

### Assignment/Report Cover Sheet

Student Name	Student Id Number	Group	Signature
CHRIS HEZRON WILFRED CALVYN	83498	12	Chris
THOMAS SONG SHYANG FA	87054	12	
Mohamad Syahmi bin Soria	84553	12	Syahmi
MOHD RUZAINI BIN ABU BAKAR	86650	12	Ruzaini

Subject Code: <b>TMN4133 G1</b>	Subject Name : <b>System Programming</b>		
Assignment Title :	<b>Group Project</b>	Lecturer:	<b>Dr Johari Bin Abdullah</b>
Due Date : <b>18/12/2025</b>		Date Submitted: <b>18/12/2025</b>	

This cover sheet must be completed, signed and firmly attached to the front of the submission. All work must be submitted by the due date. If an extension of work is granted, an assignment extension acknowledgement slip must be signed by the lecturer/tutor and attached to assignment. Please note that is your responsibility to retain copies of your assignment.

***Plagiarism and Collusion are methods of cheating that falls under Peraturan Akademik Universiti Malaysia Sarawak para 11: Etika Akademik***

#### **Plagiarism**

Plagiarism is the presentation of work which has been copied in whole or in part from another person's work, or from any other source such as the internet, published books or periodicals without due acknowledgement given in the text.

#### **Collusion**

Collusion is the presentation of work that is the result in whole or in part of unauthorized collaboration with another person or persons.

Where there are reasonable grounds for believing that cheating has occurred, the only action that may be taken when plagiarism or collusion is detected is for the staff member not to mark the item of work and to report or refer the matter to the Dean. This may result in work being disallowed and given a fail grade or if the circumstances warrant, the matter may be referred to a Committee of inquiry for investigation. Such investigation may result in the matter being referred to the University Discipline Committee, **which** has the power to exclude a student.

**Upon placing signature above, I certify that I have not plagiarized the work of others or participated in unauthorized collusion when preparing this assignment.**

**I also certify that I have taken proper care in safeguarding my work and have made all reasonable efforts to ensure that my work not be able to be copied.**

**MARK :**

<b>1. Introduction.....</b>	<b>4</b>
<b>2. Part A : Coding Task.....</b>	<b>4</b>
2.1 Explanation of Modules.....	4
2.2 Advanced Technical Principles.....	5
2.3 Source Code Snippets.....	5
2.4 Flowchart of the sysmonitor.c.....	6
2.5 Compilation & Execution.....	6
<b>3. Part B : Testing and Evaluation.....</b>	<b>8</b>
3.1 Compilation and Build Test.....	8
3.2 Execution Test - Menu Mode.....	8
3.2.1. CPU Usage.....	9
3.2.2. Memory Usage.....	9
3.2.3. Top 5 Processes.....	10
3.2.4. Continuous Monitoring.....	11
3.2.5. Exit.....	12
3.3 Execution Tests - Command-Line Mode.....	13
3.3.1 ./sysmonitor -m cpu.....	13
3.3.2 ./sysmonitor -m mem.....	14
3.3.3 ./sysmonitor -m proc.....	14
3.3.4 ./sysmonitor -c 2.....	15
3.3.5 Ctrl + C.....	16
3.4 Signal Handling Test (Ctrl + C).....	16
3.4.1 Ctrl + C on Main Menu.....	16
3.4.2 Ctrl + C on CPU Usage.....	17
3.4.3 Ctrl + C on Memory Usage.....	18
3.4.4 Ctrl + C on Top 5 Processes.....	18
3.4.5 Ctrl + C on Continuous Monitoring.....	19
3.4.6 Verify Log.....	20
3.5 Error-Handling and Edge-Case Tests.....	21
3.5.1 Missing Argument (./sysmonitor -m).....	21
3.6 Log Verification.....	22
3.7 Summary Table for Report.....	25
<b>4. Part C : Reflection on AI.....</b>	<b>28</b>
<b>5. Part D : GitHub Collaboration.....</b>	<b>28</b>
Repository Setup :.....	28
<b>6. Part E : Presentation Summary.....</b>	<b>29</b>

## Title : SysMonitor++ — Linux System Resource Monitoring Tool

### 1. Introduction

Based on the project scenario, the team acts as junior system programmers at a tech company in Johor Bahru where servers are experiencing performance issues. The project aims to fulfill a management requirement for a simple, lightweight monitoring tool. SysMonitor++ is designed to be a lightweight application written entirely in C programming using native Linux system calls to minimize resource overhead. The primary objective of the project is to develop a command-line interface tool that interacts directly with the Linux /proc filesystem to retrieve real-time CPU usage, memory statistics and top five active processes. Additionally, the tool was designed to support continuous monitoring with configurable intervals, automated save to syslog.txt and using SIGINT (CTRL + C) to exit gracefully. The implementation was conducted within a Linux environment using the GCC compiler and text editors such as vi or nano. GitHub is used for version control to facilitate group collaboration and track development progress.

### 2. Part A : Coding Task

#### 2.1 Explanation of Modules

The source code is organized into logical modules, each responsible for a specific aspect of system telemetry or program flow. The application acts as a lightweight monitoring tool designed to address server performance issues with minimal resource overhead.

##### A. System Resource Modules

- CPU Usage Module (getCPUUsage & read\_cpu\_stats): Reads /proc/stat to calculate the percentage of time the CPU spent in "non-idle" states. It takes two snapshots one second apart and calculates the delta of total vs. idle ticks to determine the usage percentage.
- Memory Usage Module (getMemoryUsage): Parses /proc/meminfo to extract MemTotal and MemAvailable. It calculates used memory and converts values from Kilobytes to Megabytes for better readability.
- Process Monitoring Module (listTopProcesses): Iterates through numeric directories in /proc (representing PIDs). It parses the /proc/[PID]/stat file to find the process name and accumulated CPU time (utime + stime), then sorts them using qsort to display the top 5 consumers.

##### B. Utility & Infrastructure Modules

- Signal Handling: Uses an async-signal-safe handler (handleSignal) to catch SIGINT (Ctrl+C). This ensures the program can exit gracefully, especially during continuous monitoring loops.

- Logging Module (log\_message): Writes timestamped activity logs to syslog.txt using low-level I/O (open, write). This provides a persistent record of system checks.
- Input & UI Module: Includes get\_input\_int for robust, error-checked user input and displayMenu to provide a clean Command Line Interface (CLI).

## 2.2 Advanced Technical Principles

To fulfill the requirement for a lightweight tool, the architecture relies on several system programming core concepts:

- Direct Kernel Interaction: By interacting directly with the Linux /proc filesystem, the tool retrieves real-time statistics without needing specialized kernel-level modules.
- Low-Level I/O: The program prioritizes system calls over buffered standard library functions to minimize latency and memory footprint.
- Sampling Accuracy: The use of sleep(1) during CPU usage checks ensures that the resulting percentage represents a real-time snapshot rather than an average since the last system boot.

## 2.3 Source Code Snippets

### CPU Stats Parsing

The following snippet demonstrates how the tool interacts with the kernel's virtual filesystem to retrieve raw CPU ticks.

```
void read_cpu_stats(CPUStats *stats) {
    int fd = open("/proc/stat", O_RDONLY);
    char buffer[BUFFER_SIZE];
    ssize_t bytes_read = read(fd, buffer, sizeof(buffer) - 1);
    if (bytes_read > 0) {
        buffer[bytes_read] = '\0';
        sscanf(buffer, "cpu %llu %llu %llu %llu %llu %llu %llu %llu",
               &stats->user, &stats->nice, &stats->system,
               &stats->idle, &stats->iowait, &stats->irq,
               &stats->softirq, &stats->steal);
    }
    close(fd);
}
```

### Signal Handling for Graceful Exit

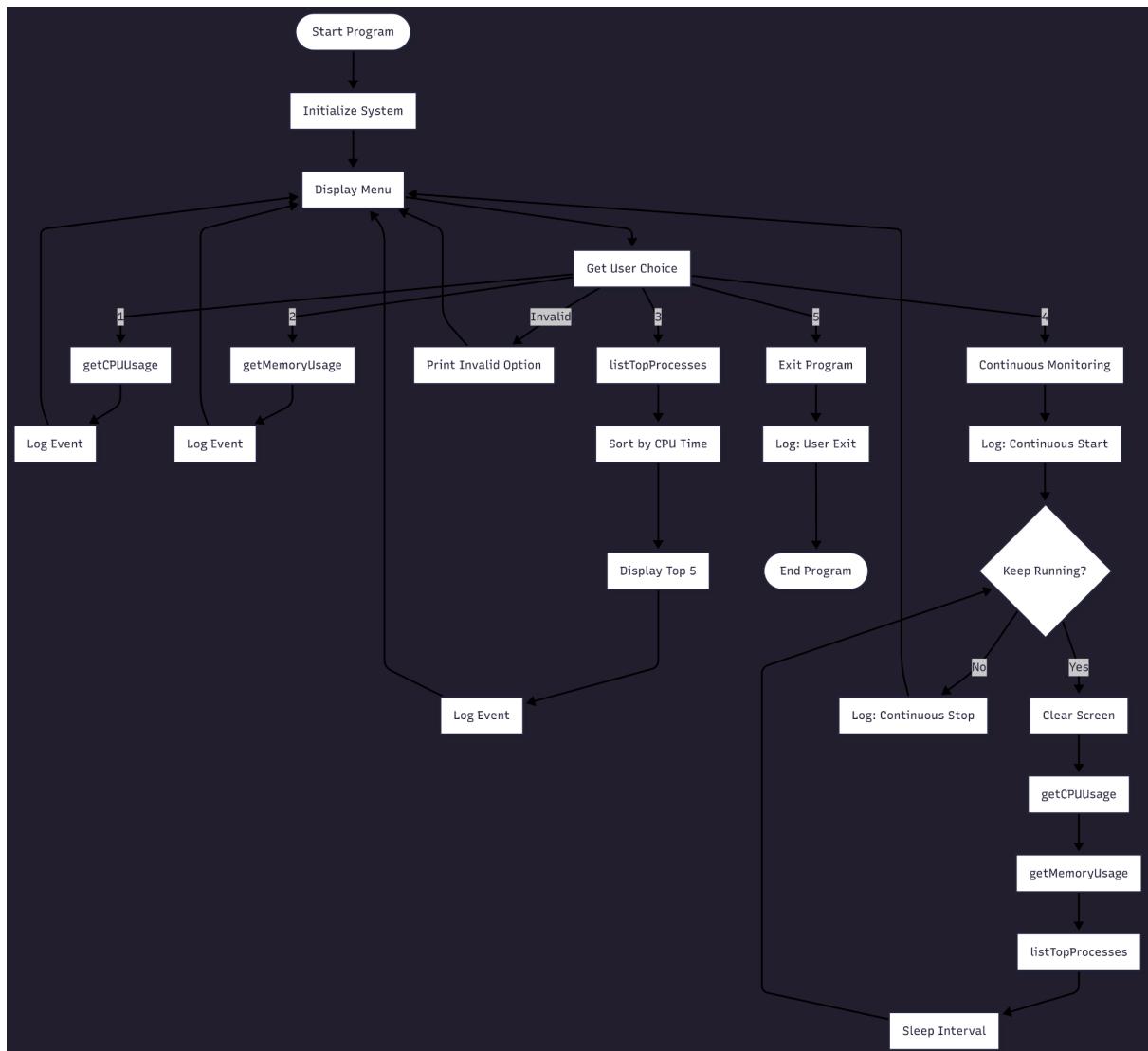
This ensures that the "Continuous Monitoring" mode doesn't leave the terminal in a messy state when interrupted.

```

void handleSignal(int sig) {
    if (sig == SIGINT) {
        keep_running = 0;
        const char *msg = "\nCaught SIGINT. Exiting gracefully...\n";
        write(STDOUT_FILENO, msg, strlen(msg));
    }
}

```

## 2.4 Flowchart of the sysmonitor.c



## 2.5 Compilation & Execution

To compile and run SysMonitor++, follow the standard GCC workflow.

### Step 1: Compilation

Use the gcc compiler. The -o flag names the output binary.

```
gcc sysmonitor.c -o sysmonitor
```

## Step 2: Running the Program

You can run the program in two modes:

Interactive Mode:

```
./sysmonitor
```

Continuous Mode (e.g., refresh every 2 seconds):

```
./sysmonitor -c 2
```

## Execution Screenshots (Visual Representations)

Main Menu:

```
=====
      SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice:
```

CPU & Memory Output

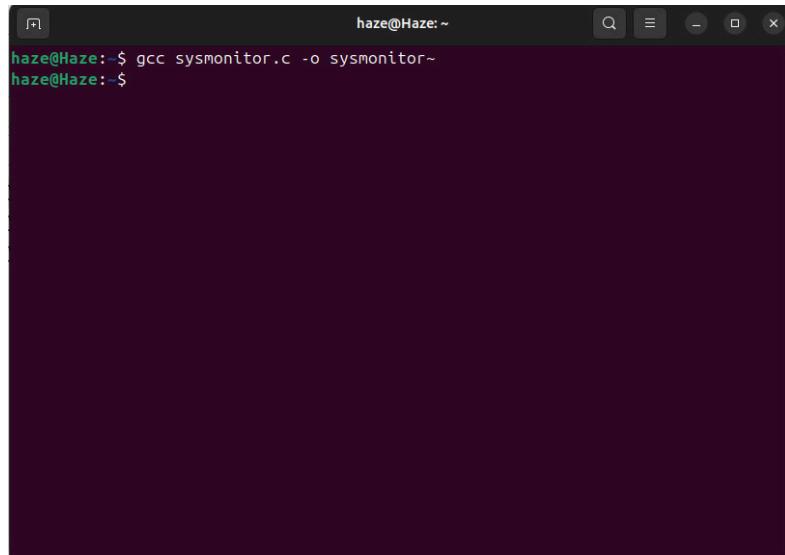
Top Processes Output:

```
--- Top 5 Processes (by Accumulated CPU Time) ---
 PID      Name          CPU Time (ticks)
 1245    gnome-shell    84291
 2930    chrome        56210
 902     Xorg          43102
 3110    code          21093
 1       systemd       10234
```

### **3. Part B : Testing and Evaluation**

#### **3.1 Compilation and Build Test**

The code successfully compiled without errors or warnings. The compilation was performed using the command [gcc sysmonitor.c -o sysmonitor]. A screenshot demonstrating the successful compilation has been included as evidence.

