Lab Assignment Submission Guidelines

Provided Programs and Tools

The following programs and tools are provided as part of this lab assignment:

- `cpu.c`: A program that runs in an infinite loop, used to observe CPU behavior.
- `cpu-print.c`: A program that prints output in an infinite loop for process monitoring.
- `pointers.cpp`: A program with pointer operations containing a segmentation fault.
- `fibonacci.cpp`: A program with a logical error in computing Fibonacci numbers.
- `memory_bugs.c`: A program with memory leaks and uninitialized variables.
- `disk.c` and `disk1.c`: Programs that simulate disk read operations.
- `make-copies.sh`: A script to create multiple copies of a file for disk operations.

Before You Begin

Before starting, ensure you are comfortable with the following:

- **Writing and Compiling Code**: Understand how to write C and C++ programs and compile them using `gcc` or `g++`. Learn how to use the `-g` flag to include debugging symbols in compiled executables.

- **Linux Commands**: Familiarize yourself with basic Linux commands like `ls`, `cat`, `cp`, `mv`, and `grep`.
- **Process Monitoring Tools**: Explore `top` and `ps` commands to monitor running processes. Understand the fields they display, such as PID, CPU usage, memory consumption, and process state.
- **The /proc Filesystem**: Learn how Linux exposes system and process information through the `/proc` filesystem. Use commands like `more /proc/cpuinfo` to view details about the CPU.

Help File

A **Help Me** file is provided for each of the questions in the assignment. Please refer to this file for step-by-step guidance on solving the problems.

Commands and Superuser Access

Ensure you are executing the commands in the Linux terminal. For tasks that require elevated permissions, use `sudo` to obtain superuser access. For example:

```
sudo command
```

You will be prompted to enter your password to proceed.

Report Submission Instructions

- Take **necessary screenshots** of your terminal commands, outputs, and any important results.
- Provide all the **commands** you executed.
- Compile all information into a **single PDF report** with your answers to Q1-Q7.
- The PDF file should be named according to your group number (e.g., `Group_4.pdf`).
- Only one member from each group should submit the report.

Submission Process

- Upload the PDF report as described above.
- The assignment will be evaluated offline and through a **viva voce** during your lab session, where you will need to explain your source codes and execute them in front of the evaluator.

Submission Instructions

- Submit all source code files and a report documenting your approach and observations.
- Name your submission directory with your roll number (e.g., `12345678/`).
- Compress the directory using the command:

```
tar -zcvf 12345678.tar.gz 12345678
```

- Upload the compressed file to the submission portal.

Important Notes

- Write your own source codes and **do not copy from any source**.
- A **plagiarism detection tool** will be used, and any detection of unfair means will result in penalties.