Exercise 4 Solutions

1. Number of pages the process is divided
= Process size / Page size
= 256 MB / B bytes
= 2^28 / 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
= (2^28 / B) x 4 bytes
= (2^30 / B) bytes

Now,
According to the above condition, we must have-
(2^30 / B) bytes <= B bytes
B^2 >= 2^30
B >= 2^15
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

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

= 252 B

Thus, Number of bits in physical address = 52 bits

Number of Bits in Page Offset: Page size = 1 MB = 220 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

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

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:

The size of inner page table is greater than the frame size (1 MB).

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

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

One page of inner page table contains 218 entries.

Thus, Number of bits required to search a particular entry in one page of inner page table = 18 bits.

Outer Page Table-1 Size: Outer page table-1 is required to keep track of the frames storing the pages of inner page table.

Outer page table-1 size = Number of entries in outer page table-1 x Page table entry size

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

= 226 x 4 bytes = 228 bytes = 256 MB

Now, we can observe:

The size of outer page table-1 is greater than the frame size (1 MB).

Thus, outer page table-1 can not be stored in a single frame.

So, outer page table-1 has to be divided into pages.

Number of Pages of Outer Page Table-1

Number of pages the outer page table-1 is divided = Outer page table-1 size / Page size = 256 MB / 1 MB = 256 pages

Now, these 256 pages of outer page table-1 are stored in different frames of the main memory.

Number of Page Table Entries in One Page of Outer Page Table-1

Number of page table entries in one page of outer page table-1= Page size / Page table entry size = 1 MB / 4 B = 220 B / 22 B = 218 entries

Number of Bits Required to Search an Entry in One Page of Outer Page Table-1

One page of outer page table-1 contains 218 entries. Thus, Number of bits required to search a particular entry in one page of outer page table-1 = 18 bits

Outer Page Table-2 Size:

Outer page table-2 is required to keep track of the frames storing the pages of outer page table-1.

Outer page table-2 size = Number of entries in outer page table-2 x Page table entry size

= Number of pages the outer page table-1 is divided x Page table entry size

= 256 x 4 bytes = 1 KB

Now, we can observe:

The size of outer page table-2 is less than the frame size (16 KB).

Thus, outer page table-2 can be stored in a single frame.

In fact, outer page table-2 will not completely occupy one frame and some space will remain vacant.

So, for given system, we will have three levels of page table.

Page Table Base Register (PTBR) will store the base address of the outer page table-2.

Number of Bits Required to Search an Entry in Outer Page Table-2

Outer page table-2 contains 256 = 28 entries.

Thus, Number of bits required to search a particular entry in outer page table-2 = 8 bits

Virtual address: | 8bit | 18bit | 18bit | 20bit |

physical address: | 32bit | 20bit |

4. FIFO: 6 page faults; hit ratio: 0.4; miss ratio:0.6

LRU: 6 page faults; hit ratio: 0.4; miss ratio:0.6

Optimal: 5 page faults; hit ratio: 0.5; miss ratio:0.5

5. It is a two level paging scheme. Thus,

Effective Access Time

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

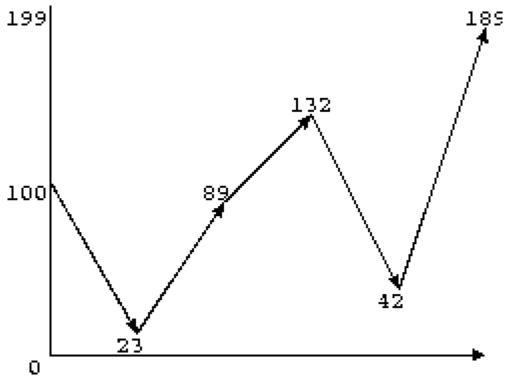
= 0.8 x 120 ns + 0.2 + 320 ns

= 96 ns + 64 ns

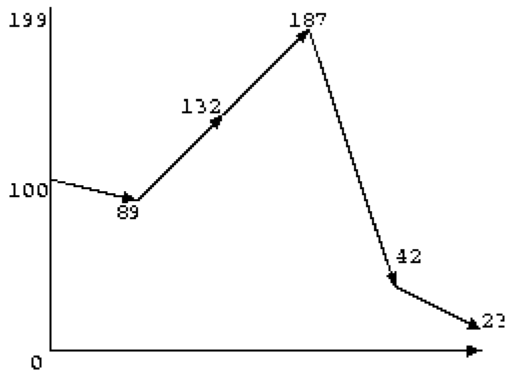
= 160 ns

Thus, effective memory access time = 160 ns.

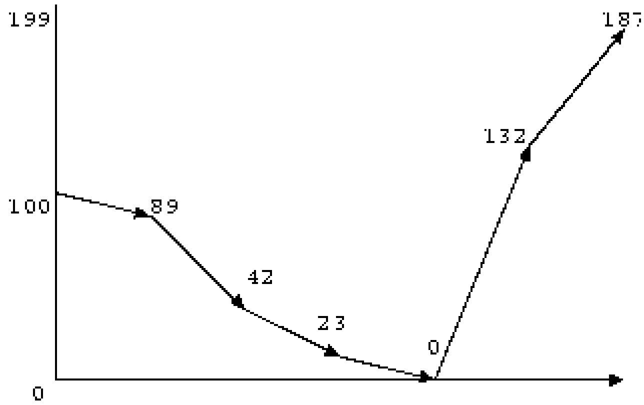
6. FCFS: 77+66+43+90+145=421



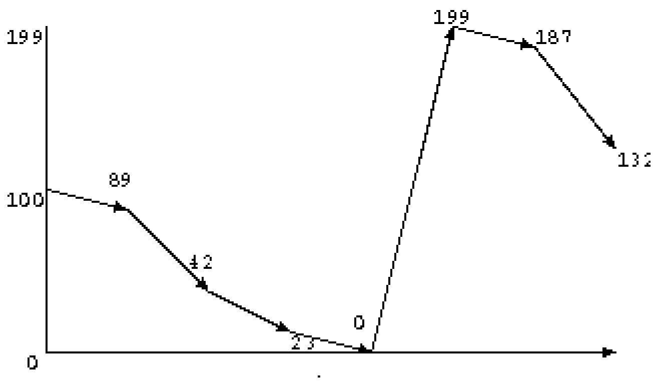
SSTF: 11+43+55+45+19=273



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



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



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