Operating Systems Project - Exercise 5

I have made some fixes and improvements to the original code provided which helped me understand and work with the code better: I have removed the unnecessary #include "stdafx.h", fixed and improved the main function(scnaf_s changed to scnaf, included the print of the menu in the loop, made it so the data init is only called once in the code), I added more printing to help me explain the workings of the algorithms better, fixed some printing with the inclusion of \n, replaced static numbers with their corresponding variables such as “4” in some segments with PM_PAGE.

Common explanations of required results: Page faults are the number of times the operating system must load a page into physical memory because it was not already there when required by a program. This can be loading a page into empty memory or replacing an already existing value. Better page replacement algorithms lead to less page faults. The number of replacements is always less than a number of page faults by the size of memory blocks, which is 4 in this case. The page fault rate is calculated from the number of page faults divided by the number of page replacements. The replacement rate is the same, but we substitute the page faults with the number of replacements. In this case as well, the lower the better.

```
pence@BenceLaptop:~/OPsystems/Exercise 5$ ./pageframereplace
Please choose page replacement algorithm : 1.FIFO, 2.LRU, 3.OPT, 0. quit:        1
The number of page fault(s) of FIFO is: 13       Page fault rate: 0.722222       Number of replacement(s) :  9       Replacement rate :  0.500000
Please choose page replacement algorithm : 1.FIFO, 2.LRU, 3.OPT, 0. quit:        2
The number of page fault(s) of LRU is: 10        Page fault rate: 0.555556       Number of replacement(s) :  6       Replacement rate :  0.333333
Please choose page replacement algorithm : 1.FIFO, 2.LRU, 3.OPT, 0. quit:        3
The number of page fault(s) of OPT is: 7         Page fault rate: 0.388889       Number of replacement(s) :  3       Replacement rate :  0.166667
Please choose page replacement algorithm : 1.FIFO, 2.LRU, 3.OPT, 0. quit:        0
Quitting...
```

This is the required output for this exercise, however to the ease of explanation of the algorithms’ inner workings and how they function, I added more printing for each algorithm:

FIFO:
A quite simple algorithm, where we replace the page which contains the oldest added algorithm. This is one of the least efficient algorithms. We can see that if the page is not already in memory, the memory blocks get replaced in a cyclical manner.

```
Please choose page replacement algorithm : 1.FIFO, 2.LRU, 3.OPT, 0. quit:        1
The virtual page access order is: 1 2 3 4 2 6 2 1 2 3 7 6 3 2 1 2 3 6
Added page 1 to physical block 0
Added page 2 to physical block 1
Added page 3 to physical block 2
Added page 4 to physical block 3
Page 2 already exists in physical block 1
Replaced page 1 with page 6 in physical block 0
Page 2 already exists in physical block 1
Replaced page 2 with page 1 in physical block 1
Replaced page 3 with page 2 in physical block 2
Replaced page 4 with page 3 in physical block 3
Replaced page 6 with page 7 in physical block 0
Replaced page 1 with page 6 in physical block 1
Page 3 already exists in physical block 3
Page 2 already exists in physical block 2
Replaced page 2 with page 1 in physical block 2
Replaced page 3 with page 2 in physical block 3
Replaced page 7 with page 3 in physical block 0
Page 6 already exists in physical block 1
The number of page fault(s) of FIFO is: 13       Page fault rate: 0.722222       Number of replacement(s) :  9  Replacement rate :  0.500000
```

LRU:

Quite similar to FIFO with the difference that the time gets updated every time a page is accessed. This means that often accessed pages stay in the memory more time. For example, 2 stays in memory for the full runtime after its added, as it is accessed every 4th or less iteration. This results in 3 less page faults then FIFO.

```
Please choose page replacement algorithm：1.FIFO, 2.LRU, 3.OPT, 0. quit:        2
The virtual page access order is: 1 2 3 4 2 6 2 1 2 3 7 6 3 2 1 2 3 6
Added page 1 to physical block 0
Added page 2 to physical block 1
Added page 3 to physical block 2
Added page 4 to physical block 3
Page 2 already exists in physical block 1, updating time accessed
Replaced page 1 with page 6 in physical block 0
Page 2 already exists in physical block 1, updating time accessed
Replaced page 3 with page 1 in physical block 2
Page 2 already exists in physical block 1, updating time accessed
Replaced page 4 with page 3 in physical block 3
Replaced page 6 with page 7 in physical block 0
Replaced page 1 with page 6 in physical block 2
Page 3 already exists in physical block 3, updating time accessed
Page 2 already exists in physical block 1, updating time accessed
Replaced page 7 with page 1 in physical block 0
Page 2 already exists in physical block 1, updating time accessed
Page 3 already exists in physical block 3, updating time accessed
Page 6 already exists in physical block 2, updating time accessed
The number of page fault(s) of LRU is: 10        Page fault rate: 0.555556       Number of replacement(s) :  6  Replacement rate :  0.333333
```

OPT:

This is not actually a viable replacement algorithm, as we need to see the full virtual page access order before runtime, but it is a great way to see what the most efficient algorithm could be. It works in a way that any time a page is added, the queue of virtual pages is checked to see when the next occurrence is. When a page fault occurs, the page which has its occurrence the latest (or does not occur) gets replaced with the new one. This results in the lowest possible number of page faults and is therefore the golden standard that we can measure actually viable algorithms to.

```
Please choose page replacement algorithm：1.FIFO, 2.LRU, 3.OPT, 0. quit:        3
The virtual page access order is: 1 2 3 4 2 6 2 1 2 3 7 6 3 2 1 2 3 6
Added page 1 to physical block 0
Page 1 will be used next at 7
Added page 2 to physical block 1
Page 1 will be used next at 7
Page 2 will be used next at 4
Added page 3 to physical block 2
Page 1 will be used next at 7
Page 2 will be used next at 4
Page 3 will be used next at 9
Added page 4 to physical block 3
Page 2 was accessed, but no page fault occurred.
Page 1 will be used next at 7
Page 2 will be used next at 6
Page 3 will be used next at 9
Replaced page 4 with page 6 in physical block 3, since this is the last accessed
Page 2 was accessed, but no page fault occurred.
Page 1 was accessed, but no page fault occurred.
Page 2 was accessed, but no page fault occurred.
Page 3 was accessed, but no page fault occurred.
Page 1 will be used next at 14
Page 2 will be used next at 13
Page 3 will be used next at 12
Page 6 will be used next at 11
Replaced page 1 with page 7 in physical block 0, since this is the last accessed
Page 6 was accessed, but no page fault occurred.
Page 3 was accessed, but no page fault occurred.
Page 2 was accessed, but no page fault occurred.
Page 2 will be used next at 15
Page 3 will be used next at 16
Page 6 will be used next at 17
Replaced page 7 with page 1 in physical block 0, since this is the last accessed
Page 2 was accessed, but no page fault occurred.
Page 3 was accessed, but no page fault occurred.
Page 6 was accessed, but no page fault occurred.
The number of page fault(s) of OPT is: 7         Page fault rate: 0.388889       Number of replacement(s) :  3  Replacement rate :  0.166667
```