Computer Science & Engineering Department, IIT Kharagpur
CS31202 Operating Systems
Spring 2024-2025

Class Test 2

12th April, 2025, Time: 5:00pm–6:00pm Maximum marks: 20

Roll no: ______________________ **Name:** ____

[Write your answers in the question paper itself. Be brief and precise. Answer all questions.]

1. Consider a memory-management unit implementing 46-bit virtual address, 32-bit physical address, and a three-level paged page-table organization. The page table base register stores the base address of the outer page table (PT1), which occupies exactly one page. Each entry of PT1 stores the base address of a page of the second-level page table (PT2). Each entry of PT2 stores the base address of a page of the third-level page table (PT3). Each entry of PT1, PT2 and PT3 is of size 4 bytes, and composed of a frame number, one valid/invalid bit, one dirty bit, and a few protection bits.
(i) What is the page size in KB in this memory-management unit? (ii) How many bits are available for storing protection in each page-table entry? (iii) Give a diagram showing how a virtual address gets translated to a physical address for this memory-management unit.

[3+2+1]

Page size P bytes
Each page table entry is 4 bytes
Number of entries per page table P/4
Offset bits Log(P)
Length of each page table index n bits
Hence 2^n=P/4 =>P=2^n*4
Total bits in virtual address = 46 = (bits for PT1 index) + (bits for PT2 index) + (bits for PT3 index) + offset
Since each page table index contains n bits
46=3n+Log(P)
Log(P)=46-3n
P=2^(46-3n)
2^n*4=P=2^(46-3n)
n=11
(i) Page size=2^11*4=2^13=8K
(ii) number of frames=2^32/2^13=2^19
19 bits for frame number.
Each PT entry: 32 bits
Protection bits=x
32=19+1+1+x
x=11 bits

2. Consider a demand paged system. Page tables are stored in main memory which has an access time of 100 nanoseconds. The TLB can hold 8 page-table entries, and has an access time of 10 nanoseconds. During the execution of a process, it is found that 80% of the time, a required page-table entry exists in TLB, and only 2% of the **total** memory references cause page faults. The average time to service page fault is 2 milliseconds. Compute the effective memory access time. Assume that at any point of time, all TLB entries refer to valid pages.

[2]

TLB hit does not result in any page fault.

0.80(10+110)+0.2(.9(10+100+100)+.1(10+100+2000000+100)= 8138

80% time there will be a TLB hit so it will take (10+100) ns.

20% of the time two things can happen:

X% of the time page fault will occur that is it will take (10+100+2000000+100) ns.

(1-X)% of the time no page fault so it will take (10+100+100) ns.

Page fault occurs 2% of the total memory accesses. So X% of 0.2 (TLB miss) should be equal to .02 (i.e 2%)

3(a) Consider a demand paged system with 16-bit virtual addresses and 256-byte pages, and has 1KB of main memory. LRU page replacement is implemented using a stack, whose current status (page numbers are in decimal) is [63, 1, 17, _], where page 63 points to the top of the stack (which stores the most recently used page number). Suppose that the following sequence of virtual addresses is subsequently generated: 0x00FF, 0x010D, 0x10FF, 0x11B0.

After each generated address, show (i) the new state of the stack, (ii) whether page fault occurs, (iii) page replacements (if there is a page fault).

[2+2+2]

Page size is 256 bytes
RAM: 2^10
Number of frames=2^10/256=4
One frame is empty
16 bit virtual address
8 bit for page#, 8 bit for offset.

First address: 0x00FF (page #0), 01DD (page# 1), 0x10FF (page 16), 0x11B0 (page 17)

[0, 63, 1, 17] → PF
[1, 0, 63, 17] → no PF
[16, 1, 0, 63] → PF (replace 17)
[17, 16, 1, 0] → PF (replace 63)

(b) A memory-management unit implements the FIFO policy for page replacement. A process has 4 frames with no pages loaded to begin with. The process first accesses 100 distinct pages in some order and then accesses the same 100 pages but now in the reverse order. How many page faults will occur in total? Justify.

[2]

Frame size=4
First access=100 pf
Second access=96 pf
Total=196

4. Consider a handheld system with a main memory of size 1MB. A customized operating system of size 130 KB has been loaded in the lower side of the memory. The contiguous memory allocation (variable partition) scheme is used with the first-fit strategy. The processes arrive and terminate in the following sequence: P1 (250KB) arrives, P2 (100KB) arrives, P3 (150KB) arrives, P2 terminates, P4 (220KB) arrives. All the processes are loaded from the lowest address of the allocated hole.