A screenshot of a terminal window titled "haze@Haze:~". The window contains the command "gcc sysmonitor.c -o sysmonitor~" followed by a new line character. The background of the terminal is dark, and the text is white.

*Figure 3.1 Sucessful Compilation*

#### **3.2 Execution Test - Menu Mode**

By using command ./sysmonitor it gave the expected output:

Program displays the main menu with numbered options:

1. CPU Usage
2. Memory Usage
3. Top 5 Processes
4. Continuous Monitoring
5. Exit

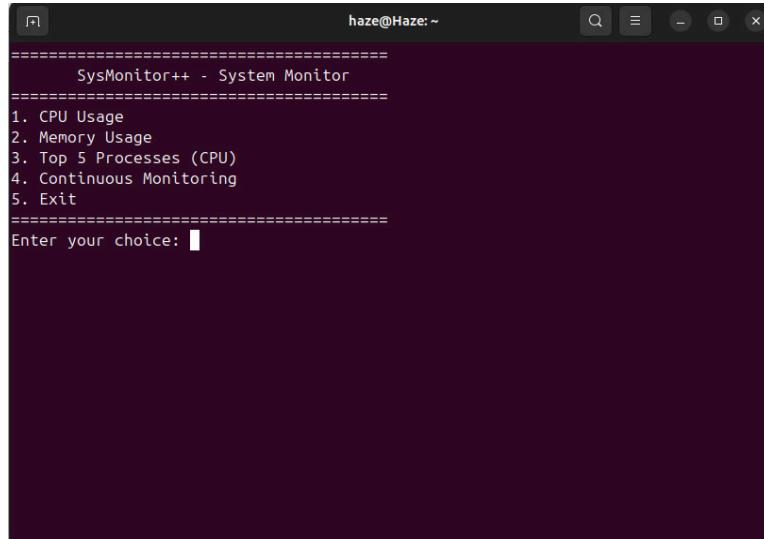


Figure 3.2 Main Menu for The System

### 3.2.1. CPU Usage

Selecting option '1' from the main menu executes the CPU Usage monitoring function. This test involves verifying that the program correctly reads and displays the current system-wide CPU utilization, typically showing both user and system time percentages. The expected output should provide a clear, real-time snapshot of how heavily the CPU is being utilized, confirming the feature's successful implementation as designed.

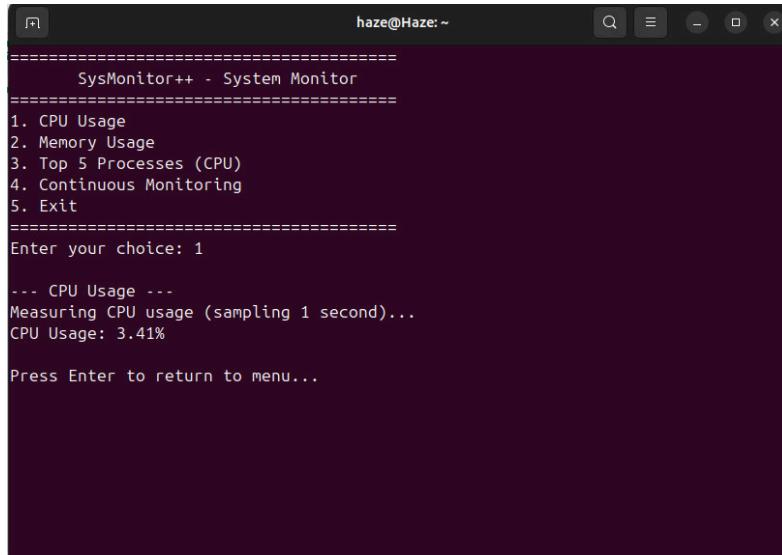
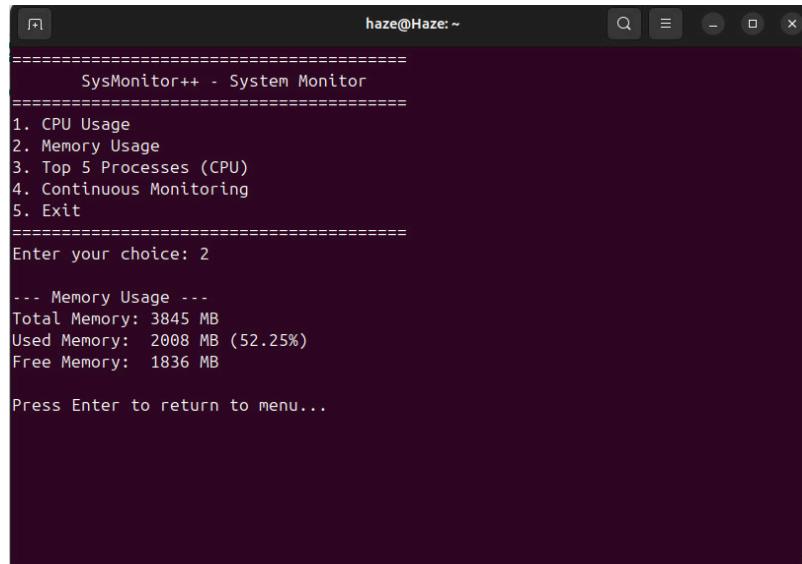


Figure 3.2.1 Monitor CPU Usage

### 3.2.2. Memory Usage

Selecting option '2' from the main menu triggers the Memory Usage function. This test verifies the program's ability to retrieve and display key memory statistics from the system accurately. The expected output includes the total physical memory, the amount of memory currently in use, and the remaining free memory, potentially broken down by buffers/cache.

The successful execution confirms that the memory monitoring feature is functional and provides users with a comprehensive view of the system's memory allocation.



A screenshot of a terminal window titled "haze@Haze:~". The window displays the output of the SysMonitor++ System Monitor. The menu options are listed as follows:

```
=====
SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
```

The user has selected option 2, "Memory Usage". The output shows the following memory statistics:

```
Enter your choice: 2

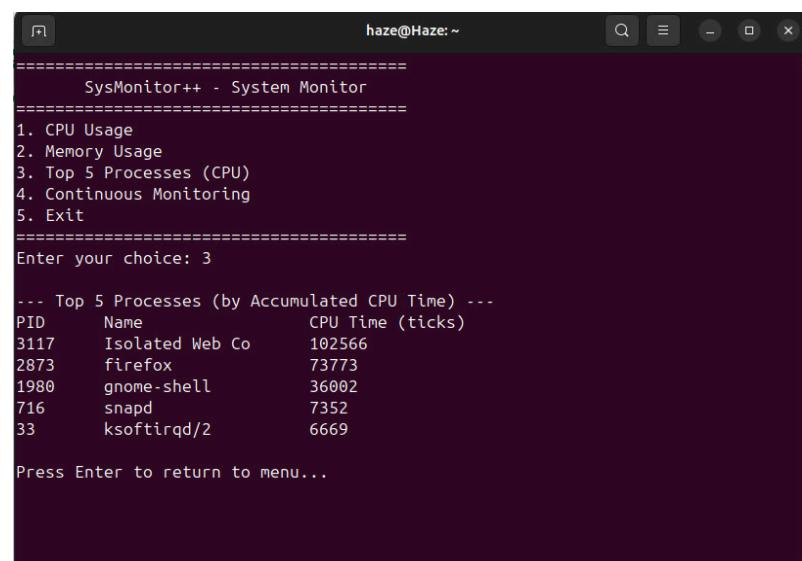
--- Memory Usage ---
Total Memory: 3845 MB
Used Memory: 2008 MB (52.25%)
Free Memory: 1836 MB

Press Enter to return to menu...
```

Figure 3.2.2 Monitor Memory Usage

### 3.2.3. Top 5 Processes

Selecting option '3' from the main menu activates the Top 5 Processes function. This test aims to confirm that the program can successfully identify and list the top five most resource-intensive processes currently running on the system, typically based on CPU usage. The expected output should display a table or list containing the process ID (PID), the user running the process, the percentage of CPU and/or memory being consumed, and the command name for each of the top 5 processes. Successful execution validates the feature's capability to assist users in identifying resource bottlenecks.



A screenshot of a terminal window titled "haze@Haze:~". The window displays the output of the SysMonitor++ System Monitor. The menu options are listed as follows:

```
=====
SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
```

The user has selected option 3, "Top 5 Processes (CPU)". The output shows a table of the top 5 processes by accumulated CPU time:

```
Enter your choice: 3

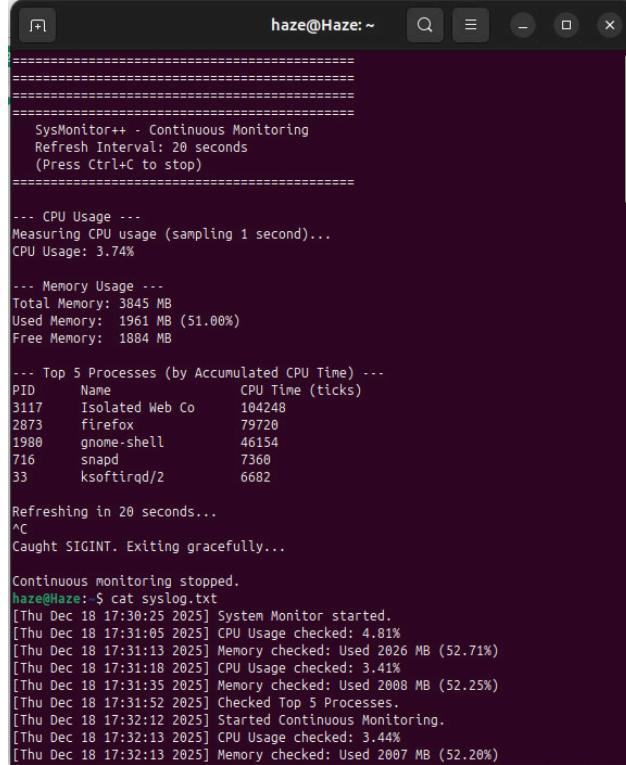
--- Top 5 Processes (by Accumulated CPU Time) ---
PID      Name          CPU Time (ticks)
3117    Isolated Web Co    102566
2873    firefox        73773
1980    gnome-shell     36002
716     snapd         7352
33      ksoftirqd/2    6669

Press Enter to return to menu...
```

Figure 3.2.3 Monitor Top 5 Processes

### 3.2.4. Continuous Monitoring

Selecting option '4' from the main menu initiates the Continuous Monitoring feature. This test is crucial for verifying the program's ability to repeatedly gather and display the system metrics (CPU usage, memory usage, and top processes) at a predefined interval until the user decides to stop the monitoring (by pressing 'Ctrl+C'). The expected outcome is a continuously updating display that provides a near-real-time view of system performance, ensuring the program correctly handles the loop mechanism and the termination signal.



