ENGG1340 Computer Programming II

**Module 1 Checkpoint Exercise**

Name: Adrio Alimin

University ID: 3035603978

**Instructions:**

For each single question or each group of questions in the Checkpoint exercise, please type your answer right after the question in this Word document. Please refer to the example below.

Checkpoint 0:

What is the meaning of the command “date”?

Ans: The “date” command prints the current date of the current machine

**Checkpoint 1.1 (Please submit your answer to Moodle)**

Now, let’s try to answer the following questions. Although you haven’t been taught the meaning of the following commands, you can display the manual page of these commands and learn their meanings by yourself

1. Why do we need to learn command line although we can use a GUI to control a computer?

Ans:

* 1. A lot of programming tools are only available through command line.
  2. Command lines allow us to be more specific with our instructions.
  3. GUIs are relatively heavy (processing, memory, and graphics wise) and thus not optimal for large iterations of a repeated task.
  4. Command lines allow us to reuse the same instruction multiple times more easily.

1. What is the meaning of **ls -t**?

Ans:

* 1. Display all files & directories in current directory, sorted by time modified (default behavior: most recently modified file comes first).

1. What is the meaning of the command **pwd**?

Ans:

* 1. Display the absolute path of the current directory.

1. What is the meaning of the command **rm**?
   1. Remove files of non-directory type from a directory.
2. What is the meaning of the command **mv**?

Ans:

* 1. Move or rename files/directories.

1. Suppose that the **fileA** does not exist in your present working directory, what is the meaning of the command **touch fileA**?

Ans:

* 1. Create a new file ‘fileA’ in the current working directory (assuming that a file with that name does not exist already).

1. What is the meaning of the command **tar**?
   1. Create and manipulate archive files (i.e. compressed files in formats such as .7z, .rar, or .zip).
2. What is the command for creating an archive **files.tar** from two files named **fileA** and **fileB**?
   1. tar -cvf files.tar fileA fileB

**Checkpoint 1.2a (Please submit your answer to Moodle)**

Assume we have logged in Ubuntu and started a bash shell. The current directory is the home directory, i.e., ~ . We want to perform the following tasks sequentially. For each of the tasks below, please state the shell command used to perform it.

1. Create a new subdirectory “***assignments***” under ~.
   1. mkdir -p assignments
2. Create a new subdirectory “***assignment 1***” under “***assignments***”. (Note that we are creating one subdirectory “***assignment 1***” but not two subdirectories “assignment” and “1” )
   1. mkdir -p “assignments/assignment 1”
3. Remove the directory “***assignments***” and all its subdirectories.
   1. rm -r assignments

**Checkpoint 1.2b (Please submit your answer to Moodle)**

1. Use SSH to login to **i.cs.hku.hk** with your CS account. The */public\_html* directory is your web directory.

If we create a file *hello.txt* under */public\_html*:

|  |
| --- |
| $ cd ~/public\_html |
| $ vi hello.txt |

With the following content

|  |
| --- |
| Hello, my name is Kit. |

We can browser the *hello.txt* using the following URL:

**http://www.cs.hku.hk/~*[yourCSID]*/hello.txt** where *[yourCSID]* is your CS account ID. (“~” should be kept)

* 1. Can we view the file *hello.txt* through a browser if we execute the following command under ~**/public\_html**? Explain your answer.

|  |
| --- |
| $chmod o-x hello.txt |

Yes. With the above chmod command, we merely remove an other user’s permission to execute the .txt file (which does nothing to affect their read permission).

* 1. What will happen if we execute the following command under **~/public\_html**? Explain your answer.

|  |
| --- |
| $chmod o-r hello.txt |

Other (non-owner) users will no longer be able to read our text file.

* 1. What is the command to set *hello.txt* to be able to be browsed by everyone, but not to altered nor executed by everyone?
     1. chmod o+r hello.txt

1. **[Self-learning question]** - You need to search for the information on the Internet to answer this question.

There is another way to modify the permission, which is called the Absolute mode.

* 1. Explain the meaning of chmod 666 hello.txt
     1. The chmod command makes changes to a file’s file mode bits. There are at least two ways to go about this. Using a symbolic representation with characters of the alphabet and mathematical operators (seen above), or changing the bits directly and absolutely with new bits (using octal; i.e. base-8 numbers). The command above changes hello.txt’s file mode to ‘all users can read/write but cannot execute the file/folder.
  2. Explain the meaning of chmod 700 hello.txt
     1. Answer above holds. But hello.txt’s file mode is now set to ‘(only) owner can read, write, and execute’
  3. What is the chmod command, in absolute mode, to set the following permission for *hello.txt*?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User** **permissions** | | | **Group Permission** | | | **Other permission** | | |
| r | w | x | - | w | - | r | - | - |

Chmod 724 hello.txt

* 1. The administrator says that “One does not simply 777 their entire server”, explain what the problem is if we chmod 777 for all the files.
     1. You make it so that every file in the server can be read, written, and executed by any user.

