HW-11 CS451 100 points Due: ondayMay, 12/7/2020

The LAST assignment!

The purpose of this program is to modify HW10 such that it makes use of the IPC facilities provided by UNIX. You must create the following programs:

Master:

- Accepts two command line inputs which will be the text file to read and output file to write.
- Creates 2 pipes, 2 semaphores, and 1 shared memory segment.
- Forks program1 and passes the input filename, write side of pipe1, and semaphore1 to Program1.
- Forks program2 and passes read side of pipe1, write side of pipe2, semaphore1, semaphore2, and the shared memory to Program2.
- Forks program3 and passes the output filename, read side of pipe2, semaphore2, and the shared memory to Program3.
- Once program1, program2, and program3 have terminated, the master must also terminate.

Program1:

- Accepts command line inputs (1 filename, 1 pipe ID, and 1 semaphore key).
- Program1 will read the input text file called and extract the words. Note that words will be separated only by spaces. Commas, periods, etc will be considered part of a word. As each word is extracted, program1 will write the words (1 word per line) into the pipe. Note that all punctuation marks will remain attached to their words (only spaces separate words).
- Program1 and program2 must synchronize the pipe using a semaphore such that only 1 word is in the pipe at a time.
- Once all of the words have been read and put into the pipe program1 terminates. Don’t forget to close the pipe!

Program2:

- Accepts command line inputs (2 pipe IDs, 2 semaphore keys, and 1 shared memory key).
- Program 2 will read from the first pipe and convert each word into Pig Latin
- (rules shown below) and write the converted word into the second pipe. Program2 will also keep track of how many words of each form (2 types possible) were converted and write those values into the shared memory (you only need 1 shared memory segment to hold multiple values).
- Program1 and program2 must synchronize the pipe using a semaphore such that only 1 word is in the pipe at a time.
- Program2 and program3 must also synchronize the pipe using a semaphore such that only 1 word is in the pipe at a time.
- Once all of the words have been read and put into the pipe program2 terminates. Don’t forget to close the pipe and release the first semaphore!

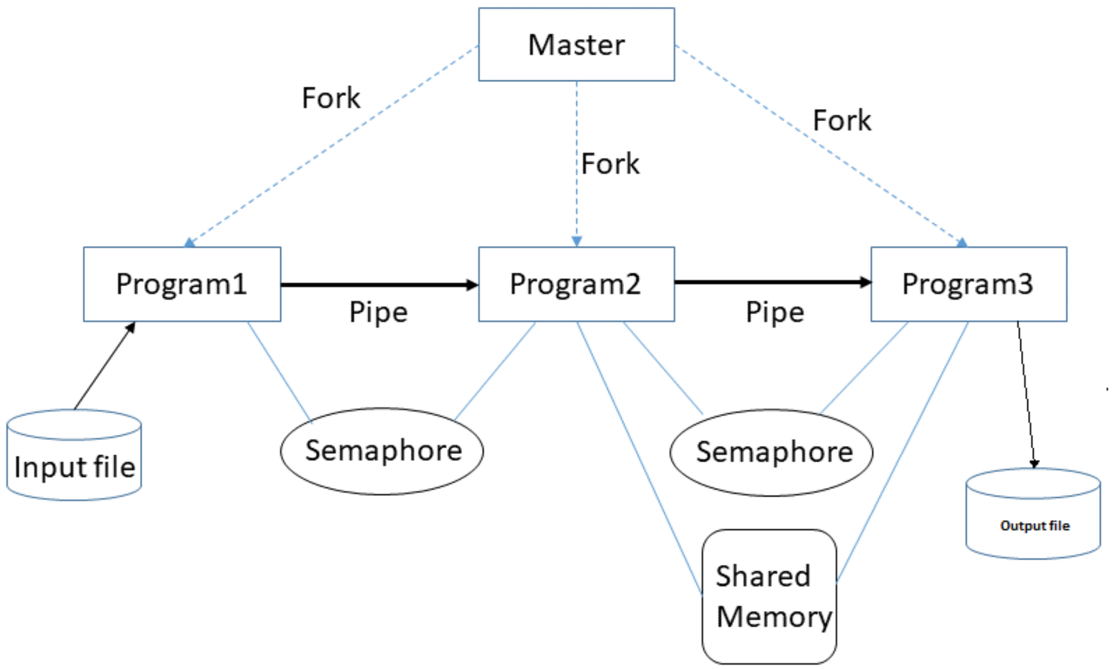
Program3:

- Accepts command line inputs (1 pipe ID, 1 semaphore key, 1 shared memory key, and output filename).
- Program3 will read from the second pipe, reconstruct the original document
- (multiple words per line) and write it out to the output file.
- Program2 and program3 must also synchronize the pipe using a semaphore such that only 1 word is in the pipe at a time.
- Once all of the words have been read and put into the output file, program2 must access the shared memory and display the values contained on the screen using the format:

Type 1: word_count
Type 2: word_count

- Once all of the words have been read program3 terminates. Don’t forget to close the pipe and release the second semaphore!

Here is a diagram of what you will create:



Pig Latin rules:
1) Type 1: For all words beginning with a vowel add the text "ray" to the end of the word. For example: "and" becomes "andray," "end" becomes "endray", etc. Note that any punctuation marks must remain at the end of the newly formed word.

2) Type 2: For all words beginning with consonants move the first letter to the end of the word and add "ay". For example, "number" becomes "umbernay," "letter" becomes "etterlay," etc. Note that any punctuation marks must remain at the end of the newly formed word.

REQUIREMENTS:
1. Your program must compile and run on Linux Mint.
2. Your full name must appear as a comment at the beginning of your program.
3. Your programs MUST be named program0, program1, program2, and program3.
   My script will assume those filenames. If your programs do not work because
   you used different filename, you will get 0 points.
4. You must include your source files and instructions on how to compile them (or a makefile).
5. Your tarball must be named hw11-yourname.tar
6. Your code must adhere to coding style 1TBS (https://en.wikipedia.org/wiki/Indentation_style)
7. Failure to follow the above directions will result in loss of points.