CSE320 : System Fundamentals II Assignment 5

DUE: 11-Nov-2024

Mastering Advanced C Programing Concepts - Exception, System Call and Process Control

Overview/Goals

The objective of this assignment is to reinforce your understanding of advanced C programming and assembly language concepts. You will work on a series of tasks that cover topics such as exception, system call, and process control.

In this assignment, you will be working reinforcing your understanding of advanced C programming and assembly language. In addition, it will give you experience with:

1. Understanding exceptions and exceptional flow control.
2. Understanding system calls.
3. Understanding process control.

Basic Program Specifications

You are provided with a simple shell program that supports pipes and redirection for basic commands. It splits user input into individual commands separated by a pipe. It also handles input and output redirection, and executes the commands using child processes. Use **execvp** to execute the command line. **execvp** and **execve** are both functions in C used for executing other programs from within a C program, but **execvp** is a simpler version of the exec family of functions. Download assignment4.zip that includes **assignment5.c** and **assignment5**.h. Compile the code with your implementation and execute it. You should capture a screenshot of the execution result from the C code.

Tasks

1. Download assignment5.zip provided in the assignment
2. Complete the TODO sections
3. Build, test, and debug the resulting code

Testing the code

Run your completed shell. Test it by executing the following commands and capturing the results in a script file.

Test these commands and capture the result.

ls > test.txt

cat < test.txt

ps -a > test.txt

tail -n 2 test.txt

ls -al | pwd

pwd | ps

ls | wc

ls -al | grep test

Verify the output 'makes sense'.

Building the Code

You may use the **Makefile** provided in the zip file to build the code. You will not have to write your own **Makefile** for this assignment.

If you need to debug code with gdb, then, on the Unix command line, type 'make debug'. You can then run app under gdb and use the debugger.

Reflection

1. Document your code thoroughly with comments explaining each section.
2. Prepare a short report PDF document summarizing your development experience including each step or function that you implemented, the challenges faced, and what you've learned during this assignment.

Submission Instructions

Deliverable Files: Completed assignment5.c, assignment5.h and the provided Makefile packaged in a single zip or tgz (gzipped tar) file.

Please follow this procedure for submission:

1. Place the deliverable files into a folder by themselves. The folder's name should be CSE320_HW5_<yourname>_<yourid>. So if your name is Alice Kim and your id is 12345678, the folder should be named 'CSE320_HW5_AliceKim_12345678'

2. Compress the folder and submit the zip file.
    a. On Windows, do this by clicking the right-mouse button while hovering over the folder. Select 'Send to -> Compressed (zipped) folder'. The compressed folder will have the same name with a .zip extension. You will upload that file to the Brightspace.
    b. On Mac, move the mouse over the folder then right-click (or for single button mouse, use Control-click) and select **Compress**. There should now be a file with the same name and a .zip extension. You will upload that file to the Brightspace.
3. Navigate to the course Brightspace site. Click **Assignments** in the top navbar menu. Look under the category 'Assignments'. Click **Assignment5.**
    a. Scroll down and under **Submit Assignment**, click the **Add a File** button**.**
    b. Click **My Computer** (first item in list).
    c. Find the zip file and drag it to the 'Upload' area of the presented dialog box.
    d. Click the **Add** button in the dialog.
    e. You may write comments in the comment box at the bottom.
    f. Click **Submit**. ← Be sure to do this so I can retrieve the submission!

Grading Criteria

The assignment will be graded for the following items:

Any errors in the compile reduce the grade by 50%.
Any warnings in the compile (but the code builds completely) reduces the grade by 30%.

If code crashes due to issues with improper use of data types, pointers, C language features, or the System API calls, this will reduce the grade by 30%.

If code is not well commented with reasonable variable names and consistent spacing and indentation, the grade may be reduced by up to 10%.

If the Assignment submission does not follow all directives (includes required files, etc., is packaging (name of zip or tgz file.)…) the assignment may lose up to 5%.

Quality and completeness of report is worth 5% of the total points.