

Operating System

Fall 2023 – Assignment 1

Instructions:

- It's a group assignment and can be done in a group of two.
- Copied assignments will be marked negatively.
- Late submissions will not be entertained in any case.
- Online resources can be used for your understanding only but basing your whole assignment on it can result in losing all your marks, with a penalty on other assessments.

Question 1:

You will create a shell script for this project to automate the management and monitoring of server resources. You will be given a scenario in which you must write a script to carry out various server resource management and monitoring duties. (10 Marks)

Scenario Description:

Imagine you are a system administrator responsible for managing a group of Linux servers. Your task is to create a shell script that will help you perform the following tasks:

1. Disk Usage Monitoring: The script should check the disk usage of the server and send an alert if the usage exceeds a certain threshold (e.g., 30%).
2. CPU Usage Monitoring: Monitor CPU usage and send an alert if the CPU usage exceeds a specified threshold (e.g., 30%).
3. Memory Usage Monitoring: Monitor memory (RAM) usage and send an alert if the available memory falls below a specified threshold (e.g., 10% free).
4. Log Rotation: Implement log rotation for a specified log file, ensuring that it does not grow beyond a certain size (e.g., 10MB).

Requirements:

1. Create a shell script named `server_monitor.sh`.
2. Implement the tasks mentioned in the scenario.
3. The script should log all activities and alerts to a log file.
4. You should use conditional statements (if-else) to check thresholds and generate alerts.
5. Use appropriate commands and tools (e.g., `df`, `top`, `ps`, `tar`, `logrotate`, etc.) to achieve the tasks.
6. Ensure that your script is well-commented to explain the purpose of each section.
7. Include error handling to gracefully handle any potential issues that may arise during script execution.

Deliverables:

You will need to submit the three files against this question including `server_monitor.sh` and log file along with one word/pdf document including report and all steps screen shots you have gone through.

Question 2:

In this assignment, you will create a program in C to simulate the instruction life cycle of a simplified computer architecture. You will implement a fetch-decode-execute cycle and demonstrate how different op codes affect the execution of instructions. (10 Marks)

Scenario Description:

Imagine you are a computer architect tasked with building a simulator for a basic computer architecture. Your goal is to create a program that can execute a series of instructions, each with its own op code, and show how the fetch-decode-execute cycle works.

Requirements:

1. Write a C program named `instruction_simulator.c`.
2. Define a simplified instruction set with at least five different op codes. Each op code should correspond to a specific operation (e.g., ADD, SUB, LOAD, STORE, etc.). Create an enum or constants to represent these op codes.
3. Implement a fetch-decode-execute cycle within a loop that processes a sequence of instructions. The cycle should include the following steps:
 - a. Fetch: Fetch the next instruction from memory.
 - b. Decode: Determine the operation to be performed based on the op code.
 - c. Execute: Perform the operation and update the program state accordingly.
4. Create a data structure (e.g., an array) to represent the memory of the computer. Initialize this memory with a sequence of instructions using the op codes defined in step 2.
5. Print the state of the computer (e.g., memory contents, registers) before and after each instruction is executed.
6. Implement at least two sample programs using your defined op codes and memory layout.
7. These programs should consist of sequences of instructions that demonstrate various operations and interactions between instructions.
8. Use comments to explain the purpose of each part of your code and how the fetch-decode-execute cycle works.

Deliverables:

1. `instruction_simulator.c`: Your C program that simulates the instruction life cycle.
2. A README file explaining how to compile and run the program and explain the outputs, as well as providing details about the op codes, memory layout, and the sample programs.