

New York University

Tandon School of Engineering

Department of Electrical & Computer Engineering

Introduction to Operating Systems (CS-GY6233)
Spring 2020

Assignment 7
(15 points)

Develop a program with two functions/subroutines that implement the LRU and the Optimal page replacement algorithms. These functions should accept a page trace and a parameter f for the number of frames allocated. You shall not implement an LRU approximation algorithm.

Your main routine shall accept two parameters n and k (i.e. when you invoke your program from the shell, you pass it two parameters, n and k , where $n \geq 16$ and $k \geq 3$). Your main routine shall generate a random page trace of size n , where the page numbers have values ranging from 0 to $2^k - 1$.

Your main routine shall then apply that page trace to each of the two subroutines implementing the algorithms, multiple times (using only one trace, randomly generated), passing a parameter f (number of page frames used) that ranges from 4 to 2^k . Your main routine shall then record the number of page faults for each run (i.e. for each f) and for each of the two algorithms.

Run your program using a page trace of length $n=64$, $k=4$ (thus f ranges from 4 to 16). Plot and submit a graph displaying the number of faults vs the number of frames allocated for each of the two algorithms.

Did your outcome exhibit the Belady's anomaly? Why?

Notes:

- For debugging, you may use the page trace given in the lectures instead of randomly generating it (with $n=20$, $k=3$ and $f=3$) and verify you obtained the same results as in the lectures.
- You may output into a text file (e.g. a comma separated file) and then use excel to plot.
- There are three ways to implement a non-approximation LRU algorithm:
 - Using a time stamp to keep track of page loading times (which may be just its index in the page trace for when it was first loaded into a page frame)
 - Using a stack (with a doubly linked list)
 - Computing the backward distance (possibly using a backwards find subroutine).
- For the optimal algorithm, you may implement it by computing the forward distance (possibly using a forward find subroutine). As such, perhaps the easiest way is to implement forward and backward search subroutines for the optimal and LRU algorithms, respectively.

What to hand in (using NYU Classes):

- Your source files (with appropriate comments). Do not attach project or make files.
- A file containing answers to questions, plots and comments (if any) in word or pdf format.
- A screen shot(s) of your terminal window (possibly in the same file) showing the current directory, the command used to compile your program, the command used to run your program and the output of your program.

RULES:

- You may consult with other students about GENERAL concepts or methods, but copying code (or code fragments) or algorithms is NOT ALLOWED and is considered cheating (whether copied from other students, the internet or any other source).
- If you are having trouble, please ask your teaching assistant for help.
- You must submit your assignment prior to the deadline.