Sardar Vallabhbhai National Institute of Technology, Surat
Department of Artificial Intelligence
Operating System Lab
Assignment - 3 and 4 (System calls in Linux)   Marks: 40

Objective:

- Explore and analyze **system calls** using different methods in Linux. [10]
- Implement C programs and debugging techniques to understand system interactions.[10]
- Use tools like `strace`, `top`, `gdb`, and `pthread` to observe system behavior.[10]
- Document findings in a **LaTeX report**. [10]

Assignment Tasks:

PROGRAM 1: Observing Variable Addresses Across Multiple Threads

Objective:

- Write a C program to print the memory address of a variable.
- Create **3-4 threads** using `pthread` and observe the printed addresses.
- Run the program on different architectures to analyze memory behavior.
- Justify the extracted values and submit the results in a **LaTeX report**.

Submission Requirements:

- **C program source code** with proper documentation.
- **Output observations** with an explanation of printed memory addresses.
- **Justification of results** based on thread memory allocation and architecture differences.
- **LaTeX report** detailing:
  - Introduction
  - Implementation
  - Experimental Results
  - Observations & Justification
  - Conclusion

PROGRAM 2: Observing System Resource Utilization Using `top` Command

Objective:

- Learn how to use the `top` command to monitor **CPU and memory usage** in real time.
- Run `top` in one terminal window and observe system statistics.
- Compile a program in another terminal and analyze its impact on system resources.
- Answer the following questions:
  - How much **free memory** is available in the system?
  - Which process is consuming the **most CPU**?
  - Which process has the **highest memory usage**?
- Document the findings in a **LaTeX report**.

Submission Requirements:

- **Screenshots or logs of `top` output** before and after compiling the program.
- **Analysis of CPU and memory usage changes** due to compilation.

- **Justification of observations** based on system behavior.
- **LaTeX report** including:
  - Introduction
  - Implementation Steps
  - Experimental Results
  - Observations & Justifications
  - Conclusion

PROGRAM 3: Exploring strace for System Call Tracing in Linux

Objective:

- Use `strace` to trace system calls of an **executing process**.
- Identify a **shell command that does not make a system call** using `strace`.
- Observe how **Bash** utilizes system calls to read commands from the console and echo them back to the screen.
- Document findings in a **LaTeX report**.

Submission Requirements:

- **Log files or screenshots** of `strace` output.
- **Analysis of system call behavior** for different commands.
- **Explanation of observed APIs (`read()`, `write()`, etc.).**
- **LaTeX report** including:
  - Introduction
  - Implementation Steps
  - Experimental Results
  - Observations & Justifications
  - Conclusion

PROGRAM 4: Debugging a C Program with Loops, File I/O, and Memory Tracing

Objective:

- Write a **C program** with:
  - **Multiple loops** (e.g., `for`, `while`).
  - **File I/O with output buffering**.
  - **Functions for operations** like finding the **largest of two numbers** and **swapping values**.
  - **Memory referencing** and pointer manipulation.
- Compile and **run the program in debug mode** using **GCC and GDB**.
- **Trace variable values and memory locations** during execution.
- Submit results in a **LaTeX report**.

Submission Requirements:

- **Source code** of the C program.
- **Screenshots or logs** from the GDB session.
- **Analysis of memory locations, function calls, and variable values.**
- **LaTeX report** including:
  - Introduction
  - Implementation Steps
  - Debugging Process
  - Observations & Justifications
  - Conclusion