PROCESS CONTROLLER

Software Requirement Specification (SRS) Document

Sprint 2 Implementation

Project Timeline: 02.01.2023 to 14.01.2023

INDEX

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 Purpose: -

The system works as a client - server based application. User/Client is required to enter the user input and then they can select the options to perform task of the selected program, they can view the statical information by giving the process name and can get the various information of the selected program such as memory occupied space and CPU time and the user can add or delete the program and can exit when the process is completed.

1.2 Intended Use: -

- Development Team
- Maintenance Team
- Clients

Since this a general-Purpose Software Thus any one Can access it.

1.3 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.

2. Overall Description: -

2.1 Project Features:

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

2.2 User Needs

1. User Characteristics: The user should be familiar with menu-driven Applications.
2. General Constraints: A full internet connection is required for Linux (Operating System).
3. Intended audience:
    a. Developers
    b. Project Manager

2.3 Operating Environment

The operating environment for the application is listed below

- Operating system: Any Linux-based OS

2.4 Assumptions and Dependency: -

- System should have any flavour of Linux installed.
- System should have either 4GB or more RAM.
- The service is used preferably on a desktop or laptop.

3. System Features and Requirements: -

3.1 Functionality: -

3.1.1 G6_FR01-> 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 them and dispatch them one after another.so the process manager monitors and controls different processes.

3.1.2 G6_FR02-> 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.

3.1.3 G6_FR03-> User input: A list of programs can be given as an input by the user. The user can enter the program name which they wish to run and get the required information.

3.1.4 G6_FR04-> 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 to the user.

3.1.5 G6_FR05-> Appropriate menu options: The process manager program will provide menu options for the user to 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.

3.1.6 G6_FR06-> Statistics information: The process manager program will display the statistics information of the program which is entered by the user such as the information about the current running process, stopped process and paused process.

3.1.7 G6_FR07-> 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 programme the user cannot exit from the process.

3.1.8 G6_FR08-> Display statistics of 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.

3.1.9 G6_FR09-> 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.

3.1.10 G6_FR10-> 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, This is the optional requirement.

3.1.11 G6_FR11-> 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 programme especially when we want to control the program from outside instead of hardcoding those values inside the code.

3.1.12 G6_FR12-> 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.

3.1.13 G6_FR13-> 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.

3.1.14 G6_FR14-> 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 System Requirements: -

System Requirements are types of functional requirements. These are features that are required for a system to function.

Software Interface:

- Operating System: Linux OS which supports networking.
- Connect protocol: TCP protocol

Hardware Interface:

Hardware requirements are:

- Processor: i3 or above
- ROM: 1TB (SSD/HDD)
- RAM: 4 GB or above

3.3 Functional Requirements:

3.3.1 G6_TR01 - Process Synchronization: It is the way by which processes that share the same memory space are managed in an operating system. Here, it ensures that multiple clients accessing the common data i.e., hotel data is synchronized, thereby avoiding conflicts.

3.3.2 G6_TR02 - Shared Memory in Linux: All data related to the hotel such as room types, price, etc. are shared by multiple clients. Mutex can be used for locking and unlocking the shared resources to avoid data corruptions and booking errors.

3.3.3 G6_TR03 - 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.

3.3.4 G6_TR04 - Support for statistics: Server is responsible for the display of statistics related to availability of rooms such as number of rooms booked and vacant.

3.3.5 G6_TR05 - I/O Multiplexing: I/O multiplexing is the ability to perform I/O operations on multiple file descriptors.

3.3.6 G6_TR06 - Logging and Debugging Framework: Linux logs provide a timeline of events for a valuable troubleshooting tool when encountering issues. When issues arise, analyzing log files facilitates debugging.