Assignment D-1

TITLE: Implementation of Page replacement algorithms.

.

PROBLEM STATEMENT:

Write a Java Program (Using OOP features) to implement paging simulation using

1. FIFO
2. Least Recently Used (LRU)
3. Optimal Algorithms

OBJECTIVE:

- Understand virtual memory management
- Analyse the need of page replacement algorithms
- Compare various page replacement algorithms

S/W and H/W

Requirements:

- 64-bit open source Linux (Fedora 20)
- Eclipse IDE, JAVA
- I3 and I5 machines

OUTCOME:

We will be able to

- Implement various page replacement algorithms like FIFO, LRU and Optimal
- Compare the page replacement algorithms based on hit ratio

Theory:

Whenever there is a page reference for which the page needed in not present memory, that event is called page fault or page fetch or page failure situation. In such case we have to make space in memory for this new page by replacing any existing page. But we cannot replace any page. We have to replace a page which is not used currently. There are some algorithms based on them. We can select appropriate page replacement policy. Designing appropriate algorithms to solve this problem is an important task because disk I/O is expensive.

First in First out (FIFO)

The oldest page in the physical memory is the one selected for replacement. Keep a list on a page fault, the page at the head is removed and the new page added to the tail of the list

LRU (Least Recently Used):

In this algorithm, the page that has not been used for longest period of time is selected for replacement. Although LRU is theoretically realizable, it is not cheap. To fully implement LRU, it is necessary to maintain a linked list of all pages in memory, with the most recently used page at the front and the least recently used page at the rear. The difficulty is that the list must be updated on every memory reference. Finding a page in the list, deleting it, and then moving it to the front is a very time-consuming operation, even in hardware (assuming that such hardware could be built).

The Optimal Page Replacement Algorithm:

The algorithm has lowest page fault rate of all algorithm. This algorithm state that: Replace the page which will not be used for longest period of time i.e. future knowledge of reference string is required. Often called Balady's Min Basic idea: Replace the page that will not be referenced for the longest time.

Algorithms:

FIFO:1- Start traversing the pages.

1.1- If set holds less pages than capacity.

1.1.1-Insert page into the set one by one until the size of set reaches capacity or all page requests are processed.

1.1.2-Simultaneously maintain the pages in the queue to perform FIFO.

1.1.3-Increment page fault

1.2 Else If

current page is present in set, do nothing.

1.3-Else

1.3.1- Remove the first page from the queue as it was the first to be entered in the memory

1.3.2- Replace the first page in the queue with the current page in the string.

1.3.3- Store current page in the queue.

1.3.4- Increment page faults.

2. Return page faults.

LRU: Let capacity be the number of pages that memory can hold.
Let set be the current set of pages in memory.
1- Start traversing the pages.
1.1-If set holds less pages than capacity.
1.1.1- Insert page into the set one by one until the size of set reaches capacity or all page requests are processed.
1.1.2-Simultaneously maintain the recent occurred index of each page in a map called indexes.
1.1.3-Increment page fault
1.2- Else If
1.2.1-current page is present in set, do nothing.
1.3- Else
1.3.1-Find the page in the set that was least recently used. We find it using index array. We basically need to replace the page with minimum index. 1.3.2-Replace the found page with current page.
1.3.3-Increment page faults.
1.3.4-Update index of current page.
2. Return page faults.

Conclusion: Thus, we successfully studied and implemented various Page Replacement algorithms.