

Develop a Multi-Threaded System Resource Monitor Using Linux and C

Project Description

Develop a C program that monitors and displays CPU usage, memory usage, and process count in real time. The program should run in the terminal and update system statistics every second. It must use multi-threading to separately handle CPU, memory, and process monitoring, ensuring efficient execution.

Functionality

1. Monitor System Resources:

- Read CPU usage statistics from `/proc/stat` and calculate the percentage of CPU utilization.
- Retrieve memory usage from `/proc/meminfo`, displaying both total and used memory.
- Count the number of currently running processes by scanning `/proc`.

2. Multi-Threading:

- Implement separate threads for CPU monitoring, memory monitoring, and process counting.
- Use pthread mutexes to ensure safe access to shared data.

3. Command-Line Options:

- Allow users to specify an update interval (e.g., `--interval 2` for updates every 2 seconds).
- Support `--log` mode to save statistics to a file (`system_stats.log`).

4. Graceful Exit:

- Handle Ctrl+C (SIGINT) and properly clean up threads before exiting.

Additional Requirements

- Use pthread library to implement multi-threading.
- Ensure proper thread synchronization using mutexes to prevent data races.
- Format the terminal output cleanly with aligned columns and percentages.
- Implement efficient file I/O for logging mode.

Example Output

System Resource Monitor

CPU Usage: 15.4%

Memory Usage: 1.2 GB / 8.0 GB (15.0%)

Running Processes: 154

(Updating every 1 second... Press Ctrl+C to exit)

If logging mode is enabled, statistics are appended to system_stats.log:

- [2025-02-03 12:00:01] CPU: 15.4%, Memory: 1.2GB/8.0GB, Processes: 154
- [2025-02-03 12:00:02] CPU: 18.2%, Memory: 1.3GB/8.0GB, Processes: 156

Mandatory Requirements

- Use C programming (no C++).
- Use Linux and GCC.
- Implement multi-threading using pthreads.
- Ensure memory safety (no leaks).
- Program must run on any Linux system without external dependencies.