CSE4509 Operating Systems

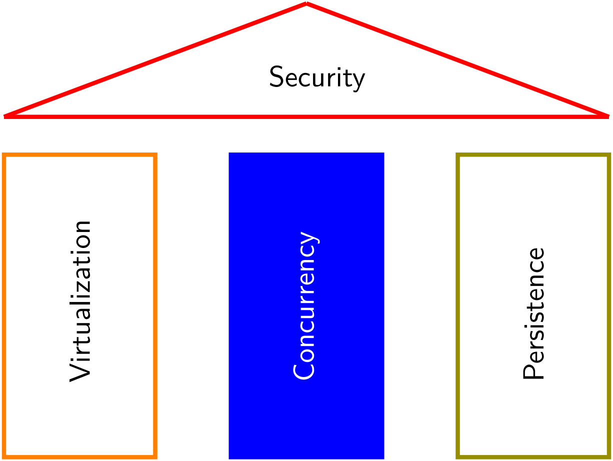
Thread

Salman Shamil

United International University (UIU)
Summer 2025

Original slides by Mathias Payer and Sanidhya Kashyap [EPFL]

Concurrency



Lecture Topics

- Thread abstraction
- Multi-threading challenges
- Key concurrency terms and definitions

This slide deck covers chapters 26 and 27 in OSTEP.

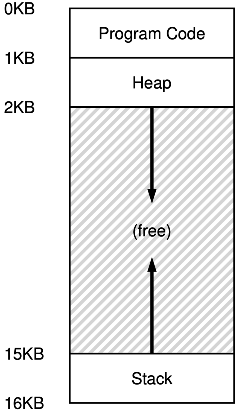
[**Credits**: Portions of the content are adapted from slides based on the OSTEP book by Prof. Youjip Won (Hanyang University) and Prof. Mythili Vutukuru (IIT Bombay), with thanks.]

Threads: Executions context

- Threads are independent execution context
  - similar to processes
  - EXCEPT they share the same address space

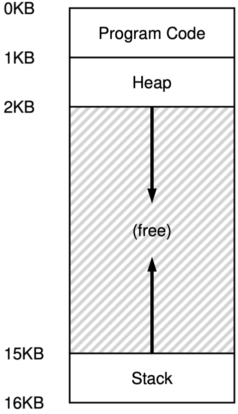
Threads: Executions context

- Threads are independent execution context
  - similar to processes
  - EXCEPT they share the same address space



Threads: Executions context

- Threads are independent execution context
  - similar to processes
  - EXCEPT they share the same address space



- We only had one thread in a process so far
  - single-threaded program
  - one Program Counter (PC)
  - one Stack Pointer (SP)

Multi-threaded Process

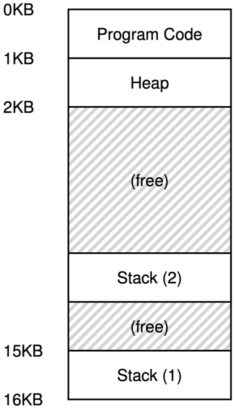
- **What happens if we want multiple threads in parallel?**
  - they must share the address space
  - they must maintain separate execution stream
  - is that possible with a shared stack or PC?

Multi-threaded Process

- **What happens if we want multiple threads in parallel?**
  - they must share the address space
  - they must maintain separate execution stream
  - is that possible with a shared stack or PC?
  - each thread has separate stack
    - leading to independent function calls
  - each thread has separate PC
    - able to execute different parts of the program
  - code and heap segments are still shared

Multi-threaded Process

- **What happens if we want multiple threads in parallel?**
  - they must share the address space
  - they must maintain separate execution stream
  - is that possible with a shared stack or PC?
  - each thread has separate stack
    - leading to independent function calls
  - each thread has separate PC
    - able to execute different parts of the program
  - code and heap segments are still shared



- **Communication between processes vs threads**
  - Processes need complicated Inter-Process Communication (IPC)
  - Extra memory footprint for IPC
  - Threads can do it by simply using global variables (shared)

To be continued