TopGear Assignments

Linux System Programming – L1

Assignments

Estimated Effort: 2 PDs

Total Points: 100

Required VDI: Linux

Author: Jayapathi Ramamohan

jayapathi.ramamohan@wipro.com

Date: 23-Apr-2018

This set of assignments tests application of following concepts of Linux System Programming

- Process creation
- Inter-process communication
- Anonymous & named pipes
- Signal handling
- Proper error reporting & exit status

To be considered as correct submission, code should meet the following guidelines:

- Source code filename : as specified in each exercise
- Input filename & output filename should not be hardcoded in program, unless it is stated as part of the problem specification
- Output should be as per the sample shown in each exercise.

Exercise – 1: Filename: convert.c

Write a Linux program as per the following specification:

1. Main program creates two processes P1 and P2
2. P1 process reads file F1 and sends the contents to process P2 using a pipe
3. Process P2 reads from pipe, converts the text into uppercase/lowercase (depending on the option –u or –l specified) and sends it back to P1 process using another pipe.
4. P1 process gets the data transformed by P2 using pipe
5. P1 saves the data into the file F2
6. Filenames F1 and F2 are given as commandline arguments.

Error handling and exit status should as per the details given below.

7. If option –u or –l is not given on commandline or an invalid option is given on commandline, the program should terminate writing error message to the standard error, as shown in the example given below, and returning 1 as exit status.
8. If program is successfully executed exit status should be 0.
9. If program terminates unsuccessfully, program should print appropriate error message and its exit status should be 1.

Usage:

$ ./convert -l x.txt out.txt

out.txt should contain the same text as in x.txt, text being in lowercase.

$ ./convert -u x.txt out.txt

out.txt should contain the same text as in x.txt, text being in uppercase.

$ ./convert x.txt out.txt

Usage: convert [-l | -u] inputfile outputfile

Exercise – 2: Filename: brokenpipe.c

Implement a program using named pipes that demonstrates broken pipe scenario.

The program should capture SIGPIPE signal and display a message indicating “broken pipe”.

Program should create required named pipe(s) within the program.

What is the procedure followed by the programmer to demonstrate the scenario, should be stated in comments at the beginning of the program.

--- END ---