Homework 4 Documentation

Abdullah TÜRKMEN

Student ID: 210104004072

Course: System Programming

Homework 4

May 12, 2025

1 Introduction

This program is a multithreaded log analyzer that reads lines from a log file, processes them using multiple worker threads, and searches for a specific term. It demonstrates the use of synchronization mechanisms, signal handling, and efficient thread communication.

2 Code Explanation

2.1 Buffer Management Functions

- **buffer_init:** Initializes the buffer, setting up its size, locks, and condition variables.
- **buffer_add:** Adds a line to the buffer, ensuring thread-safe access and handling full buffer scenarios.
- **buffer_get:** Retrieves a line from the buffer, ensuring thread-safe access and handling empty buffer scenarios.
- **buffer_destroy:** Cleans up the buffer by freeing memory and destroying locks and condition variables.

2.2 Log Analyzer Functions

- **manager_thread:** Reads lines from the log file and adds them to the buffer. Signals workers when done.
- **worker_thread:** Retrieves lines from the buffer and searches for the specified term. Updates match counts.
- **sigint_handler:** Handles SIGINT signals to gracefully stop the program and wake up waiting threads.
- **main:** Sets up the buffer, creates threads, and coordinates their execution. Displays the final results.

3 Screenshots

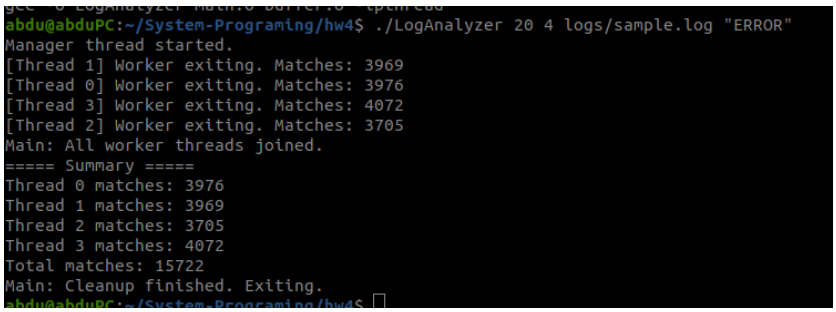

Figure 1: Program execution showing manager and worker threads.

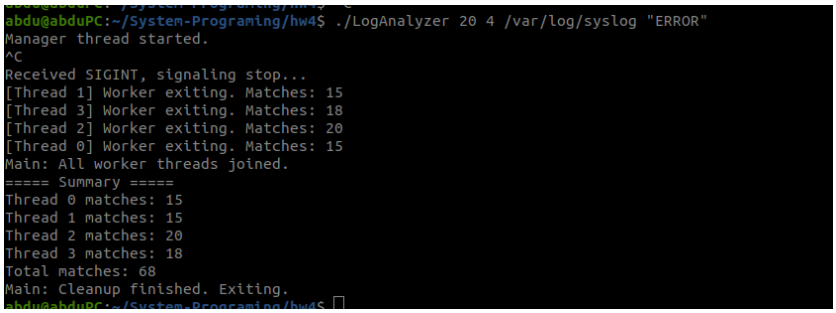

Figure 2: SIGINT signal handling and graceful termination.

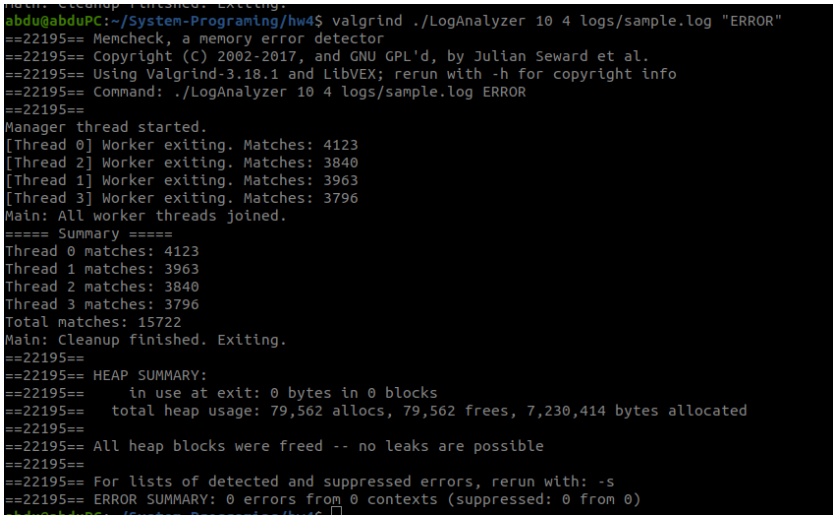

Figure 3: Program Execution with Valgrind

4 Conclusion

4.1 Challenges Faced

- Implementing thread synchronization and avoiding deadlocks.
- Handling SIGINT signals without leaving threads in a waiting state.

4.2 Solutions

- Used condition variables and mutexes to synchronize thread access to the buffer.
- Ensured all threads are woken up and terminated gracefully upon receiving a SIGINT signal.

4.3 Final Thoughts

This project provided valuable experience in multithreading, synchronization, and signal handling. The implementation successfully meets the requirements and handles edge cases effectively.