

Processes

- Process Concept
- Concurrency
- Race Conditions
- Process Creation
- Interprocess Communication
- Examples of IPC Systems

What is a Process?

- Fundamental building block of modern operating systems is the notion of a *process*
- A process is a running program (a program in execution). This includes:
 - All programs running on behalf of users (application programs)
 - Some operating system functions are also implemented using processes
- A process is a single thread of execution under control of the OS

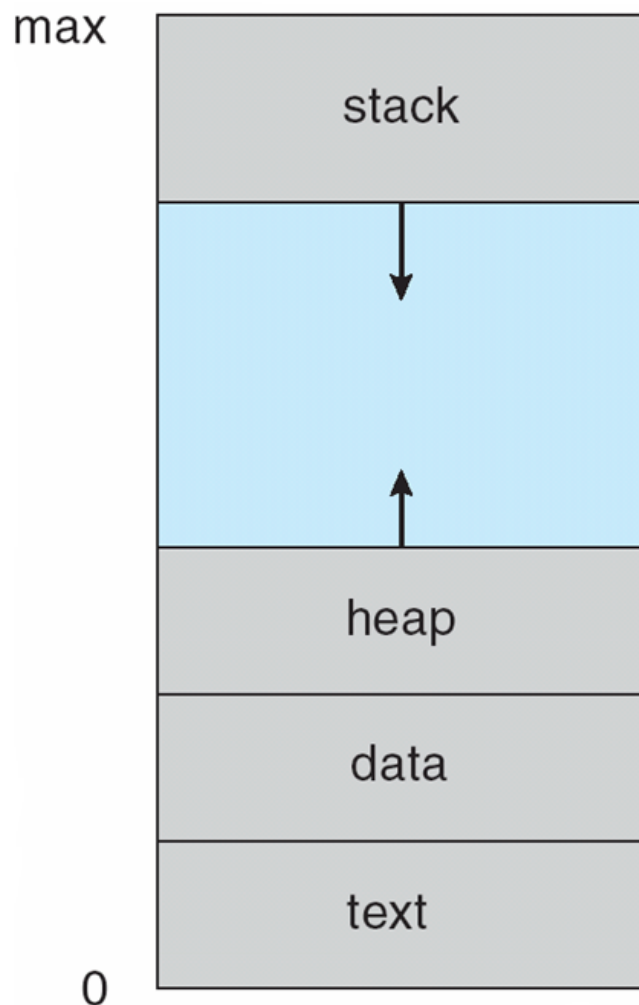
Process Details

- Much of the functionality of a modern OS is the work required to manage processes
- OS may have hundreds of processes active at the same time
 - Although only a small number of them executing at a given time on a multi-core CPU system
- Processes are not found in the operating system *kernel*

What is not a Process?

- A program by itself is not a process
- There is no one-to-one correspondence between programs and processes
 - E.g. there may be 10 people using emacs at the same time, i.e. 10 processes running emacs, but only one copy of the emacs program on disk
 - E.g. there may be many programs on disk that are not executing at the present time → these are not processes!
 - Programs are passive entities, while processes are active

A Process in Memory



- **Text**: the instructions that make up the program
- **Data**: the data the program uses
- **Heap**: used for dynamic memory
- **Stack**: used for function calls