

Threads

1. What is a thread and its properties ✓

2. Benefits of multithreading ✓

3. Process vs Threads ✓

4. Multithreading models ✓

5. Writing multithreaded code in C ✓

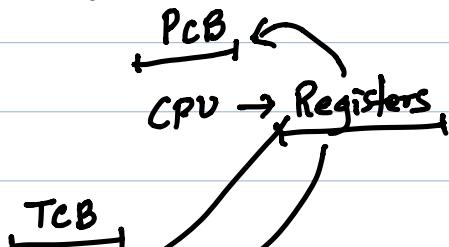
What is a thread?

A thread is a basic unit of CPU utilization.

- * CPU consists of multiple registers,
- * CPU fetches instructions using Program Counter and executes it

A thread consists of the following :

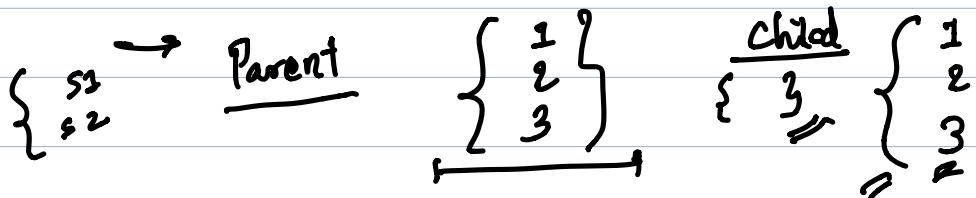
- 1. Thread id ✓
- 2. Program Counter ✓
- 3. Register set ✓



4. stack ↗ local variables
 ↘ function return code ↗ ~~Program~~
 ↗ ~~function~~

What does a thread share with other threads

- 1. Code Section of memory, Program Code,
 - 2. Data Section of memory, Global variables,
 - 3. Open files : ↗ Shared
 - 4. Signals : ↗
 - 5. Heap memory,
 Unique Id : ↗ file descriptor
- ↗ file descriptor
↗ Signals



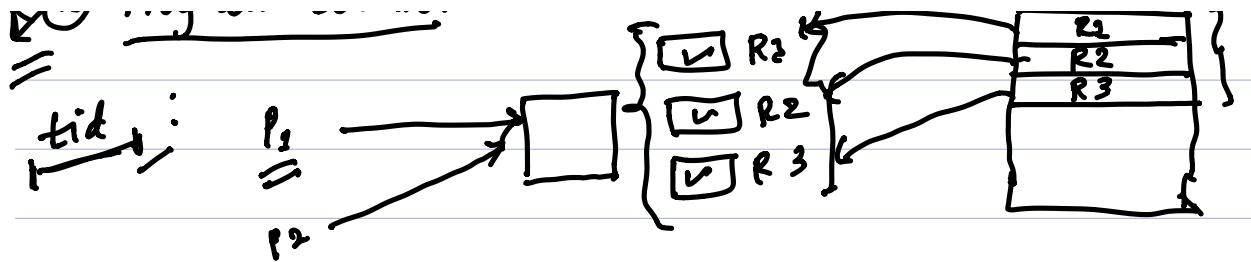
A thread is a basic unit of CPU utilization

What does CPU need to execute instructions?

