Memory Management Practice Questions

Linking, Loading and Memory Allocation

1) Consider the block sizes and the process sizes as given in the table below. Perform the memory allocation using First Fit algorithm.

Block Size (KB)	Process Size(KB)
B1: 28	P1: 46
B2: 98	P2: 89
B3: 78	P3: 27
B4: 25	P4: 76
B5: 47	P5: 25

Solution :

Block no.	Bsize	process no.	Psize
B1	28	P3	27
B2	98	P1	46
B3	78	P4	76
B4	25	P5	25
B5	47	Not allocated	

2) Consider the block sizes and the process sizes as given in the table below. Perform the memory allocation using First Fit algorithm.

Block Size (KB)	Process Size(KB)
B1: 287	P1: 461
B2: 982	P2: 891
B3: 780	P3: 279
B4: 255	P4: 763
B5: 479	P5: 259

Solution :

Block no.	Bsize	process no.	Psize
B1	287	P3	279
B2	982	P1	461
B3	780	P4	763
B4	255	Not allocated	
B5	479	P5	259

3) Consider the block sizes and the process sizes as given in the table below. Perform the memory allocation using Best Fit algorithm.

Block Size (KB)	Process Size(KB)
B1: 28	P1: 46
B2: 98	P2: 89
B3: 78	P3: 27
B4: 25	P4: 76
B5: 47	P5: 25

Solution :

Process_no	Process_size	Block_no	Block_size
P1	46	B5	47
P2	89	B2	98
P3	27	B1	28
P4	76	B3	78
P5	25	B4	25

4) Consider the block sizes and the process sizes as given in the table below. Perform the memory allocation using Best Fit algorithm.

Block Size (KB)	Process Size(KB)
B1: 287	P1: 461
B2: 982	P2: 891
B3: 780	P3: 279
B4: 255	P4: 763
B5: 479	P5: 259

Solution :

Process_no	Process_size	Block_no	Block_size
P1	461	B5	479
P2	891	B2	982
P3	279	B1	287
P4	763	B3	780

5) Consider the block sizes and the process sizes as given in the table below. Perform the memory allocation using Worst Fit algorithm.

Block Size (KB)	Process

	Size(KB)
B1: 28	P1: 46
B2: 98	P2: 89
B3: 78	P3: 27
B4: 25	P4: 76
B5: 47	

Solution :

Process No.	Process Size	Block no.
P1	46	B2
P2	89	Not Allocated
P3	27	B3
P4	76	Not Allocated

6) Consider the block sizes and the process sizes as given in the table below. Perform the memory allocation using Worst Fit algorithm.

Block Size (KB)	Process Size(KB)
B1: 287	P1: 461
B2: 982	P2: 891
B3: 780	P3: 279
B4: 255	P4: 763
B5: 479	

Solution :

Process No.	Process Size	Block no.
P1	461	B2
P2	891	Not Allocated
P3	279	B3
P4	763	Not Allocated

7) Consider the following fixed partition memory information of a certain system with four processes. Block sizes{B1,B2,B3,B4} ={200,30,700,50} and process sizes{P1,P2,P3,P4} = {20,200,500,50}. Using the first fit memory allocation, calculate the total internal fragmentation.

A) 710
B) 680
C) 700
D) 690

Solution: (B)

```
Unallocated Process P3

Memory          200     30      700     50
P. Alloc.       P1      Empty   P2      P4
Int. Frag.      180     Empty   500     0

Total External Fragmentation: 30
Total Internal Fragmentation: 680
```

8) Consider the following fixed partition memory information of a certain system with four processes. Block sizes{B1,B2,B3,B4} ={200,30,700,50} and process sizes{P1,P2,P3,P4} = {20,200,500,50}. Using the best fit memory allocation, calculate the total internal fragmentation.
A) 210
B) 180
C) 200
D) 190

Solution: (A)

```
Memory          30      50      200     700
P. Alloc.       P1      P4      P2      P3
Int. Frag.      10      0       0       200

Available Memory: 0
Total Internal Fragmentation: 210
```

9) Consider the following fixed partition memory information of a certain system with four processes. Block sizes{B1,B2,B3,B4} ={200,30,700,50} and process sizes{P1,P2,P3,P4} = {20,200,500,50}. Using the worst fit memory allocation, calculate the total internal fragmentation.
A) 710
B) 680
C) 700
D) 690

Solution: (B)

```
Memory          200     30      700     50
P. Alloc.       P2      Empty   P1      P4
Int. Frag.      0       Empty   680     0

Total External Fragmentation: 30
Total Internal Fragmentation: 680
```

10) Consider the following fixed partition memory information of a certain system with four processes. Block sizes{B1,B2,B3} ={5,10,20} and process sizes{P1,P2,P3,P4} = {10,20,30}. Using the next fit memory allocation, find the block that remains unallocated.
A) B1
B) B2
C) B3
D) All blocks are allocated.

Solution: (A)

```
Unallocated Process P3

Memory          5       10      20
P. Alloc.       Empty   P1      P2
Int. Frag.      Empty   0       0

Total External Fragmentation: 5
Total Internal Fragmentation: 0
```

Therefore, block B1=5, remains unallocated.

11) Which of the following are the advantages of dynamic linked libraries? **[MSQ]**

A) The executable is smaller.
B) When the library is changed, the code that references it does not usually need to be recompiled.
C) Every loaded program contains a duplicate copy of library routines.
D) Multiple programs can access the same library at the same time.

Solution: (A) (B) (D)
The executable accesses the libraries at run time; therefore, multiple programs can access the same library at the same time (saves memory), and hence the executable is smaller. When the library is changed, the code that references it does not usually need to be recompiled. However, in static linking every loaded program contains a duplicate copy of library routines.

12) Consider a program X with three procedures, A, B, and C, are to be linked together into one process and loaded into memory. The length of each procedure is 600 words. The memory manager uses paging where page size is 1000 words and page tables occupy 1 page each. Assume that all procedures and all tables are in memory. The total internal fragmentation with the above allocation for program X is _____ words. **[NAT]**

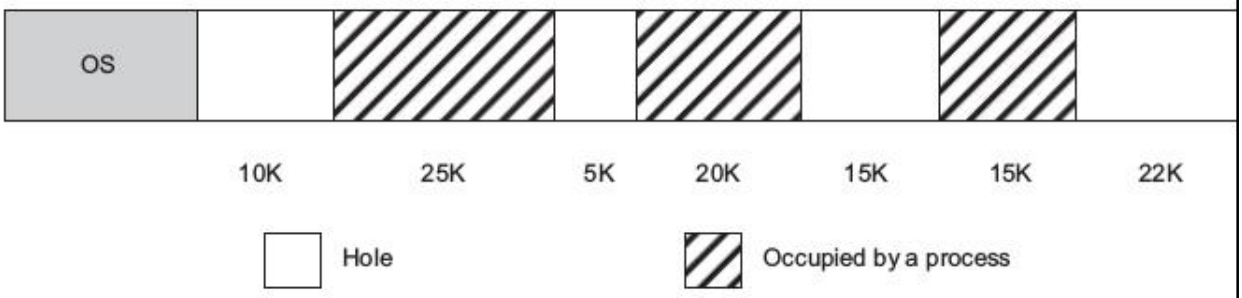
Solution: 200 words.
The page table is 1 page, and the three procedures make up a program that is 1800 words long, so it occupies 2 pages. Hence the total occupied memory is 3 pages or 3000 words. Out of which the second page that stores the page table will have only 800 words, rest 200 words space leads to internal fragmentation.

