CS & IT ENGINEERING

Operating System

Memory Management

DPP Discussion Notes

By- Anjnee Bhatnagar ma'am

[MCQ]

#Q. Consider a fixed partition contiguous memory management technique, where there are 5 partitions of size 100MB, 250MB, 200MB, 500MB and 300MB. All Partitions are initially empty. The following process requests are made in the given order:

Process	Size
P1	150 MB
P2	400 MB
P3	270MB
P4	180MB
P5	80MB

The following 2 answers are calculated for First fit, Best fit and Worst Fit policies

1. Maximum degree of multiprogramming?
2. What is the total internal fragmentation size?

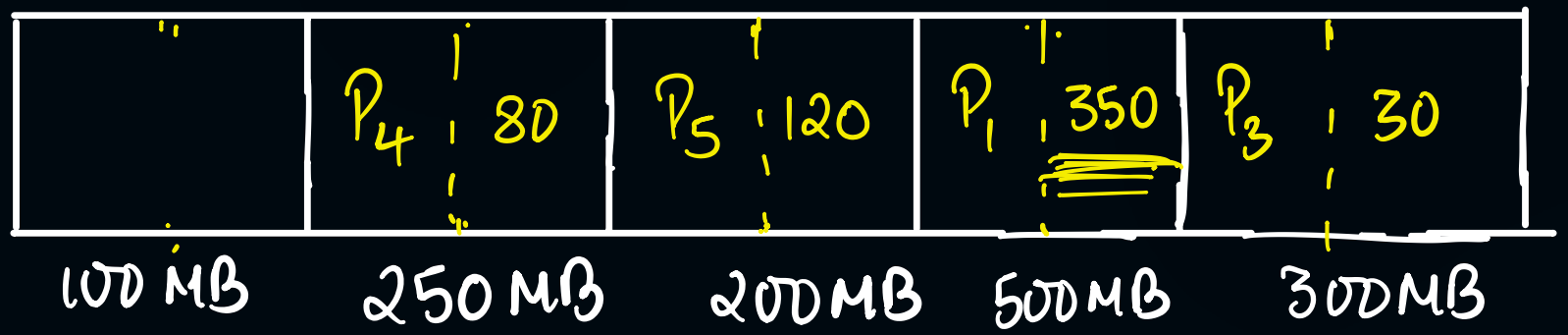
For which of the following partition allocation policy the above two calculations are not exactly same as other 2 policies?

A Best fit

B First fit

C Worst fit

D None of the above



P_1	150
P_2	400
P_3	270
P_4	180
P_5	80

X

I. F.F

1. MDM ⇒ 5
2. T.I.F ⇒ 20 + 100 + 20 + 100 + 30 ⇒ 270.

II. B.F

1. M.D.M ⇒ 5
2. T.I.F ⇒ 20 + 70 + 50 + 100 + 30 ⇒ 270.

III. W.F

1. M.D.M ⇒ 4
2. T.I.F ⇒ 80 + 120 + 350 + 30 ⇒ 570 MB

[NAT]

#Q. Consider a paged memory system where the logical address is 25 bits and physical address is 33 bits. The page size is 4KB. The approximate size of page table size is 21 k bytes ($1k = 2^{10}$)?

$4kB = 2^{12}B$

P.T.S ⇒ No. of pages * P.T.E.

$\log_2 2^{12} \Rightarrow 12$ bits.

# of pages ⇒ $\frac{2^{25}}{2^{12}} \Rightarrow 2^{13} = 8K$.

P.T.E = Translation bits + other info.

# of frames ⇒ $\frac{2^{33}}{2^{12}} \Rightarrow 2^{21} = \log_2 2^{21}$

$= 21$ bits.

P.T.S ⇒ $2^{13} \times 21$ bits ⇒ $\frac{2^{13} \times 21}{2^3} \Rightarrow 21 \times 2^{10} = 21KB$.

[NAT]

#Q. Consider a paged memory system where the page number is 12 bits and physical address is 33 bits. The page size is 2KB. The approximate size of page table size is 11 k bytes ($1k = 2^{10}$)?

