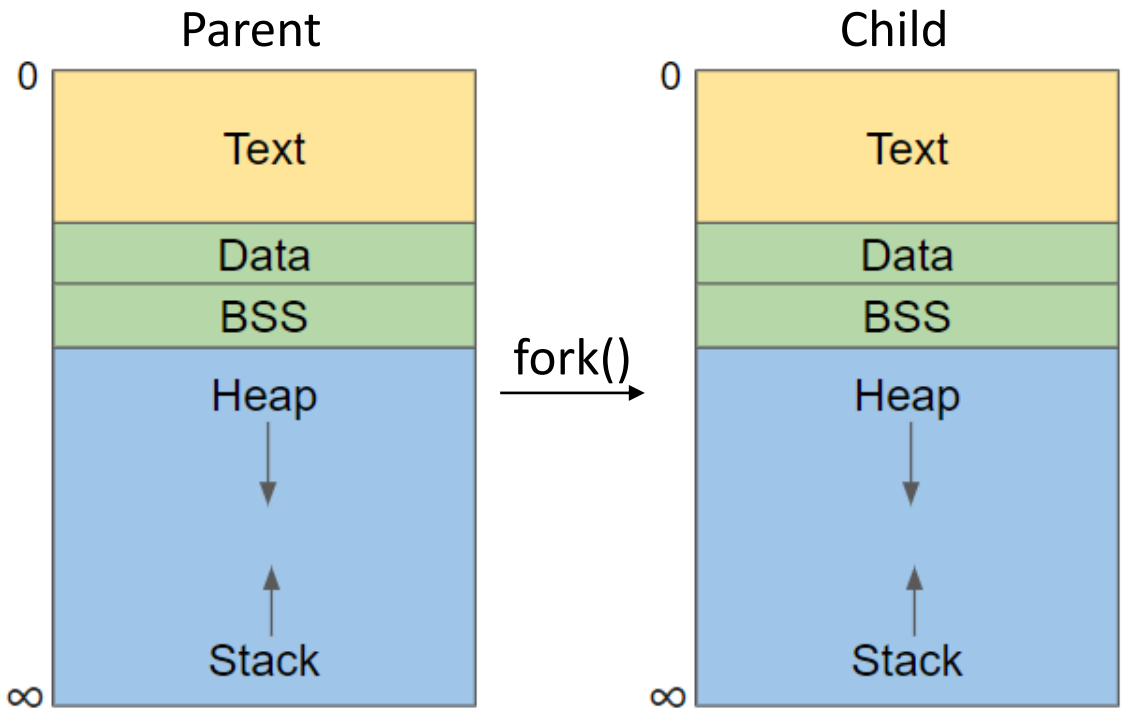
CS3841 – Design of Operating Systems
Threads

Problem

- Parent and child process do not share address spaces

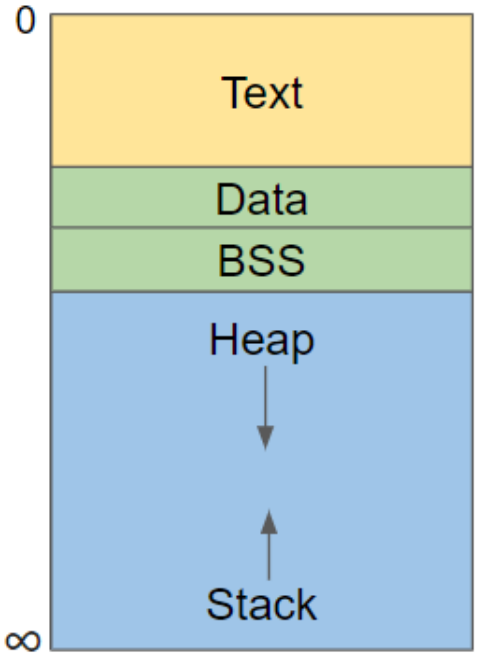
Question

- How can parent and child communicate? Inter process Communication
- Is there an easier way?



