Cmpt 214
Term 1 (Fall), 2016/17

Assignment #1

Out: September 27, 2016

Due: 23:55, October 11, 2016

Solve the following questions. A portion of your m
ark for the assignment will be for correctly follo
wing the directions. Each question specifies the s
ystem that is to be used to complete that question
..

This question can be completed on either tuxworld
or on an ismac computer. The solution should be th
e same.

Suppose you have a file named myfile in your curre
nt working directory. List three different non-err
oneous commands you could use to view the file myf
ile using the less command. The three commands mus
t exemplify three different types or forms of comm
ands, and the output from less must be made to you
r virtual terminal (and not to a file or piped to
another process). Remember that the context for th
is question is LINUX on tuxworld.

For the purposes of this question, the "form or typ
e of command" is determined by:

whether one (child) process is created to perform
the command (this is one possibility) or multiple
processes (children) are created to perform the co
mmand (the other possibility);
the presence (one possibility) or lack (the other p
ossibility) of pipes in the command;
whether or not less(1) gets its input from a file
specified as an argument (one possibility) or from
its stdin (the other possibility).
If a set of choices for the above possibilities ar

e different for one command than they are for anot
her command, then those two commands are "of a dif
ferent type". However, the options given to expand
(1) are not used to determine the "type of command
" in this question.

Also for the purposes of this question,

when less(1) executes according to your command, i
ts standard output must be (bound to) your virtual
 terminal (in which you are executing the command);
your commands cannot use command substitution subs
hells, or (the built-in command) exec, if you happ
en to already know what any of those are);
there can be only one instance of less in the pipel
ine;
whether or not the commands execute in the foregro
und or background is irrelevant to this question;
the settings of environment variables used by less(
1) are irrelevant to this question.
Remember to consult the man page for less(1) to fi
nd out more information about the command.

Submit a file q1_solution.txt containing your three
 commands.

Note that on tuxworld, there are two programs for
perusing a file: a simpler one named more, and a m
ore feature-full one named less. This is different
 than the situation on BSD UNIX systems as exhibit
ed, for example, by Mac OS X where more(1) and les
s(1) are exactly the same program.
The context for this question is Mac OS X. Therefo
re, complete this question on an ismac computer.

There is no option to the ls command telling it to
 only output the names of directories. For example
, suppose that, given the current working director
y of a user, ls behaves as follows:

```
bash-3.2$ ls -l
total 1
```

```
-rw-r--r--   1 kusalik  cmpt214  2619 13 Oct  2012
LatePenalty.html
drwxr-xr-x  10 kusalik  cmpt214   340  7 Nov  2012
assignment_1
drwxr-xr-x  12 kusalik  cmpt214   408  4 Nov  2012
assignment_2
drwxr-xr-x  31 kusalik  cmpt214  1054 20 Nov  2012
assignment_3
drwxr-xr-x  20 kusalik  cmpt214   680  4 Dec  2012
assignment_4
-rw-r--r--   1 kusalik  cmpt214   830 14 Oct  2012
external_documentation.txt
-rw-r--r--@  1 kusalik  cmpt214  9368 23 Jan  2014
extra questions.txt
drwxr-xr-x   2 kusalik  cmpt214    68 24 Sep 14:01
test this
-rw-r--r--@  1 kusalik  cmpt214  2525  1 Dec  2012
testing_documentation.html
bash-3.2$ ls
LatePenalty.html        external_documentation.txt
assignment_1            extra questions.txt
assignment_2            test this
assignment_3            testing_documentation.html
assignment_4
```

Write a UNIX command pipeline that will result in
just the directories being listed in the output th
at originates with ls. For example, in the situati
on described above, the pipeline will produce

```
assignment_1  assignment_2  assignment_3  assignmen
t_4  test this
```

As another example, suppose the current working di
rectory is the root directory of a particular Mac
OS X system and ls produces the following output:

```
bash-3.2$ cd /
bash-3.2$ ls
Applications            etc
Archive             hide
Library             home
Network             installer.failurerequests
```

```
Previous Systems        net
System              opt
Users               private
Volumes             sbin
bin             tmp
cores               usr
dev             var
bash-3.2$ ls -l
total 1
drwxrwxr-x+ 123 root  admin  4182 25 Jul 06:11 Appl
ications
drwxrwxrwx    6 root  wheel   204 17 Feb  2008 Arch
ive
drwxr-xr-x+  78 root  wheel  2652 14 Feb  2016 Libr
ary
drwxr-xr-x@   2 root  wheel    68  9 Sep  2014 Netw
ork
drwxrwxr-x   10 root  wheel   340 12 Mar  2014 Prev
ious Systems
drwxr-xr-x+   4 root  wheel   136 30 Jul  2015 Syst
em
drwxr-xr-x   11 root  admin   374 14 Feb  2016 User
s
drwxrwxrwt@   4 root  admin   136 24 Sep 10:05 Volu
mes
drwxr-xr-x@  39 root  wheel  1326 22 Aug  2015 bin
drwxrwxr-t@   2 root  admin    68  9 Sep  2014 core
s
dr-xr-xr-x    3 root  wheel  4360  2 Sep 06:50 dev
lrwxr-xr-x@   1 root  wheel    11 30 Jul  2015 etc
-> private/etc
drwxr-xr-x    2 root  wheel    68 30 Jul  2015 hide

dr-xr-xr-x    2 root  wheel     1 24 Sep 10:10 home

-rw-r--r--@   1 root  wheel   313  1 Oct  2014 inst
aller.failurerequests
dr-xr-xr-x    2 root  wheel     1 24 Sep 10:10 net
drwxr-xr-x@   4 root  wheel   136 12 Mar  2014 opt
drwxr-xr-x@   6 root  wheel   204 30 Jul  2015 priv
ate
drwxr-xr-x@  59 root  wheel  2006 22 Aug  2015 sbin
```

```
lrwxr-xr-x@   1 root  wheel    11 30 Jul  2015 tmp
-> private/tmp
drwxr-xr-x@  15 root  wheel   510  5 Nov  2015 usr
lrwxr-xr-x@   1 root  wheel    11 30 Jul  2015 var
-> private/var
```