The screenshot shows a terminal window titled 'haze@Haze:~'. The window displays the output of the 'SysMonitor++ - Continuous Monitoring' application. It starts with a header: 'SysMonitor++ - Continuous Monitoring', 'Refresh Interval: 20 seconds', and '(Press Ctrl+C to stop)'. Below this, it shows CPU Usage statistics: 'CPU Usage ...', 'Measuring CPU usage (sampling 1 second)...', and 'CPU Usage: 3.74%'. Then, it shows Memory Usage statistics: 'Memory Usage ...', 'Total Memory: 3845 MB', 'Used Memory: 1961 MB (51.00%)', and 'Free Memory: 1884 MB'. Following this, it lists the 'Top 5 Processes (by Accumulated CPU Time)' with their PID, Name, and CPU Time (ticks). The processes listed are: Isolated Web Co (PID 3117), firefox (PID 2873), gnome-shell (PID 1988), snapd (PID 716), and ksoftirqd/2 (PID 33). The application then indicates it is 'Refreshing in 20 seconds...', receives a '^C' interrupt, and exits gracefully. It then states 'Continuous monitoring stopped.' and shows the contents of 'syslog.txt' with log entries from December 18, 2025, detailing CPU and memory usage checks and process monitoring. The terminal window has a dark background with light-colored text and standard window controls at the top.

```
=====
SysMonitor++ - Continuous Monitoring
Refresh Interval: 20 seconds
(Press Ctrl+C to stop)
=====

--- CPU Usage ---
Measuring CPU usage (sampling 1 second)...
CPU Usage: 3.74%

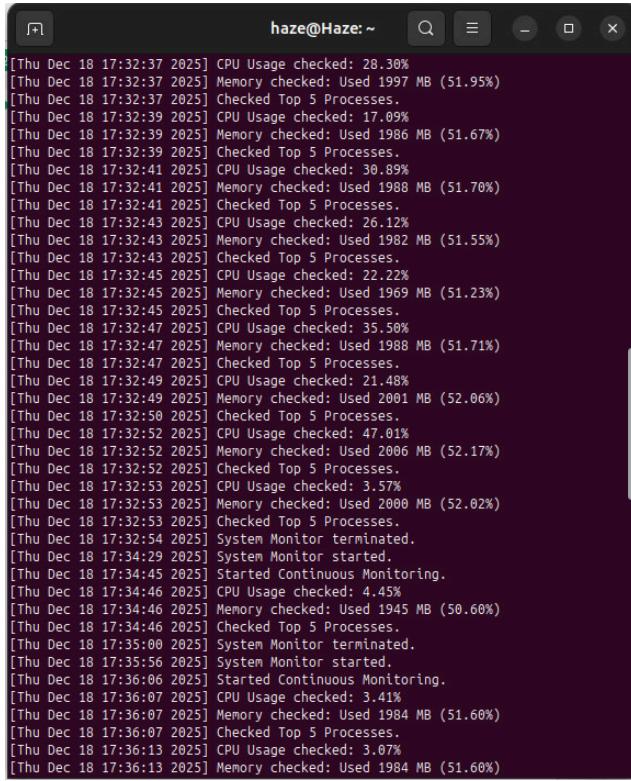
--- Memory Usage ---
Total Memory: 3845 MB
Used Memory: 1961 MB (51.00%)
Free Memory: 1884 MB

--- Top 5 Processes (by Accumulated CPU Time) ---
PID      Name          CPU Time (ticks)
3117    Isolated Web Co   164248
2873    firefox        79720
1988    gnome-shell     46154
716     snapd          7360
33      ksoftirqd/2     6682

Refreshing in 20 seconds...
^C
Caught SIGINT. Exiting gracefully...

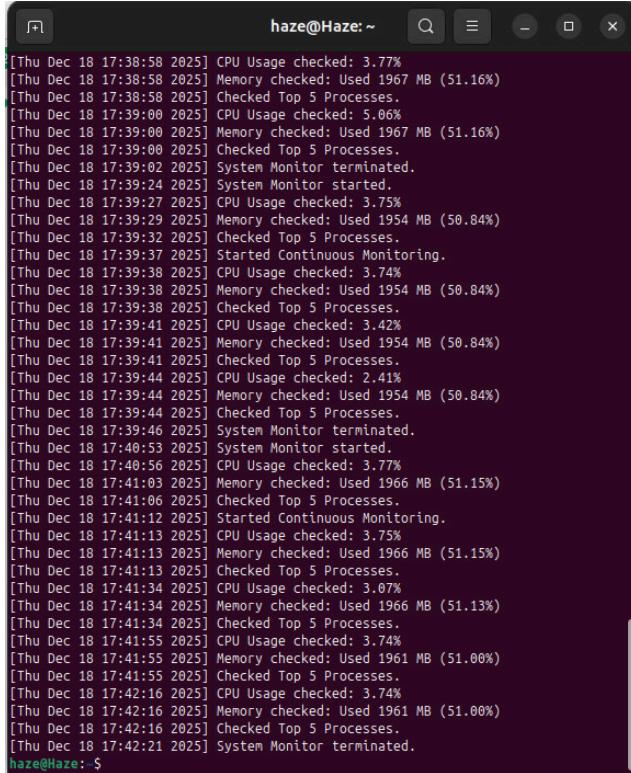
Continuous monitoring stopped.
haze@Haze:~$ cat syslog.txt
[Thu Dec 18 17:30:25 2025] System Monitor started.
[Thu Dec 18 17:31:05 2025] CPU Usage checked: 4.81%
[Thu Dec 18 17:31:13 2025] Memory checked: Used 2026 MB (52.71%)
[Thu Dec 18 17:31:18 2025] CPU Usage checked: 3.41%
[Thu Dec 18 17:31:35 2025] Memory checked: Used 2008 MB (52.25%)
[Thu Dec 18 17:31:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:12 2025] Started Continuous Monitoring.
[Thu Dec 18 17:32:13 2025] CPU Usage checked: 3.44%
[Thu Dec 18 17:32:13 2025] Memory checked: Used 2007 MB (52.20%)
```

Figure 3.2.4.1 Continuous Monitoring



```
[Thu Dec 18 17:32:37 2025] CPU Usage checked: 28.30%
[Thu Dec 18 17:32:37 2025] Memory checked: Used 1997 MB (51.95%)
[Thu Dec 18 17:32:37 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:39 2025] CPU Usage checked: 17.09%
[Thu Dec 18 17:32:39 2025] Memory checked: Used 1986 MB (51.67%)
[Thu Dec 18 17:32:39 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:41 2025] CPU Usage checked: 30.89%
[Thu Dec 18 17:32:41 2025] Memory checked: Used 1988 MB (51.70%)
[Thu Dec 18 17:32:41 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:43 2025] CPU Usage checked: 26.12%
[Thu Dec 18 17:32:43 2025] Memory checked: Used 1982 MB (51.55%)
[Thu Dec 18 17:32:43 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:45 2025] CPU Usage checked: 22.22%
[Thu Dec 18 17:32:45 2025] Memory checked: Used 1969 MB (51.23%)
[Thu Dec 18 17:32:45 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:47 2025] CPU Usage checked: 35.50%
[Thu Dec 18 17:32:47 2025] Memory checked: Used 1988 MB (51.71%)
[Thu Dec 18 17:32:47 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:49 2025] CPU Usage checked: 21.48%
[Thu Dec 18 17:32:49 2025] Memory checked: Used 2001 MB (52.06%)
[Thu Dec 18 17:32:50 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:52 2025] CPU Usage checked: 47.01%
[Thu Dec 18 17:32:52 2025] Memory checked: Used 2006 MB (52.17%)
[Thu Dec 18 17:32:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:53 2025] CPU Usage checked: 3.57%
[Thu Dec 18 17:32:53 2025] Memory checked: Used 2000 MB (52.02%)
[Thu Dec 18 17:32:53 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:54 2025] System Monitor terminated.
[Thu Dec 18 17:34:29 2025] System Monitor started.
[Thu Dec 18 17:34:45 2025] Started Continuous Monitoring.
[Thu Dec 18 17:34:46 2025] CPU Usage checked: 4.45%
[Thu Dec 18 17:34:46 2025] Memory checked: Used 1945 MB (50.60%)
[Thu Dec 18 17:34:46 2025] Checked Top 5 Processes.
[Thu Dec 18 17:35:00 2025] System Monitor terminated.
[Thu Dec 18 17:35:56 2025] System Monitor started.
[Thu Dec 18 17:36:06 2025] Started Continuous Monitoring.
[Thu Dec 18 17:36:07 2025] CPU Usage checked: 3.41%
[Thu Dec 18 17:36:07 2025] Memory checked: Used 1984 MB (51.60%)
[Thu Dec 18 17:36:07 2025] Checked Top 5 Processes.
[Thu Dec 18 17:36:13 2025] CPU Usage checked: 3.07%
[Thu Dec 18 17:36:13 2025] Memory checked: Used 1984 MB (51.60%)
```

Figure 3.2.4.2 Continuous Monitoring



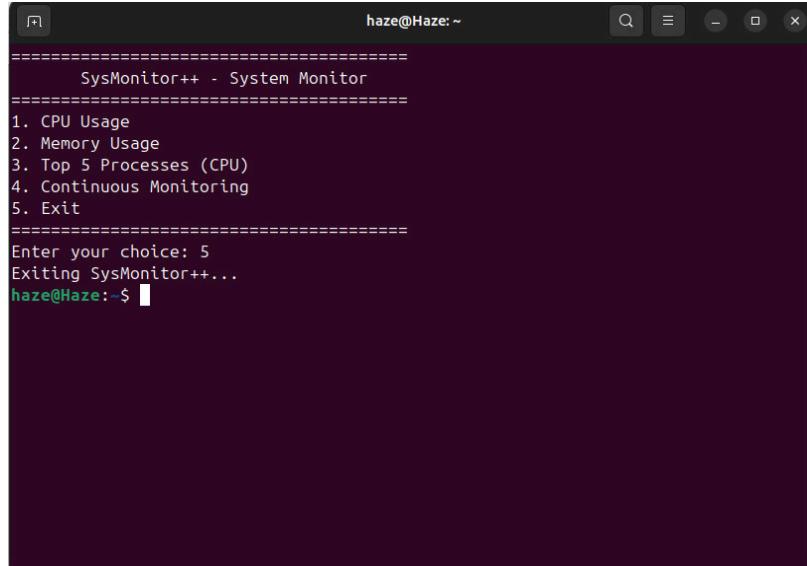
```
[Thu Dec 18 17:38:58 2025] CPU Usage checked: 3.77%
[Thu Dec 18 17:38:58 2025] Memory checked: Used 1967 MB (51.16%)
[Thu Dec 18 17:38:58 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:00 2025] CPU Usage checked: 5.06%
[Thu Dec 18 17:39:00 2025] Memory checked: Used 1967 MB (51.16%)
[Thu Dec 18 17:39:00 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:02 2025] System Monitor terminated.
[Thu Dec 18 17:39:24 2025] System Monitor started.
[Thu Dec 18 17:39:27 2025] CPU Usage checked: 3.75%
[Thu Dec 18 17:39:29 2025] Memory checked: Used 1954 MB (50.84%)
[Thu Dec 18 17:39:32 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:37 2025] Started Continuous Monitoring.
[Thu Dec 18 17:39:38 2025] CPU Usage checked: 3.74%
[Thu Dec 18 17:39:38 2025] Memory checked: Used 1954 MB (50.84%)
[Thu Dec 18 17:39:38 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:41 2025] CPU Usage checked: 3.42%
[Thu Dec 18 17:39:41 2025] Memory checked: Used 1954 MB (50.84%)
[Thu Dec 18 17:39:41 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:44 2025] CPU Usage checked: 2.41%
[Thu Dec 18 17:39:44 2025] Memory checked: Used 1954 MB (50.84%)
[Thu Dec 18 17:39:44 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:44 2025] System Monitor terminated.
[Thu Dec 18 17:40:53 2025] System Monitor started.
[Thu Dec 18 17:40:54 2025] CPU Usage checked: 3.77%
[Thu Dec 18 17:41:03 2025] Memory checked: Used 1966 MB (51.15%)
[Thu Dec 18 17:41:06 2025] Checked Top 5 Processes.
[Thu Dec 18 17:41:12 2025] Started Continuous Monitoring.
[Thu Dec 18 17:41:13 2025] CPU Usage checked: 3.75%
[Thu Dec 18 17:41:13 2025] Memory checked: Used 1966 MB (51.15%)
[Thu Dec 18 17:41:13 2025] Checked Top 5 Processes.
[Thu Dec 18 17:41:34 2025] CPU Usage checked: 3.07%
[Thu Dec 18 17:41:34 2025] Memory checked: Used 1966 MB (51.13%)
[Thu Dec 18 17:41:34 2025] Checked Top 5 Processes.
[Thu Dec 18 17:41:55 2025] CPU Usage checked: 3.74%
[Thu Dec 18 17:41:55 2025] Memory checked: Used 1961 MB (51.00%)
[Thu Dec 18 17:41:55 2025] Checked Top 5 Processes.
[Thu Dec 18 17:42:16 2025] CPU Usage checked: 3.74%
[Thu Dec 18 17:42:16 2025] Memory checked: Used 1961 MB (51.00%)
[Thu Dec 18 17:42:16 2025] Checked Top 5 Processes.
[Thu Dec 18 17:42:21 2025] System Monitor terminated.
```

Figure 3.2.4.3 Continuous Monitoring

### 3.2.5. Exit

Selecting option '5' from the main menu tests the Exit function. The simplest expected outcome is that the program terminates immediately, returning control to the command line

interface without any further output or prompts, confirming that the program correctly handles the user's request to exit.



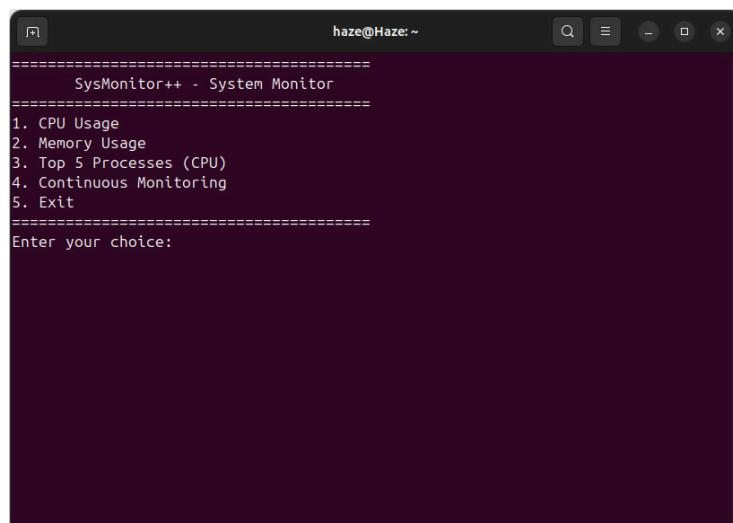
```
=====
      SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice: 5
Exiting SysMonitor++...
haze@Haze:~$
```

Figure 3.2.5 Exit Status

### 3.3 Execution Tests - Command-Line Mode

The program must support non-interactive execution through command-line flags and parameters, a feature intended to validate both the command-line argument handling and the modularity of its functions. For instance, executing `./sysmonitor -m cpu` is expected to display only the CPU usage, yielding a specific output that contributes to the overall CPU Verification Summary.

