

Discussion 13.1

Parallel Algorithms and Multithreading Q&A

| | |
|---|--|
| Course Code: CPE010 | Program: Computer Engineering |
| Course Title: Data Structures and Algorithms | Date Performed: 10/30/25 |
| Section: CPE 010-CPE21S4 | Date Submitted: 10/30/25 |
| Name(s): KERWIN JAN B. CATUNGAL | Instructor: Engr. Jimlord Quejado |

Answer the following:

1. What is Parallel Algorithm and how does it work?

A parallel algorithm is a type of algorithm that distributes a larger task into a small number of independent tasks and then solving them using multiple processing elements simultaneously. Parallel computing is more efficient than the serial approach as it requires less computation time. Instead of working step by step like a normal algorithm this one performs several tasks together to save time. Each processor handles a portion of the data and then the results are combined to get the final output.

2. What are the advantages in parallel algorithm?

Parallel algorithms are faster because they can complete multiple tasks at once instead of one after another and also, they make good use of modern multicore processors and that improves performance and efficiency. This approach is especially useful for handling large amounts of data or complicated calculations and it also helps reduce the time it takes to solve problems in areas like simulations, scientific computing, and data processing.

3. What is multi-threading?

Multithreading is the ability of a program or an operating system to enable more than one user at a time without requiring multiple copies of the program running on the computer. Multithreading can also handle multiple requests from the same user. Each user request for a program or system service is tracked as a thread with a separate identity. As programs work on behalf of the initial thread request and are interrupted by other requests, the work status of the initial request is tracked until the work is completed. Each thread performs a specific task, allowing multiple parts of a program to run at the same time. This helps the CPU stay busy and improves overall speed and responsiveness. It's commonly used to make programs run smoother, especially when dealing with tasks that can be done simultaneously.

4. Where can multi-threading be applied?

Multi-threading can be used in applications where multiple tasks need to happen at once, like Transaction Processing. Widely used in banking, online payments, and mobile recharges and also Multiple transactions can be processed simultaneously without delays like Web and Internet Applications like Threads that handles multiple client requests at the same time Improves responsiveness and efficiency of online platforms. Banking & Financial Systems here is the Online fund transfers, balance updates, and background verification tasks run in parallel. Prevents bottlenecks during high traffic Telecom & Recharge Services Multiple recharge or service requests are processed concurrently and ensures quick response and smooth functioning for millions of users.

E. External References

<https://www.geeksforgeeks.org/mobile-computing/parallel-algorithm-models-in-parallel-computing/>

<https://www.slideshare.net/slideshow/parallel-computing-advantages-and-disadvantages/47743617>

<https://www.techtarget.com/whatis/definition/multithreading>

<https://www.geeksforgeeks.org/operating-systems/multithreading-in-operating-system/>