# Virtual Memory

Vo Hieu Nghia

1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FIFO | | | | | | | | | | | | |
| time | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 | t11 |
| VM page | 3 | 4 | 6 | 10 | 3 | 5 | 3 | 5 | 4 | 5 | 3 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Main Page 0 | 3 | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 5 | 5 |
| Main Page 1 |  | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 3 |
| Main Page 2 |  |  | 6 | 6 | 6 | 6 | 6 | 6 | 4 | 4 | 4 |
| Main Page 3 |  |  |  | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

* First in first out: Which page has been loaded in first will be replaced when there is page fault

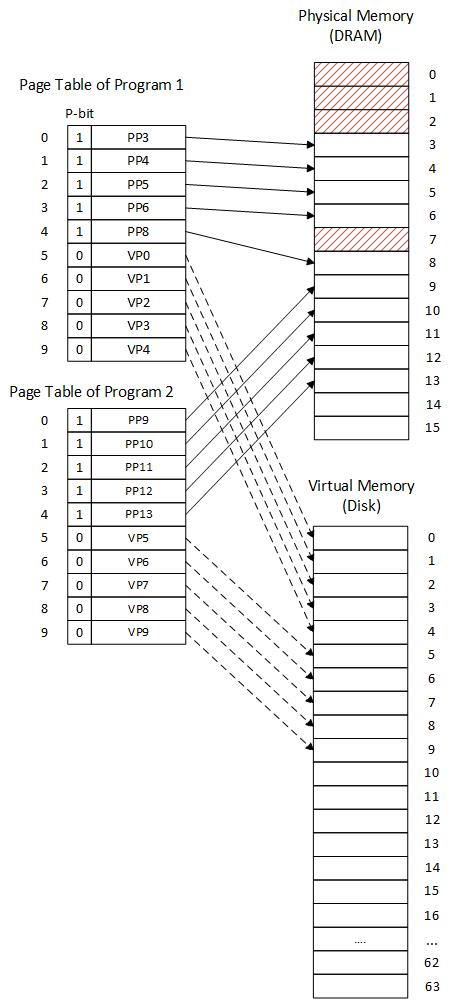
- The red marks are page faults, so the number of page fault are 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LRU | | | | | | | | | | | | |
| time | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 | t11 |
| VM page | 3 | 4 | 6 | 10 | 3 | 5 | 3 | 5 | 4 | 5 | 3 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Main Page 0 | 3 (0) | 3 (1) | 3 (2) | 3 (3) | 3(0) | 3 (1) | 3 (0) | 3 (1) | 3 (2) | 3 (3) | 3 (0) |
| Main Page 1 |  | 4 (0) | 4 (1) | 4 (2) | 4(3) | 5 (0) | 5 (1) | 5 (0) | 5 (1) | 5 (0) | 5 (1) |
| Main Page 2 |  |  | 6 (0) | 6 (1) | 6(2) | 6 (3) | 6 (4) | 6 (5) | 4 (0) | 4 (1) | 4 (2) |
| Main Page 3 |  |  |  | 10 (0) | 10(1) | 10 (2) | 10 (3) | 10 (4) | 10 (5) | 10 (6) | 10 (7) |

Least recently used:

* The numbers outside the bracket () are the virtual page which is loaded into the memory
* The numbers inside the bracket () are the reference number of Least recently used.
* If the reference of the virtual memory page is called, then the reference of that LRU will be return to 0, else it will be raised by 1
* Therefore, the number of pages fault in this case is 6

2)



3)The worst case:

- The information that 2 loading instructions is on 2 different pages, which has not been loaded to the main memory

- CPU cannot find the page table entry 🡪 Access table page

- MMU cannot find the page which have the information needed to be load in the memory 🡪 load from hard disk

- When it loads page from hard disk, the memory is full already 🡪 do page replacement

🡺 The execution time of program mostly depend on the time access I/O hardware to hard disk (because it has to access hard disk 2 times for 2 loading instructions) and the time to access hard disk is much longer than the time loading from memory