

Page Replace

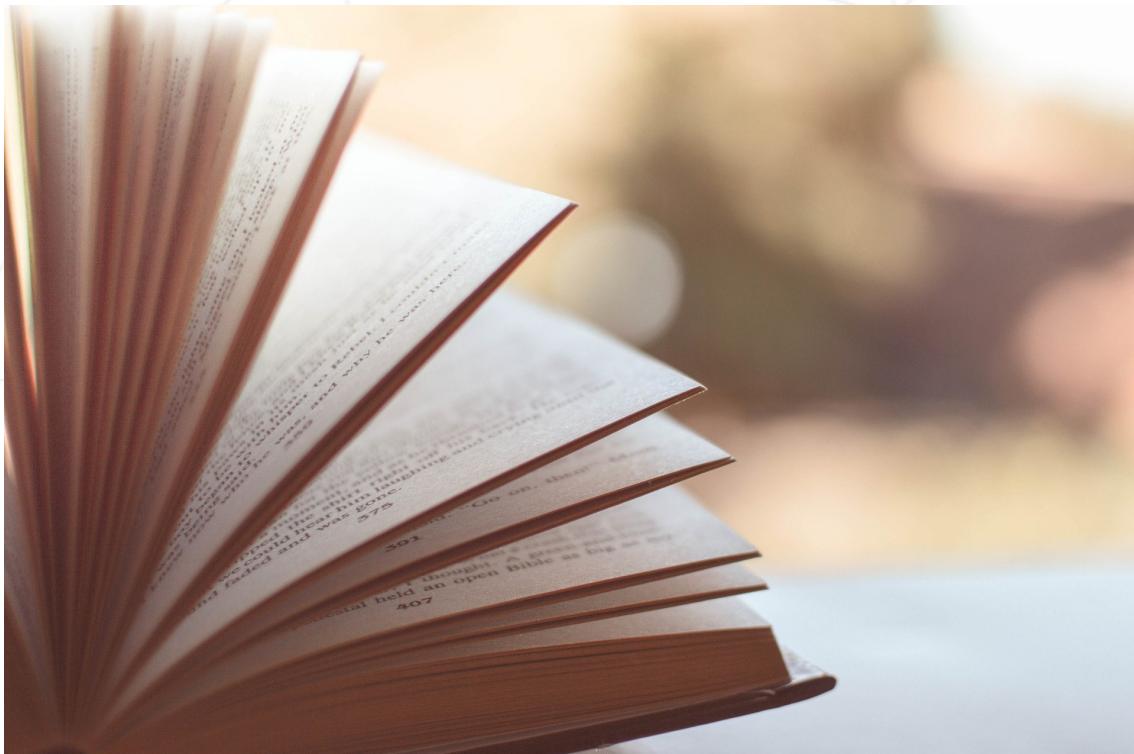
Manage your little cache memory

Contents

I	Foreword	2
II	Introduction	3
III	Common Instructions	4
IV	Mandatory Part	5
V	Bonus	7
VI	Submission and peer correction	8

Chapter I

Foreword



Chapter II

Introduction

You are working in a small cafe. All four tables in the cafe are full of customers. When a new guest comes in, you must empty a table and receive a new one. Which table should I empty? Make a choice!!

1. customer who First In
2. customer who Least Recently ordered
3. customer who Least Frequently ordered
4. customer who won't order for the longest time(You know the future)

You know what? You chose an **Algorithm**.

Now think of the customer as a different concept. For example, **page**.

If the cafe is memory, the customers are the page that saved in memory.

The page chosen by algorithm becomes victim and falls out of memory.



FIFO, LRU, LFU, OPT. These are page replace algorithms.
https://en.wikipedia.org/wiki/Page_replacement_algorithm

Chapter III

Common Instructions

Your project must be written in C.

This subject will compile your source files to the required output with the flags -Wall, -Wextra and -Werror, use cc.

Submit your work to your assigned git repository. Only the work in the git repository will be graded.

This subject simulate the operating system's page replacement algorithm.

Therefore, you should be able to explain the behavior of your algorithm.

Chapter IV

Mandatory Part

Manage your little cache	
Program name	vpdlwldkfrhflwma
Files to turn in	*.h, *.c
Allowed functions	strtol, printf, write, strlen
Arguments	list of positive integers
Description	The Optimal Algorithm

- The goal of the program is to implement a **Optimal page replacement algorithm** that creates the lowest page fault.
- Your vpdlwldkfrhflwma program accepts only positive integers as arguments
 - If it isn't, print "Error\n" to standard error and exit program.
 - If no argument is given, nothing is output and exit program.
- The input arguments are the index of the page to be referred to.
 - Each argument is separated by a space.
- Page Table required for page replacement shall be declared array of size 4
- Page Table should be printed when one page(one argument) is accessed while the program is running.
- The empty space of the page table must be initialized to '0'.
- If page replacement has the same priority, replace the element further to the left in the page table.

Example

```
bash-5.1$ ./vpdlwldkfrhflwma 1 2 5 3 6 1 2
1 0 0 0
1 2 0 0
1 2 5 0
1 2 5 3
1 2 6 3
1 2 6 3
1 2 6 3
bash-5.1$ ./vpdlwldkfrhflwma hello world!
Error
bash-5.1$ ./vpdlwldkfrhflwma
bash-5.1$
```

During the evaluation process, a binary will be provided in order to properly check your program.

It will work as follows:

```
bash-5.1$ ./vpdlwldkfrhflwma 3 1 2 4 2 | ./corrector 3 1 2 4 2
OK
bash-5.1$ ./vpdlwldkfrhflwma 3 1 2 4 2 | ./corrector 3 1 2 4
KO
bash-5.1$
```

If the program corrector displays “KO”, it means that your program came up with a list of instruction that doesn't work correctly

Chapter V

Bonus

Manage your little cache	
Program name	vpdlwldkfrhflwma_bonus
Files to turn in	*.h, *.c
Allowed functions	strtol, scanf, printf, write, strlen
Arguments	list of positive integers
Description	Paging replacement “feasible” algorithm

- The goal of the bonus program is to simulate a page replacement algorithm that is feasible (except OPT).
- Input keeps coming in until End Of File.
- The input conditions are the same as the mandatory.
- Every input comes in, It must be print of the replaced state of your page table.
- Bonus will evaluate only by peer-evaluation.

Chapter VI

Submission and peer-evaluation

Your assignment should be stored in the root directory of the Git Repository.
Check more than once to make sure it is submitted well.