cp ../textfile .
C. cp cse/textfile comp2021
D. cd ..; cp textfile comp2021

2. Unix file system and links *[10 marks]*

Look at the following screen output and answer the questions.

```
$ ls -li
total 140
273908807 -rw-r--r-- 1 chuckjee comp2021 128079 Nov 8 11:32 exam
273908808 -rw-r--r-- 1 chuckjee comp2021     15 Nov 8 11:33 lab1
273908809 -rw-r--r-- 2 chuckjee comp2021    202 Nov 8 11:33 lab2
273908809 -rw-r--r-- 2 chuckjee comp2021    202 Nov 8 11:33 lab3
273908811 lrwxrwxrwx 1 chuckjee cs            4 Nov 8 11:36 lab4 -> lab3
403453165 drwxr-xr-x 2 chuckjee comp2021      6 Nov 8 11:31 lectures
542597928 drwxr-xr-x 2 chuckjee comp2021      6 Nov 8 11:37 secret
273908810 lrwxrwxrwx 1 chuckjee cs            8 Nov 8 11:35 slides -> lectures
```

a) How many directories are there? How many links are there? *(2 marks)*

Answer: 2 directories, 3 links (or 4 links, or 8 links)

b) Name all the actual files in the list. *(1 mark)*

Answer: exam, lab1, lab2 (or lab3, but not both)

c) What will happen if this command is issued: "**rm lab3**"? *(2 marks)*

Answer: Only a link to lab3 is removed, so lab4 becomes a broken link. lab2 is not affected.

d) The **comp2021** group includes all the COMP2021 staff (including **chuckjee**). What command should be issued so that **exam** can be edited by all of them, while anyone else cannot see nor edit it? *(2 marks)*

Answer: chmod 660 exam / chmod g+w,o-r exam (plus execute permission is okay)

e) What command should be issued so that **secret** is only accessible by **chuckjee**? *(1 mark)*

Answer: chmod 700 secret / chmod go-rx secret

f) Use a single command without "**;**" to list out only the file names of all files beginning with "lab". *(2 marks)*

*Answer: ls lab**

3. Shell *[5 marks]*

a) Write down the command needed to exactly achieve the following effect: *(2 marks)*

```
$ pwd
/bin
$ 2021
$ pwd
/homes/horner/2021
```

Answer: alias 2021 "cd ~horner/2021"

b) What is the difference between these two commands? *(2 marks)*

```
firefox
firefox&
```

Answer:

*1. With the & symbol, the program firefox is run in the **background**.*

*2. The command prompt is **released** immediately.*

c) After executing "**`man ls`**", you pressed **`Ctrl+Z`** to suspend and suppress it to the background. What can you do to bring it to the foreground again? *(1 mark)*

Answer: Use command fg

4. Shell and Perl programming *[10 marks]*

You have done a "Big and Small" game in the lab. Translate this Perl program into a Shell program. Please write clearly. You may use a symbol ▽ to denote any necessary space.

Perl programming	Shell programming
`#!/usr/local/bin/perl`	`#!/bin/sh`
`srand;`	`# no srand needed`
`while (1)`	`while [ 1 ] # (1 -- while loop)`
`{`	`do`
`$dice = int(rand(6)) + 1;`	`dice=`expr $RANDOM % 6 + 1``
`if ($dice <= 3)`	`if [ $dice -le 3 ] # (1 -- cond.)`
`{`	`then`
`$result="S";`	`result="S" # (.5)`
`} else {`	`else # (1 -- if)`
`$result="B";`	`result="B" # (.5)`
`}`	`fi`
`print "B or S?\n";`	`echo "B or S?"`
`chomp($input = <STDIN>);`	`read input # (1)`
`if ($input ne "B" && $input ne "S")`	`if [ $input != "B" -a $input != "S" ] # (1 -- cond.)`
`{`	`then`
`last;`	`break # (1)`
`}`	`fi`
`print "* Result is $result and your guess is $input\n";`	`echo "* Result is $result and your guess is $input" # (1)`
`if ( $input eq $result )`	`if [ $input = $result ] # (1)`
`{`	`then`
`print "You have won!\n";`	`echo "You have won!"`
`} else {`	`else`
`print "You have lost!\n";`	`echo "You have lost!"`
`}`	`fi`
`}`	`done # (1)`

