Great observation, Gyanenjay! ✅ You're absolutely right — **we *can* create as many threads as we want**, even if we only have **1 or 2 cores**. But the important thing is to understand **how they're executed** under the hood.

Let’s break this down clearly 👇

**🤖 Yes, You Can Create Many Threads**

for (int i = 0; i < 1000; i++) {

new Thread(() -> {

System.out.println(Thread.currentThread().getName() + " is running");

}).start();

}

✅ This code will run **1000 threads** — **no problem!**

BUT...

**⚠️ Only a Limited Number of Threads Can Run *Simultaneously***

This is where **CPU Cores** come in 👇

**🧠 Example:**

| **CPU Type** | **What Happens** |
| --- | --- |
| **1 Core CPU** | Only 1 thread runs at a time. OS switches between them very fast (**context switching**) so it *seems* like multitasking. |
| **4 Core CPU** | 4 threads can actually run **at the same exact time** (true parallelism). The rest are **queued** or switched in and out by the OS. |
| **8 Core CPU** | 8 threads = full speed. 100 threads = 8 run, 92 wait and take turns. |

**🔄 Context Switching (What Happens Internally)**

* CPU **rapidly switches** between threads.
* This creates the **illusion of parallelism**.
* But context switching has a **cost** (overhead), especially when you have thousands of threads.

**🔥 Why Core Count Matters in Performance**

* If you have **more threads than cores**, threads wait.
* If threads are **CPU-bound** (heavy computation), **more threads won't help** much after a point.
* If threads are **IO-bound** (waiting for database, file, or network), you **can afford more threads** because they’re idle often.

**✅ Best Practice for Thread Pool Size (Rough Formula)**

// For CPU-bound tasks (e.g., math, computation)

int threads = cores + 1;

// For IO-bound tasks (e.g., network calls, DB access)

int threads = 2 \* cores or even more

You can get the available cores like:

int cores = Runtime.getRuntime().availableProcessors();

**📌 Conclusion**

* ✅ You **can create unlimited threads**.
* 🔄 But only as many as cores can run **truly in parallel**.
* ⏳ Extra threads are managed by the OS via **context switching**.
* ⚠️ Too many threads = overhead, performance drop.

Let me know if you want a **real demo** of this with example code showing **how many threads actually run in parallel based on your cores** 🙌