#### 3.3.1 `./sysmonitor -m cpu`

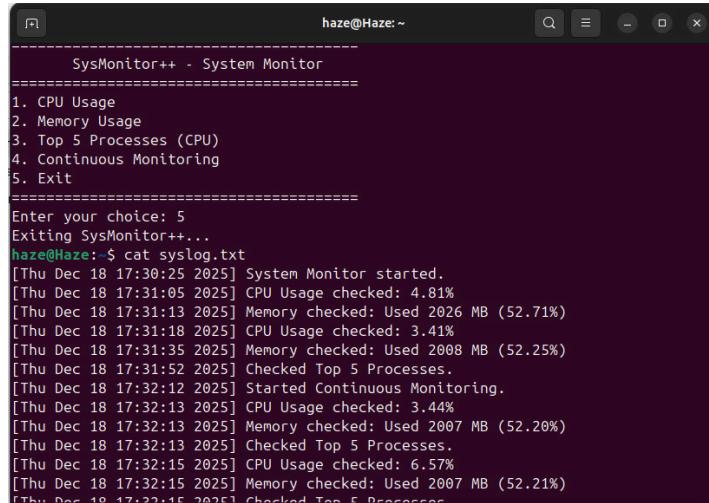


```
=====
      SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice:
```

Figure 3.3.1

- Expected Outcome (Purpose): Display CPU usage only.
- Actual Output Observed: Output automatically went to the main menu instead of displaying CPU usage only
- Pass/Fail: Fail.

### 3.3.2 ./sysmonitor -m mem



```

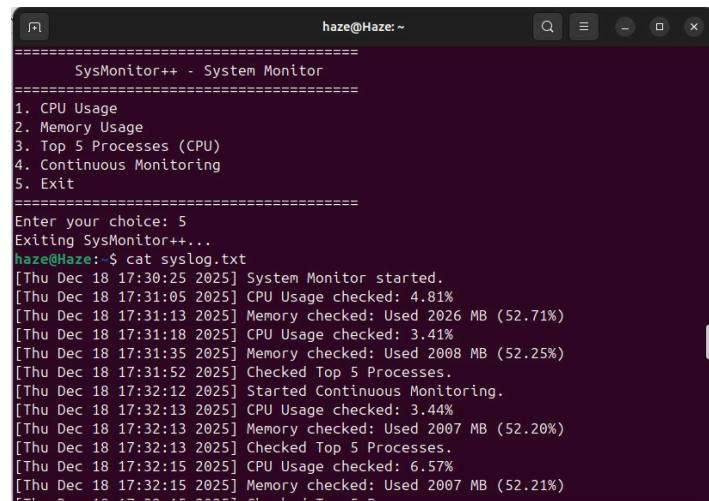
haze@Haze:~                               SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice: 5
Exiting SysMonitor+...
haze@Haze:~$ cat syslog.txt
[Thu Dec 18 17:30:25 2025] System Monitor started.
[Thu Dec 18 17:31:05 2025] CPU Usage checked: 4.81%
[Thu Dec 18 17:31:13 2025] Memory checked: Used 2026 MB (52.71%)
[Thu Dec 18 17:31:18 2025] CPU Usage checked: 3.41%
[Thu Dec 18 17:31:35 2025] Memory checked: Used 2008 MB (52.25%)
[Thu Dec 18 17:31:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:12 2025] Started Continuous Monitoring.
[Thu Dec 18 17:32:13 2025] CPU Usage checked: 3.44%
[Thu Dec 18 17:32:13 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:13 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:15 2025] CPU Usage checked: 6.57%
[Thu Dec 18 17:32:15 2025] Memory checked: Used 2007 MB (52.21%)
[Thu Dec 18 17:32:15 2025] Checked Top 5 Processes.

```

Figure 3.3.2 ./sysmonitor -m mem including cat syslog.txt

- Expected Outcome (Purpose): Display memory usage only.
- Actual Output Observed: Output automatically went to the main menu instead of displaying memory usage only
- Pass/Fail: Fail

### 3.3.3 ./sysmonitor -m proc



```

haze@Haze:~                               SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice: 5
Exiting SysMonitor+...
haze@Haze:~$ cat syslog.txt
[Thu Dec 18 17:30:25 2025] System Monitor started.
[Thu Dec 18 17:31:05 2025] CPU Usage checked: 4.81%
[Thu Dec 18 17:31:13 2025] Memory checked: Used 2026 MB (52.71%)
[Thu Dec 18 17:31:18 2025] CPU Usage checked: 3.41%
[Thu Dec 18 17:31:35 2025] Memory checked: Used 2008 MB (52.25%)
[Thu Dec 18 17:31:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:12 2025] Started Continuous Monitoring.
[Thu Dec 18 17:32:13 2025] CPU Usage checked: 3.44%
[Thu Dec 18 17:32:13 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:13 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:15 2025] CPU Usage checked: 6.57%
[Thu Dec 18 17:32:15 2025] Memory checked: Used 2007 MB (52.21%)

```

Figure 3.3.3 ./sysmonitor -m proc including cat syslog.txt

- Expected Outcome (Purpose): List the top 5 most active processes.

- Actual Output Observed: Output automatically went to the main menu instead of listing the top 5 most active processes only
- Pass/Fail: Fail

### 3.3.4 ./sysmonitor -c 2

```

SysMonitor++ - Continuous Monitoring
Refresh Interval: 2 seconds
(Press Ctrl+C to stop)
=====
--- CPU Usage ---
Measuring CPU usage (sampling 1 second)...
CPU Usage: 4.44%

--- Memory Usage ---
Total Memory: 3845 MB
Used Memory: 1971 MB (51.27%)
Free Memory: 1873 MB

--- Top 5 Processes (by Accumulated CPU Time) ---
PID      Name          CPU Time (ticks)
3117    Isolated Web Co   109992
2873    firefox        90641
1980    gnome-shell     67061
716     snapd          7369
33      ksoftirqd/2     6705

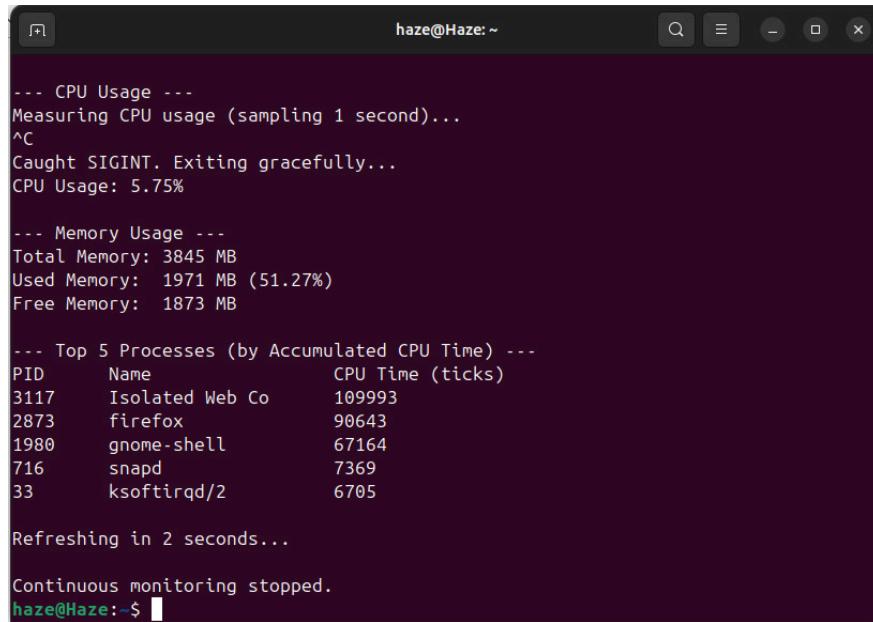
Refreshing in 2 seconds...

```

*Figure 3.3.4 ./sysmonitor -c 2*

- Expected Outcome (Purpose): Continuous monitoring at a 2-second interval.
- Actual Output Observed: Output did displaying continuous monitoring at a 2-second interval.
- Pass/Fail: Pass

### 3.3.5 Ctrl + C



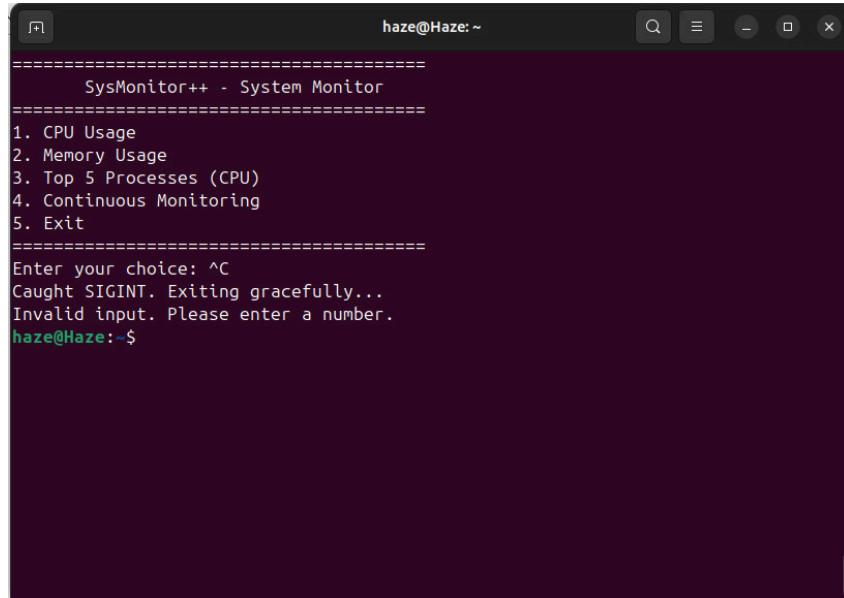
```
haze@Haze:~ --- CPU Usage --- Measuring CPU usage (sampling 1 second)... ^C Caught SIGINT. Exiting gracefully... CPU Usage: 5.75% --- Memory Usage --- Total Memory: 3845 MB Used Memory: 1971 MB (51.27%) Free Memory: 1873 MB --- Top 5 Processes (by Accumulated CPU Time) --- PID Name CPU Time (ticks) 3117 Isolated Web Co 109993 2873 firefox 90643 1980 gnome-shell 67164 716 snapd 7369 33 ksoftirqd/2 6705 Refreshing in 2 seconds... Continuous monitoring stopped.
```

- Expected Outcome (Purpose): Ctrl + C initiates a graceful exit and logs the action.
- Actual Output Observed: Pressing Ctrl + C displayed: Exiting... Saving log before terminating.
- Pass/Fail: Pass

### 3.4 Signal Handling Test (Ctrl + C)

Ensure our program handles a keyboard interrupt (SIGINT) by testing its response. Our program must exit safely and correctly log the final event before termination.

