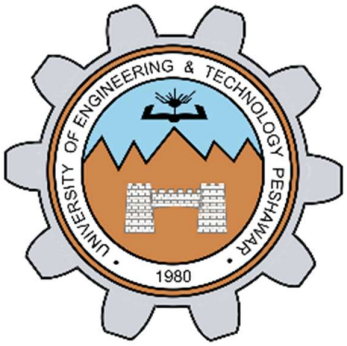
Comprehensive Process Management Tool

Project Proposal



Fall 2024

CSE-302L System Programming Lab

Group Members:

Ahsan Raza(22PWCSE2099**)**

Hassan Zaib Jadoon(22PWCSE2144**)**

Mutahhar Fayyaz(22PWCSE2176**)**

Class Section: **A**

Submitted to:

Engr. Abdullah Hamid

December 26, 2024

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Project Title:

Comprehensive Process Management Tool

Executive Summary:

The proposed project aims to develop a robust Process Management Tool in the C programming language. This tool will provide advanced functionality for managing and monitoring processes on Linux-based systems. It will cater to both system administrators and developers, offering features such as listing active processes, terminating unwanted or unresponsive processes, real-time monitoring of resource usage (CPU and memory), adjusting process priorities, retrieving detailed process information, and launching new processes. The tool will utilize Linux's /proc filesystem and system calls for efficient process management.

The tool is designed to be lightweight, efficient, and user-friendly, making it a valuable asset for system optimization and troubleshooting. It will cover all key characteristics of modern process management solutions while maintaining simplicity in its implementation and usability.

Objectives:

Core Functionalities:

List Active Processes: Provide a detailed view of all currently running processes with process IDs (PIDs) and command names.

Terminate Processes: Enable users to terminate processes by sending signals like SIGKILL or SIGTERM.

Monitor Resource Usage: Continuously display CPU and memory usage in real-time for system monitoring.

Get Process Details: Fetch and display detailed information about a specific process, including its state, memory usage, and runtime.

Start New Processes: Provide functionality to execute new commands and launch child processes.

Additional Features:

- Error handling to manage invalid inputs and unexpected system behavior.
- Clean and structured console-based interface for ease of use.
- Optimized performance for minimal resource usage by the tool itself.

Standard Libraries:

- <stdio.h>: For input and output operations.
- <stdlib.h>: For memory allocation and process control.
- <string.h>: For string manipulation.
- <unistd.h>: For POSIX API support.
- <dirent.h>: For reading directories (used in /proc).
- <signal.h>: For sending and handling process signals.
- <sys/resource.h>: For process priority management.
- <sys/types.h>: For data types used in system calls.
- <ctype.h>: For input validation.

Key Features and Functionalities:

1. List Running Processes:

Displays a list of active processes by reading /proc.

Shows PID, command name, and basic details.

2. Terminate a Process:

Sends user-defined signals (SIGKILL, SIGTERM, etc.) to a specified process.

Handles invalid PIDs and permissions gracefully.

3. Monitor Resource Usage:

Continuously tracks and displays CPU and memory usage.

Utilizes /proc/stat and /proc/meminfo for real-time data.

4. Start New Processes:

Executes user-specified commands using fork() and execlp().

Handles errors if the command is invalid or execution fails.

Benefits:

Efficient System Monitoring: The tool provides a real-time view of resource usage, enabling proactive system optimization.

Troubleshooting: Terminating unresponsive or rogue processes improves system stability.

Customizability: Modular design allows easy integration of new features.

Performance: Written in C for speed and low resource overhead.

Conclusion:

This project proposes the development of a feature-rich Process Management Tool that addresses critical system management needs. By providing comprehensive process monitoring, control, and prioritization capabilities, the tool will significantly enhance system performance and troubleshooting efficiency. With its lightweight design and advanced features, the tool will serve as a valuable resource for system administrators and developers alike.