

Operating Systems!

An Overview of Operating Systems (Part 1)

Prof. Travis Peters

Montana State University

CS 460 - Operating Systems

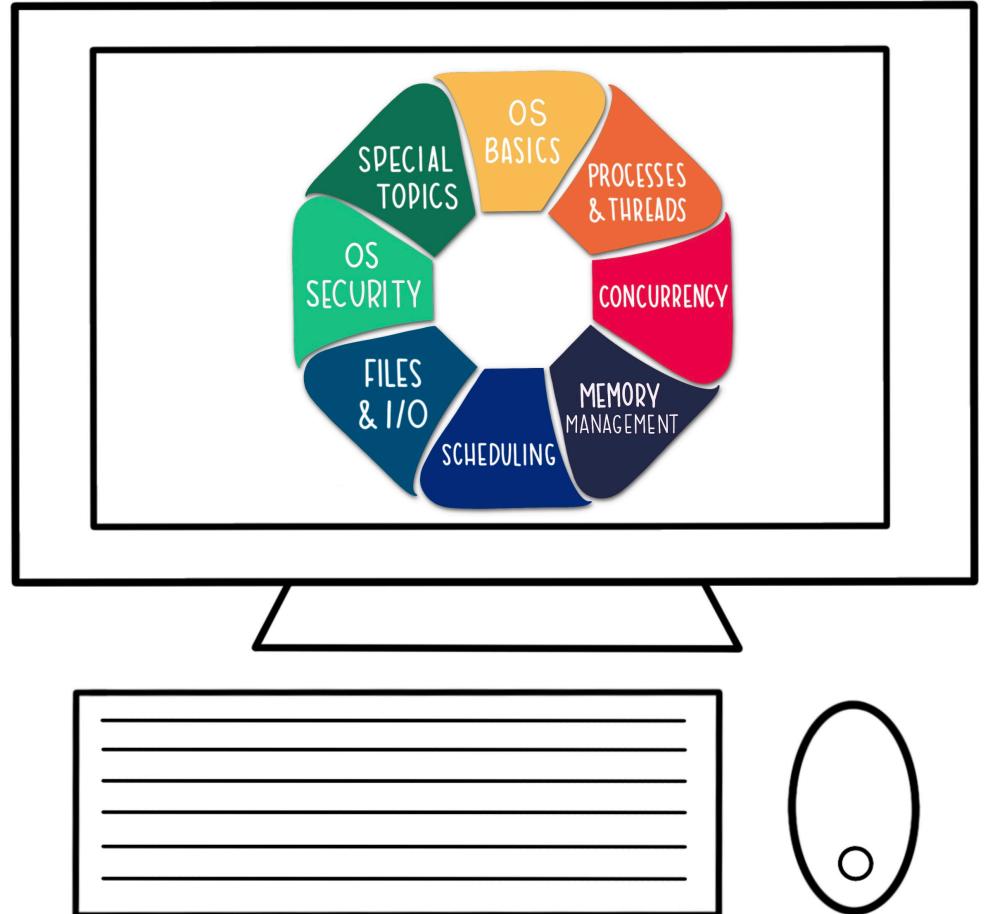
Fall 2020

<https://www.cs.montana.edu/cs460>

Today

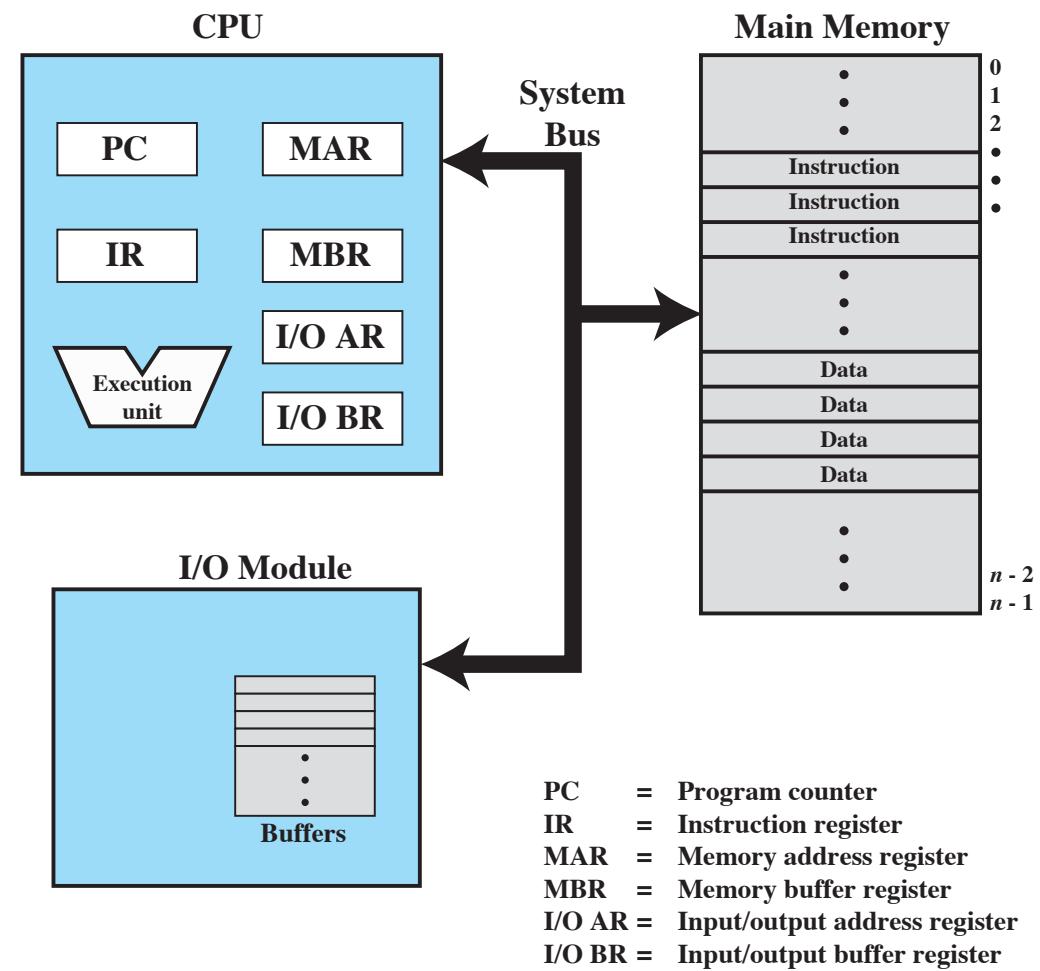
- Announcements
 - Nice job in Tech Bootcamp!
 - PA0 Due:
Sunday [09/06/2020] @ 11:59 PM (MST)
 - Schedule Updates!
More frequent assignment deadlines coming soon!
 - Vagrant/VM hooks
How I apply auto updates to VMs
- Learning Objectives
 - Understand the big ideas and constructs behind operating systems

some revised slides posted work through tutorials on gdb, valgrind, & git



Operating System (Overview)

- Objectives of an OS
 - Referee, Illusionist, Glue
vs. Convenience – Efficiency – Modular (textbook)
- Major Achievements
 - Process/Threads
 - Memory Management
 - Scheduling & Resource Management
 - Security



OS Objectives/Roles

- Referee
 - manage resources between apps
 - isolate apps and users from one another
 - facilitate communications between apps/users
- Illusionist
 - make apps believe they have the whole machine to themselves
 - create appearance of infinite processors and memory
 - abstract away complexity of storage, network communications, etc.
- Glue
 - manage hardware so apps can be machine-agnostic;
provide a set of common services



— Thanks to Adam Bates for the nice analogies:
<https://courses.engr illinois.edu/cs423/sp2018/>

