Lab03: IPC and map-reduce

Due date

Please refer to the lab assignment requirements.

Goal

The goal of this project is to practice various IPC methods (for data passing and synchronization) and learn map-reduce (parallel computing). Both are very important techniques frequently used in industry.

Details

This project consists of three independent sub-projects, each performing the same task: given a text file, the program outputs the lines that contain a given word. For example, given a line "Hello World!" and assuming the word of interest is "world", then this line should be output. (Note that if a line is "Hello worlds", then the line should not be output).
Your program execution (i.e., the parent process) will create a child process. The parent process can open the text file, read the content, and pass it to the child using one of the following methods, but cannot examine the words, while the child process cannot open that file but can examine words. Finally, the parent process (rather than the child) should output the lines in their alphabetical order. Below are the requirements for the three sub-projects:

1. Use Pipe as the method of passing the file content and the result.
2. Use Unix Domain Socket as the method of passing the file content and the result.
3. Use Shared Memory as the method of passing the file content and the result. Plus, the child process creates 4 threads, each acting as a Mapper; and the child process's main thread works as the single reducer. Map-reduce is only required for THIS sub-project. You are not allowed to use Hadoop as the map-reduce infrastructure; instead, you have to use Posix thread programming to implement map-reduce.

Sample input text files

Anna Karenina. Leo Tolstoy, 1870. (See attached file: ANNA_KARENINA.txt)
A 6.5M file. (See attached big.txt)
These are only some sample text files. Your program should accept a file path as a parameter. In another word, it should be able to work with any text file. So creating an index for the input file is not a good idea and hence not allowed.

Submission

Your submission should include (1) the code (a makefile is required), (2) a readme file describing your design and how to compile / use your code, and (3) a report for the following home assignments:

- Time the execution of your three programs, and analyze what contributes to the performance difference.
- Your design of the program
- Snapshots of experimental results(statistics) with analysis
- Problems encountered and your solution
- Summarize the different IPC methods provided by Linux, and describe when to use which.
- Write a short paragraph about map-reduce.
- Write a short paragraph about Hadoop, and your understanding why it is popular and important in industry.
- Reference materials
- Your suggestions and comments

Environment

Linux (Ubuntu 18.04/16.04 is recommended) and C/C++.

References

You may find the following articles useful

A simple makefile tutorial. Colby University. link
An Introduction to Linux IPC. Kerrisk, 2013. link
Beej's Guide to Unix IPC. Beej, 2010. link
Linux Interprocess Communications. Goldt, Meer, Burkett, and Welsh, 1995. link
MapReduce: Simplified Data Processing on Large Clusters. Dean and Ghemawat, 2004. link
POSIX Threads Programming. Barney, 2015. link
Beej's Quick Guide to GDB. Beej, 2009. link