

CSC3002F: Operating Systems

Practical Assignment 1: Memory Management

Department of Computer Science
University of Cape Town

11 April, 2023

DUE: 25 April, 2023, 9.00 AM

Assignment Description

Write a Python program that implements the *FIFO*, *LRU*, and *optimal* page replacement algorithms presented in *Chapter 10: Virtual Memory* of Silberschatz et al. [Silberschatz et al., 2018].

First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm.

Implement the *FIFO*, *LRU*, and *optimal* (OPT) replacement algorithms so that the number of page frames can vary from 1 to 7. Assume that *demand paging* is used. The *main* function should include the following.:

```
def main():
    #...TODO...
    size = int(sys.argv[1])
    print 'FIFO', FIFO(size,pages), 'page faults.'
    print 'LRU', LRU(size,pages), 'page faults.'
    print 'OPT', OPT(size,pages), 'page faults.'

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print 'Usage: python paging.py [number of page frames]'
    else:
        main()
```

Implement these FIFO, LRU and OPT algorithms as functions within one file called *paging.py*, making sure that you clearly identify yourself (name and student number) in comments at the top of the file.

The total assignment mark (100) will be calculated as follows.:

1. Correct implementation and functioning of FIFO. **(30%)**
2. Correct implementation and functioning of LRU. **(30%)**
3. Correct implementation and functioning of OPT. **(40%)**

NOTE: Values in bold parentheses are the percentage weighting of each question as a portion of the total assignment mark.

Assignment Instructions and Notes

- Note that the assignment will be manually marked – Hence it is important that your code compiles and runs so as the tutors can test that the algorithms are actually functioning properly. Your code must run on the *nightmare.cs.uct.ac.za* server. You can use SSH and STP to test it.
- You must implement this assignment in Python – All algorithms should be in the same *.py* file – the main function is given in the above assignment description.
- For each algorithm (*LRU*, *FIFO*, *OPT*): You do not need to use real virtual memory to test your algorithms, but rather a *simulation* of referenced pages. This is just an array of N integers to test N page references. For example.: the array: [8, 5, 6]: could test page numbers: 8, 5, 6. Page numbers must be values in the range of 0 to 9 (also stated in the assignment description). Reference arrays of N such page numbers can then be (randomly) generated.
- Each algorithm must also use the *frames* parameter, i.e. how many pages can be in memory at any one time. Your *LRU*, *FIFO*, *OPT* algorithms should work with any frame size, but you can test each algorithm with frame sizes between 1 and 7 (as stated in assignment description).
- Be sure to test your algorithms with various page reference sizes (e.g. 8 pages, 16 pages, 24 pages, and 32 pages) with some set frame-size (e.g. = 3), to be sure that the algorithms are working correctly.
- For reference look at the associated reading: *A Comparison of Three Page Replacement Algorithms: FIFO, LRU and Optimal* [CSC3002F, 2022], which presents an example test run done for each algorithm and output given for number of pages = 8, 16, 24, and 32. Testing of your own FIFO, OPT and LRU algorithms should do something similar, but with more pages! (e.g. $N > 32$).
- NOTE: You DO NOT have to show your testing results (this is just for your own testing purposes). You only need to hand in the *.py* file with the src for your FIFO, OPT and LRU functions.

* * *

References

- [CSC3002F, 2022] CSC3002F (2022). *A Comparison of Three Page Replacement Algorithms: FIFO, LRU and Optimal*. Department of Computer Science, University of Cape Town, South Africa.
- [Silberschatz et al., 2018] Silberschatz, A., Galvin, P., and Gagne, G. (2018). *Operating System Concepts - 10th Edition*. John Wiley & Sons, New York, USA.