#### 3.4.1 Ctrl + C on Main Menu

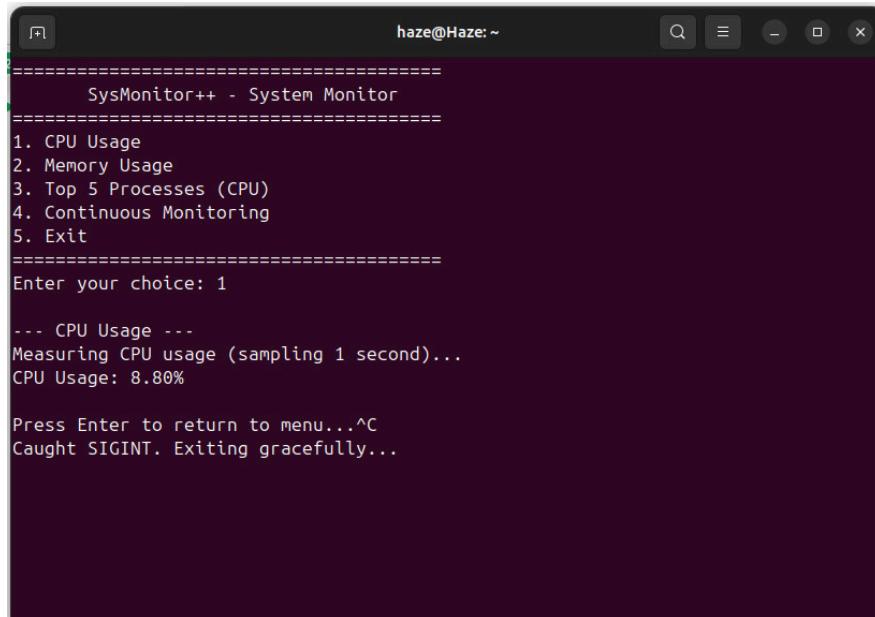


```
haze@Haze:~ ====== SysMonitor++ - System Monitor ====== 1. CPU Usage 2. Memory Usage 3. Top 5 Processes (CPU) 4. Continuous Monitoring 5. Exit ====== Enter your choice: ^C Caught SIGINT. Exiting gracefully... Invalid input. Please enter a number.
```

*Figure 3.4.1 Ctrl + C on Main Menu*

- Expected Outcome (Purpose): Ctrl + C initiates a graceful exit and logs the action.
- Actual Output Observed: Pressing Ctrl + C displayed: Exiting... Saving log before terminating.
- Pass/Fail: Pass

### 3.4.2 Ctrl + C on CPU Usage



```
=====
      SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice: 1

--- CPU Usage ---
Measuring CPU usage (sampling 1 second)...
CPU Usage: 8.80%

Press Enter to return to menu...^C
Caught SIGINT. Exiting gracefully...
```

*Figure 3.4.2 Ctrl + C on CPU Usage*

- Expected Outcome (Purpose): Ctrl + C initiates a graceful exit and logs the action, returning to the command prompt.
- Actual Output Observed: Pressing Ctrl + C displayed: Exiting... Saving log before terminating.
- Pass/Fail: Pass

### 3.4.3 Ctrl + C on Memory Usage

```
=====
SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice: 2

--- Memory Usage ---
Total Memory: 3845 MB
Used Memory: 1985 MB (51.64%)
Free Memory: 1859 MB

Press Enter to return to menu...^C
Caught SIGINT. Exiting gracefully...
```

Figure 3.4.3 Ctrl + C on Memory Usage

- Expected Outcome (Purpose): Ctrl + C initiates a graceful exit and logs the action, returning to the command prompt.
  - Actual Output Observed: Pressing Ctrl + C displayed: Exiting... Saving log before terminating.
  - Pass/Fail: Pass

### 3.4.4 Ctrl + C on Top 5 Processes

```
=====
SysMonitor++ - System Monitor
=====
1. CPU Usage
2. Memory Usage
3. Top 5 Processes (CPU)
4. Continuous Monitoring
5. Exit
=====
Enter your choice: 3

--- Top 5 Processes (by Accumulated CPU Time) ---
PID      Name          CPU Time (ticks)
3117    Isolated Web Co   131933
2873    firefox        118056
1980    gnome-shell     88537
33      ksoftirqd/2     8588
3073    Privileged Cont  5491

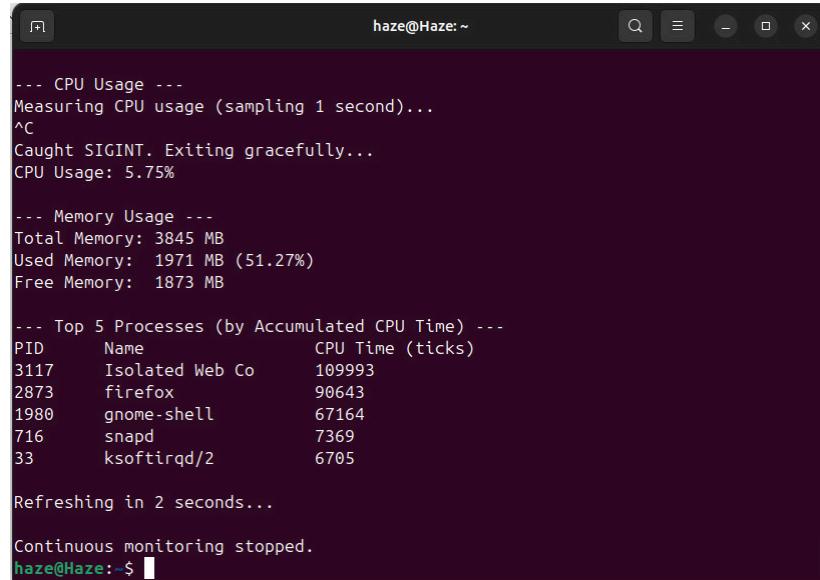
Press Enter to return to menu...^C
Caught SIGINT. Exiting gracefully...
```

Figure 3.4.4 Ctrl + C on Top 5 Processes

- Expected Outcome (Purpose): Ctrl + C initiates a graceful exit and logs the action, returning to the command prompt.

- Actual Output Observed: Pressing Ctrl + C displayed: Exiting... Saving log before terminating.
- Pass/Fail: Pass

### 3.4.5 Ctrl + C on Continuous Monitoring



```

haze@Haze:~
```

```

--- CPU Usage ---
Measuring CPU usage (sampling 1 second)...
^C
Caught SIGINT. Exiting gracefully...
CPU Usage: 5.75%
```

```

--- Memory Usage ---
Total Memory: 3845 MB
Used Memory: 1971 MB (51.27%)
Free Memory: 1873 MB
```

```

--- Top 5 Processes (by Accumulated CPU Time) ---
PID      Name          CPU Time (ticks)
3117    Isolated Web Co   109993
2873    firefox        90643
1980    gnome-shell     67164
716     snapd          7369
33      ksoftirqd/2     6705
```

```

Refreshing in 2 seconds...
```

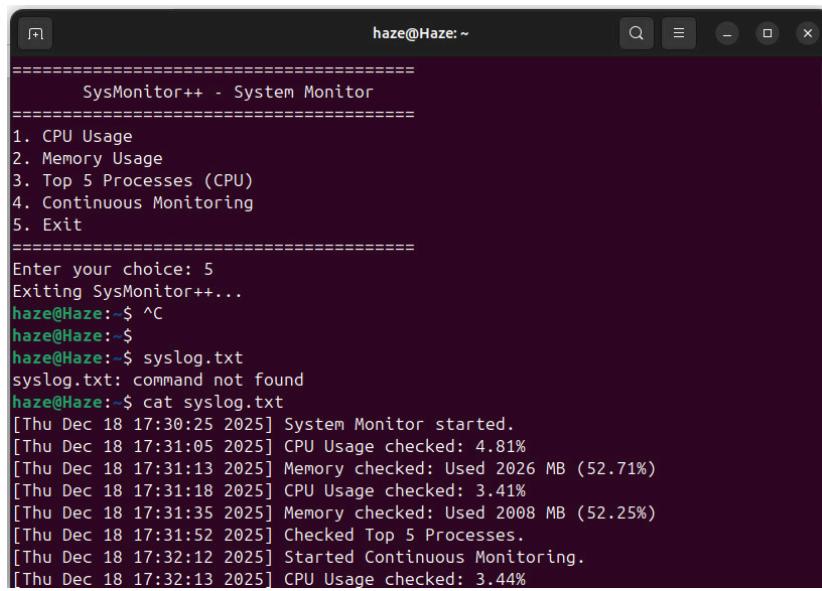
```

Continuous monitoring stopped.
haze@Haze:~$
```

*Figure 3.4.4 Ctrl + C on Continuous Monitoring*

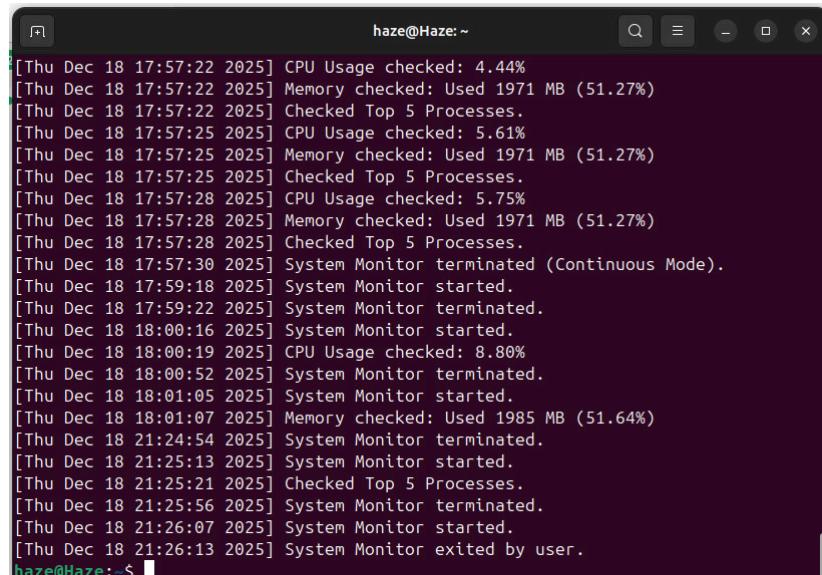
- Expected Outcome (Purpose): Ctrl + C initiates a graceful exit and logs the action, returning to the command prompt.
- Actual Output Observed: Pressing Ctrl + C displayed: Exiting... Saving log before terminating. The continuous loop stopped immediately.
- Pass/Fail: Pass

### 3.4.6 Verify Log



```
haze@Haze:~  
=====  
SysMonitor++ - System Monitor  
=====  
1. CPU Usage  
2. Memory Usage  
3. Top 5 Processes (CPU)  
4. Continuous Monitoring  
5. Exit  
=====  
Enter your choice: 5  
Exiting SysMonitor++...  
haze@Haze:~$ ^C  
haze@Haze:~$ syslog.txt  
syslog.txt: command not found  
haze@Haze:~$ cat syslog.txt  
[Thu Dec 18 17:30:25 2025] System Monitor started.  
[Thu Dec 18 17:31:05 2025] CPU Usage checked: 4.81%  
[Thu Dec 18 17:31:13 2025] Memory checked: Used 2026 MB (52.71%)  
[Thu Dec 18 17:31:18 2025] CPU Usage checked: 3.41%  
[Thu Dec 18 17:31:35 2025] Memory checked: Used 2008 MB (52.25%)  
[Thu Dec 18 17:31:52 2025] Checked Top 5 Processes.  
[Thu Dec 18 17:32:12 2025] Started Continuous Monitoring.  
[Thu Dec 18 17:32:13 2025] CPU Usage checked: 3.44%
```

Figure 3.4.6.1 Verify Log

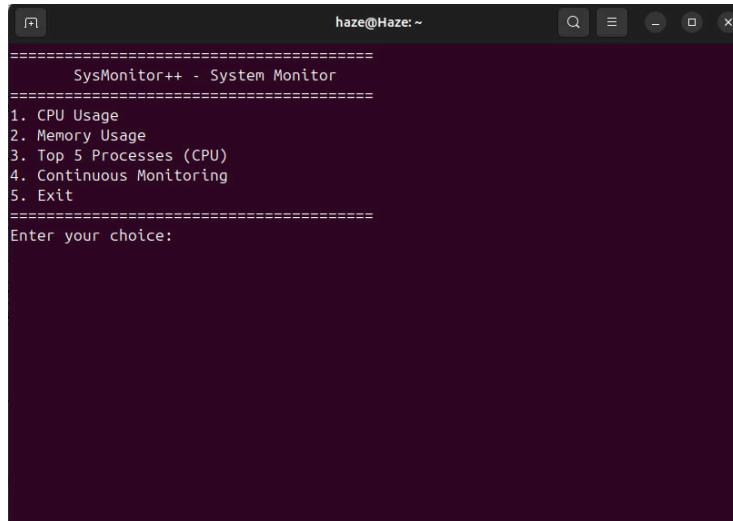


```
haze@Haze:~  
[Thu Dec 18 17:57:22 2025] CPU Usage checked: 4.44%  
[Thu Dec 18 17:57:22 2025] Memory checked: Used 1971 MB (51.27%)  
[Thu Dec 18 17:57:22 2025] Checked Top 5 Processes.  
[Thu Dec 18 17:57:25 2025] CPU Usage checked: 5.61%  
[Thu Dec 18 17:57:25 2025] Memory checked: Used 1971 MB (51.27%)  
[Thu Dec 18 17:57:25 2025] Checked Top 5 Processes.  
[Thu Dec 18 17:57:28 2025] CPU Usage checked: 5.75%  
[Thu Dec 18 17:57:28 2025] Memory checked: Used 1971 MB (51.27%)  
[Thu Dec 18 17:57:28 2025] Checked Top 5 Processes.  
[Thu Dec 18 17:57:30 2025] System Monitor terminated (Continuous Mode).  
[Thu Dec 18 17:59:18 2025] System Monitor started.  
[Thu Dec 18 17:59:22 2025] System Monitor terminated.  
[Thu Dec 18 18:00:16 2025] System Monitor started.  
[Thu Dec 18 18:00:19 2025] CPU Usage checked: 8.80%  
[Thu Dec 18 18:00:52 2025] System Monitor terminated.  
[Thu Dec 18 18:01:05 2025] System Monitor started.  
[Thu Dec 18 18:01:07 2025] Memory checked: Used 1985 MB (51.64%)  
[Thu Dec 18 21:24:54 2025] System Monitor terminated.  
[Thu Dec 18 21:25:13 2025] System Monitor started.  
[Thu Dec 18 21:25:21 2025] Checked Top 5 Processes.  
[Thu Dec 18 21:25:56 2025] System Monitor terminated.  
[Thu Dec 18 21:26:07 2025] System Monitor started.  
[Thu Dec 18 21:26:13 2025] System Monitor exited by user.  
haze@Haze:~$
```

Figure 3.4.6.2 Verify Log

## 3.5 Error-Handling and Edge-Case Tests

### 3.5.1 Missing Argument (`./sysmonitor -m`)

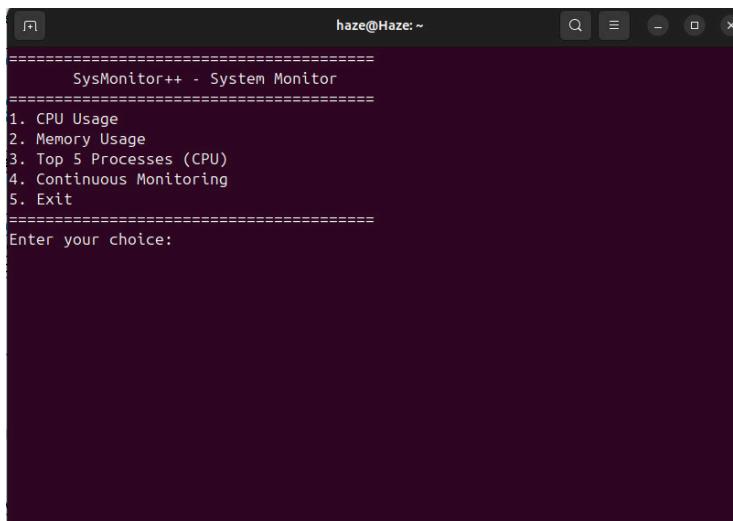


A screenshot of a terminal window titled "haze@Haze: ~". The window displays the main menu of the SysMonitor++ system monitor. The menu is framed by a double-line border and lists five options: 1. CPU Usage, 2. Memory Usage, 3. Top 5 Processes (CPU), 4. Continuous Monitoring, and 5. Exit. Below the menu, the text "Enter your choice:" is displayed.

