Author: 7121

Assignment: 8

Questions:

1. Given a 4 KB page size, what are the page numbers and offsets for the following addresses Page (329)
   1. 2375
      1. 2375/4096 = 0.5798 or 0 page number
      2. 2375 mod 4096 = 2375 page offset
      3. Verify = 4096 \* 0 + 2375 = 2375
   2. 19366
      1. 19366/4096 = 4.728 or 4 page number
      2. 19366 mod 4096 = 2982 offset
      3. Verify = 4096 \* 4 + 2982 = 19366
   3. 30000
      1. 30000/4096 = 7.3242 or 7 page number
      2. 30000 mod 4096 = 1328 page offset
      3. Verify = 4096 \* 7 + 1328 = 30000
   4. 256
      1. 256/4096 = 0.0625 or 0 page number
      2. 256 mod 4096 = 256 page offset
      3. Verify = 4096 \* 0 + 256 = 256
   5. 16385
      1. 16385/4096 = 4 page number
      2. 16385 mod 4096 = 1 page offset
      3. Verify = 4096 \* 4 + 1 = 16385
2. Given an architecture that supports Translation Look-aside Buffer (TLB), what is the effective memory access time if 1) TLB access time is 10 nanoseconds, 2) memory access time is 200 nanoseconds, and 3) the TLB hit ratio is 95%. Assume the architecture supports one-level paging. Page (334)
   1. Effective memory access time = (hit ratio \* total access time) + (1 – hit ratio) \* (memory access time \* 2 + access time)
   2. Effective memory access time = (0.95 \* (10 + 200)) + (1 – 0.95) \* (10 + 200 + 200)
   3. Effective memory access time = 220 nanoseconds
3. The x86-64 architecture only supports 48-bit addressable virtual memory space.
   1. Suppose a 4 MB (222) page size used in this 48-bit (248) virtual memory space. How many pages exist in the virtual memory space? Page (339)
      1. 248 / 222 = 226 or 67,108,864 pages
   2. Why does not this 64-bit architecture support 64-bit virtual memory space? Page (383)
      1. Because it would require the page tables to be far larger than the limit of 64-bit itself. The reason why it supports 48-bit, would be to use the extra space to store flags. Also, whereas the minimum number of frames per process is defined by the architecture, the maximum number is defined by the amount of physical memory, so 264 addressable units which can access 16 Exabyte of memory, which no hard drive has at the moment.
4. Given the following two programs, suppose they run on an architecture with a 4KB page size and 12 KB physical memory space. Please find how many page faults will occur for both programs. Hint: each integer is 4 byte long. (Page 403)
   1. In the first array of arr[ i ][ j ], the first run will be from arr[ i ][ j – j ] to to arr[ i ][ j – 1], which will not cause a page fault, but once it increments the first for loop, and arr[ i ] becomes arr [ i + 1], a page fault occurs. This means that the first for loop dictates how many page faults will be, which is: 1024 page faults.
   2. In the second array of arr[ j ][ i ], the first run will change the first variable in the beginning of each array from arr[ j ] to arr[ j – 1 ]. This means that every time it switches from each array[ j ] memory location, it creates a page fault, which is: 1024 \* 1024 = 1,048,576 page faults.
5. Given the following utilization of CPUs and paging disk, which one may indicate thrashing? Page (386)
   1. CPU (10%), paging disk (95%). This is because as the CPU scheduler sees that the CPU Utilization decreases, it increases the level of multiprogramming. This can take frames from a running process which can cause more page faults and longer queues for the paging device. Therefore the system throughput plunges.
6. Please implement the stack-based Least Recently Used algorithm. You are given two test programs: 1) PageReplacementShortTest and 2) PageReplacementLongTest. The long test should complete within less than 100 milliseconds. If your long test does not complete within half second, there must be something wrong with your programs.
   1. PLEASE SEE CODE INCLUDED IN THE PACKAGE