Description

The purpose of this project will be to gain experience with the C programming language as well as learn more about Linux processes.

There are a number of programs (`ps`, `top`, `pstree`) that display process information. This information is generally obtained from the `/proc` filesystem in Linux. For this project you will create a program to display a process tree on the terminal similar to the output of the `pstree` program.

The program will:

1. Scan the /proc filesystem pulling information for each process (directories with numeric names).
2. From the directory associated with each process obtain:
    a. The process id (pid)
    b. The parent process id (ppid)
    c. The size of the virtual address space (vsize)
    d. The name of the executable (comm)
3. Print the information obtained to the terminal in a tree format. The `pstree` program uses line drawing to show the tree, this is NOT required. What is required is that you indent levels in the tree similar to:

```
(1) systemd, 17052 kb
  (743) gdm3, 2034 kb
  (2025) bash, 20000 kb
    (2345) mypgm, 100 kb
```

   Where each line contains the PID of the process, the name of the program running and the memory in use. Lines are indented under their parent processes such that process 2345 has parent process 2025, processes 2025 and 743 have parent process 1

Hints

- You will need to open the /proc filesystem as a directory and read through it looking for numeric subdirectories (https://www.geeksforgeeks.org/c-program-list-files-sub-directories-directory/).
- All of the information required is available from the /proc/<pid>/stat file. You can open the file with the `fopen` function and use the `fscanf` functions to process the data (https://stackoverflow.com/questions/34575285/read-proc-stat-information).
- You will need to load the data into some sort of structure and figure out how to work through the tree. I would suggest first finding the root of the tree (will have a parent pid of zero) and then work down the tree setting up pointers to parents.
- There are lots and lots of other sources of information available on google.

Due Date and Deliverables

The project is due by midnight, February 4th. Late work will be accepted according to the schedule shown in the syllabus.

By Thursday, January 23rd, I will need to know the names of students that are working together (DM in slack preferably). The project is to be worked in groups of two students, not one, not three. If there are an odd number

of students participating, it will be at the instructor's discretion to allow either one individual submission or one group of three.

I will be expecting at a minimum a C source file and a `Makefile` to build the project. My preference would be a link to a GitHub (Bitlocker, GitLab, etc.) repository that I can clone to see the code but I will accept the aforementioned files submitted into the Oaks dropbox.

Rubric

The project will be worth 50 points in total allocated as follows:

1. 25 Points – Operation
    - (10 points) Does the program compile?
    - (15 points) Does the program work as specified?
2. 25 Points – Use of the "C" language
    - (10 points) Is the code well commented and readable?
    - (15 points) Does the code demonstrate an understanding of C features including simple loops and pointers.