National University of Computer & Emerging Sciences, Karachi
FAST School of Computing
Spring 2022

Course Code: CS2006	Course Name: Operating Systems
Instructor Name: Anaum Hamid	Issue Date: 16th May 2022
Assignment no. 03	Due Date: 20th May 2022

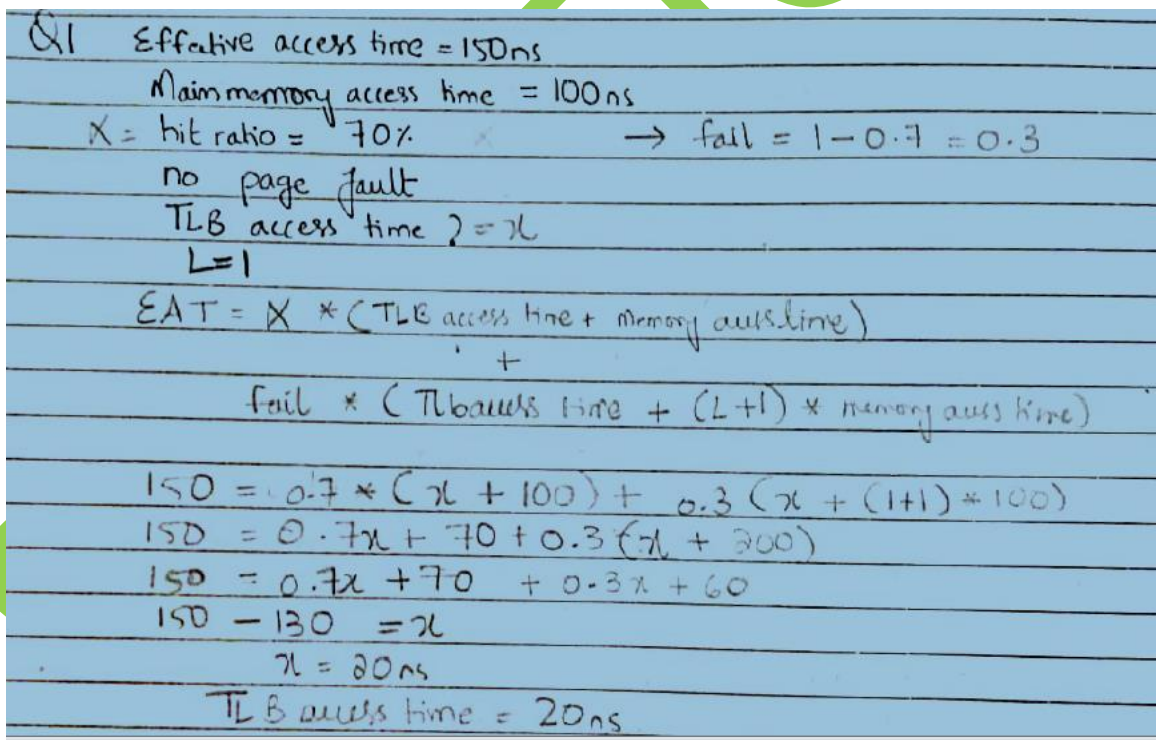
Instructions:

- Assignment should be solved handwritten on paper and their scanned soft copies are to be submitted on GCR.
- Must write your NU id on the top of each page of the assignment.
- Rough work must be solved for the clarification of the answer.
- Prints of Soft copy documented assignment cost direct zero.

All questions have equal marks (5 Each) - Total Marks-40 Marks

Memory Management

1. A paging scheme uses a Translation Lookaside Buffer (TLB). The effective memory access takes 150 ns, and the main memory access takes 100 ns. What is the **TLB access time** (in ns) if the TLB hit ratio is 70% and there is no page fault?



2. Consider a **two-level paging scheme** with a TLB. Assume no page fault occurs. It takes 25 ns to search the TLB and 100 ns to access the physical memory. If TLB hit ratio is 75%, What will be its **EAT**??

- Number of levels of page table = 2
- TLB access time = 25 ns
- Main memory access time = 100 ns
- TLB Hit ratio = 80% = 0.75

Calculating TLB Miss Ratio-
TLB Miss ratio = 1 – TLB Hit ratio
= 1 – 0.8
= 0.25
Calculating Effective Access Time-

Effective Access Time = 0.75 x { 25 ns + 100 ns } + 0.25 x { 25 ns + (2+1) x 100 ns }
= 0.75 x 125 ns + 0.25 + 325 ns
= 93.75 + 81.25
= 175 ns

3. Consider a **three-level paging scheme** with a TLB. Assume no page fault occurs. It takes 30 ns to search the TLB and 100 ns to access the physical memory. If TLB hit ratio is 70%, What will be its **EAT**??

- Number of levels of page table = 3
- TLB access time = 30 ns
- Main memory access time = 100 ns
- TLB Hit ratio = 70% = 0.7

Calculating TLB Miss Ratio-
TLB Miss ratio = 1 – TLB Hit ratio = 1 – 0.7 = 0.3

Calculating Effective Access Time-

Effective Access Time = 0.7 x { 30 ns + 100 ns } + 0.3 x { 30 ns + (3+1) x 100 ns }

= 0.7 x 130 ns + 0.3 + 430 ns = 91 + 129 = 220 ns

Thus, effective memory access time = 220 ns.