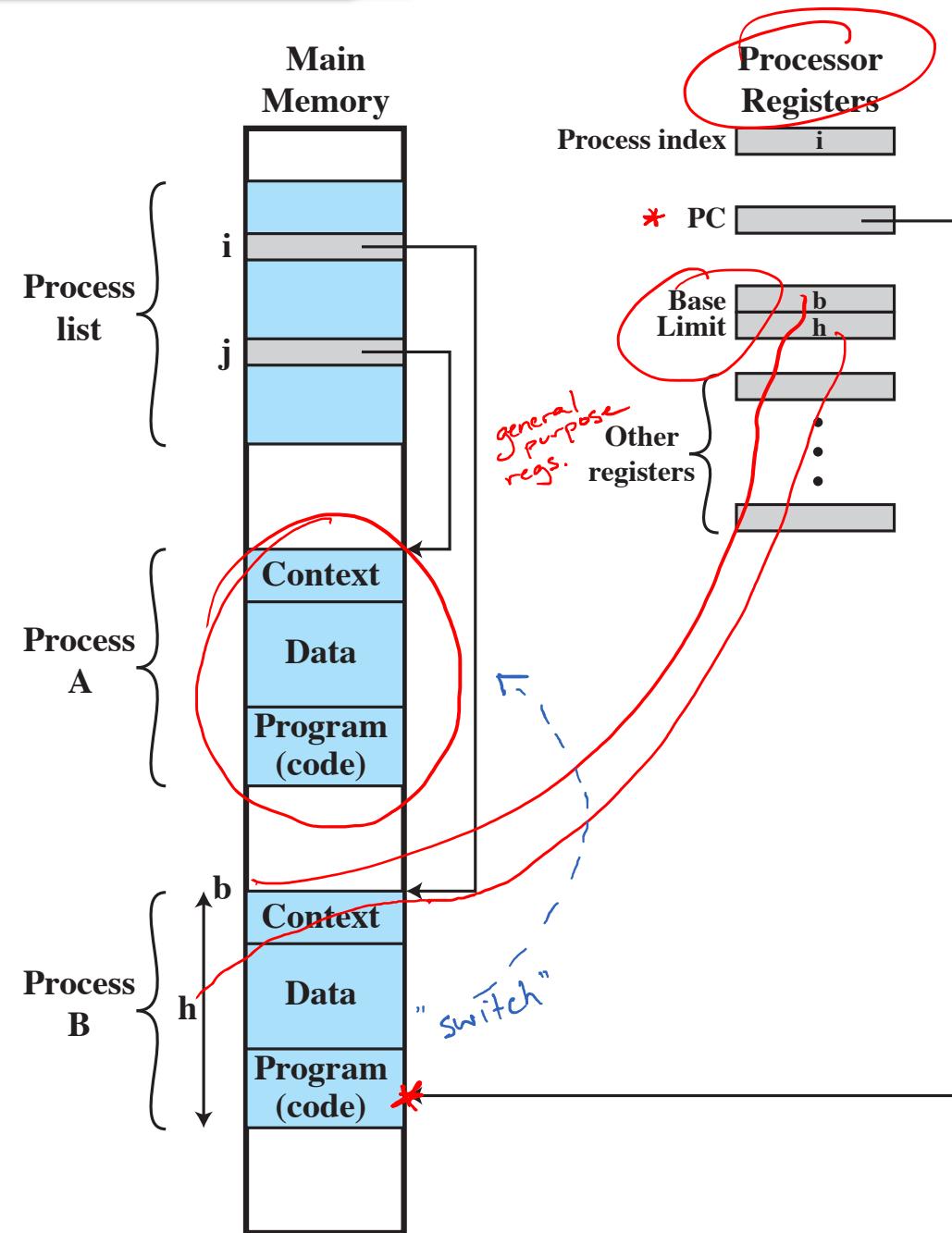
Processes & Threads

*What follows are the big ideas,
which gives us context for the role and purpose of the OS.*

The Idea of a “Process”

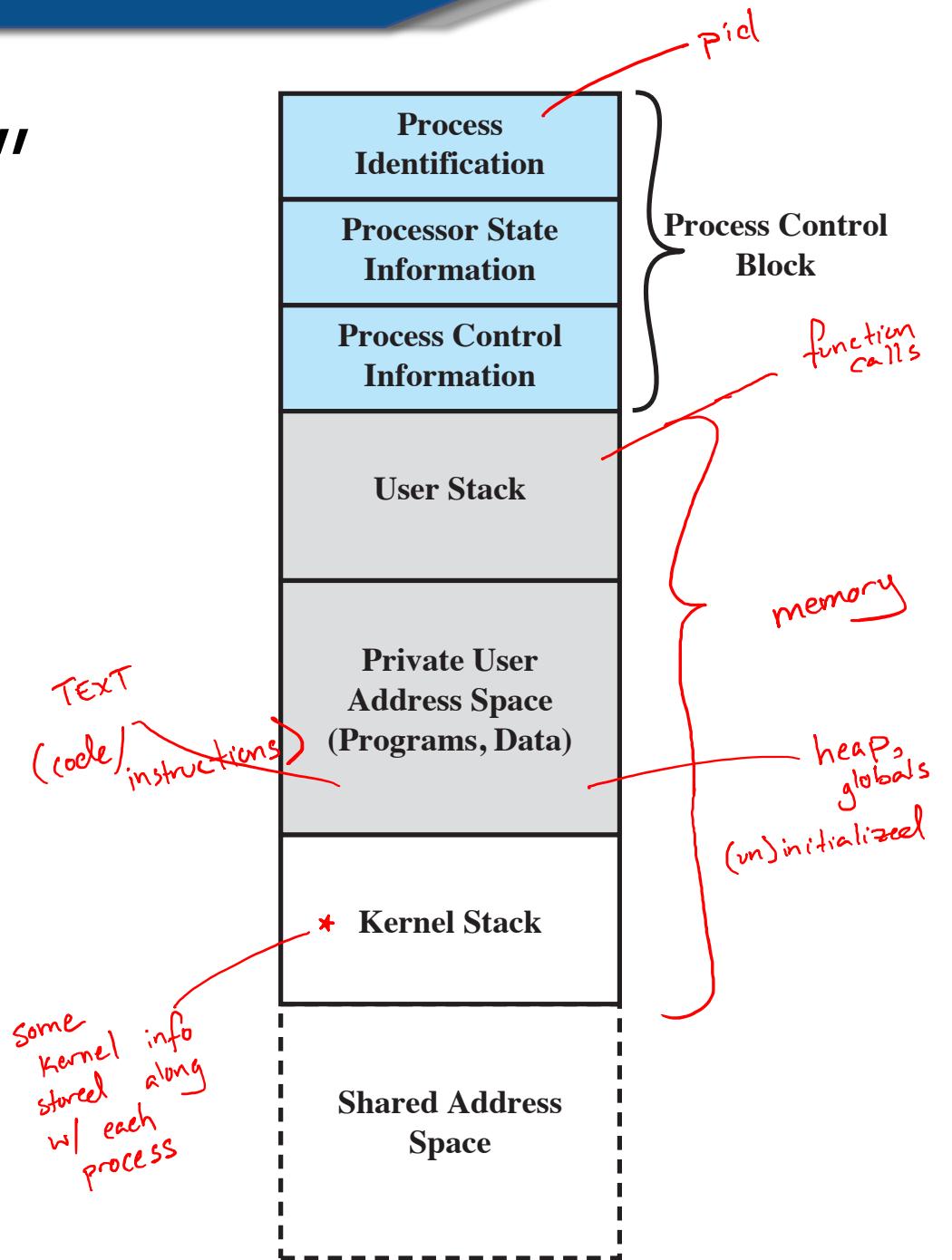
- Arose from multiprogramming, time sharing, real-time transactions
- More general than a “job”; many definitions; e.g., an instance of a program running on a computer;
- Consists of
 - (1) an executable program (code/instructions),
 - (2) associated data,
 - (3) execution context (info the OS needs to manage the process)