13) A memory management unit performs memory mapping by converting a logical address into a physical address, with the help of _____

A) base registers
B) base and limit registers
C) limit registers
D) none

Solution B) The base address of process is stored in base address register and the offset is stored in limit register of the CPU. The values of these registers are loaded during the context switch.

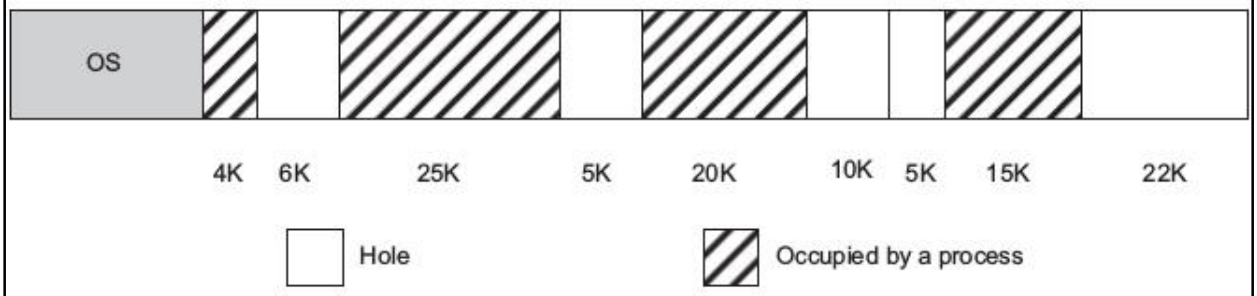
14) Consider the memory allocation scenario as shown in figure below. Allocate memory for additional requests of 4K and 10K (in this order). The total internal fragmentation for memory allocation, using first-fit, best-fit, and worst-fit allocation methods in that order is



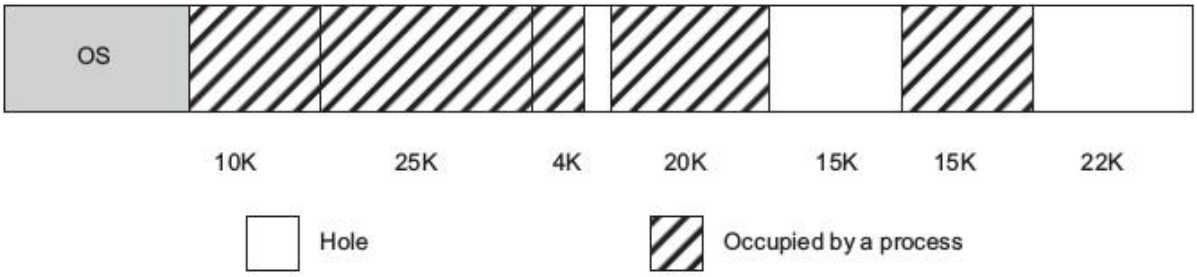
A) 1K, 11K, 8K
B) 11K, 8K, 1K
C) 1K, 10K, 18K
D) 11K, 1K, 8K

Solution (D)

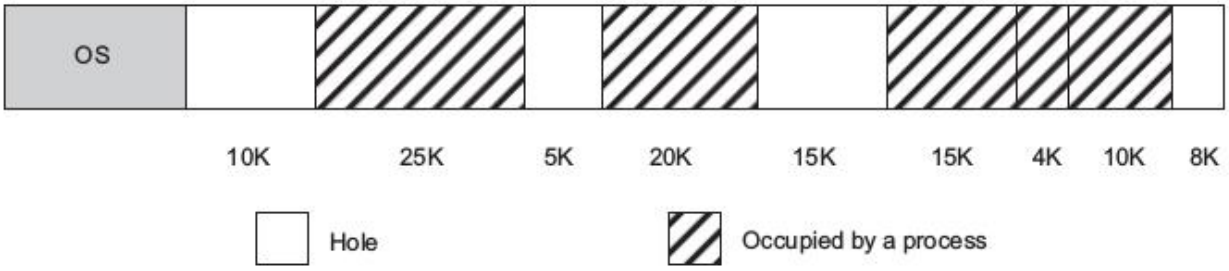
First Fit : It allocates the first hole in the list that is big enough. Hole of 10K is allocated to the process of 4K, leaving a hole of 6K in memory. The next request is of 10K. So, the next hole in the list is of 5K and it cannot accommodate the process. Hence, the next available
hole of 15K is allocated, leaving a hole of 5K . It leaves fragmentation of 6K+5K = 11K.



Best Fit : It allocates the smallest hole that is big enough. Hole of 5K is allocated to the process of 4K, leaving a hole of just 1K. The next request is of 10K. After comparing the size of all the holes, hole of 10K is allocated. It leaves fragmentation of 0K +1K = 1K



Worst Fit : It allocates the largest hole in the list. For the process of 4K, the largest hole of 22K is allocated. This leaves a hole of 18K in the memory. The next request is of 10K.Comparing the size of all holes, the 18K hole is the largest hole in the list. Hence, it is allocated to the process. It leaves fragmentation of 22K – 4K – 10K = 8K



15) Consider a machine with 1 MB physical memory and a 32 bit virtual address space. If the page size is 4 KB, the size of the page table is _____ MB. **[NAT]**

Solution: 1
Total number of frames = 1MB/4KB = 2^20/2^12 = 2^8. So, we require 8 bits for page table entry. Therefore, the page table size = (8bits*2^32)/4KB = 2^20 bytes = 1MB.

16) In a virtual memory system, size of virtual address is 32-bit, size of physical address is 30-bit, page size is 4 Kbyte and size of each page table entry is 24-bit. The main memory is byte addressable. The maximum number of bits that can be used for storing protection and other information in each page table entry is _____. **[NAT]**

Solution: 6
Total number of frames = 2^30 / 2^12 = 2^18. So, we require 18 bits for page numbering. Rest 24-18=6 bits are used or storing protection and other information in each page table entry.

17) The page size in a system is increased while keeping everything else (including the total size of main memory) the same. Which of the following is/are TRUE as a result of this increase in page size **[MSQ]**
A) Size of the page table of a process decreases.
B) Internal fragmentation of main memory increases.
C) TLB hit rate increases
D) Size of the page table of a process increases.

Solution : (A), (B) and (C)
Page table size decreases as we have fewer entries , TLB hit rate increases because of more coverage of memory pages-frames and Internal fragmentation increases as more space is wasted in a page.

18) Consider a system with byte addressable memory, 32 bit logical addresses, 4KB page size and page table entries of 4 bytes each. The size of the page table in the system in is _____MB.

Solution : 4

19) Consider a machine with 64 MB physical memory and a 32 bit virtual address space. If the page size is 4 KB, what is the size of the page table?

Solution : 2^{20} x 14 bits. (approx. 2MB)

20) In a virtual memory system, size of virtual address is 32-bit, size of physical address is 30-bit, page size is 4 Kbyte and size of each page table entry is 32-bit. The main memory is byte addressable. The maximum number of bits that can be used for storing protection and other information in each page table entry is _____.

Solution : 14 bits.

21) Given that virtual address space is 4 KB and page table entry size is 8 bytes. Calculate the optimal page size .

Solution : 256 bytes. (Optimal page size = sqrt(2*process size * PTE))

22) The virtual address space is 2^22 bytes and page size is 4KB. What is the maximum page table entry such that the entire page table fits in one page?

