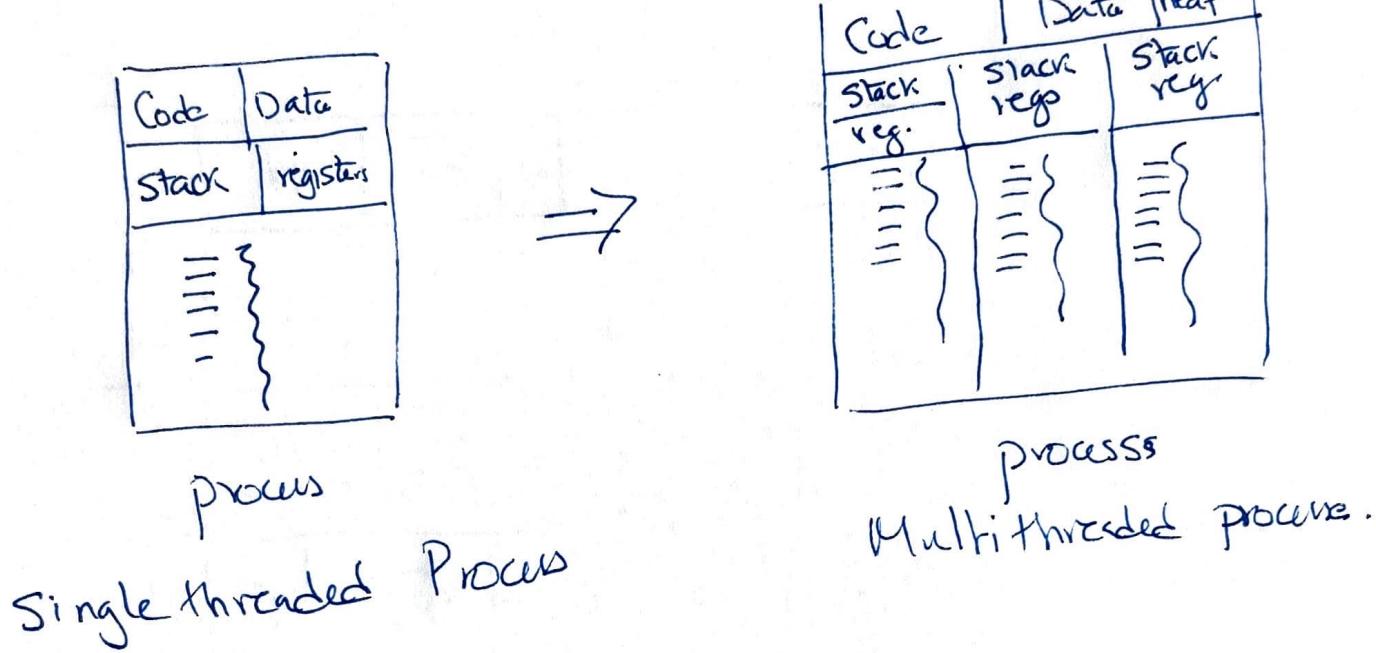


Threads

11

Goal

Support multiple paths of execution within a single process.



Why?

1) Responsiveness:

A process can continue running even if part of it is blocked (e.g. I/O)

2) Scalability:

run threads in parallel

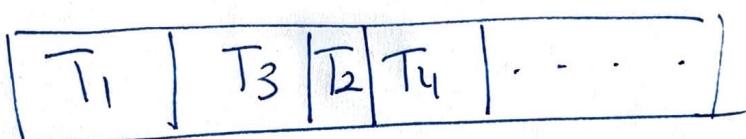
3) Resource Sharing:

Unlike shared memory or message passing, ~~not~~ threads share memory and resources by default.

