

Project 1
Process Communication
Due Date: March 2nd 2017, 10.00pm

Problem 1

Problem: Given a list of integers find the Minimum, Maximum, and the Sum of the numbers. The list will be in a text file you generate. You may use any form of interprocess communication (IPC) to partition the file/array into chunks and distribute work to more than one processes (if there are multiple ones) (e.g., pipes, shared memory, or additional (perhaps more sophisticated) inherent process system calls).

Input Format:

Input will be in a text file. Each integer will be separated by a newline character (\n).

Exp:

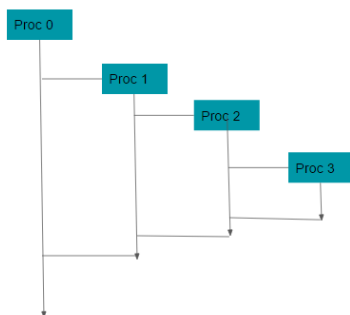
```
100
20
35
```

Output Format: You should print out the results in a text file. Every process that is created should print out their own process id and their parent's process id. Then once you have computed the final max, min, and sum, you will print those out. Please follow this format:

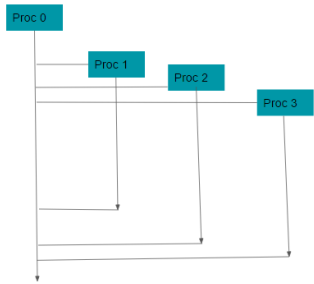
```
Hi I'm process 2 and my parent is 1.
Hi I'm process 3 and my parent is 2.
Hi I'm process 4 and my parent is 2.
Max=50
Min=1
Sum=371
```

Part A: Write a program to answer this problem using only one process.

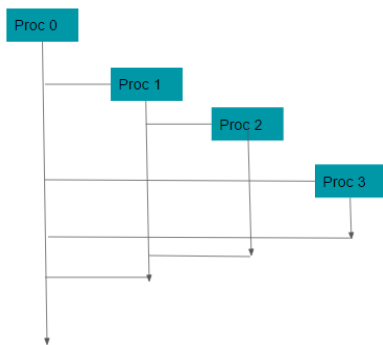
Part B: Write a program to answer this problem using multiple processes where each process spawns at most one process. (Like DFS)



Part C: Write a program to answer this problem using multiple processes where the first process spawns all other processes. (Like BFS)



Part D: Write a program to answer this problem using multiple processes where the first process spawns multiple processes and they spawn their own multiple processes. Your ultimate goal should be to produce such a hierarchy of processes so that performance compared to parts B and C, is optimized. (So basically this should be a combination of structures of Part B + Part C)



Extra Credit: Record the time it takes for each of the programs to run and comment on your observations. Try it on lists of size 10,100,1k,10k,100k.

Hint: You should first load the file into an array then start working on the data.

What to turn in:

- C files for each problem
- A makefile in order to run your programs.
- Input text file (your test case)
- Output text file (for your test case)
- Report: Explain design decisions (fewer vs. more processes, process structure, etc.). Give one thing you have learned. Include the extra credit observation in this report.

Logistics:

- For Project 1 please work in groups of 2 students.
- You are expected to work on this project using LINUX OS
- For those that do not have access to LINUX in their laptop, you may use one of the solution that I am going to post online about how to get access to a LINUX platform one way or another. If nothing works please contact me ASAP
- Make one submission per group.
- Only a couple of students that may be left without peers will be allowed to work in groups of 3.
- Do not collaborate within the groups. Groups that have copied from each other will both get zero points for this project.

