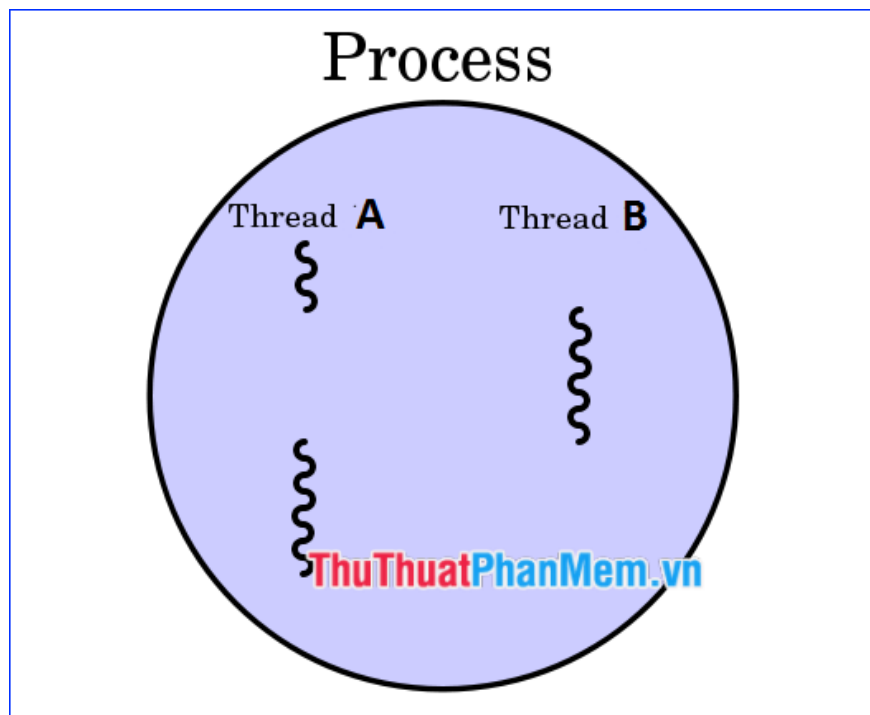


What is thread? Differences between Process and Thread

Thread and Process, these are terms that you will hear a lot if you study information technology, and need to understand when going to work. However, the definition and the difference between them is not well known, because they may not be in the knowledge taught by the school.

Thread and Process, these are terms that you will hear a lot if you study information technology, and need to understand when going to work. However, the definition and the difference between them is not well known, because they may not be in the knowledge taught by the school. To understand what Thread is, what is different from Process, please refer to the article below.

What is process?



Process, or process, is the execution of a program and performing the relevant actions specified in a program, or it is an executing unit where the program runs. The operating system creates, schedules, and terminates processes. Other processes created by the main process are called child processes.

The operation of any process is controlled by the process control block (*Process Control Block* , abbreviated as PCB). The PCB contains all the important information related to the processes, such as: process id, priority, status, CPU, etc.

The lifecycle of a process has certain states such as *ready* , *running* , *blocked* , and *terminated* . Process states are used to track the progress of the process at the present time.

The interlaced operations of processes help to increase the computing speed when I / O operations in one process overlap with the computational activities in other processes.

Attributes of a process:

- Creating each process includes system calls for each process individually.
- A process is an isolated executable and does not share data and information.
- Processes using IPC (*Inter-process Communication*) mechanism to communicate significantly increase the number of system calls.
- Manage the process of consuming more system calls.
- Each process has its own stack and heap memory, data and memory map.

What is thread?

Thread , or thread, is a small process that can be managed independently by a scheduler. All threads in a single program are logically contained within a process. The kernel allocates a stack and a flow control block (TCB) for each thread. The operating system only stores the stack pointer and CPU status at the time of switching between threads of the same process.

Threads are deployed in three different ways: kernel stream, user level thread, and hybrid thread. The stream can have three states *running* (*ready*), *ready* (*blocked*) and *blocked* (blocked); it only includes the state of unallocated computing and the state of communication reduces the cost of conversion, which increases the (parallel) concurrency so the speed also increases.

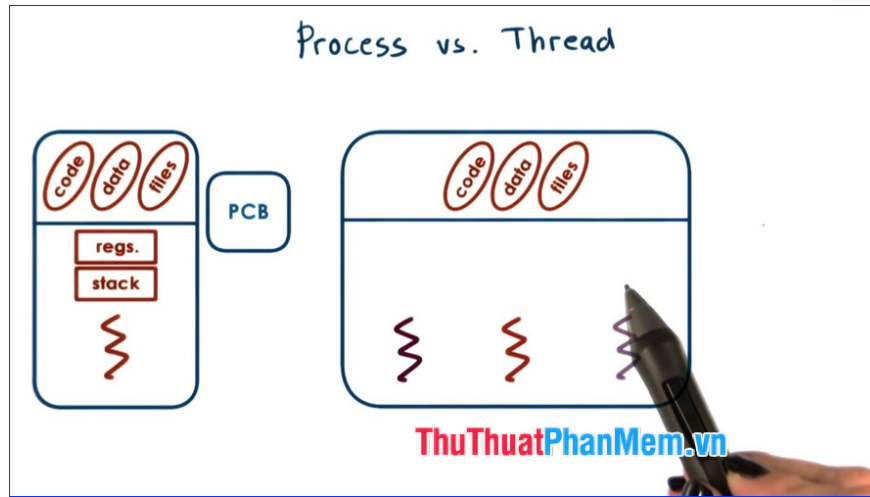
Multithreading also comes with many problems. Having multiple threads does not create complexity, but the interaction between them is not so.

A thread must have a priority attribute when there are multiple threads running. The time it takes to execute the corresponding active threads in the same process is determined by the thread priority.

Attributes of a stream:

- A system call may create more than one thread.
- Data and information sharing flow
- Shared heap memory flow but has its own stack and register.

- Manage non-consuming threads or consume less system calls because communication between threads can be achieved using shared memory.
- The isolated attribute of a process increases its cost in terms of resource consumption.



Compare Process with Thread

We have a table comparing the basic differences between *Process* (*Thread*) and *Thread* (thread) as follows:

Basis of comparison

Process

Thread (thread)

Define

Is the program being executed

Is a small process

Share memory

Completely isolated and does not share memory

Share memory with each other

Share data and code

Data and code independent

Share data segments, code segments, files, etc. with peer streams

Resource consumption

More

Less than

Time required to create

More

Less than

Time required for termination

More

Less than

Time switch context

More

Less than

Effective communication

Lower

Higher

Works when blocked

If one process is blocked, the remaining processes can continue to execute

If a user level thread is blocked, all its peer streams are also blocked

Unstable termination

The process is lost

Threads can be restored

In summary, we can summarize the most important differences as follows:

- All threads of a program are logically contained within a process.
- Threads are lighter than process.
- A process is an isolated execution unit while the thread is not isolated and has shared memory.
- A stream cannot exist separately; It is tied to a process. On the other hand, a process can exist individually.
- At the time of expiration of a thread, its associated stack can be restored because each thread has its own stack. Conversely, if a process dies, all threads also die, including the process.

Through this article, you have understood what Process and Thread are, as well as the difference between them, right? Hope the information just shared above will help you!

You finished reading the article "**What is thread? Differences between Process and Thread**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
