DESIGN DOCUMENTATION

PROCESS CONTROLLER

Table of Contents

1. Introduction

Process controller is a system software which takes the input and display the output accordingly, the user can give the program input and can perform multiple operations on the given program and can display various statistics information of that process by giving the process name.

1.1 Design Purpose

Process controller is a system software which takes the input and display the output accordingly, the user can give the program input and can perform multiple operations on the given program and can display various statistics information of that process by giving the process name.

1.2 Scope

The scope of the project is to create a process controller. This system consists of process which can be started and stopped according to user wish and perform multiple operations on it.

1.3 Functional Overview

1. **Process Synchronization**: It is the way by which processes that share the same memory space are managed in an operating system.
2. **Shared Memory in Linux:** All data related to the hotel such as room types, price, etc. are shared by multiple clients.
3. **Socket Programming in C – TCP:** Socket programming is a way of connecting two nodes, here the client and server, on a network to communicate with each other and coordinate the hotel booking activities.
4. **Support for statistics:** Server is responsible for the display of statistics related to availability of rooms such as number of rooms booked and vacant.
5. **I/O Multiplexing:** I/O multiplexing is the ability to perform I/O operations on multiple file descriptors.
6. **Logging and Debugging Framework:** Linux logs provide a timeline of events for a valuable troubleshooting tool when encountering issues.

2. Design Overview

2.1 Design Objectives

The Process Controller will comprise of modules depending on the various functionalities namely-

1. The entire process is menu driven.
2. The user can give the input and the process displays the output.
3. The client will have the menu option and the user can choose the option.
4. The server will perform the main task such as starting a process, stopping a process, temporarily suspend the program.
5. The server will also perform the task such as pause, unpause and kill the process.
5. The client can view the various statistics information of the selected program such as current running process and paused process.
6. The client can select a program to view the information such as memory occupied by that process and CPU time.
7. After completing the process, the user can exit.

2.2 Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.

2.2.1 Level-0 DFD

It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

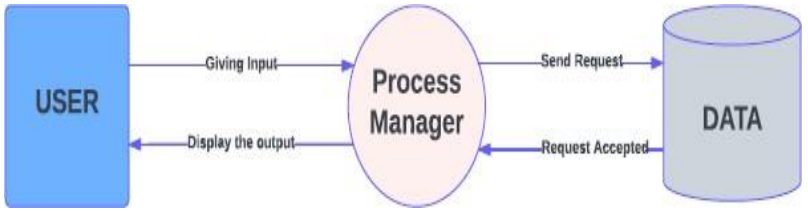

Fig 2.2 Level-0 DFD

2.2.2 Level-1 DFD

The context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

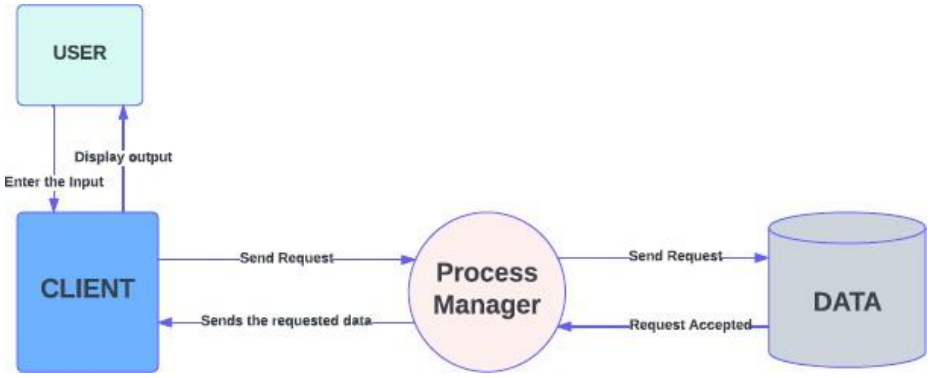

Fig 2.3 Level-1 DFD

2.3 User Interface Paradigm

- GUI: There is no GUI involved or created for the project.
- CLI: The application is wholly based on CLI, and the commands are given through it.
- The system follows a menu driven paradigm.

3. Detailed System Design

3.1 Functional Description

1. **Process Manager:** A process in Linux can go through different states after it's created and before it's terminated. The operating system must keep track of all the completing process, schedule themand dispatch them one after another.so the process manager monitors and controls different processes.