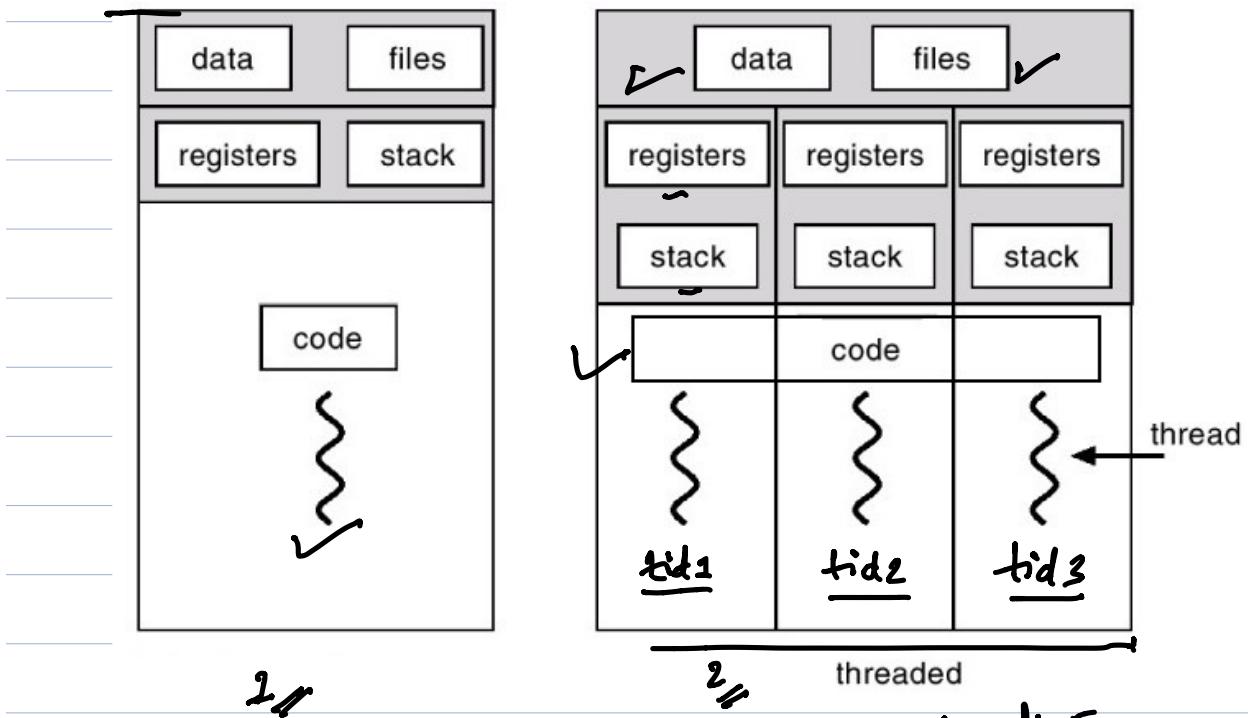
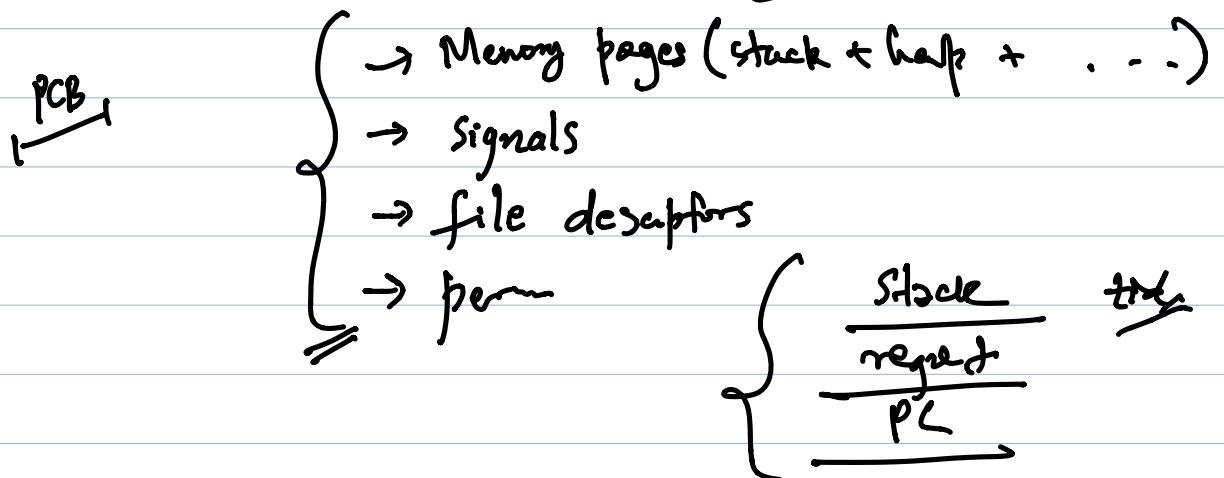
- ① Registers - includes stack register ⇔
 ② Program Counter (also stored in a register) ⇔
 Registers

What does the thread consist of?

- ① Registers - includes stack register too.
 ② Program Counter



Process is also a unit of CPU utilization



A program consist of threads. Multithreading +

By default, a program has Single thread of execution // $\xrightarrow{\quad}$ P₁ P₂ P₃

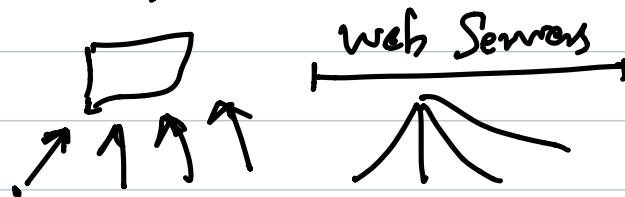
Benefits of multithreading :-

i) Responsiveness : Fast Response with interactive apps.

ii) Resource sharing : Memory sharing,

iii) Economy : Thread creation is fast.

iv) Scalability : Parallel execution,



Process vs Threads

Process	Thread (OS threads)
A program in execution	Part of a process
Takes more time for creation	Takes less time for creation
More time for context switch	Less time for context switch

Consumes more resources	Consumes less resources
Memory is not shared	Memory is shared
Process communication is complex	Easy and efficient communication
If one process crashes, it <u>doesn't affect other processes</u>	If one thread crashes, all threads crashes
* All resources are duplicated in a process. Duplication or copying consumes time;	
* Since all resources are duplicated in a process, it consumes more resources;	
* Process communicates using inter-process communication techniques like pipe, sockets and remote procedure calls (RPC) is complex.	

* Threads can easily communicate using shared memory.