Solution : approx. 4Bytes

23) Consider a virtual address with a page size of 1KB. Two-level paging is used with equal number of entries in every page table. If the page table entry is 2 bytes long, what is the maximum size of the physical address space. ?

Solution : 64MB.

24) A system has a 32 bit virtual address space, and combines both segmentation and paging as shown below. The page table entry is of 4 bytes, 4 bits for the segment, 16 for the page, and 12 for the offset. The CPU issues a read for address 0xC0DEDBAD, then the page address is _____ .

SEGMENT NUMBER| PAGE NUMBER| OFFSET

A) 0x0DED
B) 0xC0DE
C) 0xDBAD
D) 0xDEDB

Solution : (A)
The first 4 bits are segment : 0xC
The next 16 bits for page : 0x0DED
The last 12 bits for offset : BAD

25) A multilevel page table is preferred in comparison to a single level page table for translating virtual address to physical address because [MSQ]
(A) it reduces the memory access time to read or write a memory location
(B) it helps to reduce the size of page table needed to implement the virtual address space of a process
(C) it is required by the translation lookaside buffer
(D) it helps to reduce the number of page faults in page replacement algorithms

Solutions : B) If a single level is used then page table size itself may become more than the

process size. Hence we go for multilevel paging.

26) Consider a virtual memory system with 40-bit virtual addresses, 32-bit physical addresses, and 16 KB pages, the TLB stores 128 entries and the cache is directly mapped. Then, the total number of bits in TAG, INDEX and OFFSET are
A. 11, 7, 14
B. 19, 7, 14
C. 7, 19, 14
D. 7, 11, 14
Solution: (B)
Explanation:
Since the page-size is 16 KB, the page-offset field is 14 bits long, and hence the Virtual Page Number (VPN) field is 40-14 = 26 bits long.
For Direct Mapping :

Since the TLB holds 128 entries and is accessed using the VPN, the log2(128) = 7 least significant bits of the VPN will be used as the index into the TLB, and the remaining 267 = 19 VPN bits will be used as the tag.

27) Consider a 32-bit computer with page size 2KB and physical memory of 32 GB. Each virtual page has additional 8 bits for hardware control. The computer has a two-level page table. The minimum number of bits in second level page table entry is
(A) 24
(B) 32
(C) 40
(D) 16

Solution : (B) 32
There are 2^{35} bytes of memory and 2^{11} bytes/page, which means there are 2^{24} pages. The second-level page table has to have at least 24 bits to be able to select the physical page. Each virtual page has additional 8 bits for hardware control. So in total we need 24+8 = 32 bits.

28) Consider a process with 4 pages, numbered 0–3. The page table of the process consists of the following logical page number to physical frame number mappings: (0, 11), (1, 35), (2, 3), (3, 1). The process runs on a system with 16 bit virtual addresses and a page size of 256 bytes. The process accesses virtual address 770, which logical page number does this virtual address correspond to?
A) Page 3
B) Page 0
C) No entry found.
D) Page 1

Solution : (A)
770 = 512 + 256 + 2 = 00000011 00000010 = page 3, offset 2

29) Which of the following is/are TRUE ? **[MSQ]**

A) A smaller page size leads to smaller page tables
B) A smaller page size leads to more TLB misses
C) A smaller page size leads to fewer page faults
D) A smaller page size reduces paging I/O throughput

Solution B) C) and D)
A smaller page size leads to smaller page tables :False – need more entries because we have more pages.
A smaller page size leads to more TLB misses – True – Less chances that the address we are looking for is part of this page.
A smaller page size leads to fewer page faults – True – More pages can be accomodated in the main memory, reducing the page faults.
A smaller page size reduces paging I/O throughput – True – Reduced page faults decreases IO.

30) Which of the following is/are true about the logical address to physical address binding.**[MSQ]**
A) This binding can be done at compile time.
B) Converts symbolic addresses to relocatable addresses.
C) This binding cannot be done at run time.
D) Data used within the compiled source is offset within the object module.

Solution: (A)(B)(D)
Logical address to physical address binding can be done at compile/link time. Converts symbolic to relocatable. Data used within the compiled source is offset within the object module.Binding can also be done at load time (Binds relocatable to physical).Can be done at run time. (Implies that the code can be moved around during execution)

31) Which of the following is/are true.**[MSQ]**

A) Dynamic loading provides better memory-space utilization .
B) Dynamic loading does not require special support from the OS.
C) Linking cannot be postponed until execution time.
D) Dynamic linking is particularly useful for libraries.

Solution: (A)(B)(D)
Better memory-space utilization; unused routine is never loaded in dynamic loading.
No special support from the OS is required - implemented through program design.
Linking can be postponed until execution time (dynamic linking). Dynamic linking is particularly useful for libraries.

32) Which of the following is/are true.**[MSQ]**

A) Valid bit indicates that the associated page is legal.
B) Present or absent bit says whether a particular page has a frame mapped.

C) Valid/Invalid bits are part of page table entry.
D) Valid/Invalid bits are part of memory management hardware.

Solution: (A)(B)(C)
Valid bit indicates that the associated page is in the process' logical address space, and is thus a legal page. Invalid bit indicates that the page is not in the process' logical address space. In case if it is not present (present or absent bit), that is called Page Fault. It is set to 0 if the corresponding page is not in memory.

Valid/invalid basically says is this page belongs to your process or not, present absent bits represents does a frame is allotted to this page in Main memory or not.

33) Which of the following is/are true.**[MSQ]**

A) Paging suffers from internal fragmentation.
B) Page tables are the overhead on the memory.
C) Multi level paging leads to increased memory overhead.
D) Multi level paging leads to increase in memory access time .

Solution: (A)(B)(D)
Paging reduces external fragmentation, but still suffers from internal fragmentation.
Page tables should be stored in main memory, which adds to memory overhead.
Multi level paging decreases the memory overhead by reducing page table size. This in turn will make the memory access to take place via all levels of paging, which increases memory access time.

34) A system has a 32 bit virtual address space, and combines both segmentation and paging as shown below. The page table entry is of 4 bytes, 4 bits for the segment, 16 bits for the page, and 12 bits for the offset. The CPU issues a read for address 0xC0DEDBAD, then the page address is _____ .

SEGMENT NUMBER	PAGE NUMBER	OFFSET

A) 0x0DED
B) 0xC0DE
C) 0xDBAD
D) 0xDEDB

Solution : (A)
The first 4 bits are segment : 0xC
The next 16 bits for page : 0x0DED
The last 12 bits for offset : 0xBAD

SEGMENT NUMBER (4 bits)	PAGE NUMBER (16 bits)	OFFSET (12 bits)

0xC	0x0DED	BAD

Consider a process with 4 pages, numbered 0–3. The page table of the process consists of the following logical page number to physical frame number mappings:

Page No	Frame No
0	11
1	35
2	3
3	1

The process runs on a system with 16 bit virtual addresses and a page size of 256 bytes. The process accesses virtual address 770, which frame number does this virtual address correspond to?
A) Frame 3
B) Frame 0
C) No entry found.
D) Frame 1

Solution : (D)
VA = 16 bits, page size = 256 bytes => 8 bits, rest 8 bits for page number.

Virtual address (16 bits)	
PAGE NUMBER (8 bits)	OFFSET (12 bits)

770 in binary is 00000011 00000010 = page 3, offset 2. Page 3 is mapped to frame number 1. Therefore, the address 770 belongs to frame 1.

