

2)

### First Come First Serve

| Process      | P1     | P2 | P3 | P4 | P5 |
|--------------|--------|----|----|----|----|
| Burst Time   | 15     | 3  | 9  | 6  | 4  |
| Time Elapsed | 15     | 18 | 27 | 33 | 37 |
| Wait Time    | 0      | 15 | 18 | 27 | 33 |
| AWT          | 18.6ms |    |    |    |    |

### Non-Pre-emptive Shortest Job First

| Process      | P2  | P5 | P4 | P3 | P1 |
|--------------|-----|----|----|----|----|
| Burst Time   | 3   | 4  | 6  | 9  | 15 |
| Time Elapsed | 3   | 7  | 13 | 22 | 37 |
| Wait Time    | 0   | 3  | 7  | 13 | 22 |
| AWT          | 9ms |    |    |    |    |

### Non-Pre-emptive Priority Scheduling

| Process      | P1     | P4 | P5 | P2 | P3 |
|--------------|--------|----|----|----|----|
| Priority     | 5      | 4  | 3  | 2  | 1  |
| Burst Time   | 15     | 6  | 4  | 3  | 9  |
| Time Elapsed | 15     | 21 | 25 | 28 | 37 |
| Wait         | 0      | 15 | 21 | 25 | 28 |
| AWT          | 17.8ms |    |    |    |    |

### Round Robin (5ms Time Quantum)

| Process | P1 (10) | P2 (0) | P3 (4) | P4(1) | P5(0) | P1(5) | P3(0) | P4(0) | P1(0) |
|---------|---------|--------|--------|-------|-------|-------|-------|-------|-------|
| Time E  | 5       | 8      | 13     | 18    | 22    | 27    | 31    | 32    | 37    |
| Wait    | 0       | 5      | 8      | 13    | 18    | 17    | 14    | 13    | 5     |
| AWT     | 18.6ms  |        |        |       |       |       |       |       |       |

3)

| Page | Loaded | Last Ref | R |
|------|--------|----------|---|
| 0    | 140    | 270      | 0 |
| 1    | 110    | 285      | 1 |
| 2    | 126    | 280      | 1 |
| 3    | 230    | 265      | 0 |

### Second Chance

(Queue)

| Page 1 | Page 2 | Page 0 | Page 3 |
|--------|--------|--------|--------|
| 1      | 1      | 0      | 0      |

| Page 3 | Page 1 | Page 2 | Page 4 |
|--------|--------|--------|--------|
| 0      | 0      | 0      | 1      |

-The page where the reference bit not set and has been loaded least recently will be removed.  
(Page 0)

### Least Recently Used

-The page that has been referenced the longest time ago is removed. (Page 3)

### Clock

-Same as second chance. The page with reference bit that is not set and has been loaded least recently will be removed (Page 0)

4)

Given that main memory is 64K, a page/frame size is 4K and that a process's virtual memory size is 128K. (i) How many entries would a page table in this system contain? (ii) What is the size of the virtual address and a real memory address in this system? (iii) Explain how the memory management unit converts a virtual address into an address in real memory on this system.

(b) In a virtual memory management system, under what conditions are inverted page tables needed? How is such a table used?

a)

i) No. of pages =  $2^{10+7} / 2^{10(12)} = 2^{17-12} = 5$  pages  
Entries:  $2^5$

ii) Virtual Memory

Total: 17 bits

|                |                  |
|----------------|------------------|
| Page# = 5 bits | Offset = 12 bits |
|----------------|------------------|

Main Memory

Total: 16 bits

|                 |                  |
|-----------------|------------------|
| Frame# = 4 bits | Offset = 12 bits |
|-----------------|------------------|

iii) The processor would find the memory frame corresponding to the specified virtual memory page in the page table. The frame offset would be the same as the page offset.

b) Most operating systems use one page table per process in a multiprocessing environment. With a process in virtual memory, as the size of the process increases, the size of its page table also increases. A considerable amount of space can be occupied by just the page table. An inverted page table reduces the overhead of having such a big table by having each frame in main memory be represented by one page table entry. A single page table is therefore used for all processes and the number of entries in the page table is equal to the number of frames in main memory.