Project TA - Peter CHUNG (cspeter@cse.ust.hk): Handling questions before the project deadline
Grader TA - CHEN, Wei (wchenbt@cse.ust.hk): Collecting late submissions, grading and handling appeals

COMP3511 Operating System Fall 2020

Project Assignment 1 (PA1) Linux Shell - File Redirection

25 Sep 2020 (Friday) 02 Nov 2020 (Monday) 23:59

Introduction

The aim of this project is to help students understand **process management** and **inter-process communication** in Linux. Upon completion of the project, students are able to implement a useful system program using the related Linux system calls.

Overview

In this assignment, you need to implement a non-interactive command line interpreter that supports file input/output redirection. The command line interpreter is named as `cmd`. For example, suppose there are 4 files in the current working directory:

```
cmd cmd.c in.txt out.txt
```

The sample usage of the non-interactive command line interpreter is as follows:

```
$> ./cmd < in.txt > out.txt
```

`$>` represents the shell prompt. `<` means input redirection. `>` means output redirection.

The contents of `in.txt` and `out.txt` are as follows:

Content of `in.txt`	**Content in `out.txt`** (after running the above command)
`ls`	`cmd cmd.c in.txt out.txt`

You can easily use other given test cases to test your program and then use the `diff` command to compare the sample output files.

Restrictions

In this assignment, you **CANNOT** use `system` function defined in the C Standard library. The purpose of the project assignment is to help students understand process management and inter-process communication. It is meaningless to directly use the system function to process the whole command. You should use related Linux system calls such as `pipe` and `dup/dup2`. When connecting pipes, POSIX file operations such as `read, open, write, close` should be used, but not using `fread, fopen, fwrite, fclose` from C standard library. For details, please attend the PA1 related lab.

Project TA - Peter CHUNG (cspeter@cse.ust.hk): Handling questions before the project deadline
Grader TA - CHEN, Wei (wchenbt@cse.ust.hk): Collecting late submissions, grading and handling appeals

Development Environment

CS Lab 2 is the development environment. Please use one of the following machines (`csl2wk`**`XX`**`.cse.ust.hk`), where **XX**=01…40. The grader will use the same platform to grade all submissions. Please note that different `gcc` compilers and Linux platforms may produce different results.

In other words, *"my program works on my own laptop/desktop computer, but not in one of the CS Lab 2 machines"* is an invalid appeal reason. Please test your program on our development environment (not on your own desktop/laptop) thoughtfully before your final submission. Remote login is supported on all CS Lab 2 machines, so you are not required to be physically present in CS Lab 2. For details about remote login to CS Lab 2 machines, please refer to the first few labs.

Starting Point

`cmd_file_redirection_skeleton.c` is a starting point. To start your work, please rename the file as `cmd.c`

Read carefully the documentation in the base code. You are not required to start from scratch as the base file already provides you some useful features (e.g. command line parsing). Necessary programming concepts will also be introduced during the labs.

Please note that C programming language (instead of C++) must be used to complete this assignment. C is not the same as C++. Here are the commands to compile and run `cmd.c`

```
$> ls
cmd.c in.txt

$> gcc -o cmd cmd.c

$> ./cmd < in.txt > out.txt
```

File Input Redirection

Instead of typing the command on the console, the input can be redirected from a text file. The file input redirection feature can be completed by using the `dup/dup2` system call (they are discussed in the lab). The key idea is to close the default stdin (0) and replace the stdin with the file descriptor of an input file.

We can use the following command to count the number of lines of the file (cmd.c). Here is a sample input file redirection usage:

```
$> wc -l < cmd.c
```

Project TA - Peter CHUNG (cspeter@cse.ust.hk): Handling questions before the project deadline
Grader TA - CHEN, Wei (wchenbt@cse.ust.hk): Collecting late submissions, grading and handling appeals

File Output Redirection

Similar to the file input redirection, the output can also be redirected to a text file. The file output redirection feature can be completed by using the `dup/dup2` system call. The key idea is to close the stdout (1) and replace the stdout with the file descriptor of an output file.

We can use the following command to redirect the output of the ls command to an output text file (out_demo_ls.txt). Here is a sample output redirection usage:

```
$> ls -lh > out_demo_ls.txt
```

Sample Test Cases

The grade TA will use the following pattern to grade your submission

```
$> ./cmd < [Input file name] > [output filename]
```

The input test cases are provided. You can assume that each input file stores a single line of command. You can also assume that the input format is valid. Each command will have at most one character (<) and one character (>).
Here is the summary of the provided test cases:

Input	**Expected output**
`ls`	`ls` displays the filenames of the current working directory
`ls        -l   -h`	`ls -l -h` displays the filenames of the current working directory using a better format, with some empty space characters added
`ls -lh`	It displays the same output as above
`wc -l < cmd.c`	It displays the total number of lines of the file (cmd.c)
`ls -lh > out_demo_ls.txt`	Instead of displaying the result on the console, the output will be redirected to a text file (out_demo_ls.txt)
`wc -l < cmd.c > out_demo_both.txt`	The total number of lines of the file (cmd.c) should be redirected to a text file (out_demo_both.txt)
`wc -l > out_demo_both2.txt < cmd.c`	The result is the same as above, except that the file output is redirection to another text file (out_demo_both2.txt)

Project TA - Peter CHUNG (cspeter@cse.ust.hk): Handling questions before the project deadline
Grader TA - CHEN, Wei (wchenbt@cse.ust.hk): Collecting late submissions, grading and handling appeals

Please note that the exact output will be different based on the files and configuration of your machine. In addition, you can assume that the output files do not exist in the machine (Hint: you can use the command `rm -f out*.txt` to remove all output files)

Sample Linux Executable

The sample Linux executable file (which is runnable in a CS Lab 2 machine) is provided for your reference. After the executable file is downloaded to a CS Lab 2 machine, you need to add an execution permission bit to the file. For example:

```
$> chmod u+x cmd_file_redirection
```

Please note that the executable file can only be executed in one of the CS Lab 2 machines. It cannot be run in other Linux / Mac / Windows machines.

Marking Scheme

1. (30%) Explanation of `process_cmd` within the comment block. You should use point form to clearly explain how you implement this function
2. (70%) The given test cases. We do not have other hidden test cases. Please note that the grader TA will check the source codes and may shuffle the order of test cases. Thus, students who hardcoding the output of the test cases (without implementing the codes) cannot get any mark.

Plagiarism: Both parties (i.e. students providing the codes and students copying the codes) will receive 0 marks. A plagiarism detection software will be used to identify the cheating cases. We will handle the potential cheating cases near the end of the semester.

Submission

Submission platform: Canvas (https://canvas.ust.hk)
File to submit: **`cmd.c`**

Please check carefully you submit the correct file. For example, in the past semesters, some students submitted the executable file instead of the source file. Zero marks will be given as the grader TA cannot grade the executable file.

You are not required to submit other files such as the input test cases.

Late Submission

For late submission, please submit it via email to the grader TA. Suitable late penalty will be applied for the late submissions.