

Page Replace

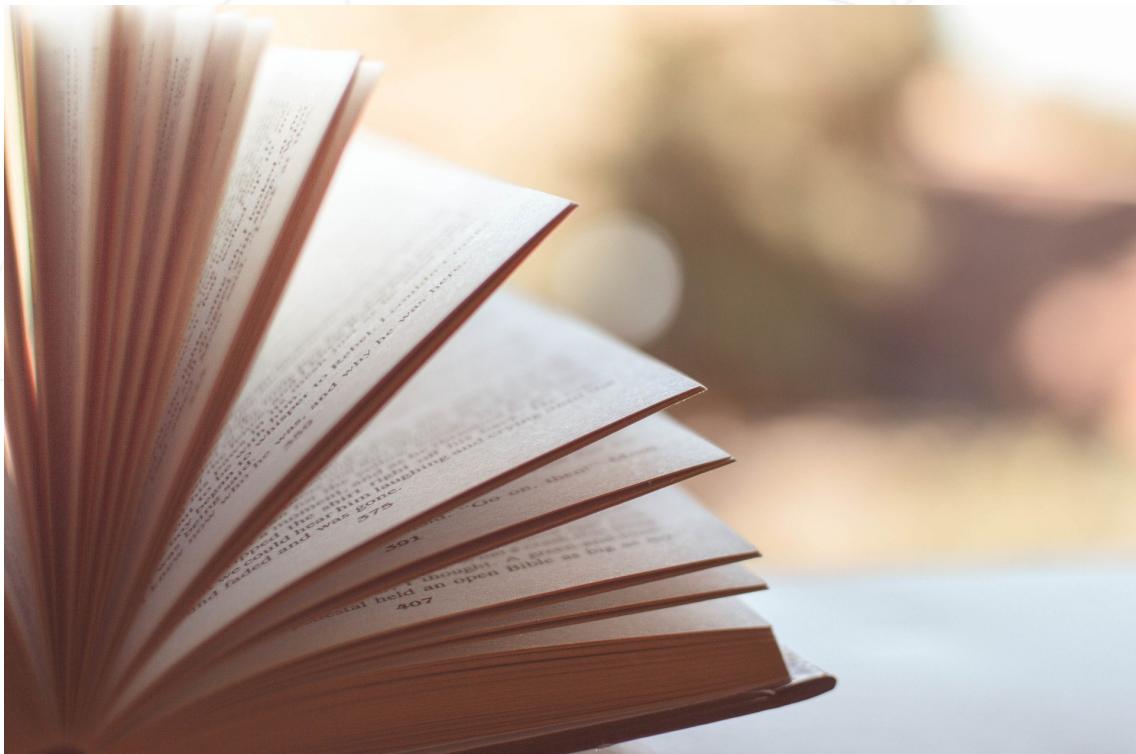
Manage your little cache memory

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

if you haven't heard 'Page replacement algorithm', this task won't be easy.

So read Wikipedia (https://en.wikipedia.org/wiki/Page_replacement_algorithm) first. Read it.

This assignment will takes 30 minutes to 2 hours

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 implements the operating system's page replacement algorithm. Therefore, you should be able to explain the implementation and behavior of your chosen 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	Paging replacement algorithm

- The goal of the program is to implement a page replacement algorithm that creates the lowest page fault rate.
- Your vpdlwldkfrhflwma program accepts only positive integers as factors:
 - If it is not a positive integer, output "Error\n" to standard error and exit
 - If no argument is given, nothing is output.
- N page index to be referenced are inserted as arguments in the program.
 - Each argument is separated by a space.
 - If the argument is not a positive integer value, it is printed as an error.
- The argument is the index of the page to be referred to, and refers to the page in the order of the argument.
- Page frames required for page replacement shall be declared array of size 4
- A page frame should be printed when one page is accessed while the program is running.
- The empty space of the page frame is initialized to '0'.

- If page replacement has the same priority, replace the element further to the left in the page frame.

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$
```

```
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$
```

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 implement a page replacement algorithm that is feasible
- input keeps coming in until End Of File or you want (ctrl-c)
- the input and print conditions are the same as the mandatory
- every input comes in It must be print of the current state that can check the behavior of your algorithm
- bonus only consisted for peer correction

Chapter VI

Submission and peer correction

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