**Log monitor**

This log monitor is designed around a producer- consumer pattern. KeywordMatcher has the filtering logics, LogMonitor has the file I/O operations and comes paired with Gsuit for testing and Cmake for easy compilation.

I took on a polling based approach of about 10ms intervals to avoid buffer overflows. Memory efficiency was also prioritized over CPU usage to maintain stable resource profiles.

C++17 features such as std::stringview, std::unique\_ptr were used to reduce memory usage and maintenance of the 64kb read buffer. Atomic also provides thread safe control of the monitoring loop to allow clean shutdowns without mutex overhead. Std::signal was also used to stop the monitor cleanly.

The processBuffer method in log\_monitor.cpp implements a state machine that handles partial lines across buffer boundaries. when a newline isn't found before the buffer ends, the partial content is accumulated in partialLine\_ until completion. If a partial line exceeds 5000 characters, it is immediately processed and truncated, preventing memory exhaustion from malformed input. This design ensures that even a 10gb single line won't consume more than 5000 bytes of memory.

Google test was used to test on 3 scenarios of basic filtering functionality, line truncation behavior, and real-time updates for a total of 11 tests.

| **#** | **Test Name** | **What It Tests** | **Expected Result** |
| --- | --- | --- | --- |
| 1 | KeywordMatcherTest.MatchesSingleKeyword | Basic single keyword detection | Finds "key1" in "This line contains key1 somewhere" |
| 2 | KeywordMatcherTest.MatchesMultipleKeywords | Multiple keywords in same line | Detects both "key1" AND "key2" present |
| 3 | KeywordMatcherTest.NoMatchWhenKeywordAbsent | No false positives | Returns false for lines without keywords |
| 4 | KeywordMatcherTest.CaseSensitiveMatching | Case sensitivity enforcement | "key1" matches, "KEY1" and "Key1" don't |
| 5 | KeywordMatcherTest.KeywordAtDifferentPositions | Position independence | Finds keyword at start, middle, and end |
| 6 | KeywordMatcherTest.PartialMatchDoesNotCount | Substring matching behavior | "key" matches both "key" and "keyboard" |
| 7 | KeywordMatcherTest.VeryLongLine | Performance with long lines | Finds keyword in 20,000 character line |
| 8 | KeywordMatcherTest.EmptyKeywordList | Edge case handling | Returns false when no keywords configured |
| 9 | LogMonitorTest.BasicKeywordFiltering | End-to-end filtering | Writes matching lines, excludes non-matching |
| 10 | LogMonitorTest.TruncatesLongLines | 5000 character limit | Truncates 10,005 char line to ≤5000 chars |
| 11 | LogMonitorTest.HandlesRealTimeUpdates | Real-time monitoring | Detects logs as they're written progressively |

Made By Nicholas Loo