35) Consider a virtual address with a page size of 4KB. Two-level paging is used with an equal number of entries in every page table. If the page table entry is 1 byte long, what is the maximum size of the physical address space._____ MB ? **[NAT]**

Solution :
PTE = 1 byte = 8 bits = Frame numbers. Therefore, we can have a maximum of 2^8 frames.
Size of main memory = No.of frames * Frame size
= 2^8* 4KB = 1MB.

Consider a computer system with a 32-bit logical address and 4KB page size. The system

supports up to 1GB of physical memory. How many entries will be there in an inverted page table?
A) 128K
B) 512K
C) 1M
D) 256K

Solution : (D)
No.of entries in inverted page table = No.of frames = Physical address space/frame size.
Therefore, entries in the inverted page table = 2^30/2^12 = 2^18 = 256K.

36) Consider a byte addressable system which uses 32 bits virtual address, 4KB page size with 2-level paging. The page table entry is 4 bytes.
The total size of the outer page table, if it is required to fit the outer table in one frame is _____ KB **[NAT]**

Solution: 4KB

No.of pages = VAS/PS = 2^32 bytes/2^12bytes = 2^20 pages.
The size of the page table = No.of page entries * page entry size = 2^20*4bytes = 2^22 bytes =4MB.
Since, 4MB > frame size (4KB), we divide the innermost page table as
4MB/4KB = 2^10 frames.
Number of page table entries in one page of inner page table
= Page size / Page table entry size
= Page size / Number of bits in frame number
= 4 KB / 32 bits
= 4 KB / 4 B
= 2^10.
Thus, Number of bits required to search a particular entry in one page of the inner page table = 10 bits.
Outer page table is required to keep track of the frames storing the pages of the inner page table.
Outer Page table size
= Number of entries in outer page table x Page table entry size
= Number of pages the inner page table is divided x Number of bits in frame number
= 2^10 x 32 bits
= 2^10 x 4 bytes
= 4 KB.

37) A program has been divided into five modules. Their lengths and base addresses are stored in the segment table, as depicted shown below :

Segment number	Length	Base address
0	200	4100
1	700	1000
2	400	3700
3	900	1800
4	1000	2700

<s,d> represents the segment number and the given offset value of the logical address. Which of the following addresses will not generate a segmentation fault ? **[MSQ]**

A) <3,906>
B) <1,665>
C) <4,770>
D) <0,127>

Solution (B)(C)(D)
<s,d> = <segment number, given offset>
<1,665> = segment 1 base address (1000) , Given offset (665); Valid (665 < 700).
<4,770> = segment 4 base address (2700) , Given offset (770); Valid (770 < 1000).
<3,906> Not a valid address, as offset is larger than the length of the segment (906 > 900).
<0,127> =segment 0 base address (4100) , Given offset (127); Valid (127 < 200).. Hence, valid.

38) Consider a system with 1KB page size and single-level paging is used. The virtual address space is 36 bits and the physical address space is 32 bits. Calculate the number of frames required to store the entire page table.

A) 2^18
B) 2^20
C) 2^26
D) 2^28

Solution: (A)

The number of page table entries = VAS/Page size = 2^36/2^10 = 2^26 entries.
Since, the physical address space is 32 bits, the page table entry is therefore 32 bits = 4 bytes long. So, the entire page table takes 2^26*4 bytes = 256MB.
The number of frames required to store this page table is thus, 256MB/1KB = 2^18.

39) Consider a system with 4KB page size and 3-level paging is used. The virtual address space is 36 bits and the physical address space is 32 bits. The page table entry at every level is 16bytes.The number of frames required to store all the level page tables, if each level uses 8 bits for page table indexing and each page table fits one page-frame_____ **[NAT]**

Solution: 65539

8bits	8bits	8bits	12bits

Level1	Level2	Level3	Offset

The number of pages in the virtual memory = 2^36/2^12 = 2^24 pages. But the level 3 page table has only 2^8 entries, so, one page table of level 3 can point to 2^8 pages only. So, we need 2^16 level-3 page tables.

Level 2: we need 2^16 entries in Level-2. But the level 2 page table has only 2^8 entries, so, one page table of level 2 can only point to 2^8 page tables of level-3, So, we need 2 level-2 page tables.

Level 1: We need 1 Level-1 page table to point to level-2 page tables.

So the total number of pages for storing all three levels page tables = (level-1 page tables) + (level-2 page tables) + (level-3 page tables) = 1+2+2^16 = 65539

40) Choose the correct statement(s) among the following: **[MSQ]**

A) Paging eliminates the problem of external fragmentation and therefore the need for compaction.
B) Paging allows sharing of code pages among processes, reducing overall memory requirements.
C) Paging enables processes to run when they are only partially loaded in main memory.
D) Paging reduces the memory access overhead.

Solution: (A)(B) and (C)
Paging
i) eliminates the problem of external fragmentation and therefore the need for compaction.
ii) allows sharing of code pages among processes, reducing overall memory requirements.
iii) enables processes to run when they are only partially loaded in main memory (on demand paging)
iv) Translating from a virtual address to a physical address (memory access overhead) is more time consuming.

41) Suppose that we have a two-level page translation scheme with 4K-byte pages and 4-byte page table entries (includes a valid bit, a couple permission bits, and a pointer to another page/table entry). What is the format of a 32-bit virtual address

A) Outer Page Table Index : 8 bits, Inner Page Table Index : 14 bits, Offset : 12 bits
B) Outer Page Table Index : 12 bits, Inner Page Table Index : 12 bits, Offset : 12 bits
C) Outer Page Table Index : 10 bits, Inner Page Table Index : 10 bits, Offset : 12 bits
D) Outer Page Table Index : 10 bits, Inner Page Table Index : 12 bits, Offset : 10 bits

Solution : (C)
Outer Page Table Index : 10 bits, Inner Page Table Index : 10 bits, Offset : 12
Page size is 4KB so, Offset is 12 bits.

Outer page table should fit in one page. 4KB / 4byte = 1K entries or 10 bits. So, remaining bits = (32-12-10) = 10 bits for inner page table.

42) A multilevel page table is preferred in comparison to a single level page table for translating virtual address to physical address because

(A) it reduces the memory access time to read or write a memory location.
(B) it helps to reduce the size of the page table needed to implement the virtual address space of a process.
(C) it is required by the translation lookaside buffer.
(D) it helps to reduce the number of page faults in page replacement algorithms

Solutions : (B)
If a single level is used then page table size itself may become more than the process size. Hence we go for multilevel paging.

43) Consider the following segment table: Which of the following logical addresses (Segment Number, Offset) may generate a TRAP?

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

A) 0,430
B) 1,10
C) 2,500
D) 3,400

Solution: (C)
The length of Offset should be less than Segment length.
Seg 0: 219 + 430 = 649
Seg 1: 2300 + 10 = 2310
Seg 2: Since 500 > 100 for Segment 2, illegal reference; traps to operating system
Seg 3: 1327 + 400 = 1727.

44) A virtual address ‘a’ in a paging system is equivalent to a pair (p, w), in which ‘p’ is a page number and ‘w’ is a byte number within the page. Let ‘z’ be the number of bytes in a page. The relation between ‘p’, ‘z’, ‘w’ and ‘a’ is
A) p=az + w
B) a=pw + z
C) z=pa + w
D) a=pz + w

Solution: (D)
The relationship is a=pz + w, 0 ≤ w < z. By definition of VA-PA mapping, virtual address is found by page number(p) x page size(z) + Offset (w)

45) Consider a page reference string for a process with a working set of M frames, initially all empty. The page reference string is of length P with N distinct page numbers in it. For any page replacement algorithm, lower bound and upper bound on the number of page faults is
A) N, P
B) N, P/2
C) N/2, P
D) N/2, P/2

Solution: (D)
Lower bound : N, since there are N distinct pages.
Upper Bound : P, all references may create page fault.

