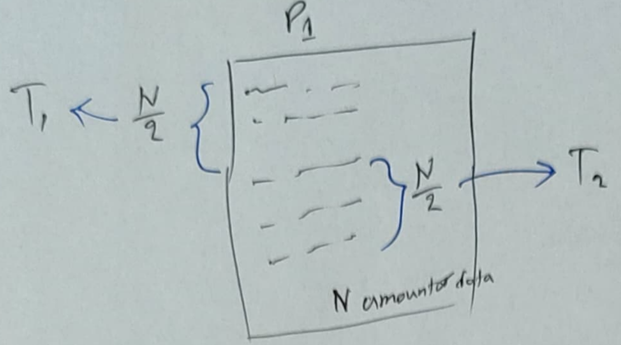
CSE 323 / L-18 / 27.04.2024 /

⊛ Data Parallelism:



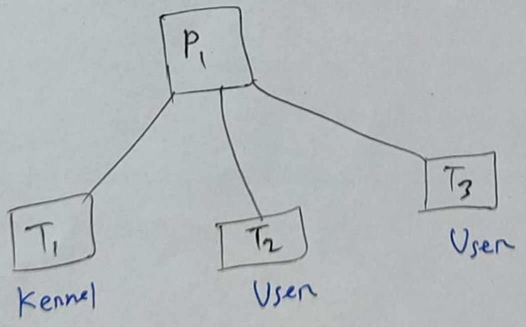
$$\left.\begin{array}{l} \text{Core-1} \Rightarrow T_1 = \frac{N}{2} \\ \text{Core-2} \Rightarrow T_2 = \frac{N}{2} \end{array}\right\} \text{N data}$$

⊛ User thread:

- created in user mode
- kernel is not aware of the existence of threads.

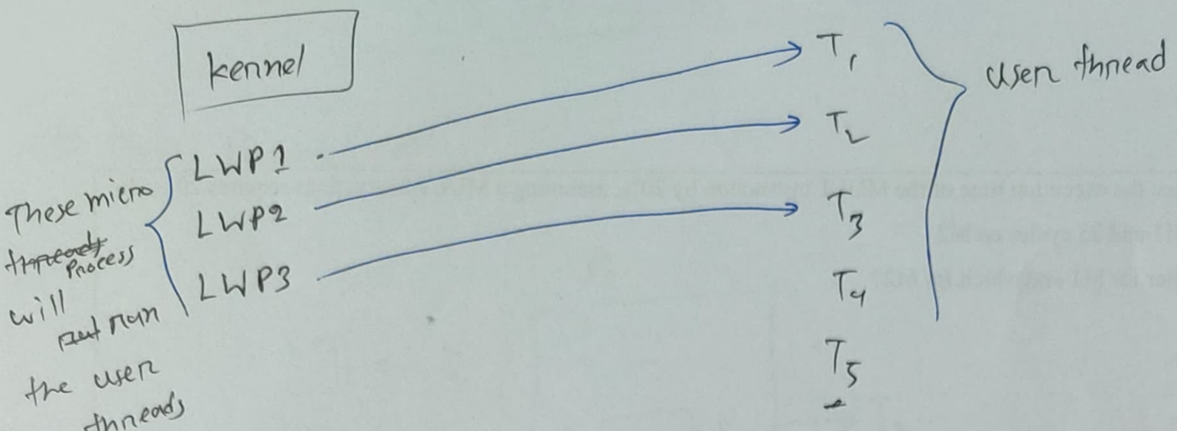
⊛ kernel threads:

- created ~~b~~ and manage by kernel



$$\left.\begin{array}{l} \text{Core-1} = T_1 \Rightarrow T_1 \\ \text{Core-2} = T_2 \Rightarrow P_1 \\ \text{Core-3} = T_3 \Rightarrow P_1 \end{array}\right\} \text{kernel view.}$$

interms of Process, it is not possible


kernel
LWP1
LWP2
LWP3
T1
T2
T3
T4
T5
user thread
These micro ~~threads~~ process will ~~but~~ run the user threads

LWP = Light Weight Process

Core1 ⇒ LWP 1 ⇒ T_1

Core-2 ⇒ LWP 2 ⇒ T_2

⊛ Advantage of user level threads:

- cheap to create and destroy
- switching threads is so fast

Drawback:

- blocking system call will immediately block the entire process.
- Solution given above.

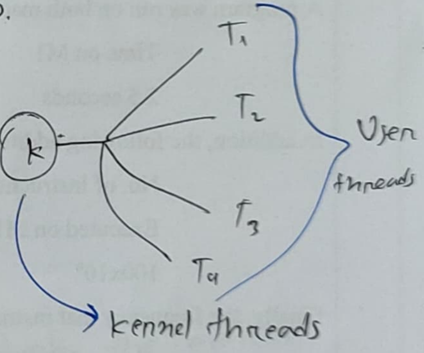
$\boxed{4.15}$

L-19/ 02. 05. 2024/

⊛ Multithreading Models:

(i.) Many to One!