In this situation, your pipeline would produce

```
Applications      System            dev
   private
Archive           Users             hide
   sbin
Library           Volumes           home
   usr
Network           bin               net

Previous Systems  cores             opt
```

However, if the current working directory contains no subdirectories, your pipeline will produce nothing or else a blank line as output.
The number of columns will be determined automatically by one of the programs in your program.

Place your pipeline in a file q2_answer.txt. Test your pipeline and place an annotated test log in a file q2_log.txt. Submit these two files.

Hints:

Use the UNIX command "rs" (with no options or arguments) in your pipeline. Make sure to check the man page for rs to find out what it does. rs(1) is not available on LINUX. If you use the -t option to rs, your output should still be placed in columns but your pipeline may not correctly handle file names with spaces in them.

Make sure to test your pipeline in directories containing subdirectories with various ownership, and in particular, variation in the length of user or group names.

In pursuing this problem, we suggest that you adhe
re to the process of incremental software developm
ent, successively building up the command, verifyi
ng that each successive addition works with a vari
ety of input, and gradually building towards the f
inal solution.

The tee command is sometimes useful for testing an
d verification of pipelines, by allowing you to re
cord the output of intermediate points in the pipe
line, while still providing that output on to late
r commands.

The fragment of Procedural C++ code in file q3_ini
tial.cc contains several violations of the program
ming style guidelines given in class. Identify as
many of these violations as you can, and then rewr
ite the code fragment using better programming sty
le. Consider for improvement not only the executab
le statements, but the comments as well. Do not ch
ange any of the program logic; i.e. the overall co
ntrol flow and design should remain the same. Plac
e your corrected or improved program in a file q3_
final.cc.

In a file name d q3_solution.txt described all the
 style violations you identified. Make each descri
ption concise, but be clear about what lines or st
atements are being referenced or discussed. If you
 find multiple instances of the same type of probl
em, report it in general terms just once.

Submit your q3_solution.txt and q3_final.cc.

Hint: how might "-x option to more(1) or less(1) be
 useful in this question?

This question can be completed on either tuxworld
or on an ismac computer. The solution should be th
e same.

In class, we used a construction such as

```
egrep '\<[0-9]{1,2}\>' <<< 0
egrep -v '\<[0-9]{1,2}\>' <<< 199
```
to test regular expressions given to grep(1) (and egrep). Suppose that you are using an old version of bash(1) wherein the construct <<< does not exis t. Another way to perform the same test, but with a different type of command would be
```
egrep '\<[0-9]{1,2}\>' << EOF
0
EOF
```
and
```
egrep -v '\<[0-9]{1,2}\>' << EOF
199
EOF
```
The above makes use of a construct called a "here document". (See pages 450 to 451 of the Sobell tex t for an explanation of "here documents".) Give ye t another type of shell command that will perform the same test as
```
egrep '\<[0-9]{1,2}\>' <<< 0
```
but this time not using <<< , << , <, or a file. T hat is, egrep(1) will test an input string provide d on its standard input against a regular expressi on given as its argument. The input, in this case "0", is provided as part of the command you need t o create/compose. The output of the command will b e "0" on the standard output if the string "0" is matched by the regular expression '\<[0-9]{1,2}\>' . If it does not match, no output appears on the s tandard output.
Then give a shell command of the identical type tha t will perform the same test as

```
egrep -v '\<[0-9]{1,2}\>' <<< 199
```
In neither case can you can use a temporary file. N or can you use the << or < constructs.

Hint: continue to use egrep (or grep) but consider using a pipe.

Hint: invoking egrep (or grep) with different opti
ons than shown above is not a solution to this que
stion. Further, the correct solution does not invo
lve any options to egrep (or grep) than those show
n above.

Submit your two commands in a file named q4_soluti
on.txt and a log of testing your commands in a fil
e named q4_log.txt.

General Notes:

Electronic submissions are to be made using the moo
dle pages for the class.

All of the files you upload as part of your assign
ment solution are to be in plain ASCII text format
.