|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative | | | | |
|  |  |  | |  |
| Page replacement algorithm with beladys anomaly | | |
|  | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative | | | | |
|  |  |  | |  | |
|  | TEAM:   1. DAKSH CHORDIYA – 18BCP014 2. HEEM AMIN - 18BCP034 3. MESHVA PATEL – 18BCP063 4. SANYAM RAINA – 18BCP100 5. DRASHTI KHENI – 18BCP145D | |  | |

|  |  |
| --- | --- |
|  |  |
| FIFO ALGORITHMEXAMPLE 1: Page reference string: 1, 3, 0, 3, 5, 6 ,3  Page frames: 3 | |
| **Explanation:**   1. Initially all slots are empty, so when P1(1), P2(3), P3(0) came they are allocated to the empty slots —> 3 Page Faults. 2. When P4(3) comes in, it is already in the memory so —> 0 Page Faults - Page Hit. 3. Then P5(5) enters, which is not available in the memory so it replaces the oldest page slot i. e, P1(1) —>1 Page Fault. 4. P6(6) enters, that is not available in memory so it replaces the oldest page slot i.e., P2(3) —>1 Page Fault. 5. Finally, when 3 comes in, which is not available presently in the page slots, therefore replacing P3(0) —> 1-page fault   Hence, the Total Page Faults: 6; Page Hits: 1; Hit Ratio: 1/7   EXAMPLE 2: Page reference string: 1,2,3,4,1,2,5,1,2,3,4,5  Page frames: 4      Hence, the graph clearly points out that there are more page faults when there is increase in the number of page frames. | |  |
|  | |  |

|  |  |  |
| --- | --- | --- |
|  |  | |
| LRU ALGORITHMEXAMPLE 1: Page reference string: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 ,3  Page frames: 4    **Explanation:**   1. Initially all slots are empty, so when P1(7), P2(0), P3(1), P4(2) are allocated to the empty slots —> 4 Page faults. 2. P5(0) is available in the memory so —> 0 Page fault - 1 Page Hit. 3. when P6(3) enters it would take the place of P1(7) as it is least recently used —>1 Page fault. 4. P7(0) is already in the memory so —> 0 Page fault - 1 Page Hit. 5. P8(4) will occupy the page slot of P3(1) —> 1 Page Fault. 6. Now for the further page reference string —> 0 Page faults - 3 Page Hits because they are already available in the memory.   Hence total Page Faults: 6; Page Hits: 8; Hit Ratio: 6/14 EXAMPLE 2: Page reference string: 5,7,6,0,7,1,7,2,0,1,7,1,0  Page frames: 3      Hence the graph points out that even after increasing the page frames the LRU algorithm doesn’t suffer from Belady’s Anomaly.  This occurs as LRU is a stacking algorithm, and using k frames will always be a subset of k + n frames for LRU. Thus, any page-faults that may occur for k + n frames will also occur for k frames, which in turn means that LRU doesn't suffer Belady's anomaly. | | |
| |  |  | | --- | --- | |  |  | | optimal page replacementEXAMPLE 1: Page reference string: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 ,3  Page frames: 4    **Explanation:**   1. Initially all slots are empty, so when P1(7), P2(0), P3(1), P4(2) are allocated to the empty slots —> 4 Page faults 2. P5(0) is there in the memory so —> 0 Page fault - 1 Page Hit. 3. when P6(3) enters it will take the place of P1(7) because it is not used for the longest duration of time in the future. —>1 Page fault. 4. P7(0) is there in the memory so —> 0 Page fault - 1 Page Hit. 5. P8(4) will take the place of P3(1) —> 1 Page Fault. 6. Now for the further page reference string —> 0 Page fault, 6 Page Hit, because they are already available in the memory.   Hence total Page Faults: 6; Page Hits: 8; Hit Ratio: 6/14 EXAMPLE 2: Page reference string: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2  Page frames: 3      Hence the graph points out that even after increasing the page frames the OPR algorithm doesn’t suffer from Belady’s Anomaly.   |  |  | | --- | --- | |  |  | | conclusion The page replacement algorithm decides which memory page is to be replaced. The process of replacement is sometimes called swap out or write to disk. Page replacement is done when the requested page is not found in the main memory (page fault).  It is very important to have the optimal frame allocation and page replacement algorithm. Frame allocation is all about how many frames are to be allocated to the process while the page replacement is all about determining the page number which needs to be replaced in order to make space for the requested page.  There are various page replacement algorithms. Each algorithm has a different method by which the pages can be replaced.   1. Optimal Page Replacement algorithm → this algorithm replaces the page which will not be referred for so long in future. Although it cannot be practically implementable but it can be used as a benchmark. Other algorithms are compared to this in terms of optimality. 2. Least recent used (LRU) page replacement algorithm → this algorithm replaces the page which has not been referred for a long time. This algorithm is just opposite to the optimal page replacement algorithm. In this, we look at the past instead of staring at future. 3. FIFO → in this algorithm, a queue is maintained. The page which is assigned the frame first will be replaced first. In other words, the page which resides at the rare end of the queue will be replaced on every page fault.   In the case of LRU and optimal page replacement algorithms, it is seen that the number of page faults will be reduced if we increase the number of frames. However, Belady found that, In FIFO page replacement algorithm, the number of page faults will get increased with the increment in number of frames.  This is the strange behavior shown by FIFO algorithm in some of the cases. This is an Anomaly called as Belady's Anomaly | | | | |  |  |

|  |
| --- |
|  |

|  |
| --- |
|  |