

CS 4348/5348 Operating Systems -- Project 1

The goal of this project is to let you get familiar with the basic commands in Unix systems and learn how to write shell scripts. You can use any Unix shell language, such as Bourne shell (sh), C shell (csh), bash, to implement this project.

Similar to procedural programming languages, shell script languages also support the basic control constructs if-then-else, while loop, foreach, etc. You can google to find tutorials for writing shell scripts. Some resources for this and subsequent projects are given at my web page:

<https://personal.utdallas.edu/~ilyen/course/os/home.html>

1 Shell Scripts

1.1 Makefile, pipe, and redirect

First, write a **makefile** for compiling the C source code files provided for the project (please see my web page given above). The generated executable should be named as “**cal.exe**”.

Note that you may face problems with the compilation of the code (the code is correct) and you need to figure out how to solve them. The code is correct and there is no need to change anything. When you compile each C source code file separately (you are required to do so in your makefile), you need to use proper options for gcc (or cc) to avoid complaints about undefined variables. Also, you need to figure out what other parts shall be included in the compilation command to properly compile the code without getting error messages.

Next, write a shell script “**exec.sh**” to compile the program using the “**make**” command and execute cal.exe with provided input file “**cal.in**”. In script “exec.sh”, you need to capture the output from cal.exe and print it to a file with name *f*. In other words, your script needs to take *f* as an input argument. Your shell script will need to execute cal.exe twice. In the second execution, you need to pipe the output of cal.exe to the “**wc**” command and append the output from wc command to *f*.

1.2 Directory traversal

Write a shell script “**traverse**” to traverse the directory structure starting from a directory *d*, where *d* is given as the input argument for your traversal script. During traversal, print out each directory you have visited on the screen. At the same time, for each directory traversed, print the directory name and the list of files in the directory (directories are also files) with file sizes less than 1KB to a log file *l*. *l* will be the second input argument for traverse.

1.3 About your scripts

Your shell scripts should allow anyone to run from any working directory (e.g., TA or instructor will download your submission from elearning and test your scripts in her/his directories). You need to test your scripts on a UTD Linux server, on which your scripts will be tested by the TA or the instructor. You can use putty to remote login and do the testing.

2 Project Submission

You need to submit your program **before** midnight (11:59pm) of the due date (check the web page for the due date). Use UTD elearning system for submission: "elearning.utdallas.edu".

Your submission should include the following:

- ✓ All shell script files discussed above.

- ✓ If you did not finish the project, you need to specify which part(s) you have completed in your *README* file.