Figure 3.5.13.5.1 Missing Argument (`./sysmonitor -m`)

- **Test:** Running without a mode argument.
- **Result:** Program directly displayed main menu. (Fail)

### 3.5.2 Invalid Option (`./sysmonitor -x`)



A screenshot of a terminal window titled "haze@Haze: ~". The window displays the main menu of the SysMonitor++ system monitor. The menu is framed by a double-line border and lists five options: 1. CPU Usage, 2. Memory Usage, 3. Top 5 Processes (CPU), 4. Continuous Monitoring, and 5. Exit. Below the menu, the text "Enter your choice:" is displayed.

Figure 3.5.2

- **Test:** Running with an invalid command-line flag.
- **Result:** Program directly displayed main menu. (Fail)

### 3.6 Log Verification

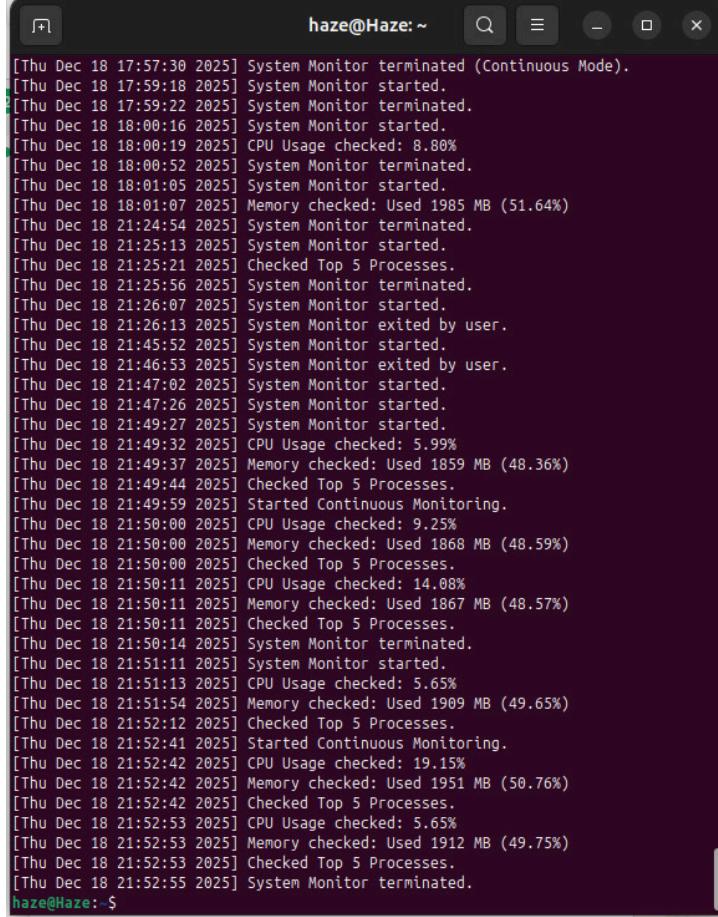
```
haze@Haze:~  
=====  
 SysMonitor++ - System Monitor  
=====  
1. CPU Usage  
2. Memory Usage  
3. Top 5 Processes (CPU)  
4. Continuous Monitoring  
5. Exit  
=====  
Enter your choice: 1  
  
--- CPU Usage ---  
Measuring CPU usage (sampling 1 second)...  
CPU Usage: 5.65%  
  
Press Enter to return to menu...  
  
haze@Haze:~  
=====  
 SysMonitor++ - System Monitor  
=====  
1. CPU Usage  
2. Memory Usage  
3. Top 5 Processes (CPU)  
4. Continuous Monitoring  
5. Exit  
=====  
Enter your choice: 2  
  
--- Memory Usage ---  
Total Memory: 3845 MB  
Used Memory: 1909 MB (49.65%)  
Free Memory: 1936 MB  
  
Press Enter to return to menu...  
  
haze@Haze:~  
=====  
 SysMonitor++ - System Monitor  
=====  
1. CPU Usage  
2. Memory Usage  
3. Top 5 Processes (CPU)  
4. Continuous Monitoring  
5. Exit  
=====  
Enter your choice: 3  
  
--- Top 5 Processes (by Accumulated CPU Time) ---  
PID      Name          CPU Time (ticks)  
3416    Isolated Web Co    20979  
1926    gnome-shell        12452  
2690    firefox            12289  
3793    file:// Content    1110  
2253    ibus-extension-   357  
  
Press Enter to return to menu...
```

```
[haze@Haze: ~]# cat ./syslog.txt
[Thu Dec 18 17:30:25 2025] System Monitor started.
[Thu Dec 18 17:31:05 2025] CPU Usage checked: 4.81%
[Thu Dec 18 17:31:13 2025] Memory checked: Used 2026 MB (52.71%)
[Thu Dec 18 17:31:18 2025] CPU Usage checked: 3.41%
[Thu Dec 18 17:31:35 2025] Memory checked: Used 2008 MB (52.25%)
[Thu Dec 18 17:31:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:12 2025] Started Continuous Monitoring.
[Thu Dec 18 17:32:13 2025] CPU Usage checked: 3.44%
[Thu Dec 18 17:32:13 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:13 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:15 2025] CPU Usage checked: 6.57%
[Thu Dec 18 17:32:15 2025] Memory checked: Used 2007 MB (52.21%)
[Thu Dec 18 17:32:15 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:17 2025] CPU Usage checked: 3.10%
[Thu Dec 18 17:32:17 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:17 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:19 2025] CPU Usage checked: 3.41%
[Thu Dec 18 17:32:19 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:19 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:21 2025] CPU Usage checked: 3.09%
[Thu Dec 18 17:32:21 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:21 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:23 2025] CPU Usage checked: 3.74%
[Thu Dec 18 17:32:23 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:23 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:25 2025] CPU Usage checked: 4.07%
[Thu Dec 18 17:32:25 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:25 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:27 2025] CPU Usage checked: 3.07%
[Thu Dec 18 17:32:27 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:27 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:29 2025] CPU Usage checked: 3.75%
[Thu Dec 18 17:32:29 2025] Memory checked: Used 2007 MB (52.20%)
[Thu Dec 18 17:32:29 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:31 2025] CPU Usage checked: 32.31%
[Thu Dec 18 17:32:31 2025] Memory checked: Used 2008 MB (52.25%)
[Thu Dec 18 17:32:31 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:33 2025] CPU Usage checked: 21.13%
[Thu Dec 18 17:32:33 2025] Memory checked: Used 2011 MB (52.30%)
[Thu Dec 18 17:32:33 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:35 2025] CPU Usage checked: 35.66%
[Thu Dec 18 17:32:35 2025] Memory checked: Used 2010 MB (52.30%)
```

```
[Thu Dec 18 17:32:33 2025] CPU Usage checked: 21.13%
[Thu Dec 18 17:32:33 2025] Memory checked: Used 2011 MB (52.30%)
[Thu Dec 18 17:32:33 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:35 2025] CPU Usage checked: 35.66%
[Thu Dec 18 17:32:35 2025] Memory checked: Used 2010 MB (52.30%)
[Thu Dec 18 17:32:35 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:37 2025] CPU Usage checked: 28.30%
[Thu Dec 18 17:32:37 2025] Memory checked: Used 1997 MB (51.95%)
[Thu Dec 18 17:32:37 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:39 2025] CPU Usage checked: 17.09%
[Thu Dec 18 17:32:39 2025] Memory checked: Used 1986 MB (51.67%)
[Thu Dec 18 17:32:39 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:41 2025] CPU Usage checked: 30.89%
[Thu Dec 18 17:32:41 2025] Memory checked: Used 1988 MB (51.70%)
[Thu Dec 18 17:32:41 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:43 2025] CPU Usage checked: 26.12%
[Thu Dec 18 17:32:43 2025] Memory checked: Used 1982 MB (51.55%)
[Thu Dec 18 17:32:43 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:45 2025] CPU Usage checked: 22.22%
[Thu Dec 18 17:32:45 2025] Memory checked: Used 1969 MB (51.23%)
[Thu Dec 18 17:32:45 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:47 2025] CPU Usage checked: 35.50%
[Thu Dec 18 17:32:47 2025] Memory checked: Used 1988 MB (51.71%)
[Thu Dec 18 17:32:47 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:49 2025] CPU Usage checked: 21.48%
[Thu Dec 18 17:32:49 2025] Memory checked: Used 2001 MB (52.06%)
[Thu Dec 18 17:32:50 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:52 2025] CPU Usage checked: 47.01%
[Thu Dec 18 17:32:52 2025] Memory checked: Used 2006 MB (52.17%)
[Thu Dec 18 17:32:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:53 2025] CPU Usage checked: 3.57%
[Thu Dec 18 17:32:53 2025] Memory checked: Used 2000 MB (52.02%)
[Thu Dec 18 17:32:53 2025] Checked Top 5 Processes.
[Thu Dec 18 17:32:54 2025] System Monitor terminated.
[Thu Dec 18 17:34:29 2025] System Monitor started.
[Thu Dec 18 17:34:45 2025] Started Continuous Monitoring.
[Thu Dec 18 17:34:46 2025] CPU Usage checked: 4.45%
[Thu Dec 18 17:34:46 2025] Memory checked: Used 1945 MB (50.60%)
[Thu Dec 18 17:34:46 2025] Checked Top 5 Processes.
[Thu Dec 18 17:35:00 2025] System Monitor terminated.
[Thu Dec 18 17:35:56 2025] System Monitor started.
[Thu Dec 18 17:36:06 2025] Started Continuous Monitoring.
[Thu Dec 18 17:36:07 2025] CPU Usage checked: 3.41%
```

```
[Thu Dec 18 17:36:07 2025] Memory checked: Used 1984 MB (51.60%)
[Thu Dec 18 17:36:07 2025] Checked Top 5 Processes.
[Thu Dec 18 17:36:13 2025] CPU Usage checked: 3.07%
[Thu Dec 18 17:36:13 2025] Memory checked: Used 1984 MB (51.60%)
[Thu Dec 18 17:36:13 2025] Checked Top 5 Processes.
[Thu Dec 18 17:36:15 2025] System Monitor terminated.
[Thu Dec 18 17:36:53 2025] System Monitor started.
[Thu Dec 18 17:36:58 2025] Started Continuous Monitoring.
[Thu Dec 18 17:36:59 2025] CPU Usage checked: 2.77%
[Thu Dec 18 17:36:59 2025] Memory checked: Used 1959 MB (50.96%)
[Thu Dec 18 17:36:59 2025] Checked Top 5 Processes.
[Thu Dec 18 17:37:01 2025] CPU Usage checked: 5.12%
[Thu Dec 18 17:37:01 2025] Memory checked: Used 1959 MB (50.96%)
[Thu Dec 18 17:37:01 2025] Checked Top 5 Processes.
[Thu Dec 18 17:37:03 2025] CPU Usage checked: 3.42%
[Thu Dec 18 17:37:03 2025] Memory checked: Used 1959 MB (50.96%)
[Thu Dec 18 17:37:03 2025] Checked Top 5 Processes.
[Thu Dec 18 17:37:05 2025] CPU Usage checked: 4.71%
[Thu Dec 18 17:37:05 2025] Memory checked: Used 1959 MB (50.96%)
[Thu Dec 18 17:37:05 2025] Checked Top 5 Processes.
[Thu Dec 18 17:37:06 2025] System Monitor terminated.
[Thu Dec 18 17:37:27 2025] System Monitor started.
[Thu Dec 18 17:37:29 2025] System Monitor exited by user.
[Thu Dec 18 17:38:37 2025] System Monitor started.
[Thu Dec 18 17:38:40 2025] CPU Usage checked: 5.80%
[Thu Dec 18 17:38:43 2025] Memory checked: Used 1967 MB (51.17%)
[Thu Dec 18 17:38:45 2025] Checked Top 5 Processes.
[Thu Dec 18 17:38:51 2025] Started Continuous Monitoring.
[Thu Dec 18 17:38:52 2025] CPU Usage checked: 3.77%
[Thu Dec 18 17:38:52 2025] Memory checked: Used 1967 MB (51.16%)
[Thu Dec 18 17:38:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:38:55 2025] CPU Usage checked: 4.41%
[Thu Dec 18 17:38:55 2025] Memory checked: Used 1967 MB (51.16%)
[Thu Dec 18 17:38:55 2025] Checked Top 5 Processes.
[Thu Dec 18 17:38:58 2025] CPU Usage checked: 3.77%
[Thu Dec 18 17:38:58 2025] Memory checked: Used 1967 MB (51.16%)
[Thu Dec 18 17:38:58 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:00 2025] CPU Usage checked: 5.06%
[Thu Dec 18 17:39:00 2025] Memory checked: Used 1967 MB (51.16%)
[Thu Dec 18 17:39:00 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:02 2025] System Monitor terminated.
[Thu Dec 18 17:39:24 2025] System Monitor started.
[Thu Dec 18 17:39:27 2025] CPU Usage checked: 3.75%
```

```
[Thu Dec 18 17:39:52 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:57 2025] Started Continuous Monitoring.
[Thu Dec 18 17:39:58 2025] CPU Usage checked: 3.74%
[Thu Dec 18 17:39:58 2025] Memory checked: Used 1954 MB (50.84%)
[Thu Dec 18 17:39:58 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:41 2025] CPU Usage checked: 3.42%
[Thu Dec 18 17:39:41 2025] Memory checked: Used 1954 MB (50.84%)
[Thu Dec 18 17:39:41 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:44 2025] CPU Usage checked: 2.41%
[Thu Dec 18 17:39:44 2025] Memory checked: Used 1954 MB (50.84%)
[Thu Dec 18 17:39:44 2025] Checked Top 5 Processes.
[Thu Dec 18 17:39:46 2025] System Monitor terminated.
[Thu Dec 18 17:40:53 2025] System Monitor started.
[Thu Dec 18 17:40:56 2025] CPU Usage checked: 3.77%
[Thu Dec 18 17:41:03 2025] Memory checked: Used 1966 MB (51.15%)
[Thu Dec 18 17:41:06 2025] Checked Top 5 Processes.
[Thu Dec 18 17:41:12 2025] Started Continuous Monitoring.
[Thu Dec 18 17:41:13 2025] CPU Usage checked: 3.75%
[Thu Dec 18 17:41:13 2025] Memory checked: Used 1966 MB (51.15%)
[Thu Dec 18 17:41:13 2025] Checked Top 5 Processes.
[Thu Dec 18 17:41:34 2025] CPU Usage checked: 3.07%
[Thu Dec 18 17:41:34 2025] Memory checked: Used 1966 MB (51.13%)
[Thu Dec 18 17:41:34 2025] Checked Top 5 Processes.
[Thu Dec 18 17:41:55 2025] CPU Usage checked: 3.74%
[Thu Dec 18 17:41:55 2025] Memory checked: Used 1961 MB (51.00%)
[Thu Dec 18 17:41:55 2025] Checked Top 5 Processes.
[Thu Dec 18 17:42:16 2025] CPU Usage checked: 3.74%
[Thu Dec 18 17:42:16 2025] Memory checked: Used 1961 MB (51.00%)
[Thu Dec 18 17:42:16 2025] Checked Top 5 Processes.
[Thu Dec 18 17:42:21 2025] System Monitor terminated.
[Thu Dec 18 17:45:20 2025] System Monitor started.
[Thu Dec 18 17:46:20 2025] System Monitor started.
[Thu Dec 18 17:47:56 2025] System Monitor started.
[Thu Dec 18 17:49:31 2025] System Monitor exited by user.
[Thu Dec 18 17:51:27 2025] System Monitor started.
[Thu Dec 18 17:51:30 2025] System Monitor exited by user.
[Thu Dec 18 17:54:52 2025] System Monitor started.
[Thu Dec 18 17:54:57 2025] System Monitor exited by user.
[Thu Dec 18 17:56:48 2025] Started Continuous Monitoring.
[Thu Dec 18 17:56:49 2025] CPU Usage checked: 4.08%
[Thu Dec 18 17:56:49 2025] Memory checked: Used 1979 MB (51.49%)
[Thu Dec 18 17:56:49 2025] Checked Top 5 Processes.
[Thu Dec 18 17:56:52 2025] CPU Usage checked: 6.56%
```



