Team 08
s192865 KATOUZIAN Pouria
s201354 Gerard Manon

INFO0940-1 - Project 1 (2024)

Overview of your approach

Explain here a short overview of your structure/code (you can include figures).

Joker

multicore CPU.

Project Comprehension

Q) Assuming that there is no context switch time (set to 0) and all processes are equally important, what do you think is the best scheduling algorithm among the four proposed? Why?

In the absence of any form of priority considerations and negligible context switch costs (context switch time set 0 as mentioned in question), the Round Robin scheduling algorithm stands out as an optimal choice.

Q) Do you think this algorithm is implemented in most operating systems? Why?

Due to our research and understanding, various scheduling algorithms are used in different operating systems based on their necessity. In most operating systems, Round Robin (RR) is used, albeit in specific cases and at a lower level within their own scheduling algorithms, which are typically more complex. These systems often combine multiple scheduling algorithms or utilize different ones altogether. For instance, the Multilevel Feedback Queue algorithm, Completely Fair Queuing (CFQ), and hybrid scheduler (combining Multilevel Feedback Queue scheduling and Round Robin (RR)) are employed in Windows, Linux, and macOS.

In all three operating systems, RR is utilized at a lower level for Time Sharing System goals. In other words, it manages processes within specific priority queues or for short-term scheduling decisions that benefit from time slices. RR's strengths in fairness and ensuring all processes get a chance to run come into play, and it plays a supporting role in most operating systems, particularly for ensuring responsiveness in interactive environments.

However, RR is avoided as a sole algorithm due to its limitations, such as context switch overhead and potential drawbacks for long or I/O-bound processes. Other algorithms might be better suited for these situations.

Feedback

- Difficulty: This project was significantly harder than past projects which we have done for other courses. We encountered numerous files containing uncommented functions, making comprehension a challenge. There was a significant disconnect between the course material and the project's difficulty level.
- Amount of work: It took us more than three weeks to understand the purpose of different files and their relationships, all in order to define distinct sub-problems. Each team member spent over four days coding due to the project's complexity and the numerous tasks required.
- Other: Requiring this type of project is unfair given our restricted time and lack of expertise. Such a project would be more appropriate for advanced operating system courses, intended for individuals with a deeper understanding of operating systems than we currently possess.