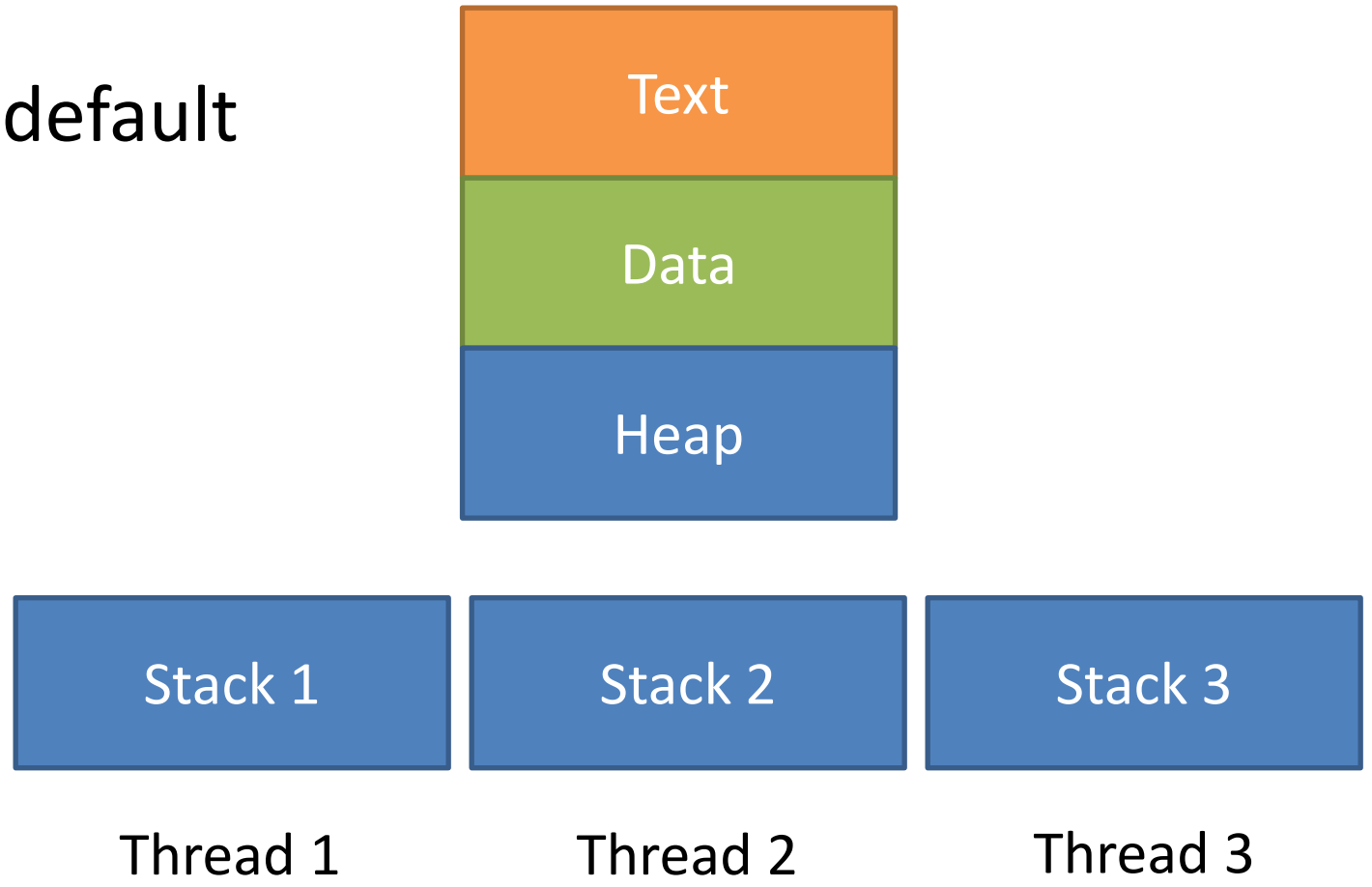
Inter Process Sharing

- Sharing variables is easier than sharing via handles or descriptors
  - Unstructured
  - Load/store
- How much do we want processes to share?
  - Text?
  - Heap?
  - Data?
  - Stack?



Threads

- "Processes" that share by default
  - Text
  - Data
  - Heap
- Each have their own stack
  - Why?
    Function calls
    Local variables

