Assignment 5

Page Replacement Strategies

Subham Rajgaria 15CS10061

Sunaina Singh Rathod 15CS30023

Page Reference Generation

The number of pages given were 4 and we took a working set size of 40. Two probability inputs were also taken one whether the next reference will be from the working set or not, taken to be 0.99 in our case. The other probability is whether it will be a read or write operation taken as 0.75 in our case. So now based on the probability the page reference is either a random number between 0 and 63 or a number from the working set, the working set moves along as the references increase. The generated page reference is stored in the instructions.txt file.

Plot is then generated using the plot.py file.

Page Replacement Code

Structures Used

Bitset used for each page table entry and number of pages equal to 64.

Helper Functions

FdGetFileSize

Returns the file size of a given file.

mapFrame

Sets the frame number in the first 29 bits of each page table entry

extractFrame

Extracts the frame number from the first 29 bits of the page entry

printStatistics

Prints the statistics like number of page faults, total execution time and number of page transfers.

printPageTable

Prints the page table that is all 64 page entries

Main Function

This is a menu driven program which loads the page references from the instructions.txt file. Initially it sets all the page table entries to be 0 and then asks for which page replacement strategy to follow. It iterates through the page references and if it finds the page already loaded then it prints a page found message else if there is space in memory it loads the frame number in the next empty page table entry else it removes one of the page table entries according to the chosen policy and adds the current entry. The program also takes in the number of free frames in memory as an input and trace mode on or off also has to be provided, for trace mode on the page table is printed after each reference.

FIFO policy

Every page reference is pushed back in a vector. When the number of free frames in memory becomes 0 , the first element of the vector is deleted and the new entry is pushed back.

Random Policy

A random index is generated and all the page references made earlier are kept in a vector sequentially. Once the number of free frames become 0 , an index is randomly chosen and that element is deleted from the vector and the new element is pushed back.

LRU Policy

In LRU every page reference is pushed back in a vector but in case whenever a page is accessed which is already there in memory , it is erased from the vector and inserted at the end. Hence the first element of the vector will always be the one which has been least recently used and thus while replacing the page the first element is deleted and replaced with the new entry.

NRU Policy

In this case we take another input as to when to set the reference bit to 0 whether at every 1000 memory references or at every 10th page fault. The NRU divides the page references into 4 classes 00 - not modified , not referenced, 10- modified but not referenced, 01- not modified but referenced , 11 - modified and referenced. Search is done sequentially through these classes and the first match is chosen as the victim page. A pointer is also maintained as to from which entry we removed the page and the next search begins from that point.

Second Chance

In second chance we normally push back the page references in a vector . But while selecting the victim page what we do is we first set all the reference bits to be 0 and then select the first page whose reference bit is set to be 0. Now say a page is referenced and was found in memory so its reference bit becomes 1 , hence now if say after this it was his turn to be replaced but since its reference bit is set to 1 due to an earlier reference it is not chosen and the next page with reference bit 0 is taken into consideration and it is given a second chance. Since this is delpoyed only if there are no free frames this algo works.

Conclusion

In the end the statistics is printed and the page table is also printed.