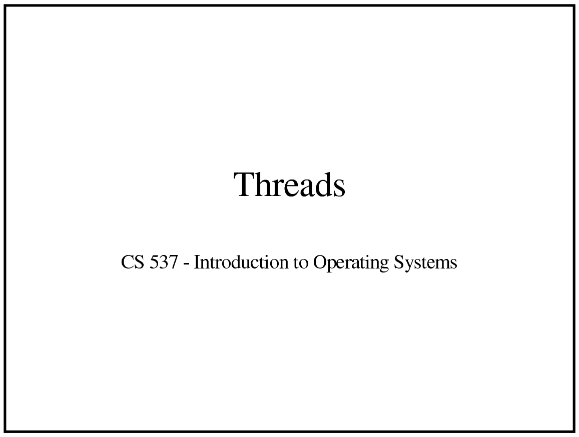
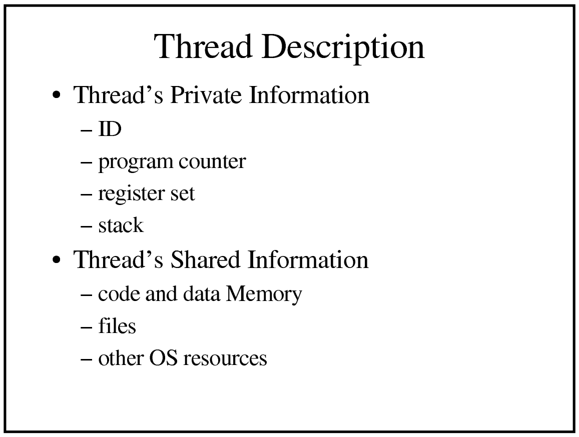


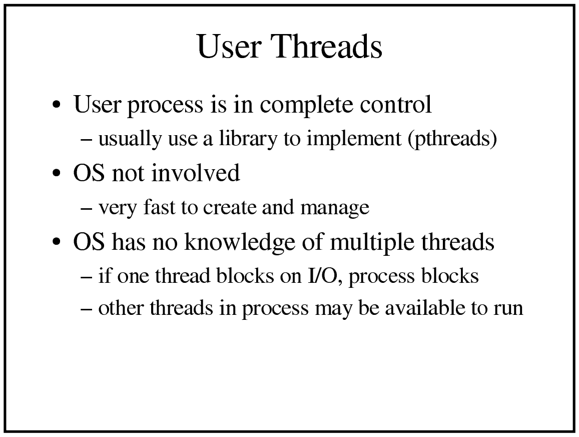
Threads

CS 537 - Introduction to Operating Systems



Thread Description

- Thread’s Private Information
  - ID
  - program counter
  - register set
  - stack
- Thread’s Shared Information
  - code and data Memory
  - files
  - other OS resources



User Threads

- User process is in complete control
  - usually use a library to implement (pthreads)
- OS not involved
  - very fast to create and manage
- OS has no knowledge of multiple threads
  - if one thread blocks on I/O, process blocks
  - other threads in process may be available to run

