SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF NETWORKING AND COMMUNICATIONS
21CSC202J-Operating Systems , Mini-Project Presentation

PROCESS MONITOR

Student 1 Reg No:RA2211056010055
Student 2 Reg No:RA2211056010070
Batch ID:AF1
Guide name and Designation:Dr.Shri Bharathi

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Table of contents

Objective

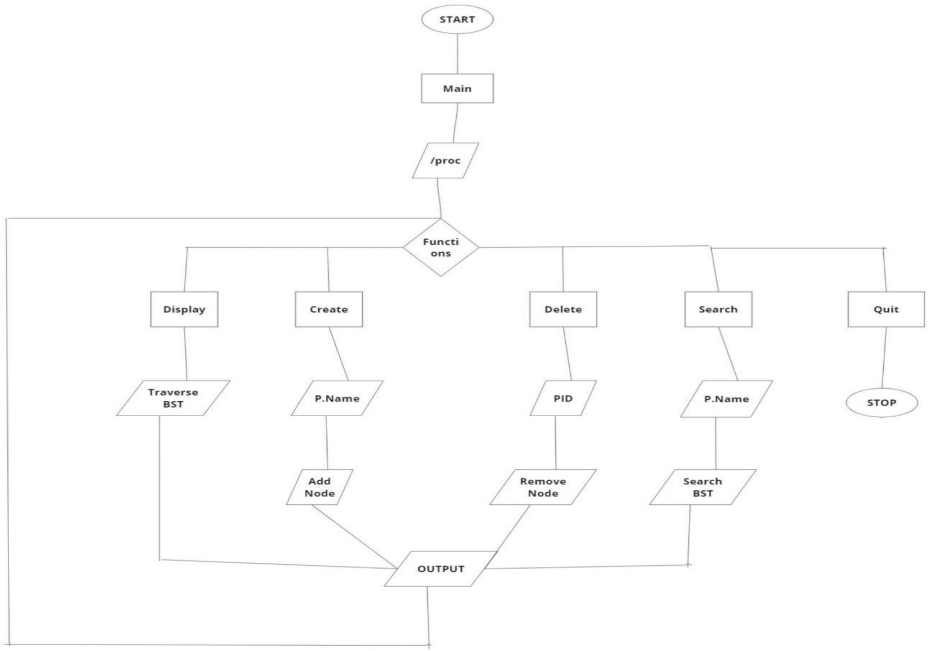
The process monitor is used to display a list of all running processes on the system. The user can use the s, d, and c keys to search for specific processes, kill processes, and start new processes, respectively.

This project will develop a useful tool for monitoring and managing processes on Linux systems. The process monitor will be easy to use and will provide users with valuable information about running processes.

Problem Statement

Process monitoring is an important task for system administrators. It allows them to identify and troubleshoot performance problems, detect and respond to security threats, and ensure that critical processes are running and available.

Architecture



Architecture

Here is a more detailed explanation of the flowchart:

The program starts by authenticating the user. If the authentication is successful, the program displays a menu to the user.

The user can then choose one of the following options:

Create a new process: The program creates a new process and adds it to a binary search tree (BST).

Delete a process: The program gets the PID of the process to delete and removes it from the BST.

Print the list of processes: The program prints a list of all the processes that are currently running.

Search for a process: The program gets the name of the process to search and searches for it in the BST. If the process is found, the program displays its details.

Quit the program: The program frees the BST and exits.

If the user enters an invalid choice, the program displays an error message.

The program repeats the loop until the user chooses to quit.

Hardware/Software requirement

Hardware requirements:

1. A computer with a Linux operating system
2. At least 2GB of RAM
3. At least 50GB of free hard disk space

Software requirements:

