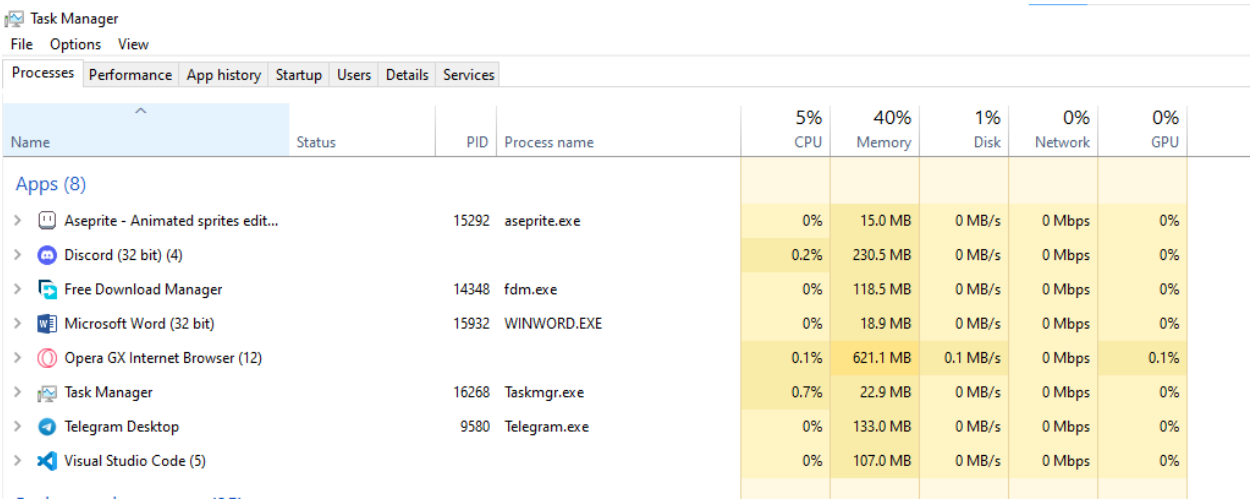
I.



The tabs CPU, MEMORY, DISK helps the user to monitor which process uses most of the memory and ram etc. task manager allows users to terminate applications and processes, adjust processing priorities and set processor affinity as needed for best performance.

II

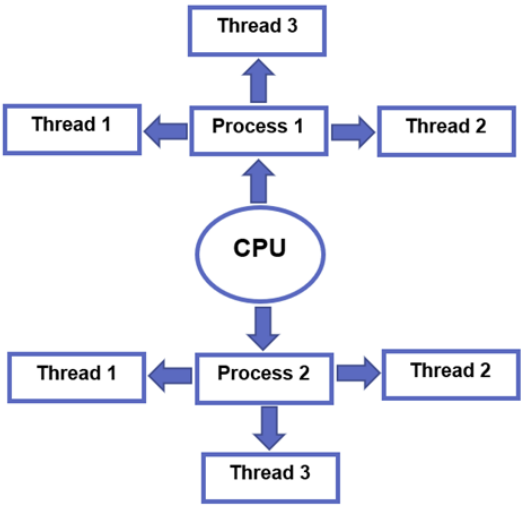
1.

LINUX

Process and threads in LINUX and Android OS

- **Provide a diagram showing the process and thread construct of the operating system.**
- **Briefly explain the diagram that you have provided.**

Linux



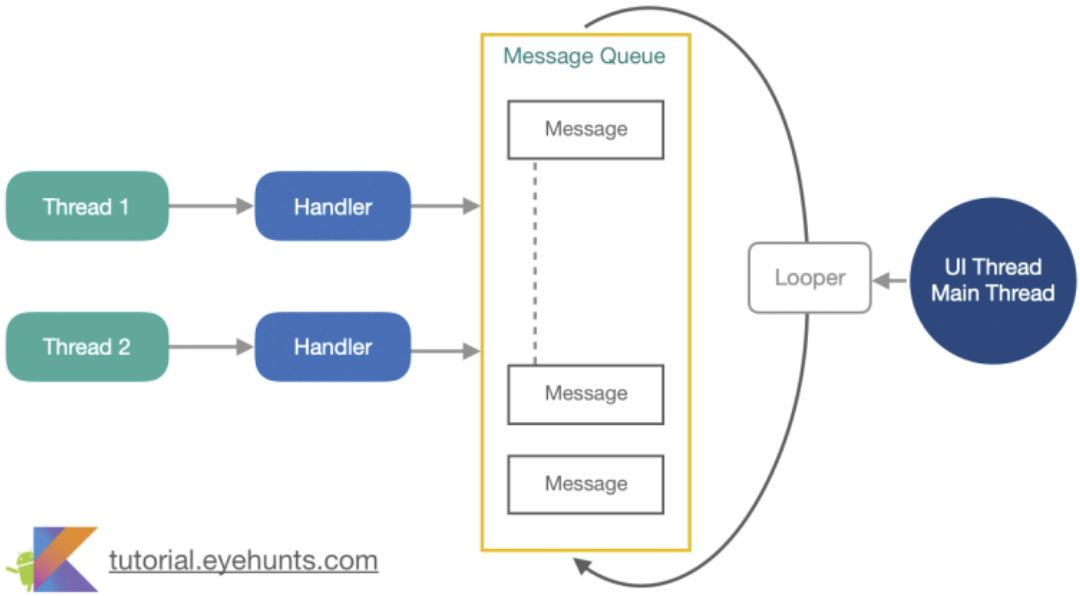
Process:

