

CS507 Computing Foundation for Computational Science –HW 1

Unix Commands

Min Long

Instruction

Due time: Sept 16, 11:59PM.

Submission command: `submit minlong CS507 HW1`

Instruction to electronic submission: <http://cs.boisestate.edu/~cs221/SubmissionProcedure.html>

- The written assignment can be done in a pure text format (*.txt, for example, problem1.txt) on Onyx.
- The programming assignment should be presented with source codes.
- Each problem should have its own working directory, such as HW2/prob1, HW2/prob2 ... For example, the following table shows the structure of HW1 from the user “student1” and how to submit HW to us through Onyx

```
1 [student1@onyx:HW1]$ ls -l
2 drwxr-x---. 2 student1 Students 13 Sep 19 2021 prob1
3 drwxr-xr-x. 3 student1 Students 14 Sep 19 2021 prob2
4 [student1@onyx:HW1]$ cd prob1/
5 [minlong@onyx:prob1]$ ls
6 -rw-r-----. 1 student1 Students 943 Sep 19 2021 problem1.txt
7 $ cd ..
8 [student1@onyx:HW1]$ pwd
9 /home/student1/CS507/HW1
10 [student1@onyx:HW1]$ submit minlong CS507 HW1
```

Listing 1: A sample structure of homework and submission procedure.

- Your source codes (if any) must compile and run on Onyx.
- Documentation is important and proper comments are expected in your source code.
 - comments giving description of: purpose, parameters, and return value if applicable
 - other comments where clarification of source code is needed
 - proper and consistent indentation
 - proper structure and modularity

Don’t ask us or your classmates directly for solutions (it happened); just try as much as possible. Be patient and enjoy coding!

Written Problems

Hint: If needed, you can type those commands on Onyx as tests to help you figure out the solutions.

P1. (10 pts) Explain the results of the following command sequences.

```
1 ls | wc -l
2 rm ???
3 who | wc -l
4 mv progs/* /users/steve/backup
5 ls *.c | wc -l
6 who | sort
7 cd; pwd
8 cp memol ..
9 date +%T
10 plotdata 2>errors &
```

Listing 2: Unix/Linux Commands. Assume “plotdata” is an executable file.

P2. (10 pts) Suppose that your HOME directory is /users/steve, which contains a directory of “documents”. In this directory, there are “letters”, “proposals” and “memos” directories.

```
1 [steve@onyx:documents]$ ls
2 letters  memos  proposals
```

Listing 3: Example of Steve’s directories.

Assuming that you just logged in to the system and executed the following commands:

```
1 $ docs=/users/steve/documents
2 $ progs=/users/steve/programs
3 $ let=$docs/letters
4 $ prop=$docs/proposals
```

Listing 4: Setting Steve’s customized environment variables.

Write the commands in terms of these variables to:

- (a) List the contents of the documents directory.
- (b) Copy all files from the letters directory to the proposals directory.
- (c) Move all files from letters directory to the current directory.
- (d) Count the number of files in the memos directory.

What would be the effect of the following commands?

```
1 ls $let/..
2 cat $prop/sys.A >> $let/no.JSK
3 echo $let/*
4 cp $let/no.JSK $progs
```

P3. (5 pts) What do you think would happen if you invoked the following command on the shell? Why?

```
1 cat infile | sort | wc -l > infile
```

P4. (5 pts) Which of the following are valid variable names? Why?

```
1  XxXxXx
2  HOMEDIR
3  file_name
4  x09
5  file1
6  Slimit
```

P5. (10 pts) “echo” is a very useful Unix/Linux command that outputs the strings that are passed to it as arguments. Given the following assignments:

```
1  $ x=*
2  $ y=?
3  $ z='one two three'
4  $ now=$( date )
5  $ symbol='>'
```

and these files in your current directory:

```
1  $ echo *
2  programs documents games homeworks
```

What will the output be from the following commands?

```
1  echo *** error ***
2  echo 'Is 5 * 4 > 18 ?'
3  echo $x
4  echo What is your name?
5  echo $y
6  echo "$y"
7  echo \*\*\*
8  echo $z | wc -l
9  echo \$$symbol
10 echo hello $symbol out
```