

Operating Systems
University at Albany
Department of Computer Science
Chongqing University of Posts and Telecommunications
Computer Science, International College
ICSI 412

Project-1

Assigned: Monday, April 14th, 2025.

Due: To be determined by your co-instructor.

Student Name:

OBJECTIVES

To develop a C program that uses `fork()`, `pipe()`, and the `exec()` family of process system calls to process simple shell commands. Your C program must be named *myShell.c* and, after compiled, the executable produced is to be named *mysh*.

PROBLEM

You are to use any distribution of the Linux operating system to create a program to parse and execute simple shell commands. The commands to be processed will have the following syntax:

MyShPrompt> command1 | command2 | command3 | command4 > file.txt

The above listed syntax illustrates the maximum sequence of commands that your shell will be able to process. Also, all commands may or may not have postfix flags.

The statements that follow illustrate the required syntax for the commands. They are also to be used for the testing of your prototype.

Prompt> ./mysh

MyShPrompt> cat country.txt city.txt | egrep 'g' | sort | more > countryCitygSorted.txt

MyShPrompt> cat country.txt city.txt | egrep 'g' | sort | wc -l > countryCitygCount.txt

In the above examples *Prompt>* is your command line prompt. *MyShPrompt>* is the new prompt created as the result of the execution of your *mysh* binary file. Files *country.txt*, and *city.txt* are input files and *countryCitygSorted.txt*, and *countryCitygCount.txt* are output files. Both input files are available in the two tables below and populated with the data to be used to verify the correctness of your solution.

WHAT TO SUBMIT

1. Your solution must be placed in a Microsoft compressed (zipped) folder (.zip). Your .zip folder must be named: *412 Project 1- Your Name*. Marks will be deducted if you do not follow this requirement. Your solution must be submitted to your co-instructor.

2. You are to use the documentation format provided below to collect all required information about the solution you developed. You are to include, in your documentation document, the source code of all your .c files as well as any output produced by your solution that clearly shows:

- a) Both *countryCitygSorted.txt*, and *countryCitygCount.txt* files.
- b) A screenshot of your command line interaction that produced the required output files.

Your program should be developed using GNU versions of the C compiler. Your solution must use **Ordinary Pipes** (Lecture 04 – Interprocess Communication; slides 37-39). It should be layered, modularized, and well commented. The following is a tentative marking scheme and what is expected to be submitted for this exercise:

1. External Documentation (as many pages necessary to fulfill the requirements listed below.) including the following:
 - a. Title page.
 - b. A table of contents.
 - c. [20%] System documentation.
 - i. [10%] A high-level data flow diagram for the system.
 - ii. [5%] A list of routines and their brief description.
 - iii. [5%] Implementation details.
 - d. [5%] Test documentation.
 - i. [3%] How you tested your program.
 - ii. [2%] A list of your test sets including the results obtained by each set.
 - e. [5%] User documentation.
 - i. [3%] How to run your program.
 - ii. [2%] Describe any parameters (if any).

2. Source Code.

- a. [65%] Correctness.
- b. [5%] Programming style.
 - i. [2%] Layering.
 - ii. [1%] Readability.
 - iii. [1%] Efficiency.
 - iv. [1%] Comments.

country.txt
zimbabwe
ruusia
australia
brazil
china
denmark
germany
france
angola
italy
japan
korea
poland
mexico
nicaragua

city.txt
miami
shanghai
albany
chongqing
tokyo
beijing
detroit
new york
hong kong
macau