“To Schedule or Not To Schedule?” Project

CMPE 322

Abdullah Umut Hamzaoğulları

11.1.2024

Introduction

The discrete event simulation project at hand is a journey into the intricacies of managing processes within a computing environment. The primary objective of this simulation is to mimic the behavior of a computer system by capturing and processing discrete events that unfold over time. The simulation is implemented in the C programming language, demonstrating a procedural approach to modeling the dynamic nature of process execution.

Simulation Architecture

Process and Program Structures

The heart of the simulation lies in the definition of two key structures: Process and Program. The Process structure represents an active form of an instruction list, complete with attributes such as arrival time, priority, and execution history. On the other hand, the Program structure is a passive representation of an instruction list, focusing on the program's name and array of instruction times. The use of these structures allows for a clear distinction between the characteristics of a process and its underlying program.

Queue and CPU Structures

The simulation employs a priority queue to manage processes, with the Queue structure encapsulating the size and an array of processes. Additionally, the CPU structure is utilized to model the central processing unit, keeping track of its occupancy status, the last execution time, and the currently executing process. This modular approach provides a flexible foundation for capturing the essential components of a computer system.

Challenges Faced

Memory Management

One of the primary challenges encountered during the project was effective memory management. The dynamic nature of processes and programs demanded careful allocation and deallocation of memory to prevent memory leaks. Despite challenges, the implemented code demonstrates a conscientious approach to memory management, emphasizing the importance of releasing allocated memory at the appropriate stages.

Process Array Decision

A significant decision in the project involved the representation of processes as an array within the Queue structure. This decision required intricate handling during processes' insertion and removal, leading to the use of memcpy and memset operations. This choice was not without its complexities, but it was too late to return to a more ideal data type because of the sunk costs. Despite its non-ideal nature, I made it work through hard work.

Simulation Workflow

Event Determination and Simulation

The simulation workflow revolves around the concept of discrete events, each event representing a significant occurrence in the system. The determination of the next event time guides the simulation. The possible next events are:

1. New process admissions
2. Instruction execution completion
3. Context switches

The careful orchestration of these events ensures an accurate representation of a computer system's dynamics.

Conclusion

Despite facing challenges in memory management and structural decisions, the project successfully simulates a dynamic computing environment. This project also had me think about how to structure a project most efficiently, and what design choices are better.