CS 588: Computer System Lab

(January-May 2023)

Assignment – 6

Submission Deadline: 11:55 pm on Tuesday, 2nd May 2023 (hard deadline)

This assignment is a programming assignment where you need to implement the given scenario in C/C++ programming language. The scenario description is given in this document.

Instructions:

- Each group needs to implement all three questions and make one single submission on shared Gmail ID. Only one member from a group needs to make the submission.
- Submit the set of source code files of the application as a zipped file on cs558.2023@gmail.com by the deadline of 11:55 pm on Tuesday, 2nd May 2023 (hard deadline). The ZIP file's name should be the same as your group number, for example, "Group_4.zip", or "Group_4.rar", or "Group_4.tar.gz".
- The assignment will be evaluated offline/through viva-voce during your lab session on Wednesday, 3rd May 2023) where you will need to explain your source codes and execute them before the evaluator.
- Write your own source codes and do not copy from any source. Plagiarism detection tool will be used and any detection of unfair means will be penalised by awarding NEGATIVE marks (equal to the maximum marks for the assignment).

Question1.

Objectives:

- Understand the problems with contiguous allocation and the need for alternative file structures.
- Implement and compare different file structures for managing files in a file system.
- Evaluate the performance and scalability of the file system with different file structures.

Requirements:

- Implement a file system that supports three different file structures for managing files:
  - ➢ Contiguous allocation: each file is allocated a contiguous block of disk space.
  - ➢ Linked allocation: each file is allocated a linked list of disk blocks.
  - ➢ Indexed allocation: each file is allocated an index block that contains pointers to the disk blocks that store the file's data.
  - ➢ Modified contiguous allocation: each file is allocated an initial contiguous area of a specified size, and overflow areas are allocated as needed and linked to the initial area.
- Implement file operations for creating, reading, updating, and deleting files, using the different file structures.
- Measure and compare the performance of the file system with the different file structures, in terms of space utilization, file access time.

Question2.

You are working as a software engineer for a social media company that provides a platform for users to upload and share photos and videos. The company is experiencing a rapid growth in the number of users and data volume, which has put a strain on the memory usage and performance of the platform. The company is looking for a solution to optimize the memory usage and performance of the platform.

Requirements:

- Implement a demand paging system that loads pages from disk to memory on demand, and unloads them when they are no longer needed.
- Develop a data generator that simulates the memory access patterns of the platform, using a realistic workload and data size.
- Evaluate the performance of the demand paging system with different configurations and workload scenarios, and compare it with a non-paging system that keeps all data in memory.
- Optimize the memory usage and performance of the platform by tuning the demand paging system parameters, such as the page size, swap space size, and page replacement algorithm.

Question 3.

You are a data analyst for a logistics company that provides a supply chain management platform. The platform allows companies to manage their inventory, track shipments, and optimize logistics operations. The platform collects and processes a large amount of data from various sources, such as sensors, GPS, and weather reports. The company is experiencing a rapid growth in the number of clients and data volume, which has put a strain on the memory usage and performance of the platform. The company is looking for a solution to optimize the memory usage and performance of the platform.

Tasks:

- Implement a swapping system that swaps out unused pages from memory to disk, and swaps in pages that are needed for processing.
- Develop a data generator that simulates the data volume and diversity of the platform, and the usage patterns of the clients.
- Evaluate the performance of the swapping system by measuring the memory usage, page fault rate, and processing time of the platform under different data loads and swapping policies.