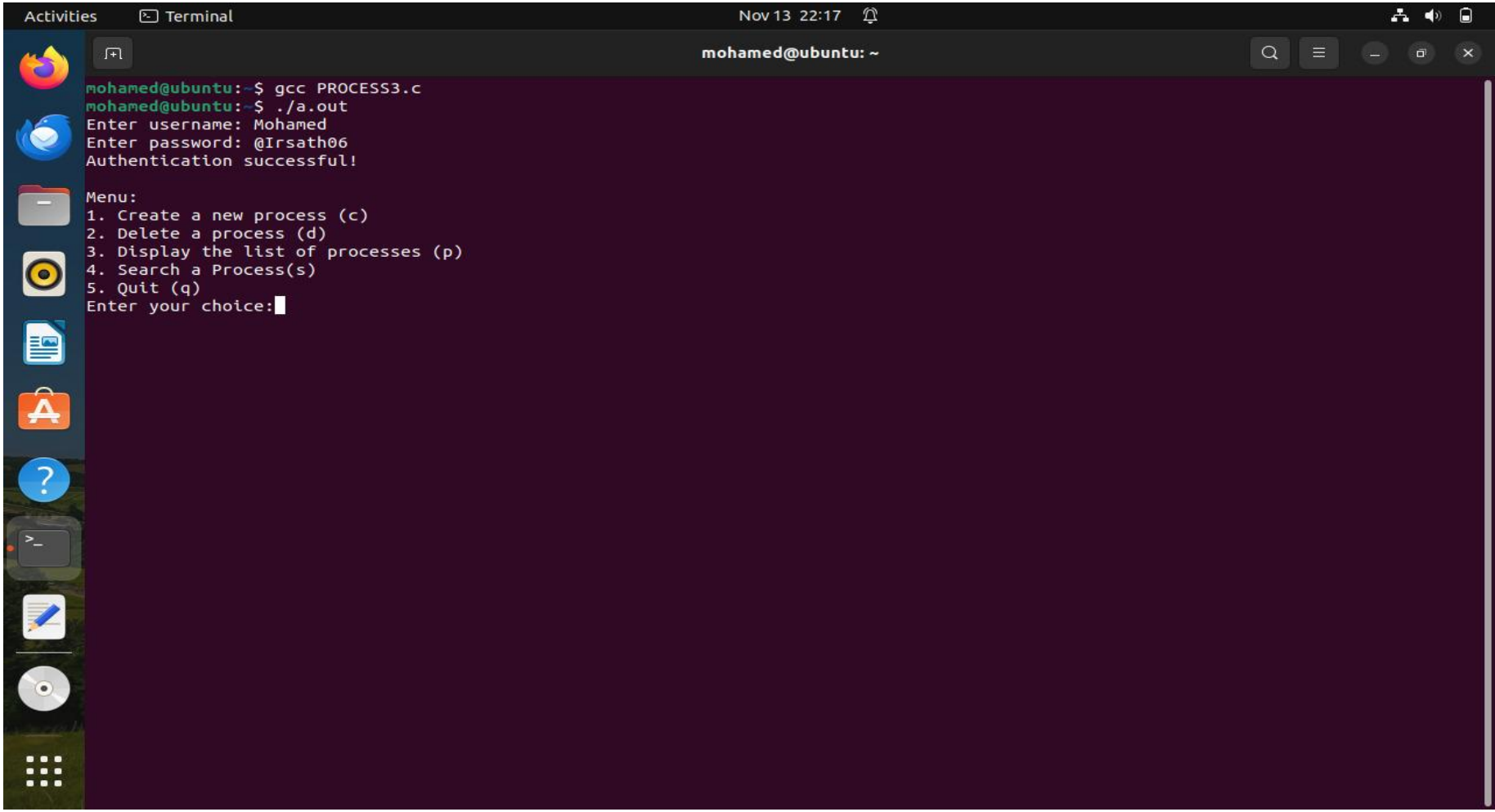
1. C compiler
2. GCC or Clang are recommended
3. Make utility
4. Libdirent library
5. Glibc library

