

Lecture 1, Jan 9, 2024

Overview of Operating Systems

- 3 core concepts:
 - *Virtualization*: sharing one resource by mimicking many independent copies
 - *Concurrency*: handle multiple things happening at the same time
 - *Persistence*: retain data consistency even without power
- A *process* is an instance of a running program – our first layer of abstraction
 - Each process needs its own virtual registers, stack, and heap
- How do we run two processes at the same time and still keep them independent?
 - Each process has its own *virtual memory* – its own independent view of the memory; this includes the stack and the heap
 - * The process thinks it has access to all the memory, and the OS maintains that illusion
 - * In reality one process cannot access the memory of another process for reasons of security
 - * The same memory address in virtual memory for two different processes is mapped to different locations in physical memory
 - For local variables, the OS allocates an independent stack in memory for each process
 - For global variables, the compiler just picks some address for each variable on compilation, and the OS ensures there are no conflicts