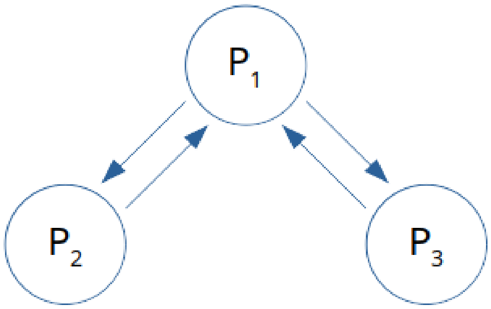
CMPE 382 Operating Systems, Spring 2023

Homework 1-Multiprocess concept: fork () and pipe() system calls

Due Date: March 27, 2024 23:55

In this homework, you are expected to write a C program, which demonstrates interprocess communication-IPC between three processes using pipes. Your program as the parent process should create two child processes and utilize four pipes, establishing a multi-directional communication as shown in the figure below:



Note that unnecessary pipe ends must be closed to prevent improper pipe usage. Each pipe needs one write one read end to remain open for two way communication.

The parent process should read a filename as a command line argument (so that different files can be used on each execution). Each file should have an unspecified number of random integers. The parent process P1 should read integers and send to BOTH child processes one by one.

- P2 receives the integers one by one, categorizes each integer according to number of digits. For readability, implement a function named "int nrDigits(int)" to do this. You may assume integer numbers in the test files be between 0 and 99999. Count the number of integers received according to the number of digits.
- P3 receives the integers one by one, computes the number of prime integers received. Implement a function named "int isPrime(int)" to do this.

  When P2 and P3 have finished their jobs, they must submit back the results they have computed back to the parent process and terminate.

  Parent process P1, after receiving results from P2 and P3 should print a nicely formatted output as given in the following example and should terminate.

Further Notes Remarks

1. No team working is allowed.
2. You may study from other sources but you are strictly forbidden to copy and/or modify from them.
3. Implement your homework in C programming language. It must compile in GNU/Linux and execute with no errors.
4. You may (modify and) use the following shell script to generate input files with random numbers:

```
for i in {1..1000}; do echo $((RANDOM)) >> numbers.txt; done
```

5. Both child processes receive unknown number of (say N) integers from the pipe in stage one (one integer per read operation), but write only once during stage two (all results in one write operation).
6. Properly comment your code. Your implementation should be understandable by reading your comments.
7. Name your program as CMPE382-HW1_YourInıtials-YourID such as “CMPE383-HW1_OKY-21326452820”
8. Upload you source program and output files to the course web page (LMS)

Sample Output for 3 input files

```
$ ./homework1 numbers.txt
Input file: numbers.txt

1 digits -    0
2 digits -    4
3 digits -    21
4 digits -    283
5 digits  -   692
Primes -     108
Nonprimes - 892
```

```
$ ./homework1 numbers2.txt
Input file: numbers2.txt

1 digits -     6
2 digits –     62
3 digits –    529
4 digits –    4875
5 digits  -   13026
Primes    - 1986
Nonprimes – 16514
```

```
$ ./homework1 numbers3.txt
Input file: numbers3.txt

1 digits -    0
2 digits -    0
3 digits -    10
4 digits –    58
5 digits  -   182
 Primes   -  34
Nonprimes – 216
```