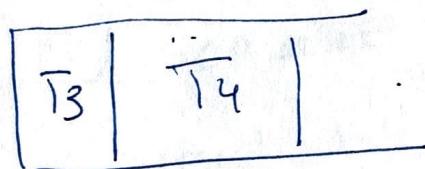
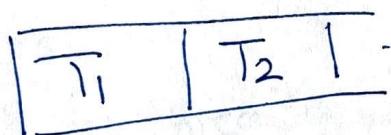
4 Economy : Thread creation ~~is~~ and management is faster than process creation and management 2

Example

1-CPU



2-CPU

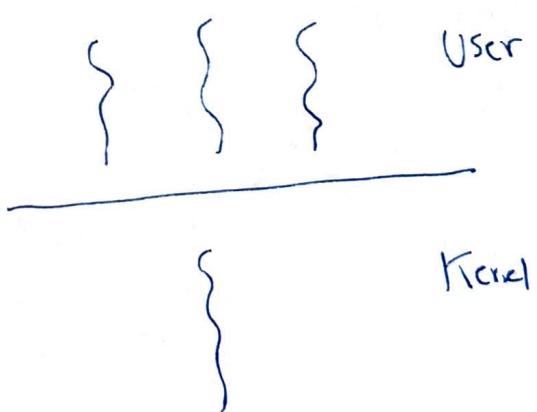


Multithreading Models

How is support for threads is provided?

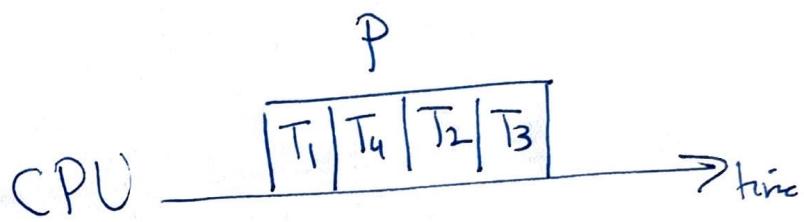
1 Many to one Model

maps many user-level threads into 1 Kernel thread.



Threading is provided by a library (e.g. Pthreads).

- * Create / delete threads
- * assign priorities
- * schedule threads.

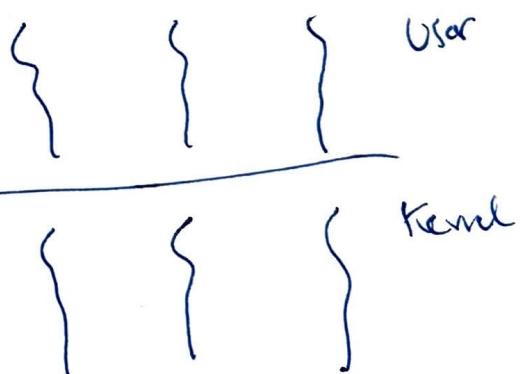


Drawbacks:

- [1] Multiple threads are unable to run in parallel on ~~a multiprocessor system~~ a multiprocessor system.
- [2] If a thread blocks, the whole process blocks.

2] One-to-One model

Map every user-level thread into a kernel-level thread



- * Can run in parallel.

Drawback:

A limit should be put in place on the number of kernel threads that can be created.

CPU1	T ₁
CPU2	T ₂
CPU3	T ₃

3) Many-to-Many

[4]



maps many user-level threads into a smaller number of kernel-level threads.

Adv Strives a good balance between the other models.

Some user-level threads can be bound into kernel-level threads.

OpenMP : Open multiprocessor.

Threading Issues

- 1 What happens if a threads calls fork()?
- a new process become a single-threaded (executing the calling thread code).
- b ~~does~~ the new process duplicates all the threads.

If exec() is called after forking (duplicating all the threads is not needed).

If exec() is not called, duplicating all threads is done.

- 2 Cancellation:
 what happens if the cancelled thread is in the middle of updating some data.
 ⇒ left in an inconsistent state.

Soln : Introduce cancellation points

3 Signal Handling :

6

Signals are used to notify the process of events.

they are delivered to the process.

Options in handling Signals:

- a Deliver signal to all threads.
- b ~ ~ to a certain thread.
- c Assign a thread to handle signals.
- d Deliver signal to the thread if applies to. (requires c).

4 Thread pools

Idea: Create a pool of threads that are ready to do work.

When a request arrives, a thread is launched to handle that request.

Adv + protect your system's resources.

* saves time creating threads when requests arrive.

Challenges in multi threading

7

- 1] Synchronization / Deadlocks
- 2] Efficiency: How to divide your process into threads?
- 3] Debugging and Testing.
hard to recreate problems that occur.