2. **PM application:** It will have a menu-based solution for accepting user input and displaying output,the user can select the options and can perform the required task and can exit the process.

3. **User input**: A list of programs can be given as an input by the user.

4. **User Output**: The user can enter the program name which they wish to run and get the requiredinformation.

5. **User display**: The user gives the input, based on the option selected by the user such as CPU time, memory occupied space and other related information will be processed by server and displayed tothe user.

6. **Appropriate menu options:** The process manager program will provide menu options for the userto select and give the input. Some of the menu options are start a given process, stop a given process, temporarily suspend the process and so on.

7. **Statistics information:** The process manager program will display the statistics information of theprogram which is entered by the user such as the information about the current running process,stopped process and paused process.

8. **PM app should be able to exit:** It should be able to exit once the current program is completed,about the completion of the current program the user cannot exit from the process.

9. **Display statistics for each process:** It should be able to display the statistical information based on the given program name, some of the statistical information are memory occupied by that process and the CPU time of the selected process.

10. **Running a process:** At any time only one process should be running. when there is a process running another process should not be running in the same process manager app.

11. **Control user program:** The PM app will be able to control any custom user programs as well as generic system programs such as ls, grep and others.

12. **Parameter's validation:** The PM app will be able to validate the parameters which are passed such as the command line argument which this very important for our program especially when we want to control the program from outside instead of hardcoding those values inside the code.

13. **Log various level of messages:** The PM app should be able to log various message logs contain messages about errors, warnings, the server including the kernel services and applications running on it.

14. **Error handling:** The PM app should be able to detect and resolve the errors which occurred during running a particular process by providing exit codes.

15. **No crashes in PM app:** The PM app should take care of the running program so that there are no crashes occurred during the process.

3.2 HLD

High Level Design in short HLD is the general system design means it refers to the overall system design. It describes the overall description/architecture of the application. It converts the Business/client requirement into High Level Solution.

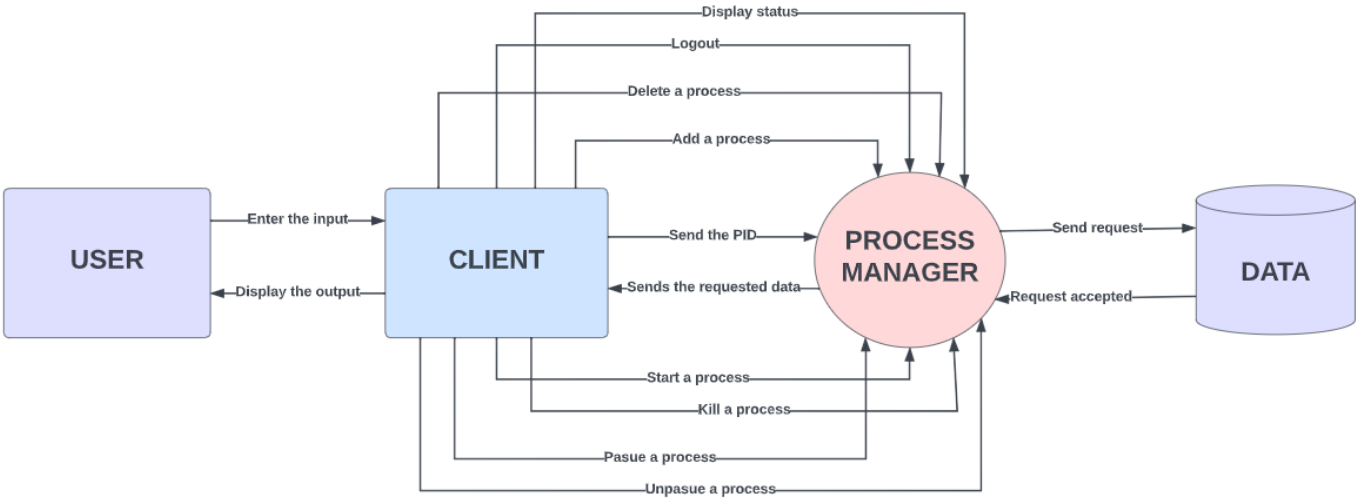

Fig 3.1 HLD

3.2 LLD

Low Level Design in short LLD is like detailing HLD means it refers to component-level design process. A detailed description of each module means it includes actual logic for every system component and it goes deep into each module's specification. It is also known as micro level/detailed design.

