

CS4540 - Operating Systems

Assignment #2

Introduction

Write a small C program using ordinary pipes to provide and filter DNA subsequences by the parent and child processes, respectively.

DNA (deoxyribonucleic acid) is a long molecule that contains our unique genetic code. Like a recipe book, it holds the instructions for making all the proteins in our bodies. DNA contains four basic building blocks or ‘bases’: adenine (**A**), cytosine (**C**), guanine (**G**) and thymine (**T**).

An application, processing and analyzing DNA subsequences, sometimes receives DNA subsequences which contain noise. (Because of several factors including converting from DNA file format to another format, preprocessing bug, etc.). For example, the following are valid DNA subsequences: **AAAACCCGGTTT**, **GTACGTACGTAC**, **CCACTTTGGG**, while the following are invalid DNA subsequences: **ACTXKG**, **4ACTGGGTAA33**, **B7ACTGACTG**, **ACTG%**.

Problem Specification

Design a program using ordinary pipes in which there are two processes (a parent and a child). The parent process should ask a user to enter a DNA subsequence (assume the minimum length is 4 and the maximum length is 100). Then, the parent sends this subsequence to the child process. The child filters the subsequence by removing noise, and returns the filtered sequence to the parent process, which prints the “clean” sequence. For example, if the user input is **YACC%TTGG4**, the output will be **ACCTTGG**, because Y, %, and 4 are filtered out.

Use two pipes (both created by the parent process); one for sending the input DNA subsequence from the parent to the child, and the other for sending the filtered DNA subsequence from the child back to the parent.

SLC Report Requirements

For each program write a full SOFTWARE LIFE CYCLE (SLC) report (analogous to the SLC report presented in class).

A few additional notes related to SLC Report:

- 1) The PROBLEM SPECIFICATION section (Step 1) might 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*).
- 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 the sufficient number of code refinement levels. Remember that in C program development, Code Refinement #1 includes the lowest-level pseudocode plus main/function headers and tails.
- 5) Due to the simple structures of the small programs, the TESTING section (Step 6) should be rather easy.
- 6) Using mono-spaced fonts like `Courier New` for source code adds a significant value for code readability and quality. Please use this type of fonts for every source code in the report.

Coding, Running and Submission Requirements

- 1) Follow *C Code Style Guide* and use 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.
- 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, each with as many pseudocode refinement steps and code refinement steps 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 are created in addition to the output to the terminal).

Submit your complete assignment package (all files) via Elearning as a zip file. Use `hw<number>_<LastName>.zip` as the format for the name of the zipped file (e.g., `hw1_Smith.zip`).

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