P.T.S = # of pages * P.T.E

# of pages => 2^{12}

# of frames => $\frac{2^{33}}{2^{11}} = 2^{22}$ => 4M

P.T.S => 2^{12} x 22 bits = 11 kB

[NAT]

#Q. Consider a paged memory system which has 4k pages each with 8kbytes size. The system maintains page table of 12k bytes. Each page table entry contains frame number, 1 valid bit, 1 modified bit and 1 replacement bit. The number of frames in the main memory is 2. $*2^{20}$?

P.T.S = # of pages * P.T.E.

12 kB = 4k * (F + 1 + 1 + 1) bits

3 12K * 8 bits = 4K (F + 3) bits

24 bits = F + 3 bits

F = 21 bits. $\Rightarrow 2^{21} \Rightarrow 2 * 2^{20}$

[MCQ]

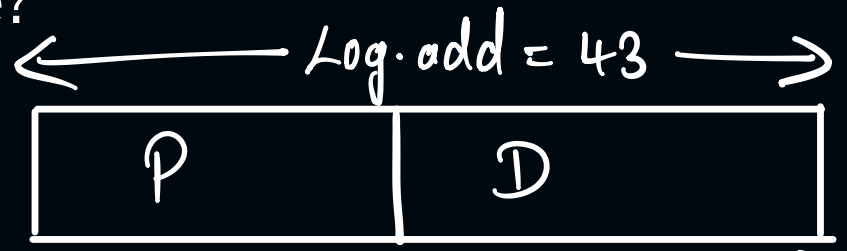
#Q. A system has 43-bit logical addresses and 51-bit physical addresses. If the pages are 8 kB in size, the number of bits required for logical page number and physical frame number will be?

A. 43 bits, 51 bits

B. 30 bits, 51 bits

C. 43 bits, 38 bits

D. 30 bits, 38 bits

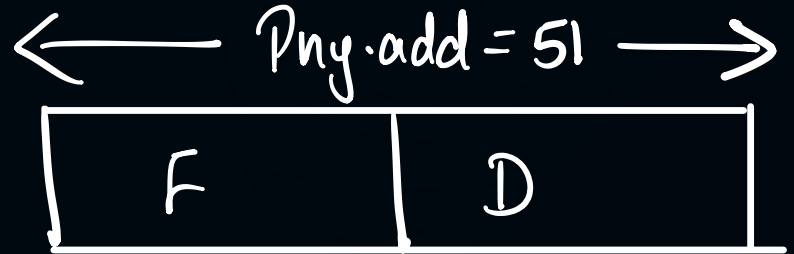


P = 43 - 13 = 30 bits

$8KB = 2^{13} B$

$D = \log_2 2^{13}$

= 13 bits



F = 51 - 13 = 38 bits

[NAT]

#Q. Consider a logical-address space of 8 pages, with page size 1024 bytes. The physical memory contains 32 frames. The page table size is ____ bits?

P.T.S = # of pages * P.T.E.

= 8 * 5 bits

= 40 bits.

$\log_2 32 = 5$ bits.

P.T.E = 5 bits

[NAT]

#Q. Consider a system using TLB for paging with TLB access time of 40ns. The hit ratio is to be used for TLB to reduce the effective memory access time from 400ns without using TLB to 280ns with using TLB, is 80 %??

Without TLB ⇒ 2 * M.M

400 ns ⇒ 2 * M.M

M.M = 200 ns

0.8 x 100 = 80%

With TLB ⇒ H (TLB + MM) + (1-H) (TLB + 2 M.M)

280 ⇒ H (40 + 200) + (1-H) (40 + 400)

280 ⇒ 240 H + 440 − 440H , H = 0.8

[NAT]

#Q. A computer system implements a 42- bit virtual address, 512GB physical address space, page size of 2KB, and an 8KB look-aside buffer (TLB) organized as direct mapped. Each page table entry contains a valid bit, a dirty bit and 2 protection bits along with the translation. The minimum length of the TLB tag in bits is 20. ?

I. Frame bits :

$$\frac{512\,GB}{2KB} \Rightarrow 256M = 2^{28}$$

$$\log_2 2^{28} = 28 \text{ bits.}$$

P.T.E = 28 + 1 + 1 + 2 ⇒ 32 bits. ⇒ 4 Bytes.

II.

$$\frac{8KB}{4B} \Rightarrow 2K \Rightarrow 11 \text{ bits.}$$

P.S = 2KB = 11 bits.

⇒ 42 − (11 + 11) ⇒ 20 bits.

THANK - YOU