Implementation

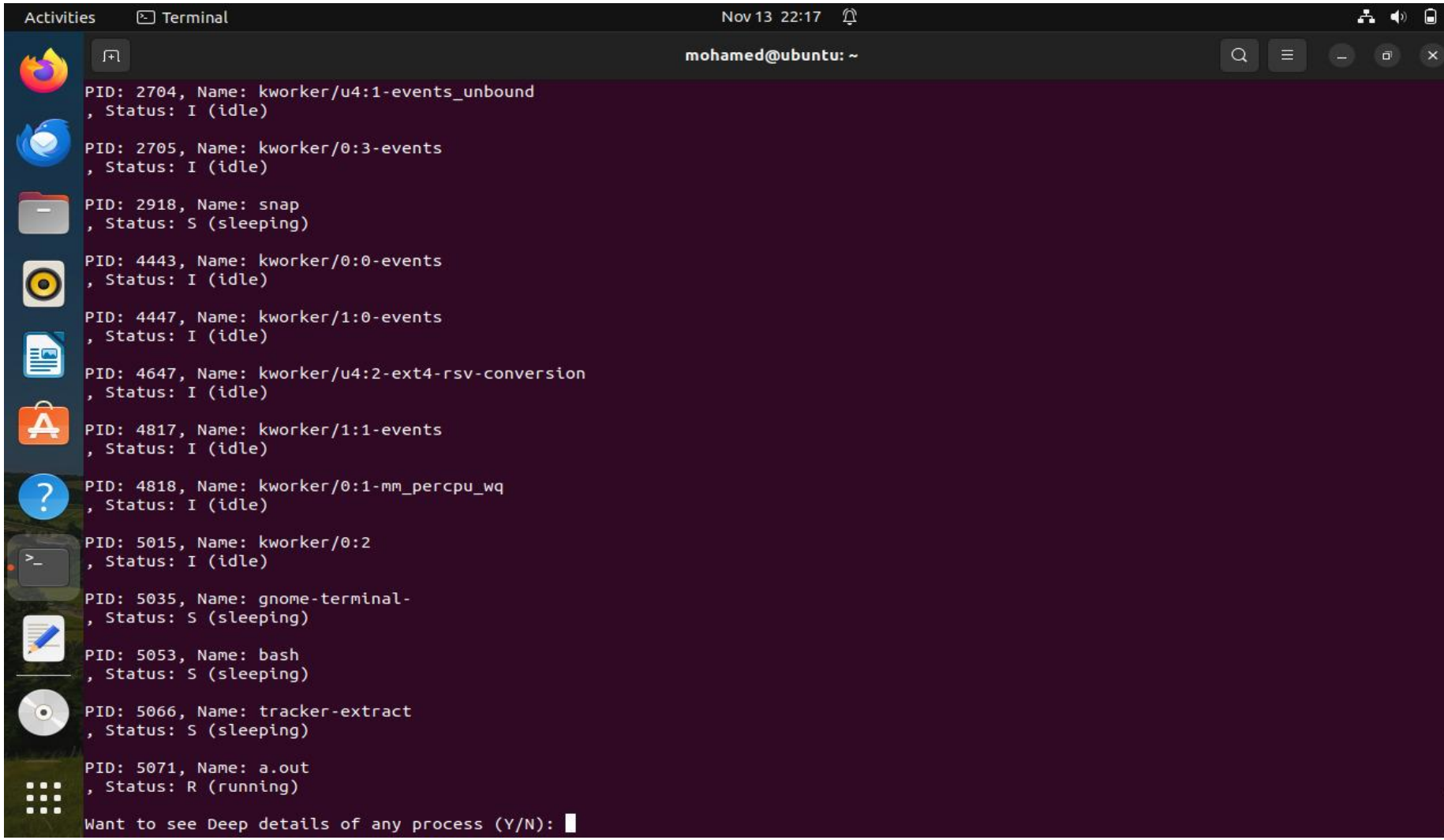
Start
-> Authenticate user
-> If authentication successful:
  -> While True:
    -> Display menu
    -> Get user choice
    -> If choice == 'c':
      -> Create a new process
      -> Add the new process to the BST
    -> Else if choice == 'd':
      -> Get the PID of the process to delete
      -> Delete the process from the BST
    -> Else if choice == 'p':
      -> Print the list of processes in the BST
    -> Else if choice == 's':
      -> Get the name of the process to search
      -> Search for the process in the BST and display the details if found
    -> Else if choice == 'q':
      -> Free the BST and exit the program
    -> Else:
      -> Display error message
  -> End While
-> End If

