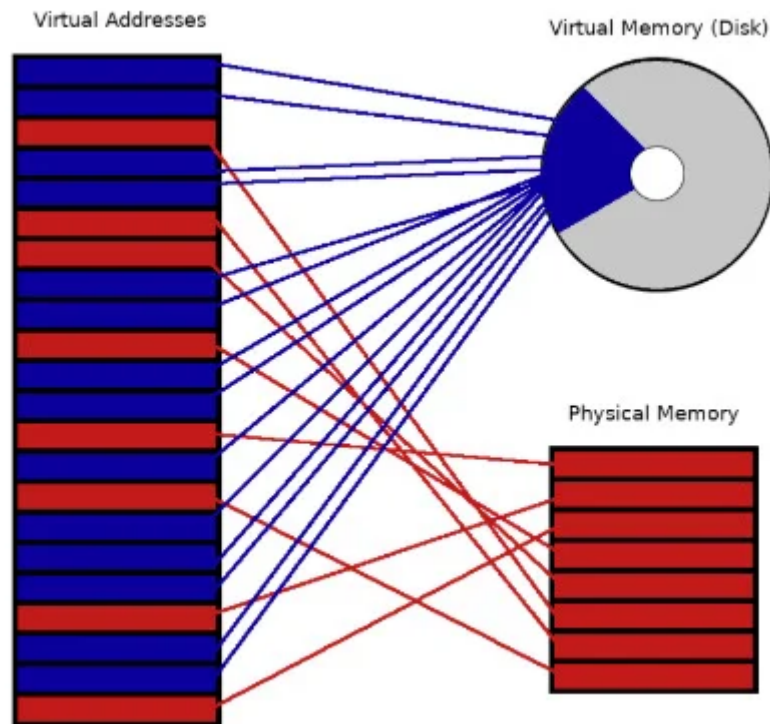


CSE 312 OPERATING SYSTEMS HOMEWORK 3 REPORT

PART 1

Implementation of The Virtual Memory

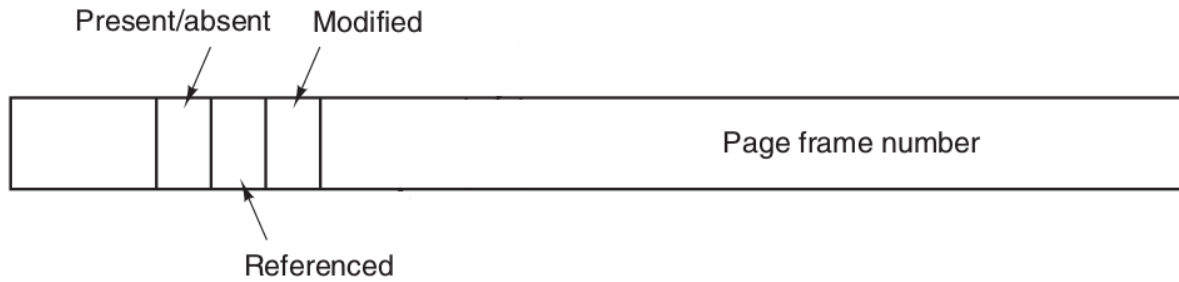


Present virtual page frames are holded in the physical memory and the others are holded in the disk.

Pages are replaced when required,

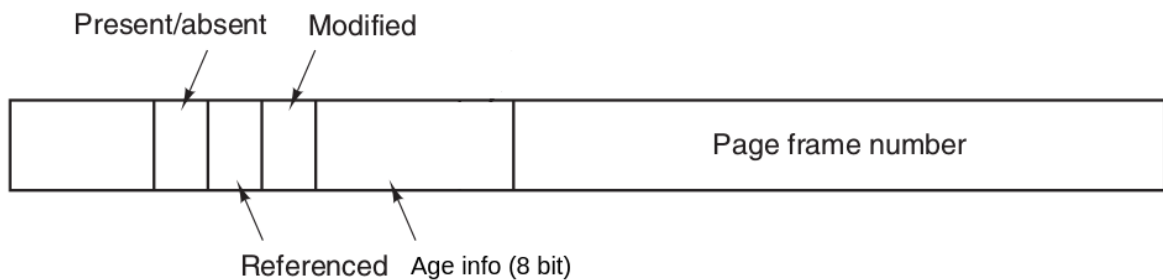
I also implemented kind of bitmap which shows the virtual page number that it is holded by or returns null.

Second Chance Algorithm



I benefit from present/absent, referenced and modified bits to implement this algorithm. I used a circular array to iterate all the present virtual pages. If the page is referenced, I switched the referenced bit to zero and increased the iterator. Unless, I replaced the frame of the physical memory that the iterated page shows, with the data of the virtual page which is looking for an empty frame in the physical memory. If the replaced page is modified, then I wrote it back to the disk.

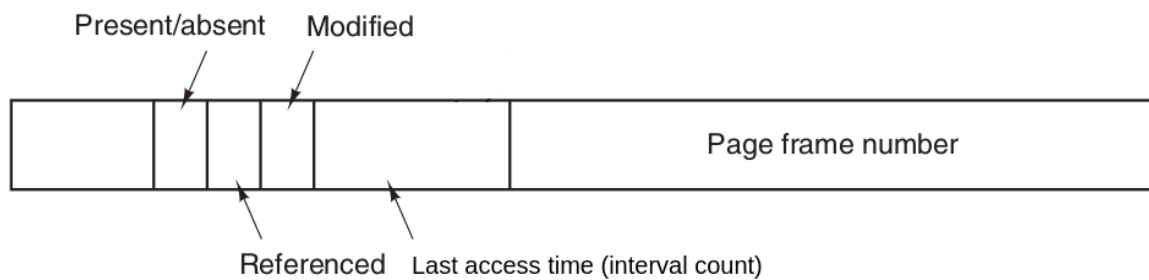
Least Recently Used Algorithm



I implemented the software simulation of that algorithm. I benefit from present/absent, referenced, modified bits, and age of the page.

I calculated the age like that: If the page is present, I shifted the age to the right by one. If the page is also referenced, I added number 10000000 to the age. While replacing the page, I found the oldest page (the number with smallest value) and I replaced the data of the virtual page in physical memory with the new one. If the replaced page is modified, I wrote it back to the disk. I also reset the age of the replaced page.

WSClock Algorithm



I benefit from the present/absent, referenced, modified bits and the last access time of the page to implement this algorithm. And I used a circular array to iterate the present pages. If the page is referenced, I switched the referenced bit to zero and kept iterating. While the page is not referenced and the last access time is no larger than threshold; If the modified bit is 1 I scheduled it to write back to disk and increased the iterator, unless I picked the page to be replaced.

PART 2

In get/set methods, firstly I didn't read any data from the disk while filling the virtual memory. If the present bit is 1, I directly get/set the integer from/to the frame that the virtual page show, and I upgraded the referenced/modified bits and access time for the WSClock algorithm if required. If not, I replaced the page with the appropriate present page by deciding that with the chosen page replacement algorithm.

I implemented the algorithms that I explained plus NRU and FIFO algorithms as replacement algorithms.

I implemented sorting/merging/searching algorithms by using get/set methods inside.

I implemented the aging algorithm / the clock in a while loop. In every clock tick, I updated the time for WSClock algorithm / aged all the present pages for LRU algorithm.