Errors: "spacing" -1, "$" -1, "brackets" -1

5. Perl basics *[5 marks]*

a) You have an array of 5 numbers. Their values are as follows: *(3 marks)*

```
@num = (10, 12, 6, 5, 20);
```

Complete the Perl program to print on the screen this chart:

```
1: ##########
2: ############
3: ######
4: #####
5: ####################
```

```
#!/usr/local/bin/perl
@num = (10, 12, 6, 5, 20);
for ($i=1; $i<=5; $i++) {
      print "$i_____$num[i-1]_____";
}
```

Answer: print "$i: "."#"x$num[i-1]."\n"; (1 mark for "#"x, 1 mark for \n, 1 mark for syntax)

b) What will be the output of the following Perl program? *(1 mark)*

```
#!/usr/local/bin/perl
@list = (0, "0", 1, 666, " ", "T", "F", "", "True", "FALSE");
for ($i = 0 ; $i < @list; $i++)
{
        print "$i " if (!$list[$i]);
}
print "\n";
```

Answer:

0 1 7 (-0.5 mark for each mistake)

c) What will be the output of the following Perl program? *(1 mark)*

```
#!/usr/local/bin/perl
@list = qw(a b c d);
print "$list[0] x @list" . "\n";
```

Answer:

a x a b c d (-0.5 mark for each mistake)

6. Perl arrays and lists *[6 marks]*

What will be the output of the following Perl program? *(2 marks each)*

a)

```
#!/usr/local/bin/perl5 -w
@numbers = (6,7,8);
@numbers = (1,2,@numbers,10);
print "@numbers\n";
```

Answer:

1 2 6 7 8 10

b)

```
#!/usr/local/bin/perl5 -w
@numbers = (6,7,8);
pop(@numbers);
@numbers = (1,@numbers);
print "@numbers\n";
```

Answer:

1 6 7

c)

```
#!/usr/local/bin/perl5 -w
@nums = (5,6,7);
$a = @nums;
print $a."\n";
```

Answer:

3

7. Perl control flow *[5 marks]*

a) What will be the output of the following Perl program? *(2 marks)*

```
$ cat control.pl
#!/usr/local/bin/perl5 -w
while(1){
        print "Waiting...\n";
   do{
                print "Wakeup [yes/no]? ";
                chomp($resp = <STDIN>);
                if($resp eq "yes"){
                last;
             }
          }while(1);
        print "Sleeping...\n";
}
$ control.pl
Waiting...
Wakeup [yes/no]? no


Wakeup [yes/no]? yes
```

(If you think the output is blank, please fill in [blank].)

Answer:
[blank]
[blank]

b) Rewrite the same code replacing the while loop with redo-last. *(3 marks)*

Answer (You should keep the original function of the code below by only one redo-last and the while loop is required to be replaced. However, you will still get 2 marks if you also rewrite the do-while loop.):

```
#!/usr/local/bin/perl5 -w
{
      print "Waiting...\n";
      do{
            print "Wakeup [yes/no]? ";
            chomp($resp = <STDIN>);
            if($resp eq "yes"){
            last;
         }
       } while(1);
      print "Sleeping...\n";
      redo;
}
```

8. Perl I/O *[5 marks]*

a) What will be the output of the following Perl program? *(2 marks)*

```
$ cat line1.pl
#!/usr/local/bin/perl5 -w
while(defined(<>)) {
        print "Loop 1\n";
        print <>;
}
while(<>) {
        print "Loop 2\n";
        print "$_";
}
$ cat file1
1
2
$ line1.pl file1
```

Answer: Loop 1 #1 marks
2 #2 marks

b) What will be the output of the following Perl program? *(1 mark)*

```
$ cat while.pl
#!/usr/local/bin/perl5 -w
while(<STDIN>) {
      print $_;
}
$ cat file1
Bill
Gates
Steve
Jobs
$ while.pl file1
```

(If you think the output is blank, please fill in [blank].)

