Assignment 4

Adding Redirection to your Shell

Author: Mike Izbicki

This project must be done in a group of two

Coding Requirements

Extend your `rshell` program so that it properly handles input redirection `<`, output redirection `>` and `>>`, and piping `|`. This will require using the Unix functions `dup` and `pipe`. You can find help on how to use these functions in the man pages.

As an example, after this assignment, your program should be able to successfully handle the following command:

```
$ cat < existingInputFile | tr A-Z a-z | tee newOutputFile1 | tr a-z
A-Z > newOutputFile2
```

IMPORTANT: This is a necessary but not sufficient test case. You must come up with others on your own.

Bash has an extensive syntax for redirection, and you are not required to implement all of it. But if you're curious, see the linux documentation project's bash io-redirection tutorial for details.

Submission Instructions

Add this new code to your rshell project from homework 3. Create a branch called test. Do all of your work under this branch. When finished, merge the test branch into the master branch, and create a tag called hw4. Remember that tags and branches in git are case sensitive!

NOTE: git push will not automatically push tags to your repository. Use git push origin hw4 to update your repository to include the hw4 tag.

To download and grade your homework, the TA will run the following commands from the hammer server:

```
$ git clone  https://github.com/yourusername/rshell.git
$ cd rshell
$ git checkout hw4
$ make
$ bin/rshell
```

You should ssh into hammer.cs.ucr.edu and run the above commands to verify that you've submitted your code successfully. If you forget how to use git, two students from previous cs100 courses (Rashid Goshtasbi and Kyler Rynear) made video tutorials on the git commands needed to submit your assignments via Github.

Do not wait to upload your assignment to Github until the project due date. You should be committing and uploading your assignment continuously. If you wait until the last day and can't figure out how to use git properly, then you will get a zero on the assignment. NO EXCEPTIONS.

You will also need to create a file for submitting your partner and github information. Create a text file using vim with the following information: you and your partner's name, you and your partner's net IDs, and the github url of your assignment's repository.

Follow this format EXACTLY:

```
name1=
ucrnetid1=
name2=
ucrnetid2=
repourl=
```

Save the file as hw3 (**WITH NO EXTENSION**) and submit it to iLearn's to Assignment 4 submission link.

Here's an example file:

```
name1=Busra Celikkaya
ucrnetid1=bceli001
name2=Amirali Darvishzadeh
ucrnetid2=adar001
repourl=https://www.github.com/busrac/rshell.git
```

Your repository should be public, however if you prefer to have a private repository please email the instructor for additional information on adding collaborators.

Project Structure

There are no changes to your project structure.

Coding Conventions

Your code must not generate any warnings on compilation.

You must follow the CalTech coding guidelines, as stated in the syllabus.

Your final executable must have no memory leaks.

Testing

Again, the tests you choose will be the most important part of your grade.

You should carefully consider: which redirections can be legally combined together, and which cannot? Does order matter? Also make sure to test that you are parsing the command correctly.

IMPORTANT: If you are unsure if your test cases are sufficient, ask one of the instructors to review them *before the deadline*.

Collaboration Policy

You MAY NOT look at the source code of any other student.

You MAY discuss with other students in general terms how to use the unix functions.

You are ENCOURAGED to talk with other students about test cases. You are allowed to freely share ideas in this regard.

You are ENCOURAGED to look at bash's source code for inspiration.

Grading

Rubric

25 points for input redirection <
25 points for output redirection > and >>
50 points for piping |