Lecture 2: Processes

Outline of activities

1. Bare machine components
2. Characteristics of applications written for bare machines
   - Extremely difficult to implement
   - Not portable
   - Many repeated routines in different applications
   - Poor CPU utilization
3. Need for operating system (a layer between applications and hardware)
   - Common routines and utilities to provide uniform interface and abstract representation of resources (read, write)
   - Management of these resources
   - Multiprogramming to improve CPU utilization
4. OS support for multiprogramming
   - Data structures to allow sharing of processor(s)
   - Algorithms to utilize resources (I/O, processor cores, etc.)
   - Mechanisms for data and I/O protection
   - Mechanisms for communication among programs in execution
5. What is a *process*?
   (a) A program in execution
   (b) An instance of a program running on a computer
   (c) The entity that can be assigned to and executed on a processor
   (d) A unit of activity characterized by the execution of a sequence of instructions, a current state, and an associated set of system resources
6. A program in execution
   - Address space
     (a) Code
     (b) Data
     (c) Stack
     (d) Heap
     (e) I/O devices
   - Registers

7. Managing processes

   - Aviation example
     - Planes
     - Air traffic controllers
     - Abstractions of planes
   - Components
     (a) Core image
     (b) Process Control Block (PCB)