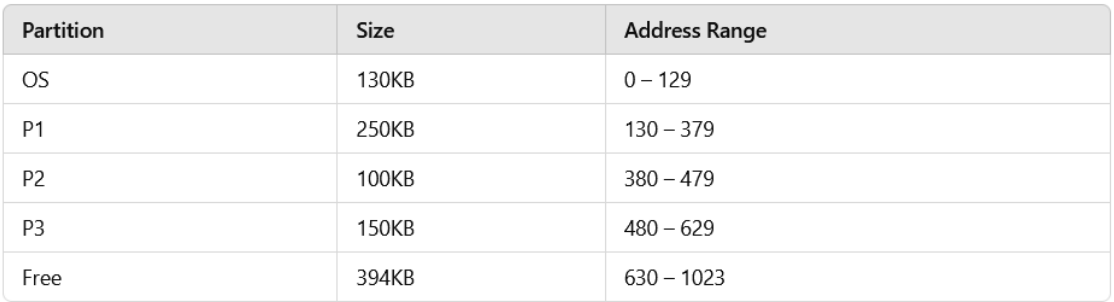
(i) Show the state of the memory after each process arrival and termination.

(ii) Now if process P5 of size 200KB arrives in the system, would it be possible to satisfy the request immediately? Give reason.

(iii) Suppose that the OS can migrate an existing process from its current location to a new hole. With this strategy in place, can the OS accommodate P5 with minimum cost? If yes, show the state of the memory and the corresponding cost.

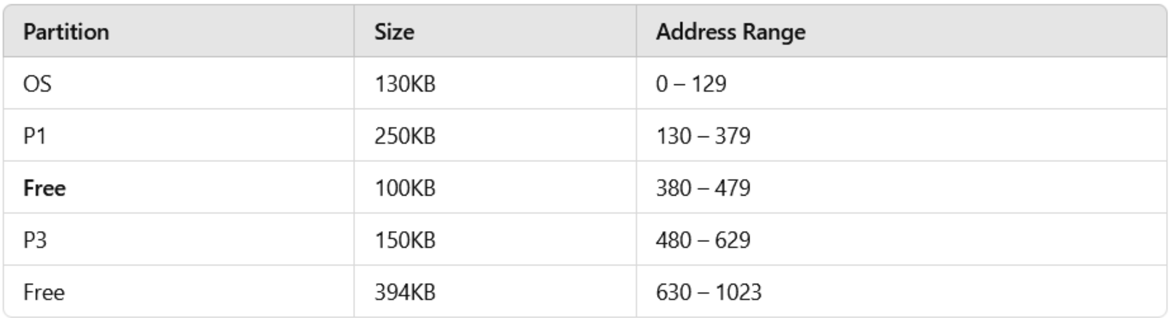
[2+1+1]

After P1, P2, P3 arrives



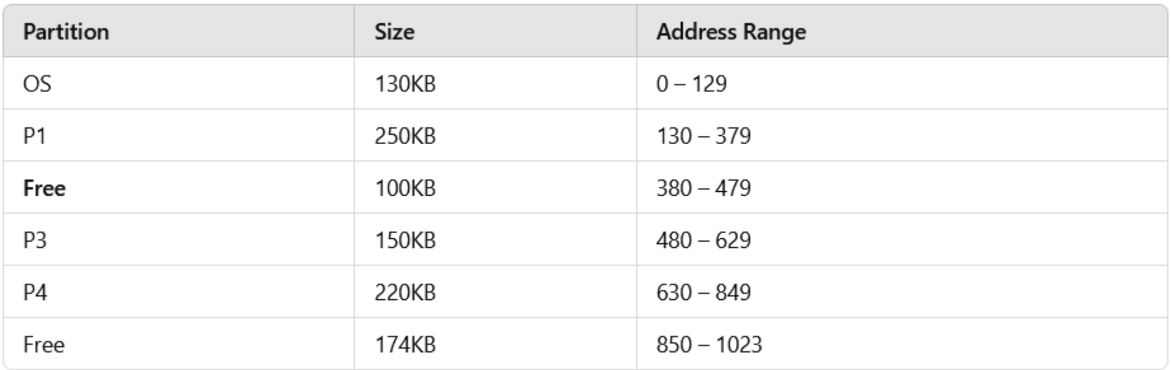
Partition	Size	Address Range
OS	130KB	0 – 129
P1	250KB	130 – 379
P2	100KB	380 – 479
P3	150KB	480 – 629
Free	394KB	630 – 1023

P2 terminates



Partition	Size	Address Range
OS	130KB	0 – 129
P1	250KB	130 – 379
Free	100KB	380 – 479
P3	150KB	480 – 629
Free	394KB	630 – 1023

P4 arrives



Partition	Size	Address Range
OS	130KB	0 – 129
P1	250KB	130 – 379
Free	100KB	380 – 479
P3	150KB	480 – 629
P4	220KB	630 – 849
Free	174KB	850 – 1023

No, P5 (200KB) cannot be loaded. not enough contiguous space due to fragmentation

Move P3 (150KB) to the last hole (174KB).

Cost is 150KB