46) A page table entry provides _____
A) offset
B) limit address
C) base address
D) None

Solution C) A page table entry provides the frame number which acts as the base address for the instruction issued by the CPU. In simple terms, the address issued by the CPU is present in this particular frame for which the entry was made in page table. Therefore, this page entry acts as the base address for this instruction.

47) Memory mapping through TLB is known as _____
A) associative mapping
B) TLB mapping
C) physical mapping
D) none

Solution (A) High speed associative cache memory is also known as Translation Look-aside Buffer (TLB)

48) Rather than having the page table entry for a virtual page, _____ is taken as a page table entry in the inverted page table.

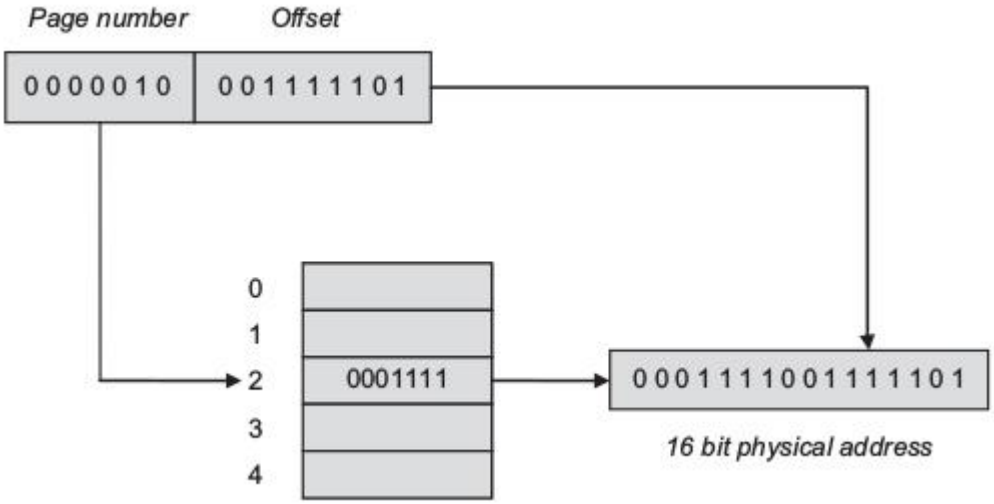
A) Process Id
B) Page table base register (PTBR)
C) Real page frame
D) None

Solution (A) An inverted page table has <process_id, frame_number> as the page table entry.

49) In a paging scheme, 16-bit addresses are used with a page size of 512 bytes. If the logical address is **0000010001111101**. What will be the physical address, if the frame address
corresponding to the computed page number is 15.

A) 0001011001101101
B) 0000111001100101
C) 0000111001111101
D) 0001111001111101

Solution (D)
Logical address space = 2^16
Page size = 512 = 2^9
Number of bits required for the page number = 16 - 9 = 7
Number of bits required for offset, b = 9
Page number is obtained by considering the leftmost 7 bits of the logical address
i.e., Page number = (0000010) = 2
Offset is obtained by considering the rightmost 9 bits of the logical address
i.e., Offset = (001111101) 2 = 125
Frame address corresponding to the second page in bits = (15) = 0001111
Appending the frame address to the offset, we get
the physical address = 0001111001111101

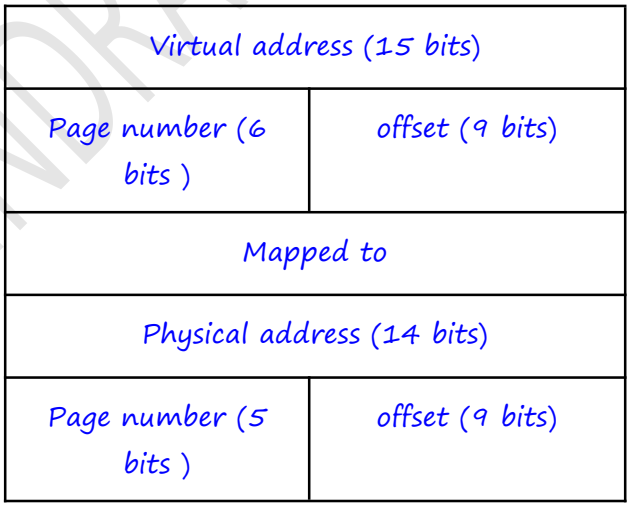


:

50) There is a system with 64 pages of 512 bytes page size and a physical memory of 32 frames. How many bits are required in the logical and physical address?

A) 15, 15
B) 15, 14
C) 14, 15
D) 14, 14

Solution (B)
Given, page size = 512 bytes = 2^9 i.e., b = 9
No. of pages = 64 = 2^6 i.e., p = 6
Since p = a – b, a = 15
Therefore, 15 bits are required in the logical address.
Since the number of bits required for offset, b = 9 and
The number of bits required to address 32 (2^5) frames = 5
Therefore, after appending the number of bits required for the frame size to the offset, 14 bits are required in the physical address.



51) A program has been divided into five modules. Their lengths and base addresses are stored in the segment table, as depicted shown below :

Segment number	Length	Base address
0	200	4100
1	700	1000
2	400	3700
3	900	1800
4	1000	2700

<s,d> represents the segment number and the offset of the logical address. Which of the following address will generate a segmentation fault ?

A) <3,906>
B) <1,665>
C) <4,770>
D) None

Solution (A)

<1,665> = 1000 + 665 = 1665 ; Valid (665 < 700)

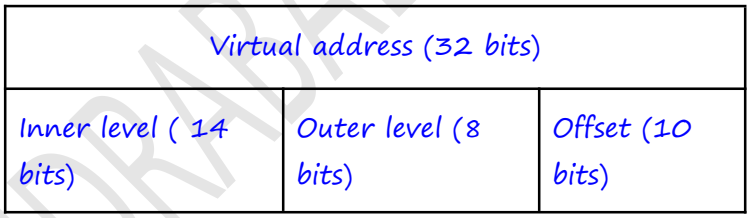
<4,770> = 2700 +770 = 3470; Valid (770 < 1000)

<3,906> Not a valid address, as offset is larger than the length of the segment (906 > 900).

52) A system with 32-bit logical address uses a two-level page table structure. It uses page size of 2^10 bytes . The outer page table is accessed with 8 bits of the address. How many entries are there in the inner page table?

A) 256 entries
B) 8192 entries
C) 16384 entries
D) 4096 entries

Solution (C)
Since the page size is 2^10 bytes, therefore, 10 bits are required for the offset. It is given that 8 bits are required to access the outer page table, so bits required to access the inner page table = 32 – (10 + 8) = 14. Since inner page table is accessed with 14 bits, it can have 2^14 = 16, 384 entries.



Reference : https://elearning.ravindrababuravula.com/materials/course/14/2394

53) A computer with a 32-bit address uses a two-level page table. Virtual addresses are split into a 9-bit top-level page table field, an 11-bit second-level page table field, and an offset. How many pages are there in the address space?

A) 2^20
B) 2^12
C) 2^9
D) 2^11

Solution (A)

20 bits (9+11) are used for the virtual page numbers, leaving 12 over for the offset. Therefore, page size is 4KB. Twenty bits for the virtual page implies 2^20 pages.