Results

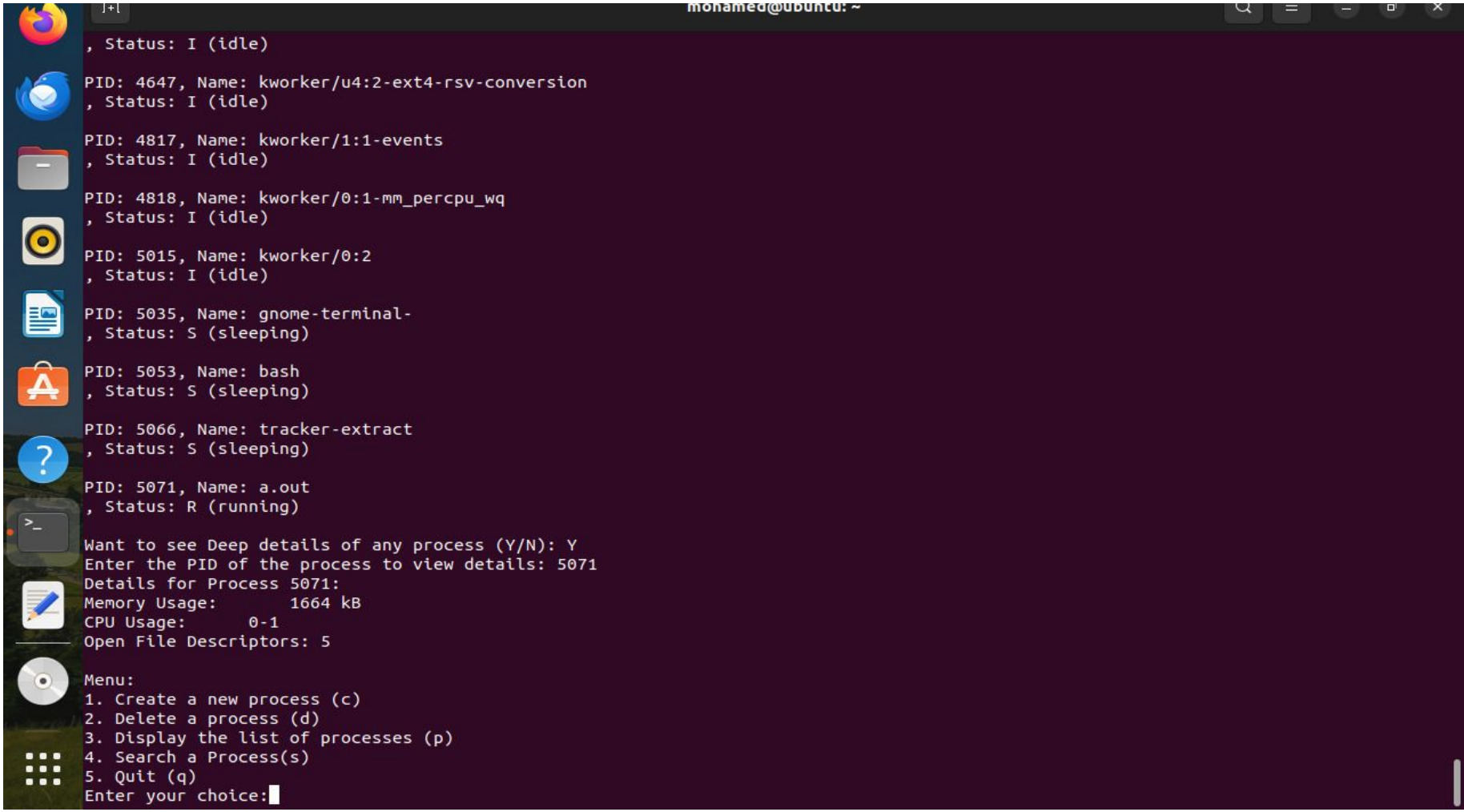
User authentication



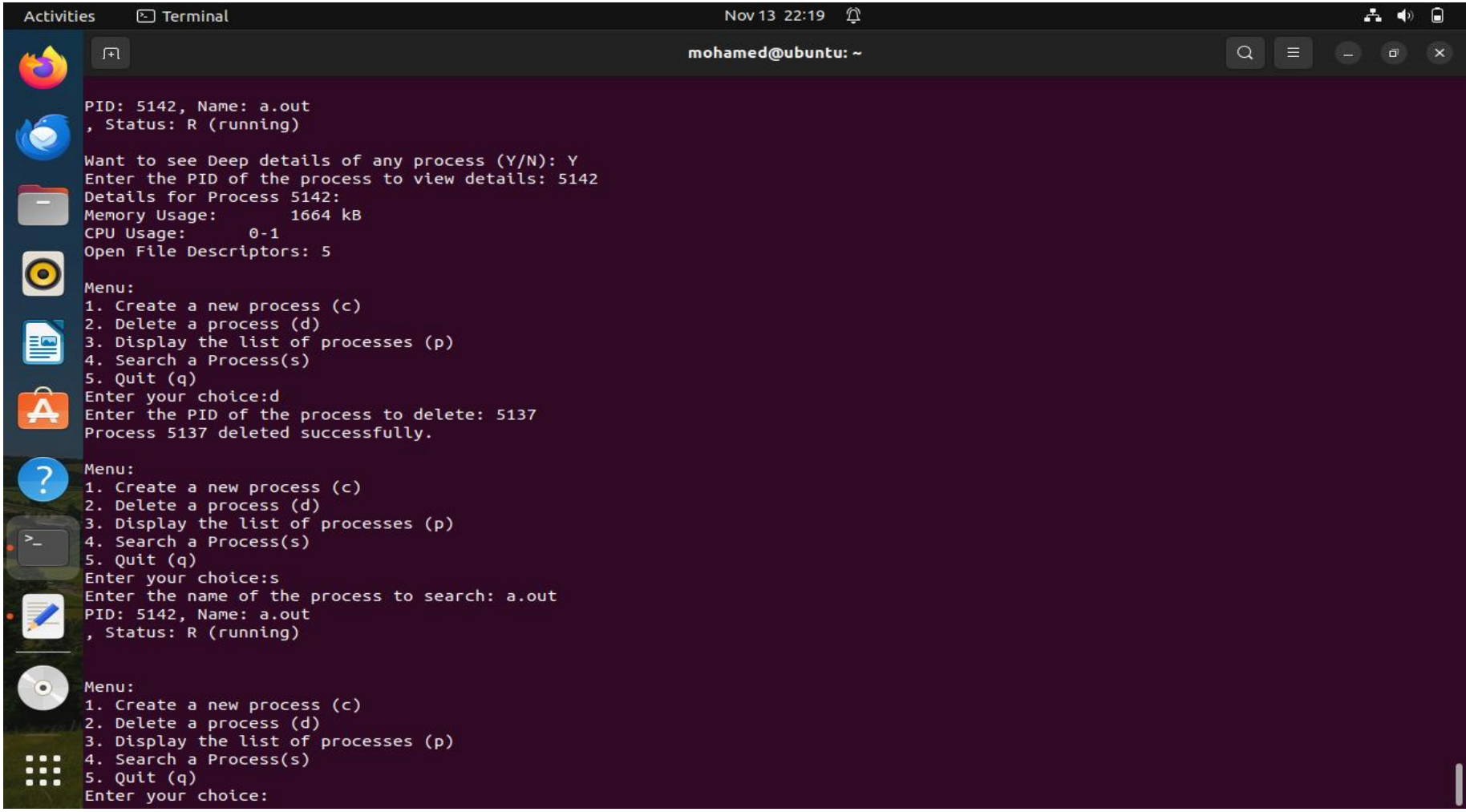
Displaying running process



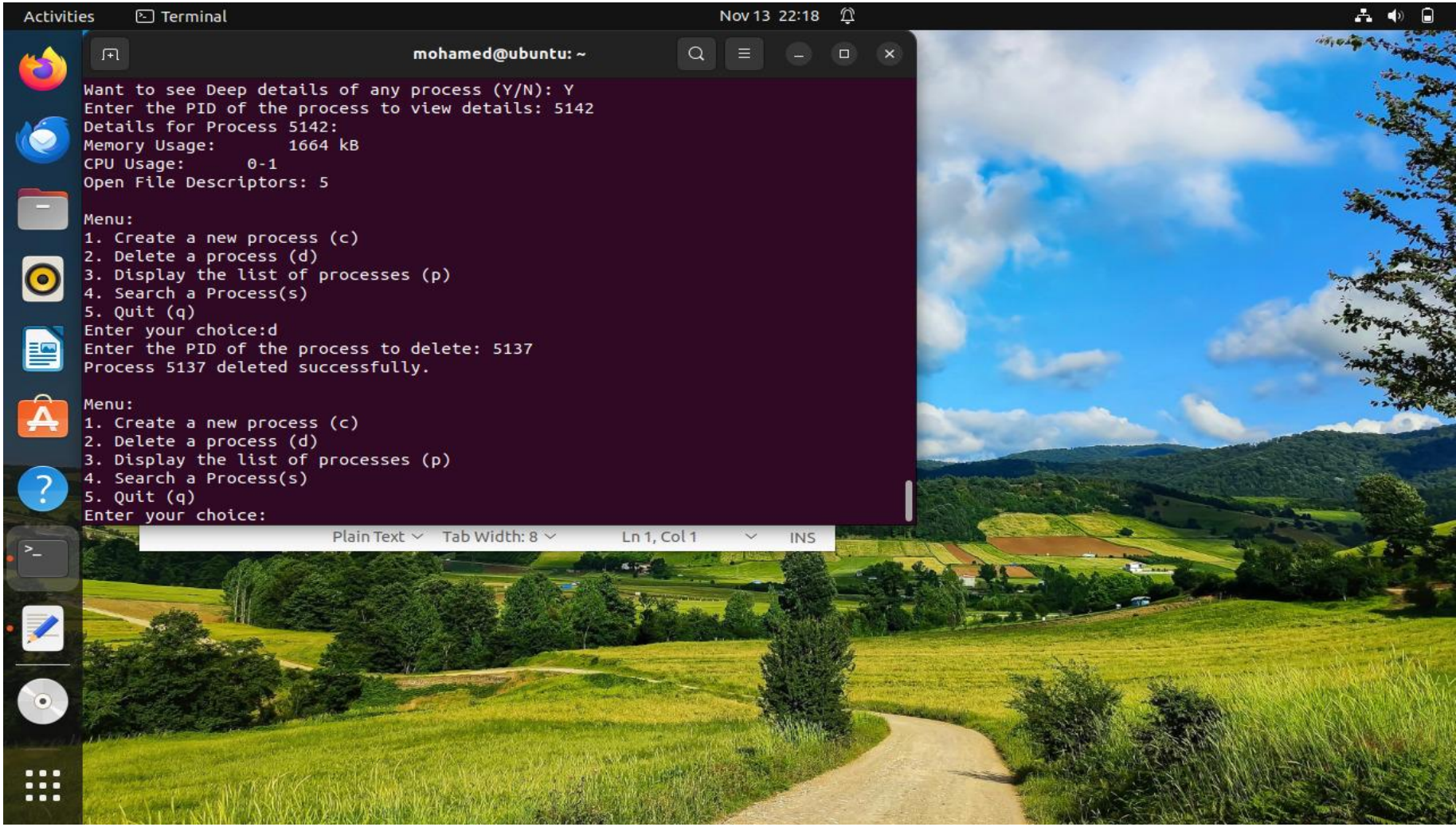
Deep details of specific process