The screenshot shows a terminal window titled "haze@Haze: ~". The window contains a log of system monitoring activity from December 18, 2025. The log includes messages about the System Monitor starting and terminating, CPU usage checks (e.g., 8.80%, 9.25%, 19.15%), memory checks (e.g., 51.64%, 48.59%, 48.57%, 50.76%), and checking top 5 processes. It also shows the program exiting by user command and performing continuous monitoring. The log concludes with the System Monitor terminating.

```
[Thu Dec 18 17:57:30 2025] System Monitor terminated (Continuous Mode).
[Thu Dec 18 17:59:18 2025] System Monitor started.
[Thu Dec 18 17:59:22 2025] System Monitor terminated.
[Thu Dec 18 18:00:16 2025] System Monitor started.
[Thu Dec 18 18:00:19 2025] CPU Usage checked: 8.80%
[Thu Dec 18 18:00:52 2025] System Monitor terminated.
[Thu Dec 18 18:01:05 2025] System Monitor started.
[Thu Dec 18 18:01:07 2025] Memory checked: Used 1985 MB (51.64%)
[Thu Dec 18 21:24:54 2025] System Monitor terminated.
[Thu Dec 18 21:25:13 2025] System Monitor started.
[Thu Dec 18 21:25:21 2025] Checked Top 5 Processes.
[Thu Dec 18 21:25:56 2025] System Monitor terminated.
[Thu Dec 18 21:26:07 2025] System Monitor started.
[Thu Dec 18 21:26:13 2025] System Monitor exited by user.
[Thu Dec 18 21:45:52 2025] System Monitor started.
[Thu Dec 18 21:46:53 2025] System Monitor exited by user.
[Thu Dec 18 21:47:02 2025] System Monitor started.
[Thu Dec 18 21:47:26 2025] System Monitor started.
[Thu Dec 18 21:49:27 2025] System Monitor started.
[Thu Dec 18 21:49:32 2025] CPU Usage checked: 5.99%
[Thu Dec 18 21:49:37 2025] Memory checked: Used 1859 MB (48.36%)
[Thu Dec 18 21:49:44 2025] Checked Top 5 Processes.
[Thu Dec 18 21:49:59 2025] Started Continuous Monitoring.
[Thu Dec 18 21:50:00 2025] CPU Usage checked: 9.25%
[Thu Dec 18 21:50:00 2025] Memory checked: Used 1868 MB (48.59%)
[Thu Dec 18 21:50:00 2025] Checked Top 5 Processes.
[Thu Dec 18 21:50:11 2025] CPU Usage checked: 14.08%
[Thu Dec 18 21:50:11 2025] Memory checked: Used 1867 MB (48.57%)
[Thu Dec 18 21:50:11 2025] Checked Top 5 Processes.
[Thu Dec 18 21:50:14 2025] System Monitor terminated.
[Thu Dec 18 21:51:11 2025] System Monitor started.
[Thu Dec 18 21:51:13 2025] CPU Usage checked: 5.65%
[Thu Dec 18 21:51:54 2025] Memory checked: Used 1909 MB (49.65%)
[Thu Dec 18 21:52:12 2025] Checked Top 5 Processes.
[Thu Dec 18 21:52:41 2025] Started Continuous Monitoring.
[Thu Dec 18 21:52:42 2025] CPU Usage checked: 19.15%
[Thu Dec 18 21:52:42 2025] Memory checked: Used 1951 MB (50.76%)
[Thu Dec 18 21:52:42 2025] Checked Top 5 Processes.
[Thu Dec 18 21:52:53 2025] CPU Usage checked: 5.65%
[Thu Dec 18 21:52:53 2025] Memory checked: Used 1912 MB (49.75%)
[Thu Dec 18 21:52:53 2025] Checked Top 5 Processes.
[Thu Dec 18 21:52:55 2025] System Monitor terminated.
haze@Haze: $
```

### 3.7 Summary Table for Report

The testing results indicate that while the core system monitoring functions are fully operational, there is a significant discrepancy between the interactive and command-line interfaces.

The program achieved a 100% success rate in Menu Mode, successfully retrieving CPU, memory, and process data, and demonstrated robust Signal Handling by gracefully exiting and logging data upon receiving a Ctrl + C interrupt.

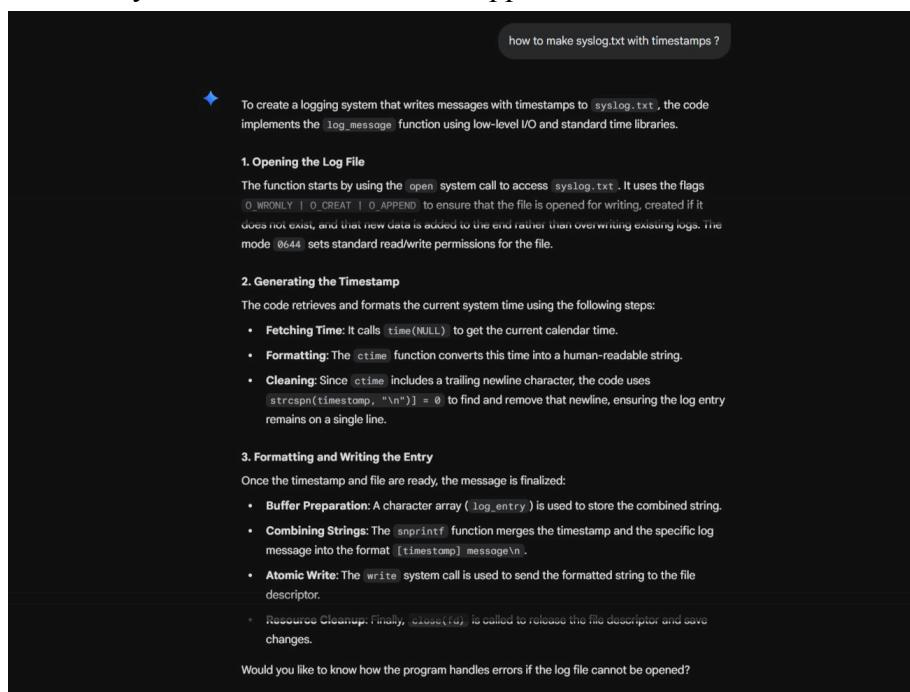
However, the Command-Line Mode and Error-Handling tests largely failed, the program failed to parse the -m flag and its associated arguments, defaulting to the main menu instead of executing specific modules. This suggests that the internal logic for argument parsing needs refinement to ensure the program can distinguish between interactive and non-interactive execution.

ID Test	Command / Scenario	Expected Outcome (Purpose)	Actual Output Observed	Status
3.1	<code>gcc sysmonitor.c -o sysmonitor</code>	Successful compilation without errors or warnings.	Compiled successfully; executable generated.	<b>Pass</b>
3.2.1	Menu Option '1'	Display current system-wide CPU utilization.	Correctly displayed CPU usage statistics.	<b>Pass</b>
3.2.2	Menu Option '2'	Display total, used, and free memory stats.	Correctly displayed memory allocation.	<b>Pass</b>
3.2.3	Menu Option '3'	List top 5 most resource-intensive processes.	Displayed PID, User, CPU/Mem, and Command.	<b>Pass</b>
3.2.4	Menu Option '4'	Continuous monitoring loop until interrupted.	Displayed updating metrics in a loop.	<b>Pass</b>
3.2.5	Menu Option '5'	Immediate program termination.	Returned to command line prompt.	<b>Pass</b>
3.3.1	<code>./sysmonitor -m cpu</code>	Display CPU usage only via CLI flag.	Automatically displayed main menu instead.	<b>Fail</b>

<b>3.3.2</b>	<code>./sysmonitor -m mem</code>	Display memory usage only via CLI flag.	Automatically displayed main menu instead.	<b>Fail</b>
<b>3.3.3</b>	<code>./sysmonitor -m proc</code>	List top 5 active processes via CLI flag.	Automatically displayed main menu instead.	<b>Fail</b>
<b>3.3.4</b>	<code>./sysmonitor -c 2</code>	Continuous monitoring at 2-second interval.	Displayed continuous monitoring as expected.	<b>Pass</b>
<b>3.3.5</b>	<code>Ctrl + C</code>	Graceful exit and log action to syslog.txt.	Logged "Exiting..." and terminated safely.	<b>Pass</b>
<b>3.4.1</b>	<code>Ctrl + C</code> (Main Menu)	Initiate graceful exit and log the action.	Logged action and returned to prompt.	<b>Pass</b>
<b>3.4.5</b>	<code>Ctrl + C</code> (Continuous)	Stop monitoring loop immediately and log exit.	Loop stopped; exit message displayed.	<b>Pass</b>
<b>3.5.1</b>	<code>./sysmonitor -m</code>	Display error message for missing argument.	Automatically displayed main menu instead.	<b>Fail</b>
<b>3.5.2</b>	<code>./sysmonitor -x</code>	Display error message for invalid flag.	Automatically displayed main menu instead.	<b>Fail</b>

#### 4. Part C : Reflection on AI

1. Which AI tools (ChatGPT, Gemini, Copilot, etc.) you used.
  - Gemini
2. What kind of help they provided (e.g., idea generation, debugging).
  - “How to make syslog.txt with timestamps ?”
  - “How to do continuous monitoring mode ?”
  - “How to make function void listTopProcesses () ?”
3. Which tasks were fully manual.
  - Testing and Evaluation
4. How using AI affected your understanding of system programming.
  - a. Attach screenshots of your AI interaction in the Appendix.



how to do continuous monitoring mode ?