Answer: [blank]

c) Fill in the blank according to result of **`printf1.pl`**. *(2 mark)*

```
$ printf1.pl
1234567890 12345 1234567.12
$ cat printf1.pl
#!/usr/local/bin/perl5 -w
$s = "1234567890";
$n = 12345;
$real = 1234567.123;

printf _____;
```

Answer: "%s %5d %10.2f\n", $s, $n, $real

9. Perl file I/O *[7 marks]*

Read the below results and fill in the blanks in the Perl program.

```
$ cat in
Bill
Gates
Steve
Jobs
$ filehandles.pl in out
There are 4 lines in 'in'
$ cat out
1: Bill
2: Gates
3: Steve
4: Jobs
```

filehandles.pl

```
#!/usr/local/bin/perl5 -w

$input = _________________; ##(Hint: the file names are supposed to be arbitrary.)
$output = _________________;
$lineno = 0;

open(INPUT, $input) || die "cannot open $input \n";
open(OUTPUT, _________________) || _________________ "cannot open $output \n";

while(<INPUT>)
{
        _________________;
        _________________;
        print _________________ "$lineno: $_\n";
}

print "There are $lineno lines in '$input'\n";

_________________;
_________________;
```

Answer:

$ARGV[0];	*#0.5 marks*
$ARGV[1];	*#0.5 marks*
">$output" #or ">>$output"	*#1 mark*
die;	*#1 mark*
$lineno++;	*#1 mark*
chomp;	*#1 mark ('chomp;' and '$lineno++' can be reversed)*
OUTPUT;	*#1 mark*
close(INPUT);	*#0.5 marks*
close(OUTPUT);	*#0.5 marks*

10. Perl functions *[10 marks]*

The following Perl program '**sub.pl**' is to check whether a 3x3 matrix has all the digits 1-9. The possible numbers are {0,1,2,3,4,5,6,7,8,9}, where 0 means this grid is not filled. If any number is missing or appears more than once, show the number(s) on the screen.

Example 1

Given the 3x3 matrix:

3	*5*	*9*
7	*6*	*2*
8	*4*	*1*

The output is:

```
$ sub.pl
Everything is OK
```

Example 2

Given the 3x3 matrix:

3	*5*	*9*
7	*6*	*2*
0	*4*	*9*

The output is:

```
$ sub.pl
Missing the following element(s):
8
1
The following element(s) appear more than once:
9
```

The program has been partially finished. The return value of sub function '**load**' is a random 3x3 matrix, of which each grid can contain any number in {0,1,2,3,4,5,6,7,8,9}.

```
$ cat sub.pl
#!/usr/local/bin/perl5 -w
use strict;
sub check_block;
sub load;
my @matrix = load();
check_block(@matrix);
```

Please finish the sub function 'check_block'

(Hint: recall the Sudoku lab session which may help!)

```
sub check_block
{
     ____________; #1 mark
     @dlist = ____________;  #1 mark
     my @tempmark = (0,0,0,0,0,0,0,0,0);
     my @missing;
     my @duplicate;
     my $flag=0;

     for (my $i=0; $i<3; $i++) #3 marks for this loop
     {
        for (my $j=0; $j<3; $j++)
        {
            ____________________________________
            ____________________________________
            ____________________________________
            ____________________________________
        }
     }

     for (my $i=0; $i<9; $i++) #3 marks for this loop
     {
           if ($tempmark[$i] ____________)
           {
               $flag=1;
               push(@missing, ____________);
           }
           elsif( ____________________ )
           {
               push(@duplicate, $i+1);
```

```
            }
        }

        if ($flag == 1) #2 marks for this 'if & else'
        {
            print "Missing the following elements:\n";
            while( ___________ )
            {
                print pop(@missing)."\n";
            }
            if( ___________ )
            {
               print "\nThe following element(s) appear more than once:\n";
            }
            while( ___________ )
            {
                print pop(@duplicate)."\n";
            }
            return;
        }
        else
        {
            print "Everything is OK\n";
            return;
        }
}
```

Answer:

1. *my @dlist; #1 mark*
2. *@_; #1 mark*
3. *if ($dlist[$i][$j]>0) #1 mark*
   {
   $tempmark[$dlist[$i][$j]-1]+=1; #2 marks
   }
4. *== 0 #1 mark*
5. *$i+1 #1 mark*
6. *$tempmark[$i]>1 #1 mark*
7. *@missing @duplicate @duplicate # 2 mark*

11. Perl hashes *[6 marks]*

a) What is the output of the following Perl program "**hash.pl**"? *(2 marks)*

```
$ cat hash.pl
#!/usr/local/bin/perl5 -w
$steve = 1;
$steve{"jobs"} = "apple";
@steve = qw(1 2 3);
print "@steve\n";
print $steve{"jobs"}."\n";
print $steve."\n";
```

Answer:

1 2 3

apple

1

b) Write a program that reads a series of words with one word per line until end-of-line, then prints a summary of how many times each word was seen. You are required to use hashes in a significant and effective way. *(4 marks)*

