Operating Systems

University at Albany
Department of Computer Science
Chongqing University of Posts and Telecommunications
Computer Science, International College
ICSI 412

Assignment-3
Assigned: Saturday, April 1st, 2023.
Due: Saturday, April 8th by 11:59 PM.

Student Name:

OBJECTIVES

To develop a C program that uses both fork() and pipe() process system calls to implement interprocess communication by means of file sharing.

PROBLEM

You are to use the Ubuntu operating system or any other Linux distribution to create two C programs.

A program *consumer.c* to

1) read from a file containing integers, and to

2) write to the display both the even numbers and

3) the sum of the odd numbers found in the input file.

A program *producer.c* to

1) create a file with 20 integers and to

2) share this file with the consumer. The file created must be named *numbers.txt.*

Both programs, the consumer, and the producer, must use an ordinary pipe to communicate. Your solution must include a child process which will execute your consumer program. You may use the *fork(),* pipe(), dup(), dup2(), read(), write(), open(), creat(), as well as any of the *exec()* family of system calls.

WHAT TO SUBMIT

The following are to be submitted to your co-instructors:

Your source code for both the producer and the consumer, as well as any output produced by your solution that clearly shows the following:

1) The contents of the file *numbers.txt,* and
2) Information about pids of both parent and child, the file descriptor *fd* of your *numbers.txt* file, and a screenshot of the output of your consumer program. You are to use the following format to structure your output:

a) File *numbers.txt* fd is: *fd-of-numbers.txt.*
b) Contents of file *numbers.txt*: *a listing of the 20 integers.*
c) A screenshot of your terminal showing both (1) the even numbers returned by your consumer program as well as (2) the sum of the odd numbers.
d) Parent Process: My pid = *pid-of-parent*. I created child pid = *pid-of-child.*
e) Child Process: My pid = *pid-of-child*. My parent pid = *pid-of-parent.*