**Checkpoint 1.3 (Please submit your answer to Moodle)**

Now you may have a doubt: I understand how **diff** works, but why the output is claimed to be the difference between the two files?

Consider the two files below:

|  |  |
| --- | --- |
| $cat question1A  Apple  Boy  Cat  Dog  Egg | $cat question1B  Boy  Cat  Egg |

Note that file **question1B** is created by removing “Apple” and “Dog” from the file **question1A**.

1. What will be the output if we execute the following command (Please try to think about the output before trying it in the shell)? Please explain your answer.

|  |
| --- |
| $diff question1A question1B |

1d0

< Apple

4d2

< Dog

1. What will be the output if we execute the following command (Please try to think about the output before trying it in the shell)? Please explain your answer.

|  |
| --- |
| $diff question1B question1A |

0a1

> Apple

2a4

> Dog

Checkpoint 1.4 (Please submit your answer to Moodle)

This is a challenging exercise! You need to understand the shell commands and the techniques introduced in the previous sections to work on this task.

The following C++ program *gen4.cpp* reads in a 4-character string from the input and generates all possible permutations from the 4 characters.

|  |
| --- |
| //gen4.cpp  #include <iostream>  #include <string>  int main() {  std::string s;  std::cin >> s;  for (int i = 0; i < s.length(); i++) {  for (int j = 0; j < s.length(); j++) {  for (int k = 0; k < s.length(); k++) {  for (int l = 0; l < s.length(); l++) {  if (i != j && i != k && i != l && j != l && j != k && k != l) {  std::cout << s[i] << s[j] << s[k] << s[l] << std::endl;  }  }  }  }  }  return 0;  } |

To compile *gen4.cpp*

|  |
| --- |
| $ g++ gen4.cpp -o gen4 |

The input of the program should be stored in the file *gen4\_input.txt* with the following content.

lopo

***gen4\_input.txt***

1. Give **ONE** command to run the *gen4* with *gen4\_input.txt* as input and redirect the result to a file named *gen4\_output.txt*.

Hints:

|  |
| --- |
| $ cat gen\_input4.txt | ./gen4 > gen\_output4.txt  $ cat gen4\_output.txt  lopo  loop  lpoo  lpoo  …  $ wc gen4\_output.txt  24 24 120 gen4\_output.txt |

\*The output file should contain all permutations of the letters ‘l’, ‘o’, ’p’ and ‘o’. There should be 24 permutations in total.

1. Give **ONE** command to sort the words in *gen4\_output.txt* in alphabetical order, and then also remove the adjacent duplicate lines and finally store the result in a file named *sort\_uniq.txt*.

Hints: Consider the command **uniq**

|  |
| --- |
| $ sort gen\_output4.txt | uniq > sort\_uniq.txt  $ cat sort\_uniq.txt.  loop  lopo  lpoo  olop  olpo  oolp  oopl  oplo  opol  ploo  polo  pool  $ wc sort\_uniq.txt  12 12 60 sort\_uniq.txt |

\*There should be 12 unique words total.

1. Give **ONE** command to check the spelling in *sort\_uniq.txt* and store the misspelled words into another file named *misspell.txt*.

spell sort\_uniq.txt > misspell.txt

1. Now *sort\_uniq.txt* contains all distinct generated words, and *misspell.txt* contains all misspelled words. The differences between the two files are the meaningful 4-character words. Give **ONE** command to return the correctly spelled words as shown below:

|  |
| --- |
| $ diff sort\_uniq.txt misspell.txt | grep "<"  < loop  < polo  < pool |

Hints: Consider the command **diff** and **grep**.

Checkpoint 1.5 (Please submit your answer to Moodle.)

Consider the file *question1.txt (****Error! Reference source not found.****)*.

2011111111,John,M,98

2011222222,Marry,F,85

2011333333,Sally,F,85

2012111111,Kit,M,86

2012222222,Ben,M,97

2012333333,Smitty,F,92

2012444444,Jolly,F,93

2012555555,Ken,M,100

Figure 1 question1.txt

1. Give ONE command to return the lines that contain the record of Kit

Hints:

|  |
| --- |
| $ grep "Kit" question1.txt  2012111111,Kit,M,86 |

1. Give ONE command to find the students with UID begin with “**2012**” (i.e., To find the lines that begin with 2012)

Hints:

|  |
| --- |
| $ grep "^2012" question1.txt  2012111111,Kit,M,86  2012222222,Ben,M,97  2012333333,Smitty,F,92  2012444444,Jolly,F,93  2012555555,Ken,M,100 |

1. Give ONE command to return the lines that contain the record of the students who are both:

* UID start at **2012**, and
* Name starts with the characters **J** or **S**

Hints:

|  |
| --- |
| $ grep "2012.\*,[JS]" question1.txt  2012333333,Smitty,F,92  2012444444,Jolly,F,93 |