54) A computer with an 8KB page, a 256MB main memory, and a 64GB virtual address space uses an inverted page table to implement its virtual memory. If a hash table is used to map the pages, then what is the number of entries in the hash table?

A) 4M
B) 16K
C) 32K
D) 8M

Solution (C)

The main memory has 2^28 /2^13 = 32,768 pages = 32K. So hash table has 32K entries. Here page table is stored using hash table.

Page Replacement Algorithms

55) Consider the page reference string given below , assuming 3 page-frames, and all frames are initially empty. Calculate the number of page faults using Optimal algorithm : **[NAT]**

4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

Solution : 5

56) Consider the page reference string given below , assuming 4 page-frames, and all frames are initially empty. Calculate the number of page faults using FIFO algorithm : **[NAT]**

4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

Solution : 5

57) Consider the page reference string given below , assuming 4 page-frames, and all frames are initially empty. Calculate the number of page faults using LRU algorithm : **[NAT]**

4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

Solution : 5

58) A process has 5 logical pages, and accesses them in this order: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5. Which of the following statements is/are TRUE ? **[MSQ]**

A) The above referenced string suffers from Belady's anomaly when used with FIFO page replacement algorithm.
B) The above referenced string generates 9 page faults with 3 frames, and 10 page faults with 4 frames, using the FIFO page replacement.
C) The above referenced string suffers from Belady's anomaly when used with LRU page replacement algorithm.
D) In LRU, the N most recently used frames are always a subset of N+1 most recently used frames. So if a page fault occurs with N+1 frames, it must have occurred with N frames also.

Solution: (A), (B) and (D).
An algorithm that satisfies stack property property does not suffer from Belady's anomaly. The property considers the execution of an algorithm with two different page-frame sizes, say m and n where n < m. At an instant of time, all the pages that are in the memory when the page-frame size is n would also be in the memory when the page frame size is m. In other words, the set of pages when the page-frame size is n is a subset of the set of pages when the page-frame size is m. The implication of stack property is that the number of page faults in page-frame size n is more compared to that of m.

For string above, 9 faults with 3 frames and 10 faults with 4 frames when used with FIFO page replacement algorithm. Thus, it exhibits Belady's anomaly. In LRU, the N most recently used frames are always a subset of N+1 most recently used frames. So if a page fault occurs with N+1 frames, it must have occurred with N frames also. So page faults with N+1 frames can never be higher. Therefore, LRU does not suffer from Belady's anomaly.

59) Suppose you know that there will only be 4 processes running at the same time, each with a Resident Set Size (RSS) of 512MB and a working set size of 256KB. Assuming the same system with 32 bit virtual and physical address space with 4KB page size, what is the minimum amount of TLB entries that your system would need to support to be able to map/cache the working set size for one process?

A) 64
B) 128
C) 512
D) 1024

Solution : (A) 64
A process has a working set size of 256KB which means that the working set fits in 64 pages. (256KB/4KB) = 64 pages. A TLB contains page entries for translation, so A TLB should have atleast 64 entries.

60) Consider a demand paging system with four physical memory frames and the following reference string over seven pages: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6

Assuming that memory starts empty, how many page faults will occur and what will be the final contents of memory under the FIFO page replacement policy?

A) 3,7,6,1 in memory and 10 page faults
B) 3,7,6,1 in memory and 11 page faults
C) 3,7,2,1 in memory and 12 page faults
D) 3,7,6,1 in memory and 12 page faults

Solution : B) 3,7,6,1 in memory and 11 page faults
1,2,3,4 4 faults
5,6,2,1 4 faults
3,6,2,1 1 faults
3,7,2,1 1 faults
3,7,6,1 1 faults
Total 11 faults

61) Consider a simple system running a single process. The size of physical frames and logical pages is 16 bytes. The RAM can hold 3 physical frames. The virtual addresses of the process are 6 bits in size. The program generates the following 20 virtual address references as it runs on the CPU:
0, 1, 20, 2, 20, 21, 32, 31, 0, 60, 0, 0, 16, 1, 17, 18,32, 31, 0, 61
The number of page faults generated by the accesses above, assuming a FIFO page replacement algorithm
A) 6
B) 7
C) 8
D) 9

Solution : C) 8
For 6 bit virtual addresses, and 4 bit page offsets (page size 16 bytes), the most significant 2 bits of a virtual address will represent the page number. So the reference string is 0, 0, 1, 0,1, 1, 2, 1, 0, 3 (repeated again). Page faults with FIFO = 8. Page faults on 0,1,2,3 (replaced 0), 0 (replaced 1), 1 (replaced 2), 2 (replaced 3), 3.

62) Consider the following memory reference string :

1,2,3,4,5,3,4,1,6,7,8,7,8,9,7,8,9,5,4,5,4,2
Calculate the average memory access time if memory access time is 100ns and 3000ns page fault service time if on demand Optimal page replacement algorithm is used with a total physical memory of 4 frames.

A) 3000ns
B) 1250ns
C) 1420ns
D) 1550ns

Solution (D) The optimal page replacement for the above reference gives 11 page fault out of 22 references. Therefore, page fault rate, p is 50%.

EMAT = (1-p)*(M) + p*(PFST) = 0.5*100 + 0.5*3000 = 1550ns.

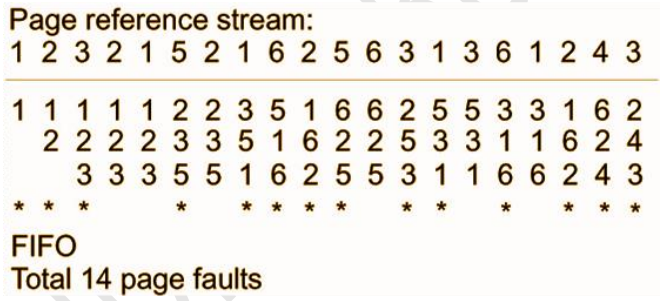
63) A certain page table entry in the page table of a process has both the valid and present bits set, now which of the following is TRUE in the context of TLB and memory access ?
A) A TLB hit occurs, always.
B) A page fault will occur.
C) A TLB hit or miss may occur, cannot say.
D) None of these

Solution : (C)
Valid and present bits in a page entry of the page table indicates that the page is valid and present in the main memory. Therefore, a page fault cannot occur. We cannot say a TLB hit or miss occurs, as the page entry may or may not be loaded into TLB.

