Today’s Plan

<table>
<thead>
<tr>
<th>Upcoming:</th>
<th>Today’s topics:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiz #2</td>
<td>From last time:</td>
</tr>
<tr>
<td>Assignment 1</td>
<td>Operating System Design and</td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
</tr>
<tr>
<td></td>
<td>Operating System Structure</td>
</tr>
<tr>
<td></td>
<td>Virtual Machines</td>
</tr>
<tr>
<td></td>
<td>Chapter 3</td>
</tr>
<tr>
<td></td>
<td>Introducing the Process Concept</td>
</tr>
<tr>
<td>Last time:</td>
<td></td>
</tr>
<tr>
<td>Quiz #1 recap</td>
<td></td>
</tr>
<tr>
<td>OS services</td>
<td></td>
</tr>
<tr>
<td>System calls</td>
<td></td>
</tr>
</tbody>
</table>
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
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
- 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
A Process in Memory

- **text:**
- **data:**
- **heap:**
- **stack:**
Modern OSes allow for more than one process to exist at the same time, and since there is usually only one processor, processes must assume different states during their lifetime:

- **Running:**

- **Blocked:**

- **Ready:**

- **Deadlocked:**

  - OS must recognize this and deal with it
Process State Diagram

READY

RUNNING

BLOCKED

DEADLOCKED
## Process Control Block (PCB)

Information associated with each process

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information
Process Control Block (PCB)

- process state
- process number
- program counter
- registers
- memory limits
- list of open files
## Process Model of an OS

- Modern OSes are a collection of cooperating processes that run on top of (and are supported by) an OS kernel.

- The kernel is responsible for the following services:
  - Creation and destruction of processes
  - CPU scheduling, memory management, device management
  - Process synchronization tools
  - Process communication tools

- OS services provided by the kernel are invoked using *system calls*.