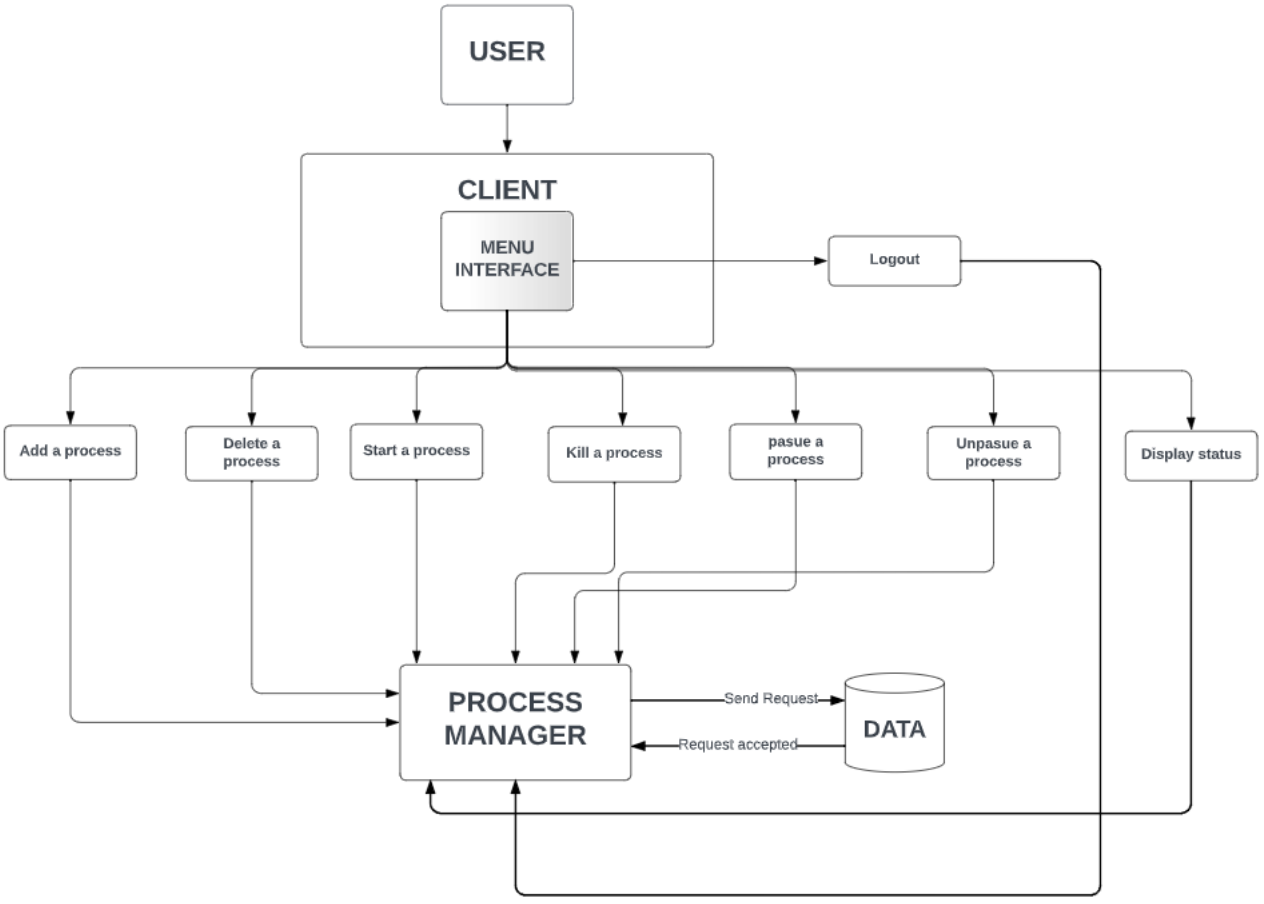

Fig: LLD Diagram

4. Environmental Description

3.3 Time Zone Support

The system supports all time zones.

3.4 Language Support

System supports only the English language.

3.5 User Desktop Requirements

Processor, RAM, Storage

4.3 Operating System Configuration

Windows and Linux OS

Ram :4 GB

Storage :256 GB SSD/HDD