Project Report

Jana Saleh, 900204192
Muhammad Azzazy, 900202821
Reem Said, 900201275

Department of Computer Science and Engineering
The American University in Cairo

Spring 2023

1 Introduction

The Linux Process Manager (LPM) we implemented is based on a Text User-Interface (TUI). It was implemented using the Rust programming language.

2 Data Structures

The various data structured utilized in the project are listed below:

- HashMap
- Vector
- str
- String

3 Features

The list below illustrates the commands that were implemented:

- desc: sorts the process table descending according to the pid and prints it

- ignite: starts a new process
- ptable: prints the process table
- find (pid): finds the process with the given pid input by the user and prints all the information related to the process
- pstree: prints the tree that shows the parent-child relationships between various processes
- kill (name): kills the process with the given name
- kill (pid): kills the process with the given pid
- network: gets the total packets received and transmitted in all network interfaces in the system
- memory: gets used,free and total memory
- df: command for retrieving disk file system information
- hddtemp: command for displaying the temperature of the Solid State Drive (SSD) or Hard Disk Drive (HDD)
- gputemp: command for displaying the temperature of the GPU
- sysinfo: command which displays basic system information
- lscpu: lists every cpu core with its frequency and other related information

4 Roles

Jana Saleh implemented the following requirements:

- descending sort
- documentations in the help command
- printing the process table and kill command
- network utilization (packets received and transmitted for network interfaces)
- Memory Information (free, used, total)
- update table which is dynamic table, worked but the printing was scattered
- sorted ptable both ascending and descending

Formatted the Added the some commands Added the help command the provides documentation for commands

Muhammad Azzazy implemented the following requirements:

- Text-User Interface (TUI)
- kill by pid command
- kill by name command
- df command for disk filesystem information
- hddtemp command for displaying the temperature of the SSD/HDD
- gputemp command for displaying the temperature of the GPU
- sysinfo command which displays basic system information
- scrolling of the ptable using the Up and Down keys. Note: this is only implemented for the ptable command
- lscpu command which lists every cpu core with its frequency and other related information
- ignite command for starting processes
- miscellaneous commands such as uname, release, and hostname which are subsets of sysinfo

Reem Said implemented the following requirements:

- infrastructure for the task manager
- process and system struct and related functions which we opted to remove in the end
- time elapsed for each process
- process table
- formatting for printing the process table
- pstree that shows the list of all process and the relationship (parent/child) between them
- code organization and implemented clean code practices
- tried to implement promoting and demoting processes

5 Issues

An attempt to implement a Graphical User Interface (GUI) was made, but it was unsuccessful. An attempt at implementing a dynamic process table was made, but it was unsuccessful due to the implementation of the TUI which restricted the output on the console from being changed as long as the function was still executing. Also, there is a problem with the pstree that it gets the parent-child relationship between all processes, but there is a trouble with printing it as it only prints some of the children not all of them. The process tree was not included in the options for the user since a function that is part of a crate was used for printing the tree.