A computer program running is referred to as a process. On Linux, several processes are active at once. Let's look at an example of using the “ps” command to observe all the processes operating on the machine to demonstrate how we may monitor them on the terminal using the “ps” command or on the System Monitor UI.

Thread:

Thread is an execution unit that is part of a process. A process can have multiple threads, all executing at the same time. It is a unit of execution in concurrent programming.

Android



Process:

By default, all components of the same application run in the same process and most applications should not change this. However, if you find that you need to control which process a certain component belongs to, you can do so in the manifest file.

Thread:

When an application is launched, the system creates a thread of execution for the application, called "main." This thread is very important because it is in charge of dispatching events to the appropriate user interface widgets, including drawing events. It is also almost always the thread in which your application interacts with components from the Android UI toolkit (components from the android.widget and android.view packages). As such, the main thread is also sometimes called the UI thread. However, under special circumstances, an app's main thread might not be its UI thread; for more information, see Thread annotations.

- **How does the operating system support or implement multithreading?**

Linux

Linux features a distinct thread implementation. All threads are implemented by Linux as ordinary processes. There are no unique scheduling semantics or data structures that represent threads in the Linux kernel. A thread is essentially a process that collaborates with other processes to share resources.

Each thread has a distinct task struct, and the kernel sees them as regular processes (which just happens to share resources, such as an address space, with other processes).

Android

Android can use multiple CPU cores for multithreading, but the kernel and JVM handle that process, not the developer himself. An internal multithreading design will improve the program's basic performance, but the device upon which it actually runs will determine its speed.

- **Is it possible to increase the number of threads within processes without affecting the average response time of this operating system? Why or why not?**

  LINUX

The maximum number of simultaneous executions that a process can support is specified by the maximum threads per process parameter in Linux. This can be changed to minimize execution latencies and throttle the operation. When a process reaches this limit, it indicates that at its busiest, it requires that many threads. But the procedure is well-tuned as long as it can respond to demands quickly.

When the cap is reached, though, threads queue up and the process can become overloaded. Until the number of active threads falls below the limit, the process postpones establishing new threads.

ANDROID

Android apps are multithreaded by developers to enhance their usability and performance. The rest of the application can continue to run while processor- or resource-intensive processes are finishing up by spinning off these tasks into separate threads. This may entail disabling the graphical user interface thread and the threads that retrieve network data. The JVM handles these numerous tasks automatically, so Android developers do not need to worry about thread performance optimization for a specific configuration of hardware.

- **What are the possible effects of multithreading in the central processing unit (CPU) utilization of this operating system? Rationalize your answers**

Linux

Multithreading allows many parts of a program to run simultaneously. These parts are referred to as threads, and they are lightweight processes that are available within the process. As a result, multithreading increases CPU utilization through multitasking.

Android

Developers' multithread Android applications in order to improve their performance and usability. By spinning off processor- or resource-intensive tasks into their own threads, the rest of the program can continue to operate while these processor intensive tasks finish working.

Properly cite all your references (i.e., books, articles, dissertations, websites, etc.)

https://developer.android.com/guide/components/processes-and-threads

https://www.wideskills.com/android/application-components/android-threads-and-processes

https://www.javatpoint.com/benefits-of-multithreading#:~:text=Multithreading%20allows%20many%20parts%20of,increases%20CPU%20utilization%20through%20multitasking.

https://smallbusiness.chron.com/android-multithreading-performance-37880.html

https://linuxhint.com/process-vs-thread-linux/