Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 07

[MCQ]

1. The page table register should be mode of _____?

(a) Very high-speed logic
(b) Secondary memory.
(c) Large memory space.
(d) Very low speed logic.

[MCQ]

2. Consider a system implementing paging hardware with a TLB. Assume the entire page table and all the pages are in the physical memory. The effective memory access time is 400 msec. with TLB hit ratio 0.6 and search time of TLB is 20 msec, the main memory access time is _____(in msec.)

(a) 275 msec. (b) 271.43 msec.
(c) 120.5 msec. (d) 240 msec.

[MCQ]

3. Consider a 2 level paging system with TLB support. The page table has divided into 4 K pages each of size 8K words. If the physical address space has 64 M words which divided into 8 K frames. TLB access time is 20 nsec. and main memory access time is 300nsec. The CPU finds 126 page reference in the TLB out of total reference of 180. Then what is the effective memory access time?

(a) 545 nsec. (b) 440 nsec.
(c) 420 nsec. (d) 500 nsec.

[NAT]

4. Consider on operating system having 32 bits virtual address space and 32 MB physical memory. If page size is 8KB. What is the approximate size of the page table in MB?

[NAT]

5. Consider a system using paging hardware with a TLB to reduce EMAT. Assume that the entire page table and all the pages are in the physical memory. It take 30 millisec to search the TLB and 110 millisec to access the physical memory. If the TLB hit ratio is 0.6, the EMAT (in millisec) is_____.

[NAT]

6. A demand paging system takes 240 time units to service a page fault and 300 time units to replace a dirty page. Main memory access time is 20 time units. The probability of page fault is 0.4. In case of a page fault the probability of page being dirty is 0.06. The average access time _____time units.

[MCQ]

7. Consider a 32 bit machine where four-level paging scheme is used. If the TLB hit ratio is 90% and it takes 30 nsec to search the TLB and main memory access time is 100 nsec so, what is the effective memory access time in nanoseconds?

(a) 115 nsec. (b) 160 nsec.
(c) 170 nsec. (d) 180 nsec.

[MCQ]

8. A processor uses 36 bit physical address and 32 bit virtual addresses, with a page frame size of 4 Kbytes. Each page table entry is of size 4 bytes. A three-level page table is used for virtual to physical address translation, where the virtual address is used as follows:

- Bits 30 –31 are used to index into the first level page table.
- Bits 21 – 29 are used to index into the 2nd level page table.
- Bits 12 – 20 are used to index into the 3rd level page table.
- Bits 0 — 11 are used as offset within the page.

The number of bits required for addressing the next level page table(or page frame) in the page table entry of the first, second and third level page tables are respectively

(a) 20,20,20 (b) 24,24,24
(c) 24,24,20 (d) 25,25,24

Answer Key

1. **(a)**
2. **(b)**
3. **(d)**
4. **(1 to 1)**
5. **(184 to 184)**
6. **(103.44 to 103.44)**
7. **(c)**
8. **(d)**

Hints & Solutions

1. (a)

Page table base register holds the base address for the page table of the current process. Every process is associated with its own page table.

While context switching the process table PTBR has to switch to the next base address of the process immediately.

2. (b)

- Effective memory Access time = TLB_{hit} (TLB access time + Main memory access time)

  + TLB_{miss} (TLBaccess time + Page table access time + mm access time)

  EMAt = 0.6 (20 + x) + (1 – 0.6) (20 + x + x)

  400 = 0.6 (20 + x) + 0.4 (20 + 2x)

  400 = 12 + 0.6x + 8 + 0.8x

  0.6x + 0.8x = 400 – (12 + 8)

  1.4x = 380

  $x = \frac{380}{1.4}$ msec.

  x = 271. 43

3. (d)

- 126-page reference in the TLB out of total reference of 180:

  TLB hit ratio = $\frac{126}{80} = 0.7$

- Effective memory access time = TLB_{hit} (TLB_{time} + mm_{time}) + TLB_{miss} (TLB_{time} + 3*mm_{time})
  = 0.7 (20 + 300) + (1 – 0.7) (20 + 3 × 300)
  = 0.7 × 320 + 0.3 × 920
  EMAT = 500 nsec.