4. Consider a demand-paging system with a paging disk that has average access and transfer time of 25 milliseconds. Addresses are translated through a page table in the main memory, with an access time of 1 microsecond per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time, we have added an associative memory that reduces access time to one memory reference if

the page-table entry is in the associative memory. Assume that 80 percent of the accesses are in the associative memory and that, of those remaining, 10 percent (or 2 percent of the total) cause page faults. What is the effective memory access time in milliseconds?

Access Time = (0.8) * (1 microsecond) + (0.18) * (2 microsecond) + (0.02) * (25000 microsecond)
= 0.8 + 0.36 + 500
= 501.16 microsecond
= 0.5 millisecond

Paging

1. Calculate the size of memory if its address consists of 22 bits and the memory is 2-byte addressable.
   Number of locations possible with 22 bits = 222 locations
   It is given that the size of one location = 2 bytes
   Thus, Size of memory
   = 222 x 2 bytes
   = 223 bytes
   = 8 MB

2. Calculate the number of bits required in the address for a memory having a size of 16 GB. Assume the memory is 4-byte addressable.

   Size of memory = 2^n x 4 bytes.
   Since, the given memory has size of 16 GB, so we have-
   2^n x 4 bytes = 16 GB
   2^n x 4 = 16 G
   2^n x 22 = 234
   2^n = 232
   ∴ n = 32 bits

3. Consider a system with byte-addressable memory, 32-bit logical addresses, 4-kilobyte page size, and page table entries of 4 bytes each. Calculate the page table size in the system.

   Number of bits in logical address = 32 bits
   Page size = 4KB
   Page table entry size = 4 bytes
   Process Size- Number of bits in logical address = 32 bits
   Process size = 2^{32} B = 4 GB

Number of Entries in Page Table

Number of pages the process is divided = Process size / Page size
= 4 GB / 4 KB = 220 pages

Number of entries in page table = 220 entries

Page Table Size

Page table size = Number of entries in page table x Page table entry size
= 220 x 4 bytes = 4 MB

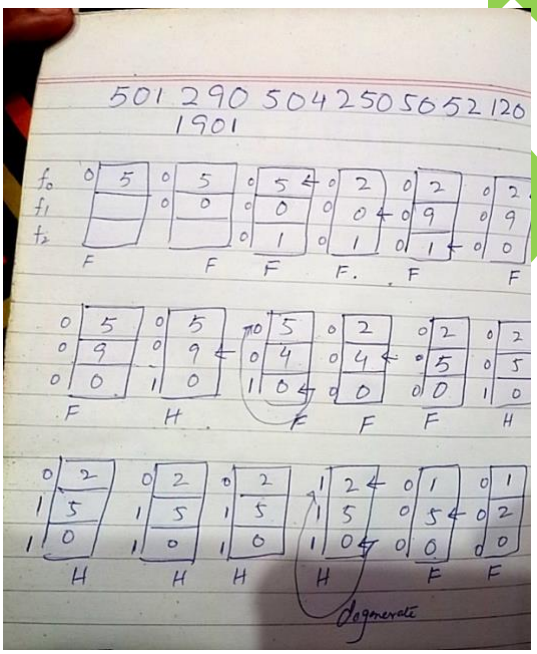
Page Replacement Algorithm

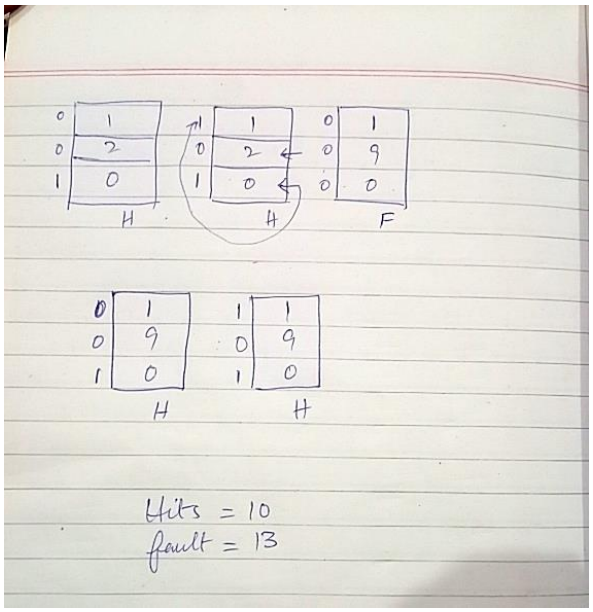
Consider the following page reference string:

5, 0, 1, 2, 9, 0, 5, 0, 4, 2, 5, 0, 5, 0, 5, 2, 1, 2, 0, 1, 9, 0, 1

Assuming demand paging with three frames and all frames are initially empty. Hence, allocation of the first unique pages will cost one fault each. Calculate fault ratio by

Second Chance Algorithm?





Fault ratio = 13/23 = 0.56 = 56%

**********End**********