Anisha Aggarwal
Assignment 4
CIS 415

1. OSC 10.11 (Currently at 2150; Previously 1805; Ranges from 0-4999;
   FIFO: 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681)
   a. FCFS
      Order: 2150, 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681
      Distance: (2150-2069) + (2069-1212) + (2296-1212) + (2800-2296) + (2800-544) + (1618-544) + (1618-356) + (1523-356) + (4965-1523) + (4965-3681) = **13011 cylinders**

   b. SSTF
      Order: 2150, 2069, 2296, 2800, 3681, 4965, 1618, 1523, 1212, 544, 356
      Distance: (2150-2069) + (2296-2069) + (2800-2296) + (3681-2800) + (4965-3681) + (4965-1618) + (1618-1523) + (1523-1212) + (1212-544) + (544-356) = **7586 cylinders**

   c. SCAN
      Order: 2150, 2296, 2800, 3681, 4965, 4999, 2069, 1618, 1523, 1212, 544, 356, 0
      Distance: (2296-2150) + (2800-2296) + (3681-2800) + (4965-3681) + (4999-4965) + (4999-2069) + (2069-1618) + (1618-1523) + (1523-1212) + (1212-544) + (544-356) = **7492 cylinders**

   d. LOOK
      Order: 2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356
      Distance: (2296-2150) + (2800-2296) + (3681-2800) + (4965-3681) + (4965-2069) + (2069-1618) + (1618-1523) + (1523-1212) + (1212-544) + (544-356) = **7424 cylinders**

   e. C-SCAN
      Order: 2150, 2296, 2800, 3681, 4965, 4999, 0, 356, 544, 1212, 1523, 1618, 2069
      Distance: (2296-2150) + (2800-2296) + (3681-2800) + (4965-3681) + (4999-4965) + (356-0) + (544-356) + (1212-544) + (1523-1212) + (1618-1523) + (2069-1618) = **4918 cylinders**

   f. C-LOOK
      Order: 2150, 2296, 2800, 3681, 4965, 356, 544, 1212, 1523, 1618, 2069
      Distance: (2296-2150) + (2800-2296) + (3681-2800) + (4965-3681) + (544-356) + (1212-544) + (1523-1212) + (1618-1523) + (2069-1618) = **4528 cylinders**

2. Consider a file system that uses inodes to represent files.
    a. Max size of disk (in bytes) for which one can use this file system?
       The inode layout does not impact the max size of the disk; it only impacts the max size of the file.

    b. What is the max size of file (in bytes) that can be stored in this file system?
       8 direct block pointers -> 8*1024 = 8kB
       1 single indirect -> (1024/4)*1024 = 256kb
       1 double indirect -> (1024/4)*(1024/4)*1024) = 2^16kB = 2^6MB = 64MB
       1 triple indirect -> (1024/4)*(1024/4)* (1024/4)*1024 = 2^24kB = 16GB
       total: 8*1024 + (1024/4)*1024 + (1024/4)*(1024/4)*1024 + (1024/4)*(1024/4)* (1024/4)*1024 = **17247248384 bytes**

3. The processor for which you are designing your application as L1i and L1d virtual caches
    a. Type of data does cache hold?
       L1i -> holds instruction cache
       L1d -> holds data cache

    b. Describe in detail the activities of the cache + memory system when executing the instruction
       To get the contents of "virtual address", CPU will check the cache first. If not in cache, the OS will need to the corresponding page table to find out the physical frame needed. If the page table entry is not in cache, CPU will access memory to read the page table entry to find the physical frame number. If the physical frame is not in memory, OS will page-fault to read the memory frame, and then fetch the contents of "virtual address" – this data will also be cached for future references.

    c. Assume that the above instruction is executed many times in a loop, and that the instruction itself is in the cache. Also assume that memory access costs τ μs, and cache access costs $\tau/15$ μs. What cache hit rate ρ for "virtual address" is required for the memory system to run 5 times faster than with no caching at all? Show your work.
       $(\rho * \tau/15) + ((1 - \rho)* \tau) = \tau/5$
       $5(\rho/15) + 5(1 - \rho) = 1$
       $\rho/3 + 5 - 5\rho = 1$
       $\rho + 15 - 15\rho = 3$
       $12 = 14\rho$
       $\rho = .857 =$ **85.7%**

    d. Suppose we have a memory system that has a main memory, a single-level cache, and demand paging virtual memory. The three levels of the memory system have the following access times
        i. effective memory access time: (0.95*2) + (0.05*100)= **6.9ns**

ii. new effective memory access time: (0.95*2) + (.05*.00001*(100 + 10x10^6) + (.05*0.9999*100) = **11.9 ns**