L8-Memory System II Paging

1. In an OS that uses paging to manage its virtual memory, which part of the virtual address is used to index the page?
A) frame number
B) page offset
C) page number
D) frame offset
ANS: C) page number

2. If the size of virtual address is m bits, the page size is $2^n$ Bytes. How many bits are in the virtual address for the page number and the page offset, respectively?
A) n, m
B) m, n
C) m-n, n
D) m-n, m
ANS: C) m-n, n

3. Which of the following statements are TRUE for paging?
A) TLB reduces the number of memory accesses for paging.
B) Paging may incur external fragmentation.
C) Multi-level paging reduces the size of page table.
ANS: A) and C)

4. Assume that we run a 64-bit OS on a device with limited physical memory, with a small page size of 2KB.
1. How many bits are needed in the virtual address to represent the page offset?
ANS: $2KB = 2^{11}$, so 11 bits are needed.

2. Given that the size of a page table entry is 4 bytes, How many page entries can be stored in a page?
ANS: Each page is 2KB, and each page entry is 4 bytes. $2^{11}/2^2 = 512$ or $2^9$.

3. We use only 38 bits out of 64 bits for paging (i.e., a 38-bit virtual address). How many pages can be indexed by a 38-bit address space?
ANS: $2^{38}/2^{11} = 2^{27}$

4. 3-level paging is implemented in this system to reduce the page table size. The three page directories have the same size. How many bits are used for each level of the page directory?
ANS: 27 bits for page index, and each page can store $2^9$ page table entries. Additionally, three levels have the same size, hence 9 bits are used for each level of the page directory.

5. Which of the following statement is true about page replacement policy?
A) A multi-level page table is used to reduce performance overhead.
B) LRU always performs better than FIFO in terms of the number of page faults for all workloads.
C) For a given page access sequence, increasing the number of page frames always reduces page faults for all replacement policies.
D) Upon the first access of a page, two memory references are made.
ANS: D) Upon the first access of a page, two memory references are made.: The page table is stored in memory, so the first time access always needs to access memory to find the page table in memory, followed by another memory access to locate the page in memory.

6. Which of the following statements are true?
A) If the dirty bit of a page is set, it means the page has been modified and is different from the one on disk.
B) Demand paging loads required pages in advance.

C) Each process has its own page table.
D) Even using swapping, the virtual address size must be smaller than the physical address size.
ANS: A), C). Demand paging loads pages only when they are about to be executed, i.e., only when the pages are actually requested. Each process has its own page table. Virtual memory can be larger than the physical memory by using swapping.

7. Assume
Virtual address space size: 64 KB
Physical memory size: 64 KB
Page size: 256 bytes
7.1 How many bits for the VPN and the page offset in the virtual address and physical address?
ANS: Page size is 256 bytes = $2^8$, hence offset is 8 bits. Virtual address space size is 64 KB = $2^{16}$, hence virtual address length is 16 bits. VPN length is 16-8=6 bits.
Physical address space size is 64 KB = $2^{16}$, hence physical address length is 16 bits. PPN length is 16-8=6 bits.

7.2 Given the following page table, what is the corresponding physical address for the virtual address 0x0bbc?

| Page number VPN (Decimal) | Frame number PPN (hexadecimal) |
|---|---|
| 0 | 0x51 |
| 1 | 0x6f |
| 2 | 0xa5 |
| 3 | 0x3c |
| 4 | 0xea |
| 5 | 0xc9 |
| 6 | 0x10 |
| 7 | 0x27 |
| 8 | 0xb4 |
| 9 | 0x64 |
| 10 | 0x1f |
| 11 | 0x9a |

ANS: 0x9abc. Since the least significant 8 bits are used for the page offset, other bits are used for the page index (VPN). The virtual address 0x0bbc is split into two parts, where 'bc' is the page offset and '0b' is used to index the physical frame. '0b' is 00001011 in binary and 11 in decimal, and the corresponding frame is 0x9a as seen from the page table. Finally, adding the offset to it results in the correct solution 0x9abc.

7.3 Assuming that the page table entries above have been loaded into the TLB. How many TLB misses will occur when the process accesses the following 14 virtual addresses?

| |
|---|
| 0x04db |
| 0x07ad |

| |
|---|
| 0x03ef |
| 0x06b8 |
| 0x0Ad7 |
| 0x0985 |
| <span style="color:red">0x0D17</span> |
| 0x0B62 |
| 0x0264 |
| 0x007d |
| 0x05fb |
| 0x0876 |
| 0x012f |
| <span style="color:red">0x0C15</span> |

ANS: 2 TLB misses. As the question states, the page table has already been loaded into the TLB, so we just need to identify the pages that are not in the TLB or will be accessed for the first time, i.e., the addresses not provided in the page table. We can use the same method as in the previous question to determine which virtual addresses will be newly accessed. These addresses are 0x0D17 and 0x0C15, because VPNs C and D in hex are equivalent to 12 and 13 in decimal, respectively, and they are not in the TLB.

7.4 If TLB lookup time = 1 cycle, Memory access time = 100 cycles
Assuming there is no cache. What is Effective Access Time with TLB for the above address sequences?
Reminder:
$EAT$ = Hit Time * Hit Rate + Miss Time * Miss Rate
$= (m + \sigma)\, \eta + (2m + \sigma)(1 - \eta) = 2m + \sigma - m\, \eta$

ANS: $EAT = 2m + \sigma - m\, \eta$ = 2*100 + 1 – 100 * (12/14) = 129.6 cycles
Since the hit rate is 12/14.

8. Multi-level cache. Consider a two-level cache L1 and L2 with the following timing parameters:
L1 Hit Time: 1 cycle, L1 Miss Rate: 10%
L2 Hit Time: 10 cycles, L2 Miss Rate: 1%
Main Memory access time: 100 cycles.
a) What is the global miss rate?
b) Calculate the Average Memory Access Time (AMAT).

ANS:
a) Global miss rate = L2 Local Miss Rate × L1 Local Miss Rate = 0.1 * 0.01 = 0.001
b) L1+L2 caches: AMAT = L1 Hit Time + L1 Local Miss rate × (L2 Hit Time + L2 Local Miss rate × L2 Miss penalty)

AMAT = 1 + .1*(10 + .01*100) = 2.1 cycles