1. So sánh giữa Thread và Process:

	Process	**Thread**
Định nghĩa	Là chương trình (program) đang được thực thi	Là một "tiến trình" con
Tác vụ	Thực hiện các nhiệm vụ lớn	Thực hiện các nhiệm vụ nhỏ
Chi phí chuyển ngữ cảnh	- Cao hơn	- Thấp hơn
Tài nguyên cá biệt	- Cần không gian vùng nhớ riêng, không chia sẻ tài nguyên với nhau ➔ Heavy weight ➔ Độc lập với nhau	- Chia sẻ một số không gian vùng nhớ (code, data, files) → Light weight ➔ Phụ thuộc vào nhau
Thời gian tạo và kết thúc	- Chậm hơn	- Nhanh hơn
Khởi tạo và xóa	- Khó và chậm hơn	- Dễ và nhanh hơn
Giao tiếp	- Phức tạp (IPC làm tăng đáng kể số lượng system calls)	- Đơn giản hơn
Tính bảo mật	- Cao hơn	- Thấp hơn
- Process là tiến trình hoạt động một ứng dụng, Thread là một bước điều hành bên trong một process. - Một Process có thể chứa nhiều Thread - Khi 1 thread kết thúc, các vùng nhớ riêng (register, stack, …) sẽ được giải phóng, không ảnh hưởng đến Process chứa nó. Ngược lại, khi 1 process kết thúc, tất cả các threads bên trong cũng phải kết thúc.		

2. Khi nào nên sử dụng Multithreading và Multiprocessing?

Để biết khi nào nên sử dụng Multithreading và Multiprocessing, cần lấy đặc điểm của chúng làm cơ sở để so sánh và chọn lựa:

Sử dụng multiprocessing khi	Sử dụng multithreading khi
– Muốn cho người dùng thấy sự phản hồi của máy tính: Các chương trình này cần phải đợi người dùng để tương tác với nó, nên sử dụng threads sẽ hợp lý hơn. *Ví dụ phần mềm đánh máy Word, khi ta gõ thì sẽ có thread đảm nhận vai trò in chữ ra trên trang giấy, có thread đảm nhận vai trò nhận biết được người dùng gõ chữ nào.*	– Muốn phát triển các ứng dụng cần CPU nhiều: Multiprocessing sẽ giúp gia tăng tốc độ của tiến trình khi mà 1 chương trình nào đó cần CPU xử lý nhiều tác vụ. – Muốn xử lý dữ liệu trong 1 thời gian ngắn: Vì nhiều CPU có khả năng tính toán nhanh hơn.
– Tuy nhiên, tạo nhiều process thì càng tốn nhiều tài nguyên và quá trình quản lý không được linh hoạt.	– Tuy nhiên, tạo nhiều thead thì xử lý càng phức tạp. – Cần giải quyết tối ưu và hiệu quả vấn đề tranh chấp vùng nhớ trong critical section → cơ chế đồng bộ. – Cần phát hiện các deadlock

3. So sánh giữa Multithreading và Multiprocessing

	Multiprocess	**Multithread**
Cách nhận biết	Multiprocess sử dụng 2 hoặc nhiều CPU hơn để tăng tốc độ xử lý trên máy tính	Multithread sử dụng 1 process nhưng bao gồm nhiều câu lệnh code trong process đó để xử lý.
Điểm khác biệt	Multiprocess chú trọng vào	Multithread chú trọng 1

	việc thêm nhiều CPU để gia tăng tốc độ máy tính	process gồm nhiều dòng code trong đó để gia tăng tốc độ máy tính.
Chi phí chuyển ngữ cảnh	- Cao hơn	- Thấp hơn
Yêu cầu bộ nhớ	- Nhiều hơn	- Ít hơn
Thời gian tạo và kết thúc	- Chậm hơn	- Nhanh hơn
Khởi tạo và xóa	- Khó và chậm hơn	- Dễ và nhanh hơn
Giao tiếp	- Phức tạp (IPC làm tăng đáng kể số lượng system calls)	- Đơn giản hơn
Tính bảo mật	- Cao hơn	- Thấp hơn

- Process là quá trình hoạt động một ứng dụng, Thread là một bước điều hành bên trong một process.
- Một Process có thể chứa nhiều Thread
- Khi 1 thread kết thúc, các vùng nhớ riêng (register, stack, …) sẽ được giải phóng, không ảnh hưởng đến Process chứa nó. Ngược lại, khi 1 process kết thúc, tất cả các threads bên trong cũng phải kết thúc.