64) Consider the following page references, with 3 frames and FIFO page replacement policy is used. Calculate the total number of page faults. **[NAT]**

1, 2, 3, 2, 1, 5, 2, 1, 6, 2, 5, 6, 3, 1, 3, 6, 1, 2, 4, 3

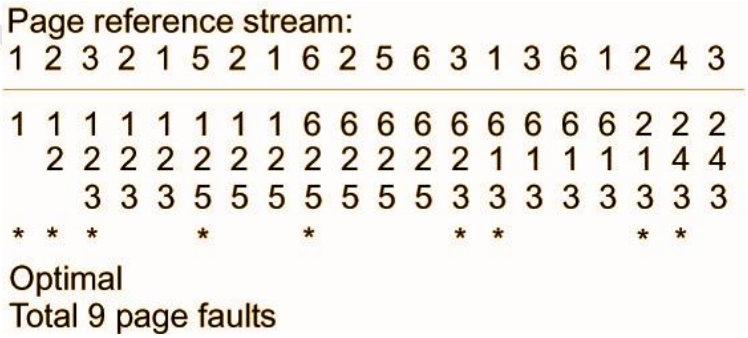
Solution: 14



65) Consider the following page references, with 3 frames and OPTIMAL page replacement policy is used. Calculate the total number of page faults. **[NAT]**

1, 2, 3, 2, 1, 5, 2, 1, 6, 2, 5, 6, 3, 1, 3, 6, 1, 2, 4, 3

Solution: 9

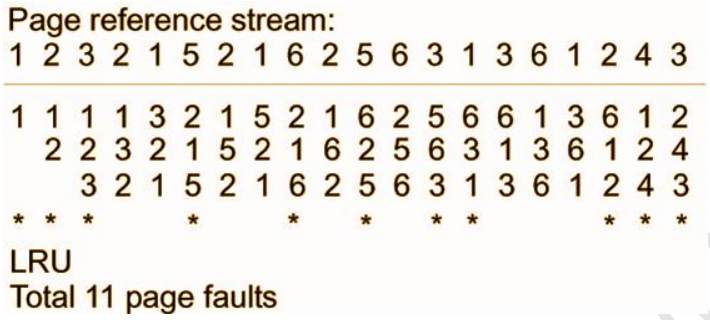


66) Consider the following page references, with 3 frames and LRU page replacement policy

is used. Calculate the total number of page faults. **[NAT]**

1, 2, 3, 2, 1, 5, 2, 1, 6, 2, 5, 6, 3, 1, 3, 6, 1, 2, 4, 3

Solution: 11



67) Consider the following code in which the integer is of size 4 bytes and the page size is 4KB, and on demand paging is used with 3 frames initially empty. The number of page faults generated by the code using a FIFO page replacement policy is:_____ **[NAT]**

```
int A[4000];

int main()
{
for(i=0;i<4000;i++){ //Row major order access
   A[i] = A[i]+1;
}
return 0;
}
```

Solution: 4
In the given code, we have 4000 integers each of size 4 bytes. Page size is 4KB. Therefore, a frame can hold 4KB/4 bytes = 1024 integers. Therefore,
Page 0: holds 1024 integers A[0-1023].
Page 1: holds 1024 integers A[1024-2047]
Page 2: holds 1024 integers A[2048-3071]
Page 4: holds the last 928 elements A[3072-3999]

The above three frames are initially empty, so it causes 3 page faults. Using FIFO page replacement the frame containing Page 0 is replaced to contain Page 4, hence one more page fault. Hence, a total of 4 page faults.

68) Consider the following sequence of page accesses:
A, B, D, C, B, A, B, A
If there are three frames of physical memory then the number of page faults using FIFO and LRU page replacement policies are ..... and ..... respectively.

A. 5,6
B. 6,5
C. 5,5
D. 6,6

Solution: (B)

	FIFO			LRU		
	F1	F2	F3	F1	F2	F3
A	A			A		
B		B			B	
D			D			D
C	C			C		
B						
A		A				A
B			B			
A						
# Page Faults	6			5		

69) The recent page references for three processes with a window size of 10 page references is given as follows:

Process	Reference window (Size =10)
P1	0, 1, 2, 0, 2, 4, 5, 1, 3, 4
P2	1, 3, 2, 5, 2, 4, 2, 1, 3, 1
P3	1, 3, 2, 0, 2, 4, 5, 1, 2, 0

Calculate the minimum number of frames required in the main memory to avoid “thrashing”

A) 15
B) 16
C) 17
D) 18

Solution: (C)
Working Set of (P1) = {0,1,2,3,4,5} => Requires 6 frames
Working Set of (P2) = {1,2,3,4,5} => Requires 5 frames
Working Set of (P3) = {0,1,2,3,4,5} => Requires 6 frames

To avoid thrashing, the sum of the working set for all the processes <= number of available frames (m).

Therefore, 6+5+6 <= m. => m>=17. So, the minimum is 17.

70) Which of the following statements is/are TRUE? **[MSQ]**

A) FIFO page replacement suffers from Belady's anomaly.
B) LRU page replacement suffers from Belady's anomaly.
C) Second Chance page replacement suffers from Belady's anomaly.
D) An algorithm suffers from Belady's Anomaly if and only if it does not follow stack property.

Solution: (A) (C) (D)
An algorithm suffers from Belady's Anomaly if and only if it does not follow stack property. This is because these algorithms assign priority to a page for replacement that is independent of the number of frames in the main memory.
The second chance algorithm (the clock algorithm) works just like FIFO, but it skips over any pages with the use bit set (and clears the use bit).

TLB and Memory Performance Equation

71) Consider a paging system that uses a 3-level page table residing in main memory and a TLB for address translation. Each main memory access takes 150 ns and TLB look up takes 10 ns. Assume that the TLB hit ratio is 90%, the average memory access time in ns (round off to 2 decimal places) is _____. **[NAT]**

Solution : 205.00 ns
EMAT = h*(t + M) + (1 - h)*(K*M + M + t)
EMAT=205.00 ns

72) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. The page fault service takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible.
The average memory access time in ns (round off to 2 decimal places) is _____ .
[NAT]

Solution : 149.50 ns
EMAT = h*(t+M) + (1-h)*(t + K*M + (1-P)*(M) + P*(PFST))
EMAT=149.50 ns

73) Consider a system with 8-bit virtual and physical addresses, and 16 byte pages. A

process in this system has 4 logical pages, which are mapped to 3 physical pages as follows:

Page table

Page	Frame
0	6
1	3
2	11
5	-

TLB

Index	Frame
0	6
2	11

Which of the following is/are TRUE ? **[MSQ]**

A) CPU access to virtual address 7 leads to a TLB hit.
B) CPU access to virtual address 20 leads to a TLB hit.
C) CPU access to virtual address 70 leads to a trap to OS.
D) CPU access to virtual address 80 leads to a page fault.

Solution: (A) (C) (D)
Virtual address 7 = 0000 (page number) + 0111 (offset) = logical page 0. TLB hit.
Virtual address 20 = 0001 0100 = logical page 1. TLB miss.
Virtual address 70 = 0100 0110 = logical page 4. TLB miss. MMU accesses the page table and discovers it is an invalid entry. MMU raises a trap to the OS.
Virtual address 80 = 0101 0000 = logical page 5. TLB miss. MMU accesses page table and discovers page not present. MMU raises a page fault to the OS.

74) Consider the following combinations and identify which of the following are possible? **[MSQ]**

A) TLB: Hit, Page: Hit, Cache: Hit
B) TLB: Hit, Page: Hit, Cache: Miss
C) TLB: Miss, Page: Miss, Cache: Hit
D) TLB: Miss, Page: Miss, Cache: Miss

Solution: (A) (B)

TLB	Page	Cache	Remarks
hit	hit	hit	Possible, but page table not checked on TLB hit, data from cache
hit	hit	miss	Possible, but page

			table not checked, cache entry loaded from memory
hit	miss	hit	Impossible, TLB references in-memory pages
hit	miss	miss	Impossible, TLB references in-memory pages

75) In a paging system with TLB, it takes 30 ns to search the TLB and 90 ns to access the memory. If the TLB hit ratio is 70%, find the effective memory access time (in ns).

