Process Manager

Project 1: COMPSC 472

By Fernanda Martins

Implemented Functionalities

Create Process: This functionality allows users to enter a command in a graphical user interface (GUI) and create a new process using the subprocess module. The created process runs a simulated producer-consumer scenario with a separate thread.

List Processes: Users can view a list of all running processes, including their Process ID (PID), associated command, active threads, and the size of their respective queues.

Terminate Process: This feature allows users to terminate a specific process by providing its PID.

Installation and Usage

Installation

No specific installation is required for this project, as it utilizes the Python standard library. However, you should have Python 3.9 installed on your system.

Usage

Run the Python script provided.

The GUI application will open, providing three main options:

Create Process: Enter a command and click the "Create Process" button to initiate a new process.

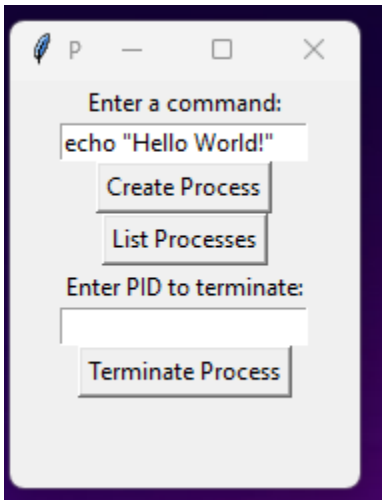
List Processes: Click the "List Processes" button to view information about all currently running processes.

Terminate Process: Enter the PID of the process you want to terminate and click the "Terminate Process" button.

Test Results

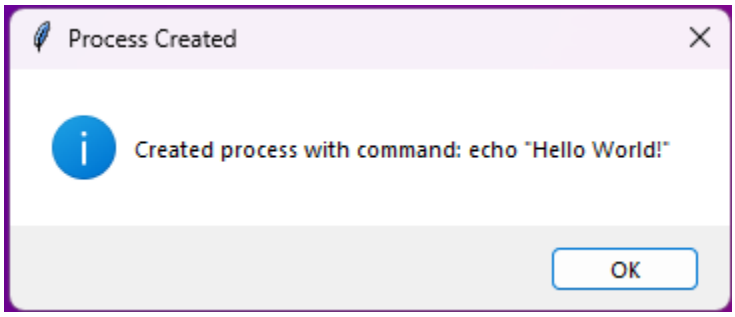
1. Create Process

Command: echo "Hello, World!"



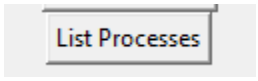
Expected Result: A new process is created, and the GUI displays a success message.

Actual Result: The process is successfully created, and a confirmation message is displayed.



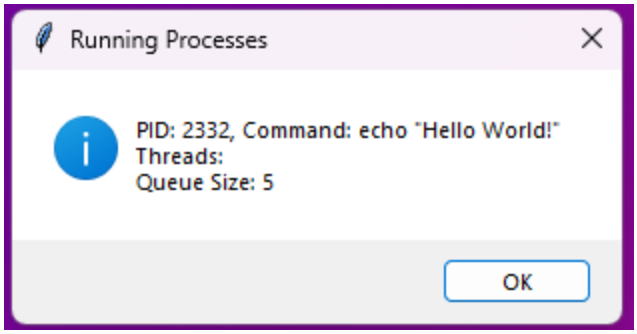
2. List Processes

Command: click “List Processes”



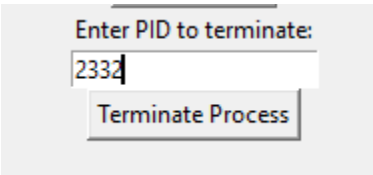
Expected Result: A list of all running processes with their PIDs, commands, thread IDs, and queue sizes.

Actual Result: The list of processes is displayed as expected.



3. Terminate Process

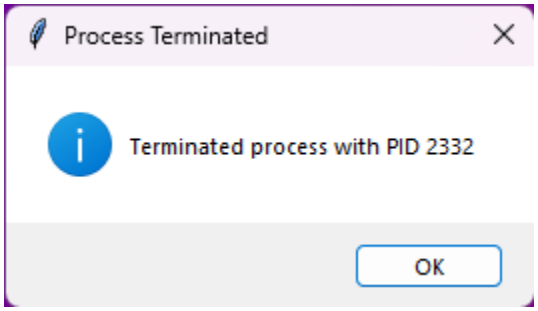
Command: Enter in the correct PID you'd like to run and click "Terminate Process"



Expected Result: Termination of a specific process with the provided PID.

Actual Result: The process is successfully terminated, and a confirmation message is displayed.

Discussion on the Project Result



The "Process Manager GUI" project provides a user-friendly interface to create, list, and terminate processes. It effectively demonstrates the use of threads, queues, and subprocesses in Python. Here are some points to consider:

Ease of Use: The GUI makes it easy for users to interact with running processes, which can be particularly useful in managing and monitoring background tasks.

Error Handling: The project incorporates error handling for various scenarios, such as invalid PIDs or command creation failures. Error messages are displayed to the user when necessary.

Multi-Threading: The project uses multiple threads to simulate a producer-consumer scenario for each created process, allowing for better resource utilization and showcasing multithreading capabilities.

Logging: The use of logging allows for easy tracking of activities and errors related to process creation and termination.

Improvement Suggestions:

-The ability to pass arguments and parameters to the command could enhance the functionality.

-Implementing a feature to monitor the output of running processes in real-time could be beneficial.

-Adding process priorities or resource limits could be considered for more advanced use cases.

-Creating a better GUI that looks better for users

-Better error handling for users if they are confused on what a command or PID should look like that they should enter into the prompt.

Overall, the "Process Manager GUI" is a functional project that can help users manage and interact with processes in a user-friendly manner. As I worked on this project, I gained valuable insights into process management in Python, threading, and GUI development. Further enhancements and customization options could make it even more powerful and versatile, expanding its utility in both personal and professional environments. Whether used for process monitoring, automation, or debugging, this project not only serves as a practical tool but also

provided me with a profound learning experience. It offers a solid foundation for continued development and refinement to meet a wide range of user requirements.