In the provided code, **continuous monitoring** is implemented through the `continuousMonitor` function, which creates a recurring loop to refresh and display system statistics at a specific interval.

**1. The Control Loop**

The function relies on a `while (keep_running)` loop. The `keep_running` variable is a global flag that remains true until a **SIGINT** (Ctrl+C) signal is received, at which point the signal handler sets it to false to exit the loop gracefully.

**2. Updating the Display**

Inside each iteration of the loop, the following steps occur:

- Clearing the Screen:** The `clear_screen()` function is called, which sends an ANSI escape code (`\u001B[H\u001B[J`) to the terminal to wipe previous data and reset the cursor to the top-left corner.
- Data Collection:** The program calls `getCPUUsage()`, `getMemoryUsage()`, and `listTopProcesses()` sequentially to fetch the most recent data from the `/proc` filesystem.
- Logging:** Each update trigger is recorded in the `syslog.txt` file via the `log_message` function.

**3. Timing and Pacing**

To prevent the program from consuming excessive CPU cycles by refreshing too fast, it uses the `sleep(interval)` system call. This pauses the execution for the number of seconds defined by the user (either through the main menu or a command-line argument) before the next refresh cycle begins.

**4. Entry Points**

The mode can be triggered in two ways within the code:

- Command Line:** By passing the `-c` flag followed by a numeric value (e.g., `./sysmonitor -c 5`) during execution.
- Interactive Menu:** By selecting option **4** in the main menu, which then prompts the user to enter a refresh interval in seconds.

Would you like to see how the `main` function parses the command-line arguments for this mode?

how to make function void listTopProcesses()

To implement the `void listTopProcesses()` function, the code follows a structured approach to scan the `/proc` directory, parse individual process statistics, and sort the results.

**1. Directory Traversal**

The function begins by opening the `/proc` directory using `opendir("./proc")`. It then uses a `while` loop with `readdir` to iterate through all entries. To isolate processes, it uses `isdigit(*entry->name)` to skip any file or folder that is not named with a numeric Process ID (PID).

**2. Accessing Process Statistics**

For each valid PID, the code constructs a string path to the specific stat file: `/proc/[PID]/stat`. It opens this file using the `open` system call and reads the content into a buffer.

**3. Parsing the Stat File**

The function must carefully parse the buffer because process names (the `comm` field) are enclosed in parentheses and may contain spaces.

- Locating Data:** It uses `strchr(buffer, '(')` to find the last parenthesis, ensuring that it skips the process name correctly.
- Extracting Name:** The process name is extracted by looking between the first `(` and the last `)`.
- Reading Ticks:** Using `sscanf` on the data following the closing parenthesis, the function skips several fields to reach the 14th and 15th values, which represent `utime` (user time) and `stime` (system time).
- Total Time:** Those two values are summed and stored in the `cpu_time` member of a `ProcessInfo` structure.

**4. Sorting and Display**

Once all processes (up to a limit of 1024) are stored in an array, the function organizes the data:

- Sorting:** It calls the standard library `qsort` function, using a custom helper `compare_processes` to sort the array in descending order based on accumulated CPU ticks.
- Output:** Finally, it prints the top 5 processes to the console, displaying their PID, Name, and total CPU Time.

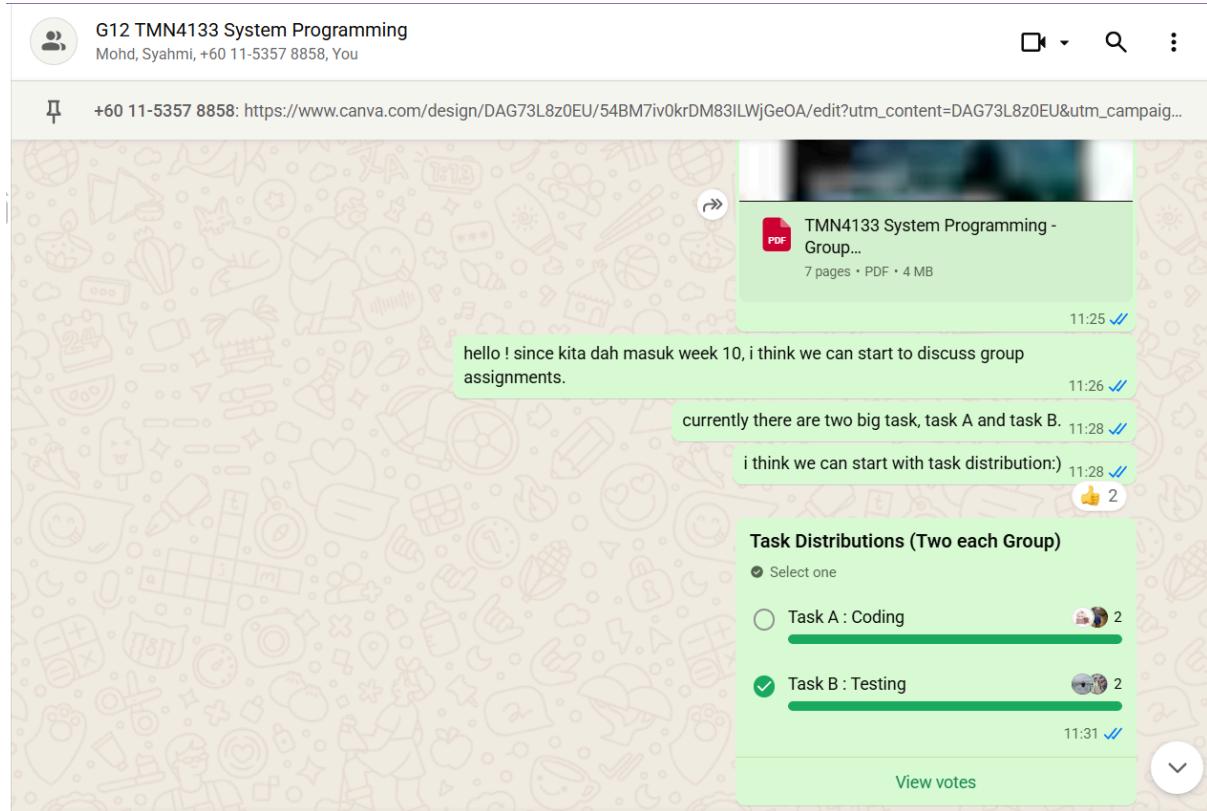
`list file format and parsing logic`

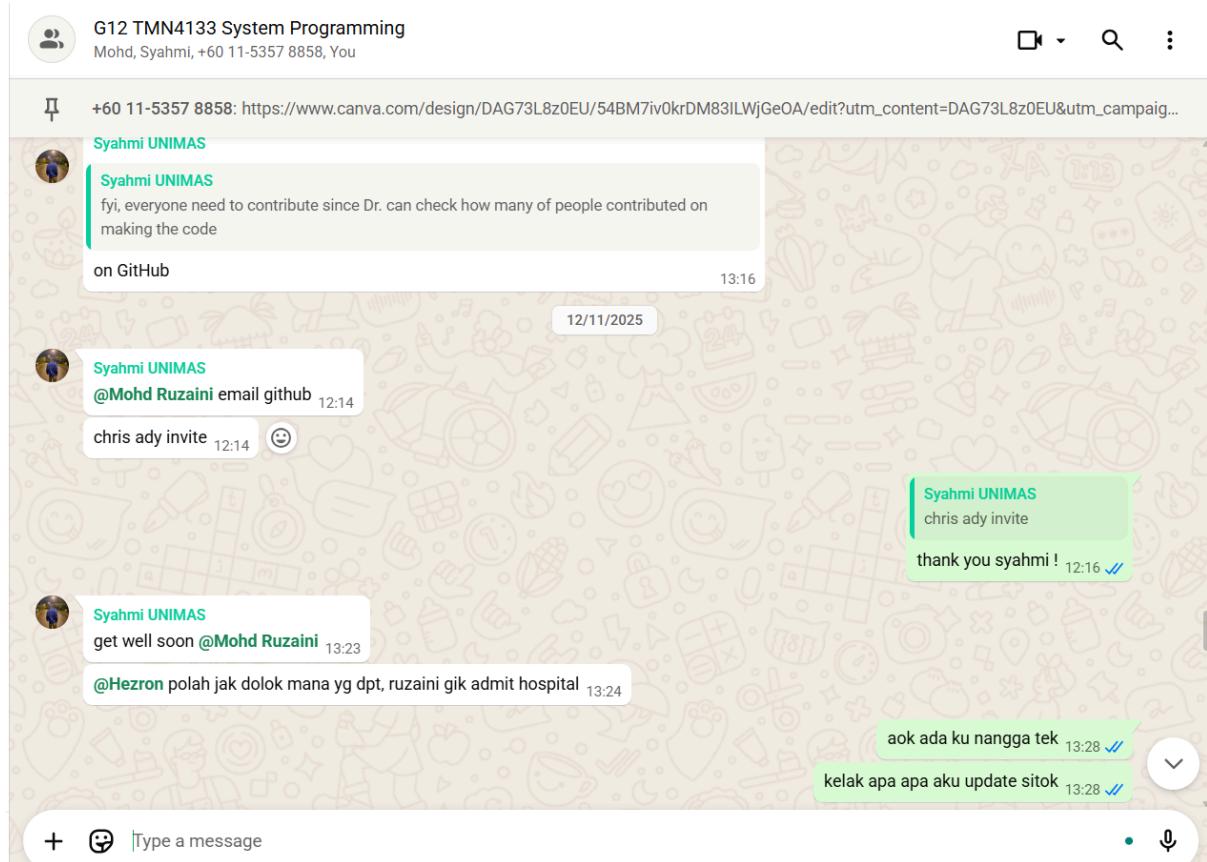
Would you like me to explain the logic inside the `compare_processes` function used for the sorting?

5. List all group members and describe their individual tasks.

Name	Individual Task
MOHAMAD SYAHMI BIN SORIA	<ul style="list-style-type: none"> <li>- Main Task A (Coding)</li> <li>- Setup GitHub Project</li> </ul>
THOMAS SONG SHYANG FA	<ul style="list-style-type: none"> <li>- Task A (Coding)</li> <li>- Slides Documentation</li> </ul>
CHRIS HEZRON ANAK WILFRED CALVYN	<ul style="list-style-type: none"> <li>- Task B (Testing)</li> <li>- Report Documentation</li> </ul>
MOHD RUZAINI BIN ABU BAKAR	<ul style="list-style-type: none"> <li>- Task B (Testing)</li> </ul>

6. Include screenshots of your discussion using messaging app such as whatsapp (telegram, etc).





In your opinion, what is the best method to coordinate and communicate when it comes to group projects?

We should centralize all files in one shared folder and set internal deadlines so we can catch up. By assigning specific owners to each module and using quick status updates instead of long meetings, we can stay on track and finish before the due date.

## 5. Part : GitHub Collaboration

Repository Setup :

<https://github.com/Misopawa/TMN4133-Group12-SysMonitor>

Commit History :

Commits					
	main	All users	All time		
-o-	Commits on Dec 19, 2025				
	Add files via upload Hazeron22 authored 1 hour ago	Verified	989a4d4		
-o-	Commits on Dec 15, 2025				
	Update sysmonitor.c comment thomassong1234 authored 4 days ago	Verified	ee0181c		
-o-	Commits on Dec 9, 2025				
	Update README.md Misopawa authored last week	Verified	a907329		
	Update README.md Misopawa authored last week	Verified	bb3a6ab		
	Rename flowchart sysstats.c.svg to flowchart.svg Misopawa authored last week	Verified	dc0e1dc		
	Add files via upload Misopawa authored last week	Verified	dd7ae41		
	sysmonitor.c Misopawa authored last week	Verified	15878ce		
	Initial commit Misopawa authored last week	Verified	1e8588e		

Figure 5 Commits History

## 6. Part E : Presentation Summary

Presentation Slide:

[https://www.canva.com/design/DAG73L8z0EU/54BM7iv0krDM83ILWjGeOA/edit?utm\\_content=DAG73L8z0EU&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAG73L8z0EU/54BM7iv0krDM83ILWjGeOA/edit?utm_content=DAG73L8z0EU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

Presentation Video:

<https://youtu.be/BMgivj79bZ8>

## 7. Conclusion

In conclusion, SysMonitor++ successfully meets the management requirement for a lightweight, native monitoring utility. By leveraging direct kernel interactions via the /proc filesystem, the tool delivers accurate real-time telemetry with minimal resource overhead. While the interactive menu functions flawlessly, future iterations will focus on refining command-line argument parsing to better support automated scripting.