$64 = 2^6 \times 4$

CS2302 Operating System Exercises Spring 2022

Exercise 4

Due date: Apr. 25, 2022

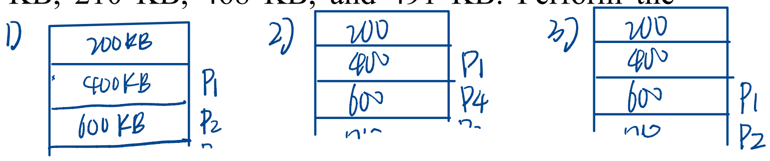
$256MB = 2^{28}B$ $\frac{2^{28}}{2^x}$ 页表项

1. Consider a single level paging scheme. The virtual address space is 256 MB and page table entry size is 4 bytes. What is the minimum page size, such that the entire page table fits well in one page?

   x

   page size $= 2^{15}$ $\therefore \frac{2^{28}}{2^x} \cdot 4 = 2^x$ $\therefore 2^{30} = 2^{2x}$ $x = 15$

2. Consider six memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB, and 250 KB. These partitions can be allocated to four sequentially arrived processes with sizes of 357 KB (P_1), 210 KB (P_2), 468 KB (P_3), and 491 KB (P_4). Perform the allocation of processes using
   1) First Fit Algorithm
   2) Best Fit Algorithm
   3) Worst Fit Algorithm



3. Consider a system using multilevel paging scheme. The page size is 1 MB. The memory is byte addressable and virtual address is 64 bits long. The page table entry size is 4 bytes.
   1) How many levels of page table will be required?
   2) Please give the structures of physical address and virtual address.

4. A system uses 3 page frames for storing process pages in main memory. It uses the
   1) First in First out (FIFO) page replacement policy 0.4
   2) Least Recently Used (LRU) page replacement policy 0.4.
   3) Optimal page replacement policy 0.5

   Assume that all the page frames are initially empty. What is the total number of page faults that will occur while processing the page reference string given below?

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

   Please also calculate the hit ratio and miss ratio.

   2级页表要访问2次页表. TLB → outer page table → inner page table → memory.

5. Consider a two level paging scheme with a TLB. Assume no page fault occurs. It takes 20 ns to search the TLB and 100 ns to access the physical memory. If TLB hit ratio is 80%, what is the effective memory access time?

   hit: 20+100 miss: 20+100+100+100 $0.8 \times 120 + 320 \times 0.2 = 160$ ns

6. Consider a disk queue with requests for I/O to blocks on cylinders 23, 89, 132, 42, 187. The head is initially at cylinder number 100. Assume we are going inwards (i.e., towards 0). The cylinders are numbered from 0 to 199. Calculate the total head movement (in number of cylinders) incurred while servicing these requests. Please apply FCFS, SSTF, SCAN, C-SCAN, and C-LOOK, respectively.

Exercise 4 Solutions

1.

Number of pages the process is divided

= Process size / Page size

= 256 MB / B bytes

= 228 / B

Page table size

= Number of entries in the page table x Page table entry size

= Number of pages the process is divided x Page table entry size

= (228 / B) x 4 bytes

= (230 / B) bytes

Now,

According to the above condition, we must have-

(230 / B) bytes <= B bytes

B2 >= 230

B >= 215

Thus, minimum page size possible = 2^15 bytes or 32 KB.

2.

In First Fit Algorithm,

	P1				
200KB	400KB	600KB	500KB	300KB	250KB

	P1	P2			
200KB	400KB	600KB	500KB	300KB	250KB

	P1	P2	P3		
200KB	400KB	600KB	500KB	300KB	250KB

P4 can not be allocated the memory.

In Best Fit Algorithm,

	P1				
200KB	400KB	600KB	500KB	300KB	250KB

	P1				P2
200KB	400KB	600KB	500KB	300KB	250KB

	P1		P3		P2
200KB	400KB	600KB	500KB	300KB	250KB

	P1	P4	P3		P2
200KB	400KB	600KB	500KB	300KB	250KB

In Worst Fit Algorithm,

		P1			
200KB	400KB	600KB	500KB	300KB	250KB

		P1	P2		
200KB	400KB	600KB	500KB	300KB	250KB

P3 and P4 can not be allocated the memory.

3.

Given: Virtual Address = 64 bits, Page size = 1 MB, Page table entry size = 4 bytes

Number of Bits in Frame Number: Page table entry size= 4 bytes= 32 bits

Thus, Number of bits in frame number = 32 bits

Number of Frames in Main Memory: We have, Number of bits in frame number = 32 bits

Thus, Number of frames in main memory = 232 frames 2^{32} frames

$2^{32} \times 1MB$

Size of Main Memory: Size of main memory = Total number of frames x Frame size = 232 x 1 MB

|| page size

= 252 B $= 2^{52} B$

Thus, Number of bits in physical address = 52 bits

Number of Bits in Page Offset: Page size = 1 MB = 220 B $2^{20} B$

Thus, Number of bits in page offset = 20 bits

Alternatively,

Number of bits in page offset = Number of bits in physical address – Number of bits in frame number = 52 bits – 32 bits = 20 bits

Process Size: Number of bits in virtual address = 64 bits

Thus, Process size = 264 bytes $2^{64} B$

Number of Pages of Process: Number of pages the process is divided= Process size / Page size

= 264 B / 1 MB = 264 B / 220 B = 244 pages $= 2^{44}$ pages.