4. (1 to 1)

- Page Table size = (Number of entries in page table) × (Page Table Entry Size)
- Number of entries in Page Table

  $= \frac{\text{Virtual address space}}{\text{Page size}}$

  $= \frac{2^{32}}{2^{13}} = 2^{19}$ entries

- Page size is equal to frame size

∴ Number of frames in physical

$= \frac{\text{Physical memory}}{\text{Frame size}} = \frac{2^{25}}{2^{13}} = 2^{12}$ Frames

- Now, Frame bits locate page in physical memory = $\log_2 2^{12}$ = 12 bits

∴ Page Table Entry Size = Frame Bits

≈ 12 bits ≈ 2 bytes

- Page Table Size = $2^{19} \times 2^1$
  = 2^{20} Bytes
  = 1 MB.

5. (184 to 184)

EMAT = TLB hit × (TLB access time + MM access time) + TLB miss (TLB access time + Page Table access time + MM access time)

= 0.6 × (30 + 110) + (1 – 0.6) (30 + 110 + 110)

= 84 + 100

= 184 millisec.

6. (103. 44 to 103.44)

- Average access time = Page fault (% dirty × (Time to replace dirty page) + (1 – % dirty) × (Page fault services time) + (1 – Page fault) × MM access time

  = 0. 4 (0.06 × 300 + 0.94 × 240 + 0.6 ×10)

  = 0.4 (18 + 225.6) + 6

  = 97.44 + 6

  = 103.44-time units

7. (c)

- 4 level paging is used in the given system.

∴ EmAt = TLB_{hit} (TLB_{time} + MM_{time}) + (1– TLB_{hit}) (TLB_{time} + (n + 1) MM_{time})

EmAt = 0.9 (30 + 100) + (1– 0.9) (30 + 5 × 100)

= 0.9 × 130 + 0.1 × 530

= 117 + 53

= 170 nsec.

8. (d)

Physical address is 36 bits. So, number of bits to represent a page frame = 36 – 12 = 24 bits (12 offset bits as given in question to address 4 KB assuming byte addressing). So, each entry in a third level page table must have 24 bits for addressing the page frames.

A page in logical address space corresponds to a page frame in physical address space. So, in logical address

space also we need 12 bits as offset bits. From the logical address which is of 32 bits, we are now left with 32 — 12 = 20 bits', these 20 bits will be divided into three partitions (as given in the question) so that each partition represents which entry' in the i^{th} level page table we are referring to.

- An entry in level *i* page table determines 'which page table' at $(i-1)^{th}$ level is being referred.

Now, there is only 1 first level page table. But there can be many second level and third level page tables and "how many" of these exist depends on the physical memory capacity. (In actual the no. of such page tables depend on the memory usage of a given process, but for addressing we need to consider the worst case scenario). The simple formula for getting the number of page tables possible at a level is to divide the available physical memory size by the size of a given level page table.

Number of third level pages tables possible

$$= \frac{\text{Physical memory size}}{\text{Size of a third level page table}}$$

$$= \frac{2^{36}}{\text{Number of entries in a singal third level page table} \times \text{Size of an entry}}$$

$$= \frac{2^{36}}{2^9 \times 4} \therefore (\text{bits 12-20 gives 9 bits})$$

$$= \frac{2^{36}}{2^{11}} = 2^{25}$$

So, we need 25 bits in second level page table for addressing the third level page tables.

Similarly, we need to find the no. of possible second level page tables and we need to address each of them in first level page table.

Now,

Number of second level page tables possible

$$= \frac{\text{Physical memory size}}{\text{Size of a second level page table}}$$

$$= \frac{2^{36}}{\text{Number of entries in a single second level page table} \times \text{size of an entry}}$$

$$= \frac{2^{36}}{2^9 \times 4} Q\,(\text{bits 21– 29 gives 9 bits})$$

$$= \frac{2^{36}}{2^{11}}$$

$$= 2^{25}$$

So, we need 25 bits for addressing the second level tables as well.

So, answer is (D).