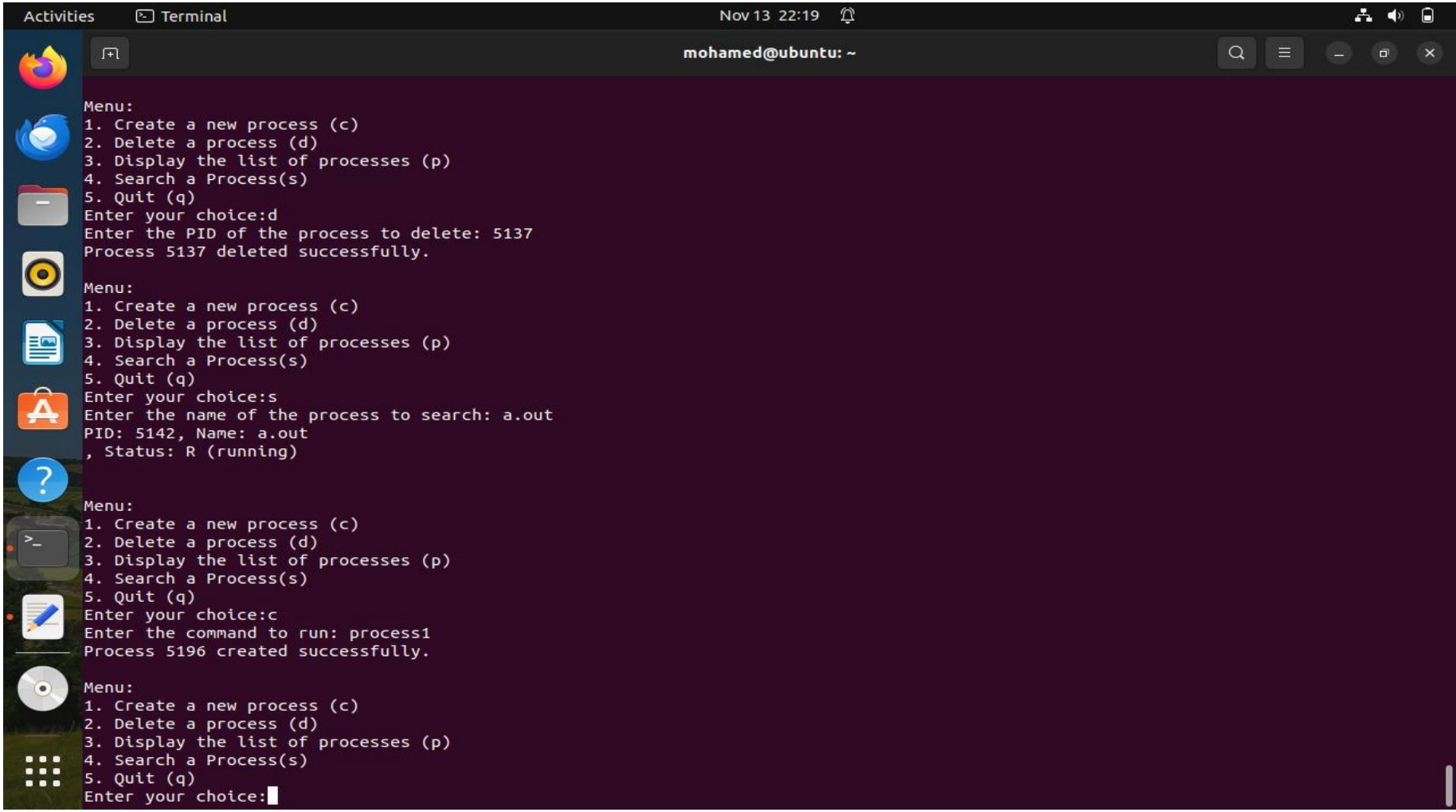
Searching a Process



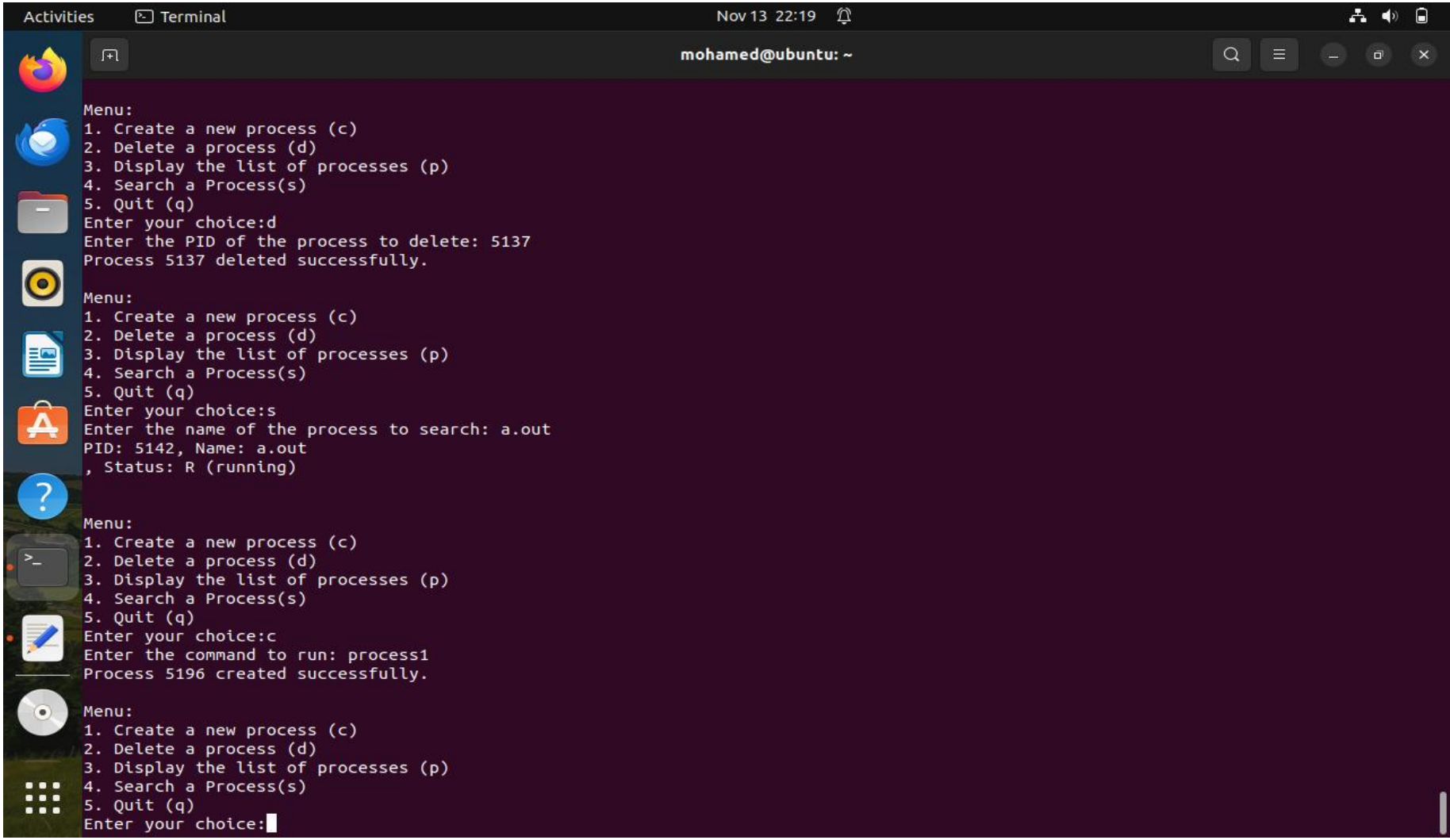
Deleting a process



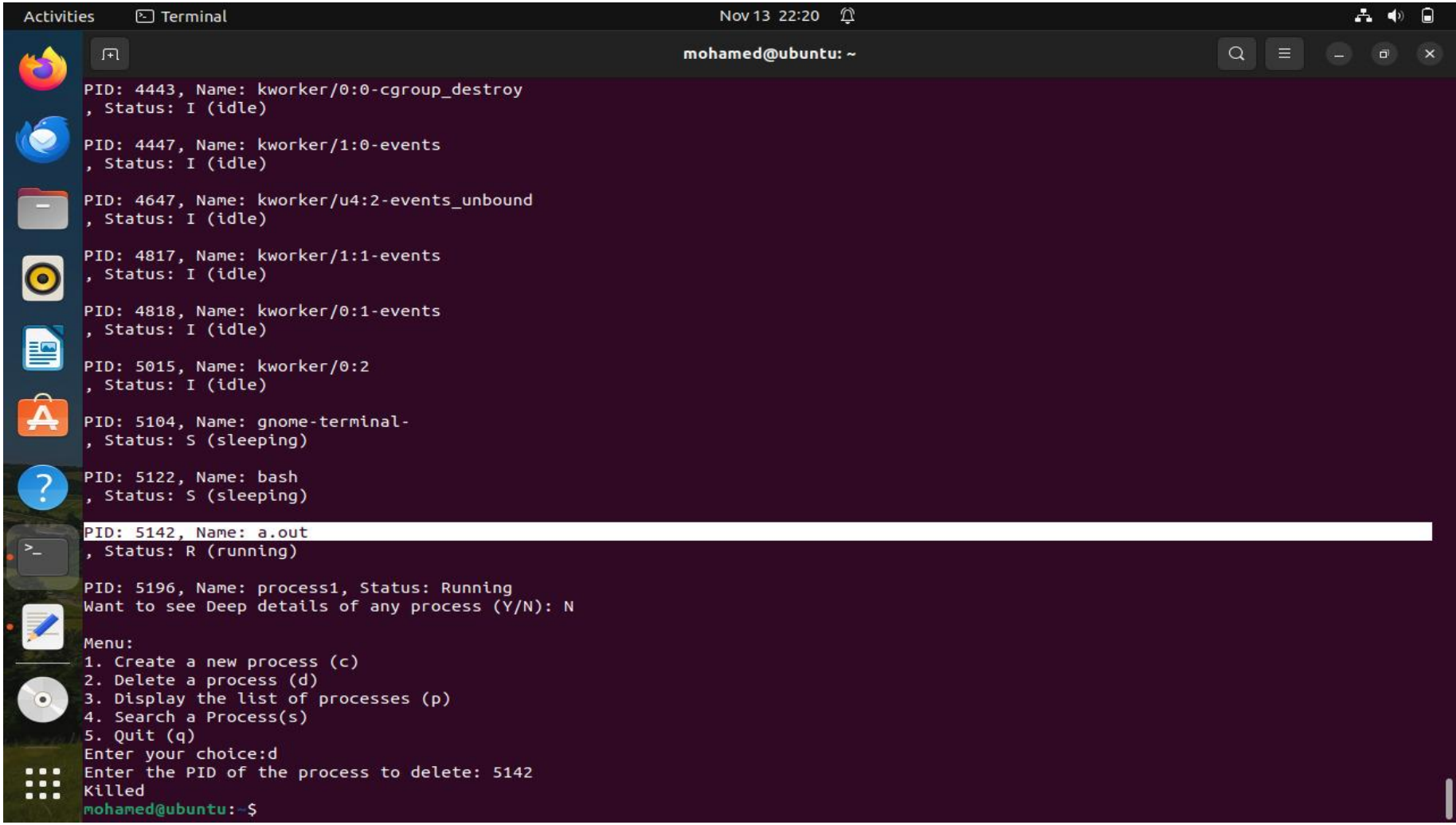
Creating a new process



Displaying after creating new process



Deleting the same process



Conclusion

In conclusion, the project provides a foundational implementation of a process management system with user authentication. It demonstrates fundamental concepts of process handling in a Unix environment and serves as a starting point for further development and refinement. The inclusion of an authentication mechanism adds a layer of security, making the system suitable for practical use in controlled environments. Further development could expand its functionality and usability.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be **University** *u/s 3 of UGC Act, 1956*

Thanks