National University of Computer & Emerging Sciences
Karachi Campus

Comparison between Process and Threads in Android

Project Report
Operating System
Section: CS-4K

Ali Shaz
Ali Zain

- ## Introduction

  In Android, processes and threads are used for concurrent execution. Processes are isolated units of execution with their own memory space and communicate with each other using inter-process communication. Threads are lightweight units of execution that share the same memory space and communicate with each other using shared memory. Processes provide more security but are slower, while threads are faster but can lead to performance issues if not managed properly. The main thread is responsible for updating the UI, while background threads are used for other tasks. The choice between processes and threads depends on the specific needs of the application.

- ## Project Specification

Software Specs:

We will be executing sorting algorithms using both threads and process and will aim towards finding the difference if any and reaching a valid conclusion.

Tools, and Technologies:

Programming Language: C language
Platform: Ubuntu

- ## Problem Analysis

Problem: how sorting algorithms react by executing in a thread or a process?

Process: run each algorithm with thread and process and find the difference.

- ## Solution Design

We will run each algorithm firstly as a thread then a process and after each algorithm is executed, we will analyze the results to reach a conclusion whether it is efficient to run such algorithms on thread or process

- ## Project Breakdown

Project planning:
Identify the sorting algorithms to be used and the input data to be sorted.
Determine the desired level of concurrency and the hardware requirements for the application.
Define the communication mechanisms to be used between threads or processes.

Design:
Define the software architecture, including the thread or process model, and the data structures to be used.
Design algorithms for data partitioning, sorting, and merging.
Develop synchronization mechanisms to avoid data races and ensure correct results.

Implementation:
Implement the software architecture and algorithms using the chosen programming language and tools.
Develop the communication mechanisms between threads or processes.
Write unit tests to verify the correctness of the sorting algorithms and the concurrency model.

Testing:
Conduct system testing to verify the functionality and performance of the application.
Conduct stress testing to determine the maximum size of input data that can be sorted efficiently.
Conduct integration testing to verify the communication mechanisms between threads or processes.

Deployment:
Deploy the application to the target environment.

Monitor the application for errors and performance issues.
Provide user documentation and support for the application.
Overall, the project breakdown for using sorting algorithms with threads and processes requires careful planning and design, as well as thorough testing and deployment to ensure that the application meets the requirements for performance, scalability, and correctness.

- **Results**

The results of using sorting algorithms with threads and processes will depend on several factors, such as the size of the input data, the number of sorting algorithms being used, and the hardware resources available.

Using threads can provide higher performance and efficiency than using processes, as threads can share memory and have lower overhead. However, the use of threads can also lead to synchronization issues and data races, which can affect the correctness of the sorting results.

Using processes can provide greater isolation and security between the different sorting algorithms being executed but can also have higher overhead due to the need for inter-process communication mechanisms.

Overall, the choice between threads and processes will depend on the specific requirements of the application, such as the size of the input data, the number of sorting algorithms to be used, and the desired level of security and isolation.

With careful design and implementation, using sorting algorithms with threads or processes can provide significant performance benefits for large-scale data processing applications, such as data analytics, scientific computing, and financial modeling.

• Acknowledgement

We would like to acknowledge the following GitHub repository and different articles that helped us in understanding the problem better.

https://github.com/kboranbayev/processes-vs-threads