Virtual Memory

* Describe the "problem" that virtual memory is designed to solve?
  + What if program memory is greater than main memory?
* When considering the relative advantages and disadvantages of (programmer) overlays and virtual memory, describe one advantage of overlays? Describe one advantage of virtual memory?
  + Overlays are program controlled.
    - This means something
  + Virtual Memory is OS controlled
    - With help from the architecture
    - Allows physical memory to be *shared* by multiple processes
      * Processes are *isolated* from each other.
    - Impossible for a process to access memory outside of its own access space
* Given a CPU clock speed, the number of cycles needed for a successful access of main memory, the amount of time needed to access a disk (when a miss occurs in main memory), and the page fault rate, compute the average number of cycles required to access main memory.
  + 200 ns clock rate
  + Access of 5 cycles -> 1000 ns
  + 25 millisecond page service time
  + MAT = (1 – p) \* 1000 + p \* 25,000,000
  + E.g. assume page fault rate of .001
  + MAT = .999 \* 1000 + .001 \* 25,000,000
  + MAT = 999 + 25,000 -> 26x penalty
* We discussed 3 scheduling policies recommended for choosing a page to remove from memory when bringing a new page in. Those are FIFO, Optimal, and LRU. Describe how each of those policies work. Give relative advantages and disadvantages of each.
  + FIFO – Literally replaces the first in by shifting all others down and moving the next fault to the last space. See the table below.
  + Optimal – Actually looks ahead (which is why it’s not implementable) and determines how the table should be filled *before* it gets to a fault.
  + LRU – Just checks to see which one has been used less frequently and replaces it.
* Given a fixed number of pages, and a trace of which pages are accessed when during "program execution", show, for each of the replacement policies listed below, both 1) how the contents of the fixed number of pages would change during the time indicated by the page trace, and 2) the number of page faults that would occur in the given page trace.
  + 70120304230321201701 – 4 Pages
  + FIFO
    - **7012**0**3**0**4**23**0**32**12**01**7**01
    - 7 0 1 2 3 4 0
    - 0 1 2 3 4 0 1
    - 1 2 3 4 0 1 2
    - 2 3 4 0 1 2 7
  + Optimal
    - **7012**0**3**0**4**23032**1**01**7**01
    - 1 3 3 3 0
    - 0 0 0 0 2
    - 1 1 4 4 7
    - 2 2 2 1 1
  + LRU
    - **7012**0**3**0**4**23032**1**01**7**01
    - 7 0 0 0 0
    - 0 1 2 2 2
    - 1 2 3 3 7
    - 2 3 4 1 1