CSE 312 HW3 REPORT

EMİRHAN UZUN
171044019

Firstly, in Figure 3.11 of book, there is a typical page table entry. It includes Caching, Referenced, Modified, Protection, Present and page frame number. But I changed this a little bit. I used isUsedRecent, Referenced, Modified, Present and pageNum variables in tableInformation struct. I don't need protection and caching variables. **This struct is in line 13 in my code.**

To print informations, I created a struct too. This is printInformation strcut and it includes numOfRead, numOfWrite, numOfPageMisses, numOfPageReplacements, numOfDiskPageWrites, numOfDiskPageReads. At the end of the program, I print these informations. **This struct is in line 20 in my code**

To hold array for memories and entries, I create 4 different dynamic array. They are physical pages and virtual pages dynamic arrays in tableInformation type. And the other 2 of them are virtual memory and physical memory dynamic arrays in integer type. According to values which is coming with command line, I allocate these array sizes. For example, to allocate for virtual memory, I find frameSize and virtual page number. I multiply that and I find the size of virtual array. The same thing is for physical memory. For physical and virtual pages arrays, I used physical page number and virtual page number. **These dynamic arrays are in line 40,41,43 and 44 in my code and I allocate these in main function.**

After that I fill the arrays. **I used srand(1000) and rand function to fill the memories.** In the fill arrays function, at the end I call **set** function and I also fill diskFileName.dat. And tableInformation variables is filled in set function also. These variables are as I said isUsedRecent, Referenced, Modified, Present and pageNum. **The fill arrays function is in line 380. Set function is in line 344.**

After filling I create 2 thread for bubble sort and quick sort. The half of them is sorted by bubble and the other half is sorted by quick. Also I use **get** function for getting the values of given index.
These created part is in main in line 515.
Sort thread void* function is in line 262.
Bubble sort is in line 213.
Quick sort is in line 252.
Partition is in line 229.
Get function is in line 158. I increase the read or writing etc. values according to sorting functions.

When the sorting finish, I create another thread for merge. I merge these arrays in one array.
Create merge thread is in line 529.
Merge thread void* function is in line 335.
Merge sort is in line 324.
Merge function is in line 297.

After the merging, I create 2 another thread for searching algorithm. First one is for linear and the other one is for binary search.
These created part is in main in line 542.
Search thread void* function is in line 275.
Linear search function is in line 59.
Binary search is in line 79.

For Fill, Quick, Bubble, Merge, Linear and Binary parts, I increments the read,writes values etc.

In page replacement algorithms I just did Second Chance Algorithm. It checks the reference bit and operate something. It checks the page which is in memory. After that it find the page which is replaced by another. And it replaces the pages.

Second Chance Algorithm is in line 129.
Physical memory check function is in line 96
Find the page function is in line 116

In second chance algorithm, I check the reference bits. If it is appropriate to replacement, then I replace.

Lastly, I print the informations about the 6 parts(Fill,Bubble,Quick,Merge,Linear,Binary)

I couldn’t do Least Recently Used and Working Set(WSClock) algoritms.