Inner Page Table Size: Inner page table keeps track of the frames storing the pages of process.

Inner page table size= Number of entries in inner page table x Page table entry size

= Number of pages the process is divided x Page table entry size= 244 x 4 bytes

= 246 bytes

Now, we can observe: so Inner page table size $= 2^{20}$.

The size of inner page table is greater than the frame size (1 MB). there are $\frac{2^{46}}{2^{20}} = 2^{26}$ inner page table

Thus, inner page table can not be stored in a single frame.

So, inner page table has to be divided into pages.

Number of Pages of Inner Page Table: Number of pages the inner page table is divided = Inner page table size / Page size = 246 B / 1 MB = 246 B / 220 B = 226 pages

Now, these 226 pages of inner page table are stored in different frames of the main memory.

Number of Page Table Entries in One Page of Inner Page Table:

Number of page table entries in one page of inner page table = Page size / Page table entry size

= 1 MB / 4 B = 220 B / 22 B = 218 entries 2^{18}

Number of Bits Required to Search an Entry in One Page of Inner Page Table:

One page of inner page table contains 218 entries. 2^{18}

4.

1) FIFO: number of page faults = 6; hit ratio = 0.4; miss ratio = 0.6;

2) LRU: number of page faults = 6; hit ratio = 0.4; miss ratio = 0.6;

3) OPT: number of page faults = 5; hit ratio = 0.5; miss ratio = 0.5;

5.

Substituting values in the above formula, we get

Effective Access Time

= 0.8 x { 20 ns + 100 ns } + 0.2 x { 20 ns + (2+1) x 100 ns }

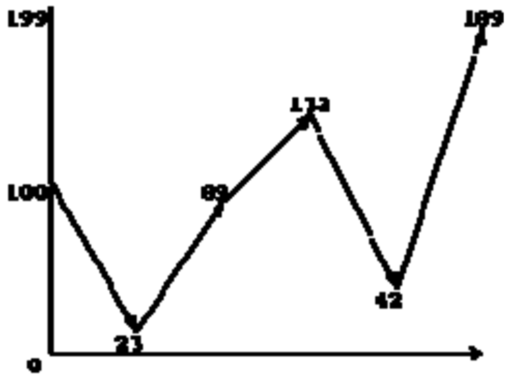
= 0.8 x 120 ns + 0.2 + 320 ns

= 96 ns + 64 ns

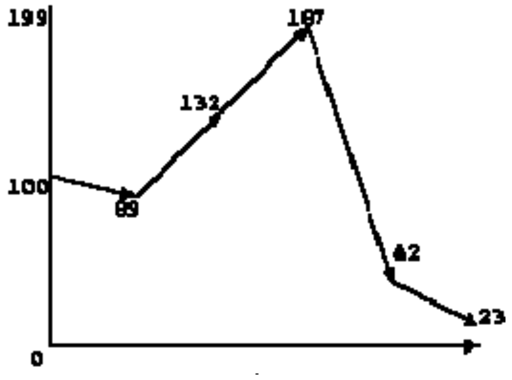
= 160 ns

Thus, effective memory access time = 160 ns.

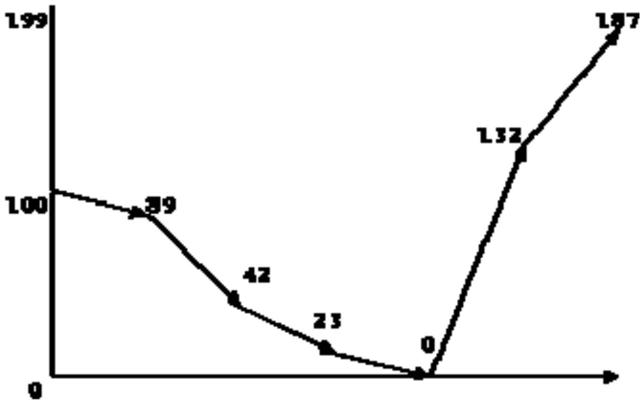
6. FCFS: 77+66+43+90+145=421



SSTF: 11+43+55+45+19=273

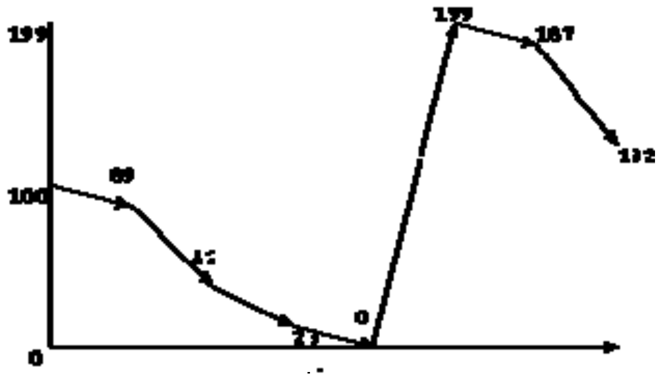


SCAN: 11+47+19+23+132+55=287



100 23 89 132 42 187

C-SCAN:11+47+19+23+199+12+55=366



C-LOOK: 11+47+19+164+55=296