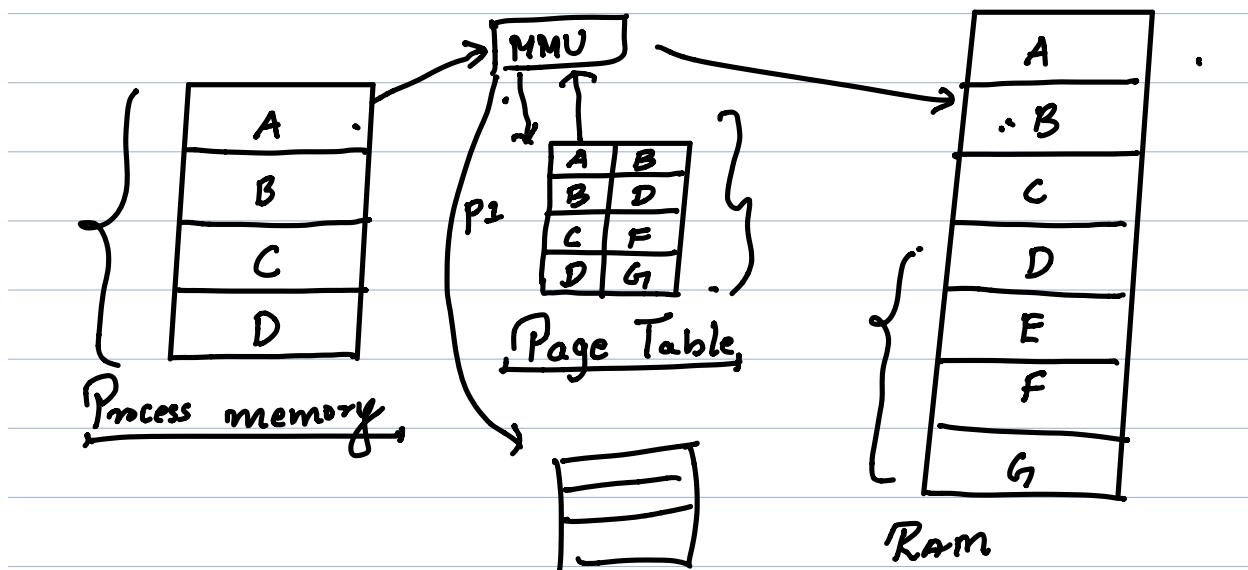
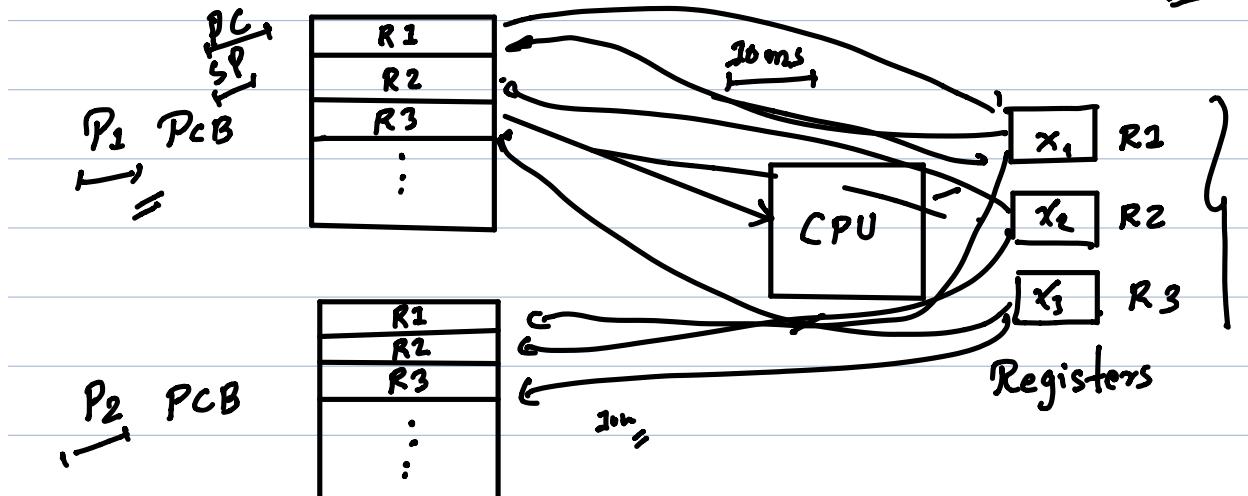
Context switch Time

Two important thing happens during context switch :-

(i) Switch the virtual memory mapping or base

tables from one process to another.

ii) Switch the processor state from one to another by
Saving and restoring information stored on registers.



When to use process vs threads :

* Threads are faster than processes

- ✓ 1 =
- * Crashing of one thread crashes other threads too
 - * No Security between threads - Can lead to race conditions.
- 2

✓ A trade off between speed vs reliability

* Chrome vs Firefox

* Web Servers

Performance

} =