Answer:

```
#!/usr/local/bin/perl5 -w

chomp(@words = <STDIN>);
foreach $word (@words)
{
    $count{$word} += 1;
}
foreach $word (keys %count)
{
    print "$word was seen $count{$word} times\n";
}
```

12. Regular expressions *[7 marks]*

Identify the strings that CAN be matched by the given regular expression patterns. There may be more than one answer. *(-1 for each wrong answer)*

a) **`/^[]aeiouAEIOU]?\D/`**

Answers: ACDEF (1 mark for each correct answer)

A. A20]

B. 12

C. sthlm

D. O9

E. e]4

F. Z[6

b). **`/(a|b)(.)[aeiou]\1/`**

Answers: BD (1 mark for each correct answer)

A. a!a!

B. a-ia_

C. _6a

D. b!ab

E. b!o!b

F. !ae

13. Perl directory access & process management *[4 marks]*

Write a program to change to a directory specified by the user, then list the names of the files in alphabetical order. (Don't show a list if the directory change did not succeed: merely warn the user). You should show the current time of the operation in the very beginning.

e.g.: '**/home**' contains '**comp4622**', '**lib**', '**lost+found**', '**vmware**'

```
$ da.pl
Sat Oct 29 17:06:13 HKT 2011
Where to?
/home  *** this line is user input ***
comp4622
lib
lost+found
vmware
```

Answer:

```
#!/usr/local/bin/perl5 -w

system("date"); # 1 mark

print "Where to?\n";
chomp($newdir = <STDIN>);
chdir($newdir || die "Cannot chdir to $newdir.\n"); # 1mark
foreach (<*>) # 1mark
{
    print "$_\n";
}

#Or
system ("ls | sort"); # 1mark

# 1 mark for correctness of program.
```

14. HTML *[5 marks]*

Mark and correct the errors in this HTML code so that the code is valid and it will be displayed as expected in any browser.


```
<html>

<head>Linking to HKUST</head>

<body>

<h1>HKUST</h1>

<img href="logo.gif" height="60px" width="246px">

<p>The HKUST website is at <link href="http://www.ust.hk">

http://www.ust.hk</link></p>

</body>

<html>
```

Answer:

<html>

<head>

<title>*Linking to HKUST****</title>***

</head>

<body>

<h1>HKUST</h1>

<img ***src****="logo.gif" height="60" width="246">*

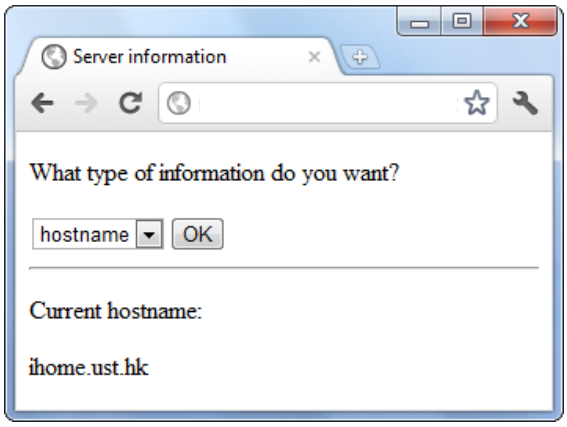
* The HKUST website is at <**a** href="http://www.ust.hk">http://www.ust.hk</**a**> *

</body>

</html>

15. CGI *[5 marks]*

Fill in the blanks to complete this CGI program. Users should be able to choose the options from a pop up menu.




```
#!/usr/local/bin/perl
use CGI qw(:standard);
print header();
print start_html("Server information");
print p("What type of information do you want?");
@choices = qw(date hostname pwd);
print start_form();
print _____;
print _____("OK");
print end_form(), hr();
if (param()) {
        $command = _____;
        $output = `$command`;
        print p("_____");
        print p($output);
}
print end_html();
```

Answer:

```
#!/usr/local/bin/perl
use CGI qw(:standard);
print header();
print start_html("Server information");
print p("What type of information do you want?");
@choices = qw(date hostname pwd);
print start_form();
print popup_menu("type", \@choices);   # 1 mark for correct popup_menu syntax, 1 mark for \@choices
print submit("OK");
print end_form(), hr();
if (param()) {
      $command = param("type");
      $output = `$command`;
      print p("Current $command:");
      print p($output);
}
print end_html();
```