# HW2

* It is the ability of a single cpu to run multiple programs at a time. It is needed to speed up computers by giving them the ability to do many things at once. Also prevents as much of a need for programs waiting on others to store their steps and relinquish control to other programs (which is a big slowdown factor).
* A process is like the parent of threads, it creates threads for various tasks it needs to complete itself.
  + Order goes:
  + Program => process => threads
  + Also threads are cheaper to create and efficiently communicate with one another.
* User Level
  + Lighter weight
  + seperate custom scheduling
* Kernel Level
  + These threads are slower
  + Less expensive (if in same process)
  + known by the os
* New
* This is were a process is created and leads out to the ready state once the process is ready to begin.
* Ready
* The process gets the resources it needs but the cpu is not actively running it so progress is not being made. This step only leads to the running state.
* Running
* This state has transitions to blocked, ready, and terminated states.  In this state, the cpu begins executing the bits of the proccess that need to happen and sends to so many other states for the various exits or waiting, or completions that this state may encounter.
* Blocked
* This state only moves a process back to the ready state and is here in the case a process needs info or is waiting for another process so that it can complete any of its steps.
* Terminated/exit
* This does not connect off to others and is where the process finishes and wraps up.
* 30/50 + 20/150 + 10/300 + x/350 <= 1
* 180/300 + 40/300 + 10/300 + x/350 <= 1
* 230/300 + x/350 <= 1
* .766666 + x/350 <= 1
* x =< .233333333 \* 350
* x =<81.66666