

Final Project

Write a Java or C/C++ (the choice is yours) program for a Demand Paging Simulator according to the following rules. The program will implement two Demand Paging algorithms: the classical OPT algorithm (described in the module 5 readings) and a NEW algorithm, described below.

The application must simulate the execution the OPT and NEW algorithms on a hypothetical computer having only N physical frames (numbered from 0 to N-1; the maximum value for the parameter N is 8), assuming that the single process that is running has a virtual memory of ten pages (numbered from 0 to 9). The number N should be a number provided as an input for the algorithms, using the program menu.

To clarify, the user should be able to run the OPT algorithm (for example) using the same reference string for a scenario where N is 4, then change N to 5 and run again, then change N to 6 and run again and in the end compare the results for each scenario. In other words, the number of physical frames should not be fixed (hardcoded) in the program, but rather considered as an input variable (parameter) for the implemented algorithms.

The program requested for this project must have a text menu like this:

```
0 - Exit
1 - Input N
2 - Input the reference string
3 - Simulate the OPT algorithm
4 - Simulate the NEW algorithm
Select option:
```

The menu is displayed and the user must select an option (a number between 0 and 4). The action corresponding to the selection is performed, then the menu is displayed again and the user can choose another option. This cycle is repeated until the user selects 0, which exits the loop and ends the program execution.

The options are:

0 – Exit

This options ends the program execution. This option should always work.

1 – Input N

The user is prompted for a [positive integer] number N. This is normally one of the first options that must be selected by the user. The number N must be at least 2 (otherwise the algorithms don't make any sense) but should not be larger than 8; the program must verify these constraints and issue an error message whenever necessary.

After performing several operations from the menu, the user should be able to select another N and work with the new N just as easily as before.

2 – Input the reference string

The user is prompted for a series of [positive integer] numbers that will constitute the reference string. The maximum length of the reference string should be 20. This is normally one of the first options that must be selected by the user. The number of elements in the reference string must be at least N

(otherwise the algorithms don't make any sense) but should not be larger than 20; each element of the reference string must be an integer number between 0 and 9 (there are 10 virtual pages). The program must verify these constraints and issue an error message whenever necessary.

After performing several operations from the menu, the user should be able to select another reference string and work with the new reference string just as easily as before.

3 – Simulate the OPT algorithm

This option must first verify that a correct N and a correct reference string were already provided by the user; otherwise an error message should be issued and the program returns to the menu.

If all the input data (N and the reference string) is available and correct, the simulation begins by printing the initial (empty) simulation table, for example:

[illegible]

Hint 1: Instead of the above reference string, the table will contain the ACTUAL reference string that was entered by the user; of course, everyone can use their reference strings for testing the correctness of the implementation. Using the reference strings from the readings to test your code is a great idea.

Hint 2: Instead of having 4 physical frames in the simulation table, we must actually have N physical frames. So if the user has entered the value 6 for N using option 1 of the menu, then the simulation table should look like this:

[illegible]

Hint 3: Being a text only interface, clearly we cannot display such a nice table, but rather an approximation of it; the table could instead look like this or similar:

| | | | | | | | | | | | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reference string | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 9 | 1 | 8 | 2 | 7 | 3 | 6 | 4 | 5 |
| Physical frame 0 | | | | | | | | | | | | | | | | | | | | |
| Physical frame 1 | | | | | | | | | | | | | | | | | | | | |
| Physical frame 2 | | | | | | | | | | | | | | | | | | | | |
| Physical frame 3 | | | | | | | | | | | | | | | | | | | | |
| Physical frame 4 | | | | | | | | | | | | | | | | | | | | |
| Page faults | | | | | | | | | | | | | | | | | | | | |
| Victim Pages | | | | | | | | | | | | | | | | | | | | |

The user must proceed with the STEP by STEP simulation of the OPT algorithm by pressing a key after each step of the algorithm; just as in the examples from the module 5 readings, each step of the simulation will fill a new column of the simulation table (previous steps must also be visible) and the table is re-printed on the screen after each step. This way, the user has an overview of all the previous steps and also of the current step. In the end, the table will be completely filled with information, and the user will have the complete view of the simulation. The total number of faults must also be displayed.

In other words, the simulation must follow **very closely** the examples of Demand Paging algorithms from Module 5 of this course. Printing the explanations is optional.

4 – Simulate the NEW algorithm

The simulation proceeds just like it did in option 4, but this time the algorithm is no longer the well known OPT algorithm, but a NEW algorithm. This NEW algorithm is a modified version of the LRU algorithm (LRU algorithm was discussed in Module 5 of this course). So how exactly does the NEW algorithm work?

The NEW algorithm works just like the LRU algorithm with just one small change, regarding the policy for page replacement, when such an event occurs, and a page must be evicted and replaced with another page. In case of the LRU algorithm, the victim page is the one that has not been accessed, or used, for the longest period of time. Instead, in the NEW algorithm, the victim page is going to be the one that has not been accessed, or used, for the **SECOND** longest period of time.

Consider the following example from the Module 5 LRU example, step 17:

- Page 4: was referenced 5 steps ago
- *Page 3: was referenced 3 steps ago*
- Page 7: was referenced 2 steps ago
- Page 2: was referenced 1 step ago

The victim page for LRU is therefore page 4, because it was not accessed for the longest time.

However, the victim page for NEW is page 3, because it was not accessed for the **SECOND** longest time.

Testing procedure for one set of input parameters (N, reference string):

0. Option 0 requires just one screenshot, for successful exit.
1. Option 1 requires one screenshot for a correct input of N and one screenshot for an incorrect N.
2. Option 2 requires one screenshot for a correct input of a reference string and several screenshots for an incorrect reference string, as several errors are possible (incorrect element smaller than 0, incorrect element greater than 9, incorrect sequence length).
3. Option 3 requires screenshots showing the simulation of the OPT algorithm. Two screenshots will suffice: one showing the “middle” of the simulation (approximately) and another showing the end of the simulation (simulation table must be full with information).
4. Option 4 requires screenshots showing the simulation of the NEW algorithm. Two screenshots will suffice: one showing the “middle” of the simulation (approximately) and another showing the end of the simulation (simulation table must be full with information).

Deliverables:

1. Source code (zipped)
2. A Test Report (called Test1.docx/Test1.pdf) showing the program performing each of the menu actions correctly (use the testing procedure described above, include all the required screenshots) for N=4 and the reference string from Homework 5.
3. A Test Report (called Test2.docx/Test2.pdf) showing the program performing each of the menu actions correctly (use the testing procedure described above, include all the required screenshots) for N=5 and the reference string from Homework 5.
2. A Test Report (called Test3.docx/Test3.pdf) showing the program performing each of the menu actions correctly (use the testing procedure described above, include all the required screenshots) for N=6 and the reference string from Homework 5.