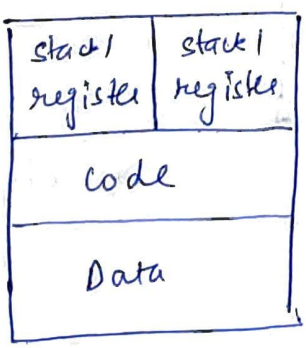
Processes v/s Threads L-1.11

→ We are talking about multitasking platform

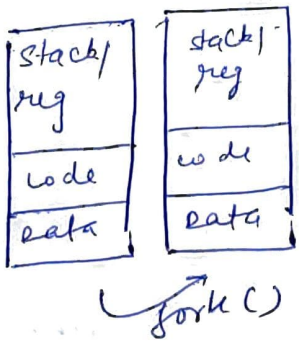
Process	Threads (user level)
(i) System calls involved in process	There is no system call involved
(ii) OS treats different process differently.	All user level threads treated as single task for OS.
(iii) Different process have different copies of data, files, code.	Threads share same copy of code and data
(iv) Context switching is slower	Context switching is faster
(v) Blocking a process will not block another.	Blocking a thread will block entire process.
(vi) Independent	Interdependent.

If one thread is blocked, all are blocked



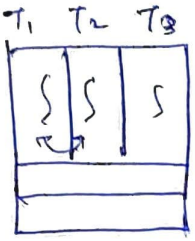
Threads

If one parent process is blocked, child processes are unaffected.



Processes

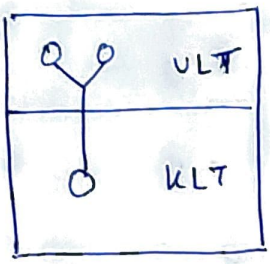
L- 1.12 User Level Threads v/s kernel level threads

	User Level Thread	kernel level thread
(i)	User level threads are managed by user level library	kernel level threads are managed by OS system calls.
(ii)	User level threads are typically fast	kernel level threads are slower than user level.
(iii)	Context switching is faster. (managed by library) 	Context switching is compatively lower. (because OS is invoked)
(iv)	If one user level thread perform blocking operation, then entire process get blocked. Because OS does not know about threading in user level process	If one kernel level thread is blocked, no effect on others.

Process > KLT > ULT

context switching time.

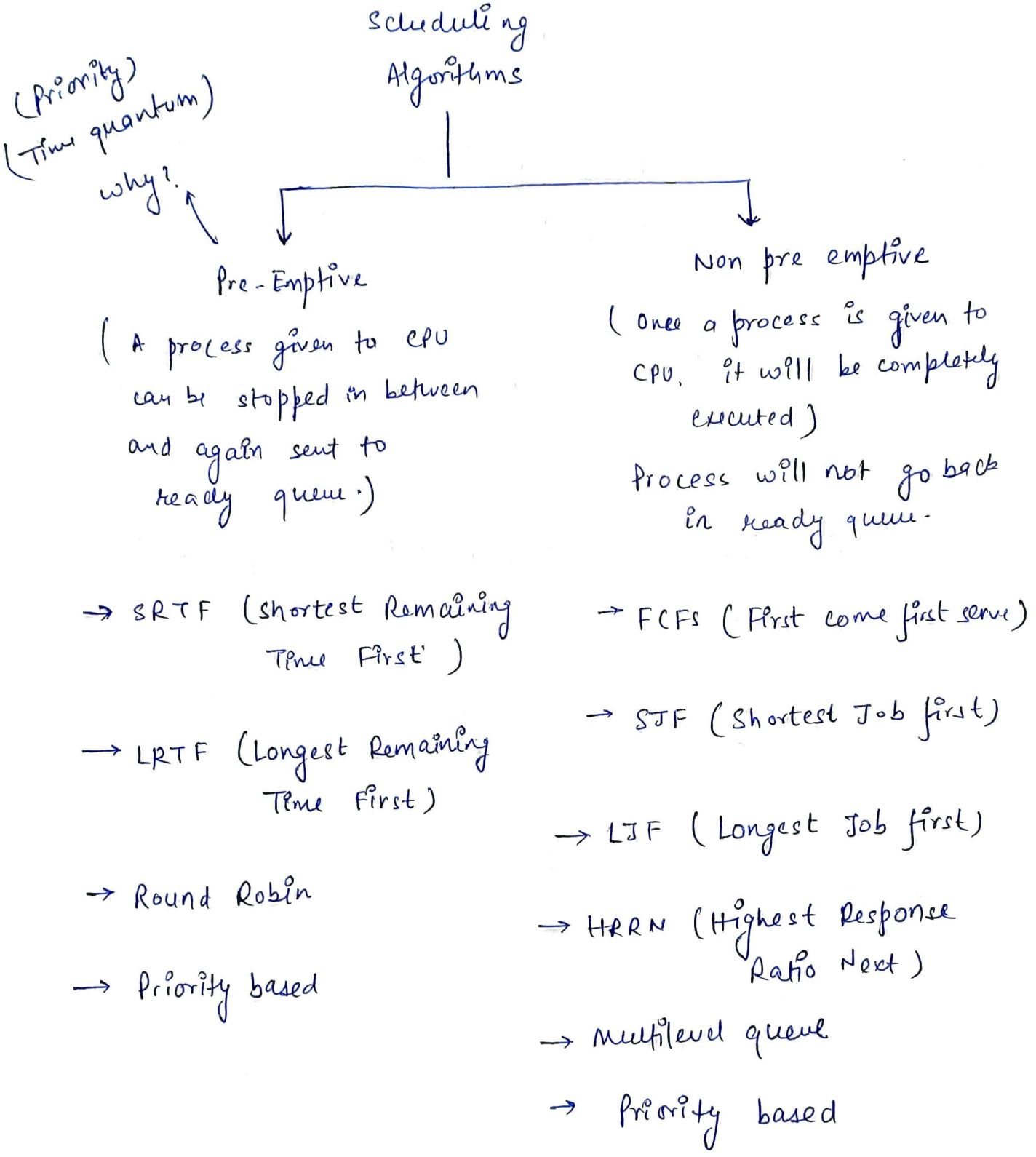
Nowadays, we use hybrid environment.



→ Threads always share code and data. That is why it is called lightweight.

L-2.1 . Scheduling Algorithms

↳ Algorithms to select processes from ready queue and giving it to CPU are called scheduling algorithms.



L-2.2 Times in CPU scheduling

(i) Arrival time : The point of time at which process enters the ready queue or state.

(ii) Burst time : Duration of time required by a process to execute on CPU.

(iii) Completion time : Point of time at which process completes its execution.

(iv) Turn Around Time : { completion time − Arrival time }

(v) Waiting time : { Turn Around time − Burst time }

CPU bound process → time it takes to execute on CPU

I/o bound → time for I/o (goes into waiting time)

(vi) Response time : { (Time at which a process gets to CPU first time) − Arrival time) }