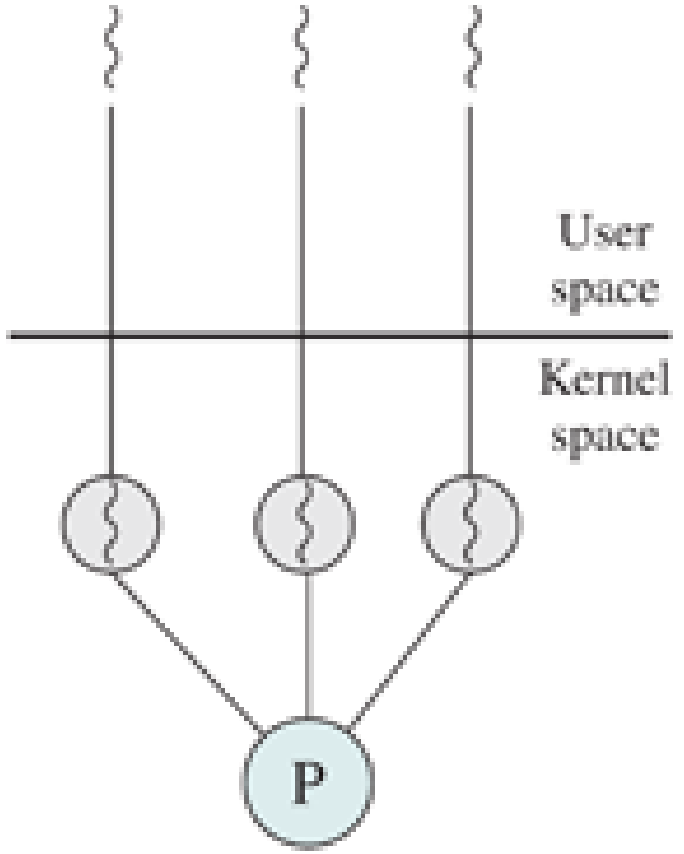


Processes vs Threads

- A process represents:
  - Address space that holds process image
  - Access to resources (I/O, file systems, etc.)
- A process possesses one or more threads, each with:
  - Thread execution state, saved context if not running
  - Execution stack
  - Per-thread static storage
  - Access to shared, process-owned memory and resources
  - Implicit IPC through shared text, data, and heap
  - Bad news – Data races

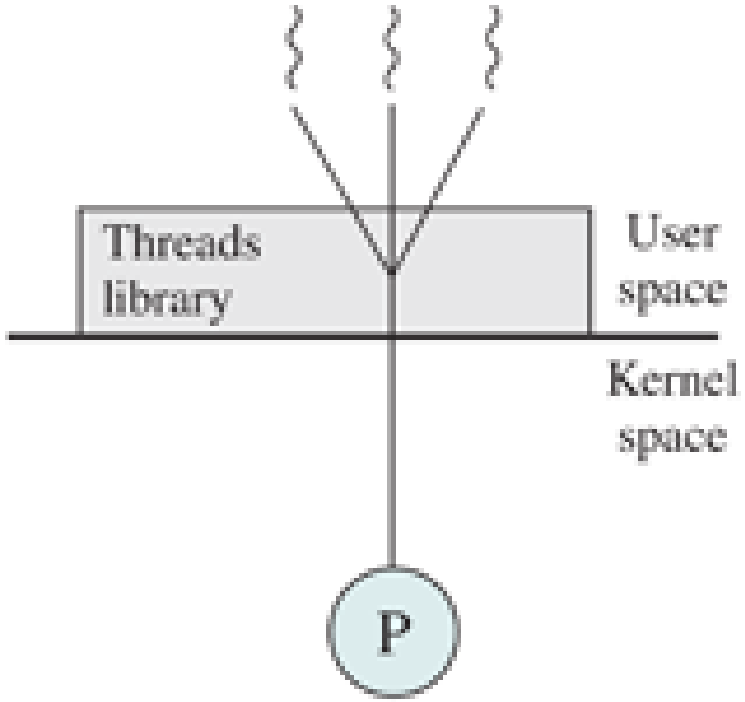
How to Implement Threads – Kernel Level

- Processes are an OS concept
- Does it make sense to implement threads in OS?
  - OS is already managing memory
  - OS is already managing scheduling
  - Blocking only stops active thread



How to Implement Threads – User Level

- Does it make sense to implement threads in user space?
  - User program may have more knowledge of how threads are used (scheduling)
  - Runs without OS awareness
  - Single process space to manage
  - Blocking stops all threads



pthreads – POSIX Threads

- Initially user level, but now are kernel level
- System calls like those for processes
  - pthread_create -> fork()
  - pthread_join -> wait()
  - pthread_exit -> exit()
- pthreads share process identifier, but have own thread identifier
- pthreads share everything parent and child processes share + text, data, and heap

Linux (not POSIX) Fine Grained Sharing

- What if we want to mix and match what is shared?
- Don’t want to share file descriptors, but want to share heap
- Linux system call -> clone()