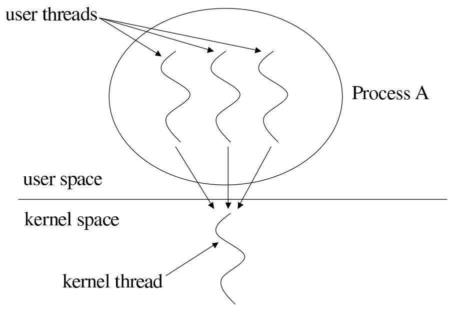
Kernel Threads

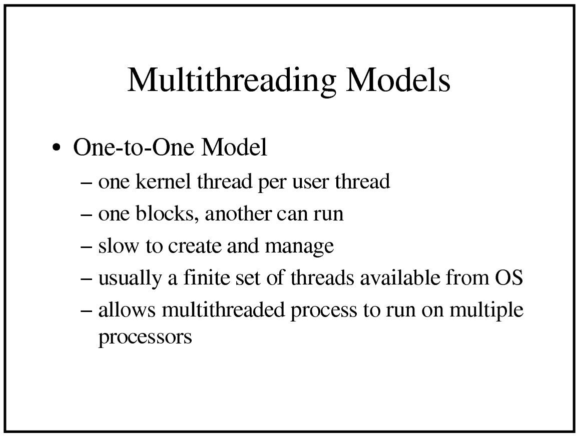
- OS devotes multiple threads to a process
- Controlled by OS
  - slow to create and manage
- OS does have knowledge of process threads
  - If one process thread blocks, OS can schedule another thread of process

Multithreading Models

- Many-to-One Model
  - one kernel thread per process
  - all process threads share a kernel thread
    - one blocks, all blocked
  - can’t run a multithreaded process on multiple processors

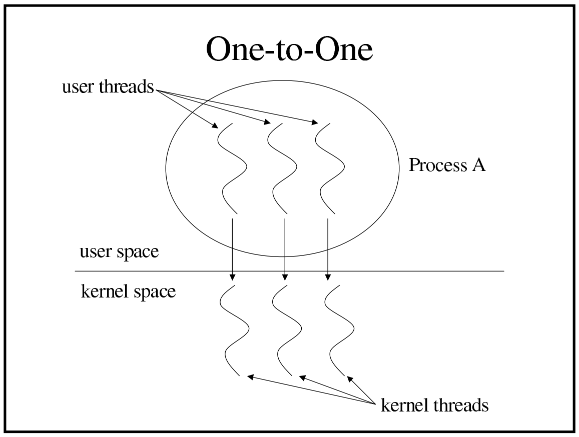
Many-to-One



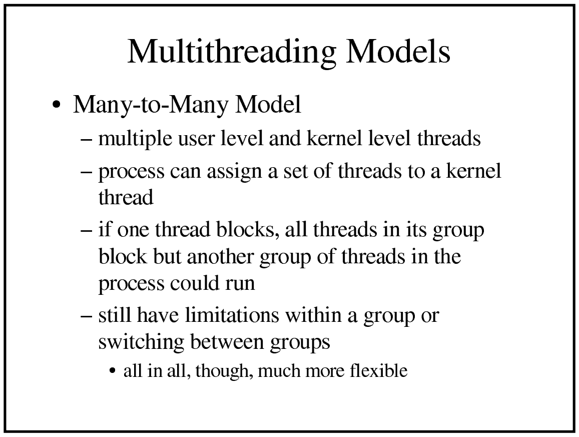


Multithreading Models

- One-to-One Model
  - one kernel thread per user thread
  - one blocks, another can run
  - slow to create and manage
  - usually a finite set of threads available from OS
  - allows multithreaded process to run on multiple processors

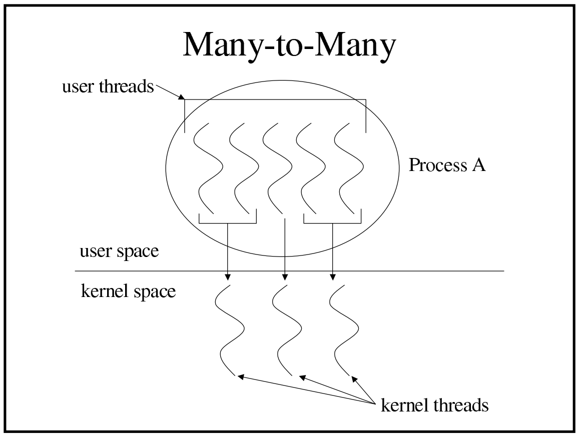


One-to-One



Multithreading Models

- Many-to-Many Model
  - multiple user level and kernel level threads
  - process can assign a set of threads to a kernel thread
  - if one thread blocks, all threads in its group block but another group of threads in the process could run
  - still have limitations within a group or switching between groups
    - all in all, though, much more flexible



Many-to-Many

user threads
Process A
user space
kernel space
kernel threads