

ICS 332 Fall 2021

[Home](#) / [Modules](#) / [Virtual Memory: Paging](#) / Homework Assignment #11

Homework Assignment #11 – Page Faults [50 pts]

You are expected to do your own work on all homework assignments. (See the statement of Academic Dishonesty on the [Syllabus](#).)

Check the [Syllabus](#) for the late assignment policy for this course.

How to turn in?

Assignments need to be turned in via [Laulima](#). Check the [Syllabus](#) for the late assignment policy for the course.

What to turn in?

You should turn in single **plain text** file named README.txt with your answers to the assignment's questions. Your file must be readable "as is" and points will be removed if the report is not readable.

Exercise #1 [42 pts]

Consider a system with four page frames and a program that uses eight pages. Consider the reference string **0 1 7 2 0 3 1 7 0 1 7** and assume that all four page frames are initially empty. Consider the three following algorithms:

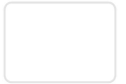
- **a. [14 pts]** Optimal page replacement
- **b. [14 pts]** FIFO page replacement
- **c. [14 pts]** LRU page replacement

In each cases show a diagram (ASCII art recommended) that shows which pages are in which frames throughout time, and page faults at the bottom. For instance:

```

  | 0  1  7  2  . . .
-----
0 | 0  0  0  0  . . .
1 | -  1  1  1  . . .
2 | -  -  7  7  . . .
3 | -  -  -  2  . . .
-----
f | X  X  X  X  . . .

```



ICS 332 Fall 2021

when and thus enter the number of page faults.

The first 4 steps are ALWAYS the same, as above (i.e., just fill the four frames, with one page fault each time)

Exercise #2 [8 pts]

Consider a system that implements demand paging without a TLB. The single-level page table is held in memory, and we assume that there are no free frames. Each memory access requires 100 **nanoseconds**. It takes 6 additional **milliseconds** to handle a page fault if the victim page is not “dirty” or 12 additional **milliseconds** if the victim page is “dirty”.

Experiments show that the probability of page faulting is 0.00003 (i.e., 0.003% chance) and that the average memory access time is 400ns.

What is the percent chance that a page is dirty at the time it is evicted?

Hint: Call the “being dirty” probability d and write a simple equation that gives the average data access time as a function of d . Solve for d (the result should be a number between 0 and 1, which you then convert to a percentage).

Powered by the Morea Framework (Theme: cerulean-green)

Last update on: 2021-12-01 23:16:05 -1000

10 modules | 10 outcomes | 16 experiences