- Multiple user threads ⇒ One kernel threads
- One kernel threads can run one user threads at a time. So multiple user threads will run one after another. Parallelism not supported
- manged by thread library in user spaces.
- if one user threads get blocked by I/O, then other user threads will be blocked too.

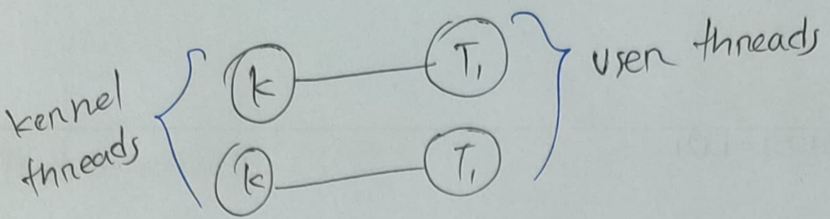


⇒ - Solaris Green threads
- GNU Portable threads

(ii) One to One

- One user threads ⇒ one kernel threads.
- better than many to One.
- allows multiple threads run in multiprocessor
- Number of threads per process is restriefed
- Number of kernel & user threads limited. ⊛ Max 100 ~~than~~ kernel threads allowed in standard.

⇒ - Windows
- Linux
- Solaris 9 and later

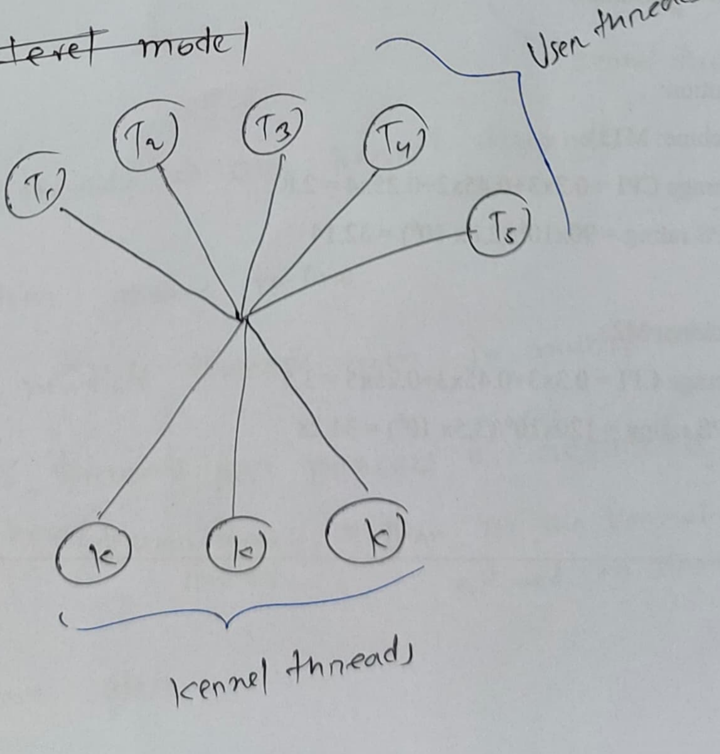


(iii) Many to many

- multiple user threads ⇒ multiple kernel threads
- there is a limit of kernel thread. No limit for user threads
- User threads count ≥ kernel threads count
- Extra user threads will wait ~~to be fr~~ for kernel threads to be free.

⇒ - Solaris prior to version 9
- Windows with the threadfiver package.

⊛ ~~Two level~~ model

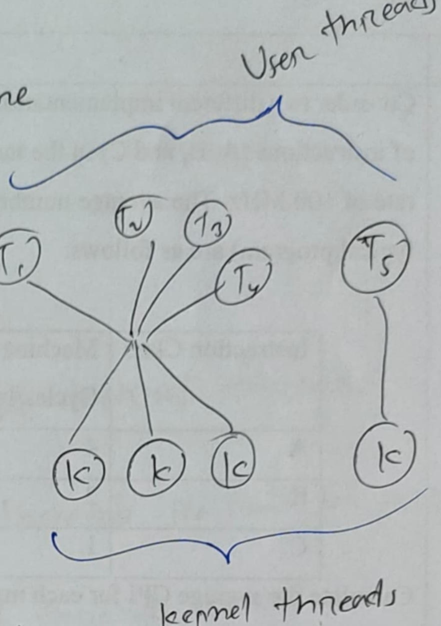


⊛ Two-level model

- many to many + one to one

⇒ - ZRIX

- HP-UX
- Tru64 UNZX
- Solaris 8 and earlier



- if there already some threads waiting for ~~kerl~~ kernel threads, in the mean time if a high priority user threads create then kernel will create ~~and~~ one kernel threads for that one user threads.

Mid-2 upto this
09.05.2024