

Project 2: UNIX Shell

CS4350 UNIX Programming

Due Date: 4/28/2016

A. Project Objectives

You have learned how to work with files and processes in UNIX. Now, you are expected to make a shell and some UNIX commands so that you can handle some basic operations in UNIX. You are going to

- a) Practice with Makefile
- b) Practice with IO operations in Unix
- c) Practice with process and thread in UNIX
- d) Learn to use POSIX library for untaught functions
- e) Learn to debug a program

B. Project Tasks

NOTE:

- You may have source code files. Please use tar and gzip to pack them into one package.
- For any function that is not discussed in class, please refer to the online POSIX library, or use man.
- Four commands and one shell program will be implemented. **Each group member must implement at least one of the required commands or the shell program. NO ONE could be only responsible for writing a group report.**
- **DO NOT use any IDE tools.** Use plain text editor to edit your source code, make your own Makefiles, and use KDBG or INSIGHT to debug your code.

Task I: Makefiles

According to task II and III, divide work load among your group members. Then, design some Makefiles so that each group member can work on their own and finally all code can be combined. You should assume you are working in a big group.

Task II: External and Buildin Unix Commands

Implement the following basic file commands and test them in **bash**. These commands should support at least the listed command syntax.

1) mycat

- mycat a_file
- mycat <a_file
- mycat <a_file >b_file

2) mycp

- mycp a_file b_file
- mycp -R a_dir b_dir

3) myls

- myls -l a_file
- myls a_dir
- mycat \$(mysls)
- myls | mycat

Implement the following buildin commands of your own shell. These buildin commands should support at least the listed command syntax.

4) mycd

- mycd .
- mycd ..
- mycd a_dir

5) pwd

- pwd

NOTE:

- The output of these commands should be the same as the standard UNIX commands. For example, **mysls -l** should display information as **ls -l**. **So, you need to learn how to use time functions and user/group id/name functions.**
- If any error happens when using your implemented commands, corresponding error message should be displayed to give an explanation of the error. For example, you may not have permission to **mycp** a file, or file does not exist for copying. **So, you need to learn how to use error functions.**
- You only need to handle text files. For example, **mycat** only needs to display a text file.
- Your commands can support any other function not specified in this project. It is up to your design.

Task III: UNIX Shell

Implement a shell program “mysh”. Run **mysh** in a terminal. Your shell program should be able to

- 1) Read your command lines, and interpret options, arguments and redirections of commands.
- 2) Output error messages for possible errors a command may encounter.
- 3) Test the myxxx commands you implemented in **mysh**.
- 4) make redirection of stdin, stdout and stderr for the required commands in Task II.
- 5) Test the following in **mysh** with the normal Unix commands.
 - Test cp, ls, cat
 - Test redirection and pipe on cat

NOTE:

- **DO NOT use system() to execute any command.**
- You must control processes when executing commands.
- You can use any function in POSIX library.
- You do not need to handle patterns, such as *, ? or [].
- **You are encouraged to learn and test anything not taught in class.**

C. Project Report

How to Deliver

A group report is needed to show what you did in the project. Please clearly state your results of this project. You are expected to hand in a report as follows:

- a) **Submit a hard copy of your group project report on the due day.**
- b) **Upload your group project report and source code in TRACS. Please tar them into one file.**
- c) **The complete code shall be separated from report. Only required code can be included in report.**

The format of group project report is

- a) A cover page with names of your group members with font size 12.
- b) Single space and single column.
- c) 7-12 pages (not including the cover page).
- d) The report file should be a PDF file, NOT a WORD file.

Note: Be sure to include the names and email addresses of all group members in the report and the code. The report and the code should be turned in before class on the specified due date. **Late grade will be deducted in case the submission is not made on time.**

What to Deliver

The report should include the following sections.

Section I (Introduction):

Clearly state the responsibility of each group member, e.g. who did which command, who wrote which part of the report, how your group was coordinated, etc.

Section II (Task I):

- 1) Please copy and paste the Makefile for the shell program, and the Makefile for one of the required commands.
- 2) Please take a screen shot of debugging a piece of code. The screen shot should show that you are looking and tracing variables at some break points. (Press the key Print_Screen in GUI, you will be prompted to save the screen shot into a picture file. Then, you can paste the picture file in your report.)

Section III (Task II):

- 1) Please describe your implementation of **mycat**, **mycp** and **mys**. For example, the functionality your commands support and the syntax of your commands. DO NOT copy and paste your source codes in your report.
- 2) List up to two issues that you must pay attention to when implementing these three commands, such as flags, permissions, exceptions, errors, etc. Describe how to handle these issues.
- 3) Show an example of error message when testing the three commands in **bash**.

Section IV (Task III):

- 1) Please describe your implementation of the shell program. Describe how you handle pipe in your shell in steps in English. DO NOT copy and paste your source codes in your report.
- 2) Show the results when testing “**mys -l**” in **mysh**.
- 3) Show the results when testing “**ls -l**” in **mysh**.
- 4) Show the results when testing “**mycat <a_file >another_file**” in **mysh**.
- 5) Show the results when testing “**mys | mycat**” in **mysh**.
- 6) Show the results when testing “**mycd ../**” in **mysh**.
- 7) Show the results when testing “**mypwd**” in **mysh**.

D. Grading Rubrics

Section I: Introduction (10% group):

Section II: Task I (20% group):

Section III: Task II (20% group):

Section IV: Task III (20% group):

Section V: Individual (30% individual):

- 1) If you finish any least one command or the shell, you get 15%.
- 2) If you write any part of the report, you get 15%.
- 3) If you only write a part of the report, you get 0%.