CS 4540 Operating Systems
Assignment #2: Ordinary Pipes

Introduction

Write a small C program using ordinary pipes to provide and accept passwords by the parent and child processes, respectively.

Problem Specification

Design a program using ordinary pipes in which there are two processes (a parent and a child). The parent process asks the user to enter a candidate password for his "username" in order to access Application X. (We do not care what application is X.) In response, the parent sends the password to the child. The child checks the strength of the user's password, and returns one of the following three decisions:

Decision	Description
STRONG PASSWORD	If the password has more than 8 characters and contains upper-case and lower-case letters as well as numbers. Examples: OPSlmr183, 187jklPNR, etc.
WEAK PASSWORD	If the password has more than 8 characters and does *not* mix upper-case letters, lower-case letters, and numbers. Examples: AAAAAAAAA, 123456789, aaaaaacccc, EERReeeee, 7878EEEE, hhhh4545, etc.
SHORT PASSWORD	If the password has fewer than 8 characters.

The program requires using two pipes: one for sending the password from the parent to the child, and the other for sending the child's decision to the parent.

SLC Report Requirements

Write a full SOFTWARE LIFE CYCLE (SLC) report (analogous to the SLC report presented in class).

Please note that your reporting job is easier due to the following simplifications or the report steps:

1) The PROBLEM SPECIFICATION section (Step 1) should include just a copy of the given problem(s).
2) The PROGRAM STRUCTURE DESIGN section (Step 2) will have a few modules to name (Substep *2.1. Modules and Their Basic Structure*), and few modules to provide pseudocode for (Substep *2.2.Pseudocode for the Modules*). You must have more than one pseudocode refinement level for all modules that are not trivially simple.
3) The sections for RISK ANALYSIS (Step 3), VERIFICATION (Step 4), REFINING THE PROGRAM (Step 7), PRODUCTION (Step 8) and MAINTENANCE (Step 9) can include just 1-2 sentences (analogous in the SLC report presented in class).
4) The CODING section (Step 5) must have at least two code refinement levels (the first will just add function headers and trailers, as shown in class).
5) Include an adequate TESTING section (Step 6).

Other reporting requirements:

1) Use a constant-width (mono-spaced) font (like Courier) for both pseudo code (which is in the form of comments) and source code; it will enhance code readability and facilitate even spacing and indentation. Please use variable-width fonts for other text in the report (so pseudocode and code versions stand out).

Coding, Running and Submission Requirements

1) Follow *C Code Style Guide* and the proper programming style it requires, including comments, blank lines, indentations, spaces, etc.
2) All programs must be compiled and executed on Ubuntu running within the Oracle VirtualBox (using the versions indicated in class handouts).
3) Remember about *Assignment Submission Instructions* (guidelines), to be followed for each program of this assignment.
4) In addition to what *Assignment Submission Instructions* require, provide a *makefile*, and, if needed a README file explaining how to compile and run your program.

Submission Checklist

For each program you need to submit:

1) the SLC report (including 9 SLC steps, with as many pseudocode and code refinements as needed);
2) *makefile*, and, if needed a README file explaining how to compile and run the program;
3) the files created by the *script* command (including program output to the terminal, if any);
4) the output files (if any in addition to the output to the terminal) and

Submit your complete assignment package (all files) via Elearning as a *zip* file. Please use **<hw#_lastname.zip>** as the format for the name of the zipped file.

-------- Good luck! -----

Prepared by the TA, Majdi Maabreh. Verified and updated by Dr. Leszek Lilien