

What is the difference between a process and a thread in Java?

[Multithreading](#)

This is the most frequently asked question during interviews. In this post we will discuss the differences between thread and process. You must have heard these terms while reading [multithreading](#) in java, both of these terms are related to each other. Both processes and threads are independent sequences of execution. The main difference is that threads (of the same process) run in a shared memory space, while processes run in separate memory spaces. Lets see the differences in detail:

Thread vs Process

- 1) A program in execution is often referred as process. A thread is a subset(part) of the process.
- 2) A process consists of multiple threads. A thread is a smallest part of the process that can execute concurrently with other parts(threads) of the process.
- 3) A process is sometime referred as task. A thread is often referred as lightweight process.
- 4) A process has its own address space. A thread uses the process's address space and share it with the other threads of that process.

5)

Per process items	Per thread items
-----	-----
Address space	Program counter
Global variables	Registers
Open files	Stack
Child processes	State
Pending alarms	
Signals and signal handlers	
Accounting information	

6) A thread can communicate with other thread (of the same process) directly by using methods like wait(), notify(), notifyAll(). A process can communicate with other process by using [inter-process communication](#).

7) New threads are easily created. However the creation of new processes require duplication of the parent process.

8) Threads have control over the other threads of the same process. A process does not have control over the sibling process, it has control over its child processes only.