mostly hidden
from
sw developer



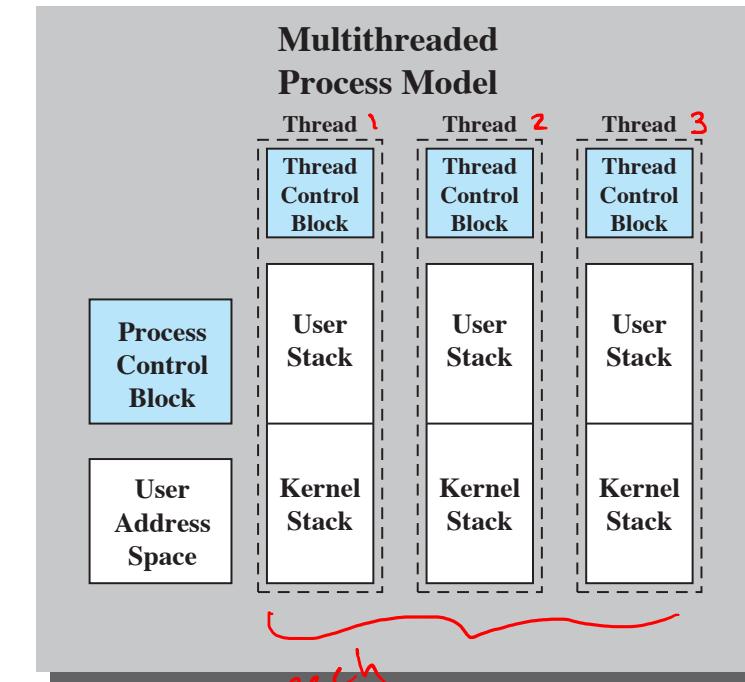
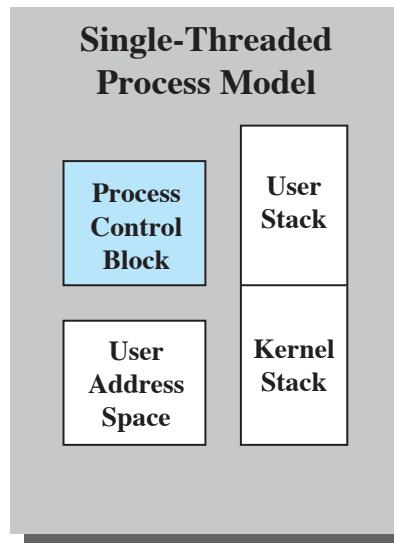
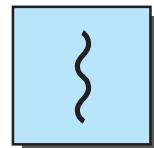
The Details of a “Process”

- Realized as nothing more than a data structure!



The Details of a “Process”

- Realized as nothing more than a data structure!
- A **thread** = the main unit of execution; enables cooperative execution within a process; use shared context.



each thread has its own execution context

The Details of a “Process”

- Realized as nothing more than a data structure!

- A *thread* = the main unit of execution; enables cooperative execution within a process; use shared context.

- Process Switching

- Interrupt = save context
(e.g., PC and other registers, memory)
 - execute interrupt handler
 - resume processing (same or different process)
- Process States = status of process
 - simple model = **executing** or **awaiting** execution

* kind of complicated...