A) 110 - 115
B) 145 - 150
C) 150 - 155
D) 55 - 60

Solution (B)
Effective memory access time (EAT) = P(H) × (Time to access TLB + Time to access the memory location) + [(1-P(H)) × (Time to access TLB + 2(Time to access the memory location))]
= 0.70 x 120 + 0.30 x 210
= 147 ns

76) A computer whose processes have 1024 pages in their address spaces keeps its page tables in memory. The overhead required for reading a word from the page table is 5 nsec. To reduce this overhead, the computer has a TLB, which holds 32 (virtual page, physical page frame) pairs, and can do a lookup in 1 nsec. What hit rate is needed to reduce the mean overhead to 2 nsec?

A)0.75-0.8
B)0.25-0.35
C)0.5 – 0.6
D)0.6-0.65

Solution (A)

The effective instruction time is , 2 = 1h + 6(1 – h), we find that h = 0.80

77) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Assume that the TLB hit ratio is 95%, the average memory access time in ns (round off to 2 decimal places) is _____.

Solution : EMAT = h*(t + M) + (1 - h)*(K*M + M + t)
EMAT=125.00 ns

78) Consider a paging system that uses 3-level page table residing in main memory and a TLB for address translation. Each main memory access takes 150 ns and TLB lookup takes 10 ns. Assume that the TLB hit ratio is 90%, the average memory access time in ns (round off to 2 decimal places) is _____.

Solution : EMAT = h*(t + M) + (1 - h)*(K*M + M + t)
EMAT=205.00 ns

79) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t + M) + (1 - h)*(t + K*M + (1-P)*(M) + P*(M+D))
EMAT=150.00 ns

80) Consider a paging system that uses 2-level page table residing in main memory and a TLB for address translation. Each main memory access takes 200 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 10000 ns. Assume that the TLB hit ratio is 99%, page fault rate is 1%. TLB update time is negligible.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t + M) + (1 - h)*(t + K*M + (1-P)*(M) + P*(M+D))
EMAT=225.00 ns

81) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. Assume that for 20% of the total page faults, a dirty page has to be written back to disk before the required page is read from disk. TLB update time is negligible.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+M) + (1-h)*(t + K*M + (1-P)*(M) + P*((1-d)*(M+D) + d*(M+2*D)))
EMAT=155.00 ns

82) Consider a paging system that uses 3-level page table residing in main memory and a TLB for address translation. Each main memory access takes 150 ns and TLB lookup takes 10 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 90%, page fault rate is 10%. Assume that for 15% of the total page faults, a dirty page has to be written back to disk before the required page is read from disk. TLB update time is negligible.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+M) + (1-h)*(t + K*M + (1-P)*(M) + P*((1-d)*(M+D) + d*(M+2*D)))
EMAT=262.50 ns

83) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. The page fault service takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+M) + (1-h)*(t + K*M + (1-P)*(M) + P*(PFST))
EMAT=149.50 ns

84) Consider a paging system that uses 2-level page table residing in main memory and a TLB for address translation. Each main memory access takes 200 ns and TLB lookup takes 20 ns. The page fault service takes 10000 ns. Assume that the TLB hit ratio is 99%, page fault rate is 1%. TLB update time is negligible.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+M) + (1-h)*(t + K*M + (1-P)*(M) + P*(PFST))
EMAT=224.98 ns

85) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%. Assume that the TLB hit ratio is 95%, and the page tables are not cached, the average memory access time in ns (round off to 2 decimal places) is _____.

Solution : EMAT = h*(t + T c) + (1-h)*(t + K*M + T c), where Tc=a*C + (1-a)*M
EMAT=102.50 ns

86) Consider a paging system that uses 2-level page table residing in main memory and a TLB for address translation. Each main memory access takes 150 ns and TLB lookup takes 10 ns. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 125 ns, with a cache hit rate of 95%.

Assume that the TLB hit ratio is 90%, and the page tables are not cached, the average memory access time in ns (round offto 2 decimal places) is _____.

Solution : EMAT = h*(t + T c) + (1-h)*(t + K*M + T c), where Tc=a*C + (1-a)*M
EMAT=166.25 ns

87) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%, and the page tables are not cached.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+Tc) + (1-h)*(t+K*M+(1-P)*Tc)+P*(M+D)), where Tc = a*C + (1-a)*M
EMAT=127.50 ns

88) Consider a paging system that uses 3-level page table residing in main memory and a TLB for address translation. Each main memory access takes 150 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 10000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 125 ns, with a cache hit rate of 95%, and the page tables are not cached. The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+Tc) + (1-h)*(t+K*M+(1-P)*Tc)+P*(M+D)), where Tc = a*C + (1-a)*M
EMAT=218.75 ns

89) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. Assume that for 20% of the total page faults, a dirty page has to be written back to disk before the required page is read from disk. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%, and the page tables are not cached.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution :
EMAT = EMAT = h*(t+Tc) + (1-h)*(t+k*M + (1-p)*Tc + P((1-d)*(M+D)+d*(M+2D))), where Tc = a*C + (1-a)*M
EMAT=132.50 ns 132.6125

90) Consider a paging system that uses 3-level page table residing in main memory and a TLB for address translation. Each main memory access takes 150 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 10000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. Assume that for 20% of the total page faults, a dirty page has to be written back to disk before the required page is read from disk.TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 125 ns, with a cache hit rate of 95%, and

the page tables are not cached. The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution :
EMAT = EMAT = h*(t+Tc) + (1-h)*(t+k*M + (1-p)*Tc + P((1-d)*(M+D)+d*(M+2D))), where Tc = a*C + (1-a)*M
EMAT=228.75 ns

91) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. The page fault service takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%, and the page tables are not cached.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+Tc) + (1-h)*(t+K*M+(1-P)*Tc)+P*(PFST)), where Tc = a*C + (1-a)*M
EMAT=127.11 ns

92) Consider a paging system that uses 2-level page table residing in main memory and a TLB for address translation. Each main memory access takes 200 ns and TLB lookup takes 20 ns. The page fault service takes 10000 ns. Assume that the TLB hit ratio is 99%, page fault rate is 1%. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 150 ns, with a cache hit rate of 95%, and the page tables are not cached.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution :EMAT = h*(t+Tc) + (1-h)*(t+K*M+(1-P)*Tc)+P*(PFST)), where Tc = a*C + (1-a)*M
EMAT=186.35 ns

93) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%. Assume that the TLB hit ratio is 95%, and the page tables are also physically cached, the average memory access time in ns (round off to 2 decimal places) is _____.

EMAT = h*(t + Tc) + (1 - h)*(K*Tc + Tc + t), where Tc = a*C + (1-a)*M
EMAT=101.38 ns

94) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%, and the page tables are also physically cached..
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t + Tc) + (1 - h)*(t + K*Tc + (1-P)*(Tc) + P*(M+D)), where Tc = a*C + (1-a)*M
EMAT=126.38 ns

95) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. Assume that for 20% of the total page faults, a dirty page has to be written back to disk before the required page is read from disk. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%, and the page tables are also physically cached..
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution :
EMAT = h*(t+Tc) + (1-h)*(t + K*Tc + (1-P)*(Tc) + P*((1-d)*(M+D) + d*(M+2*D))), where Tc = a*C + (1-a)*M
EMAT=131.38 ns

96) Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. The page fault service takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. TLB update time is negligible. To make the memory access faster, a physical level-1 cache is introduced, with no write back mechanism. The cache access time is 75 ns, with a cache hit rate of 90%, and the page tables are also physically cached.
The average memory access time in ns (round off to 2 decimal places) is _____ .

Solution : EMAT = h*(t+Tc) + (1-h)*(t + K*Tc + (1-P)*(Tc) + P*(PFST)), where Tc = a*C + (1-a)*M
EMAT=125.99 ns