

# OSN 201720: Assignment #2.1

Due on Friday, October 13, 2017

*Claudio Alvarez*

Macarena Espinosa - Alfonso Zúñiga

## Contents

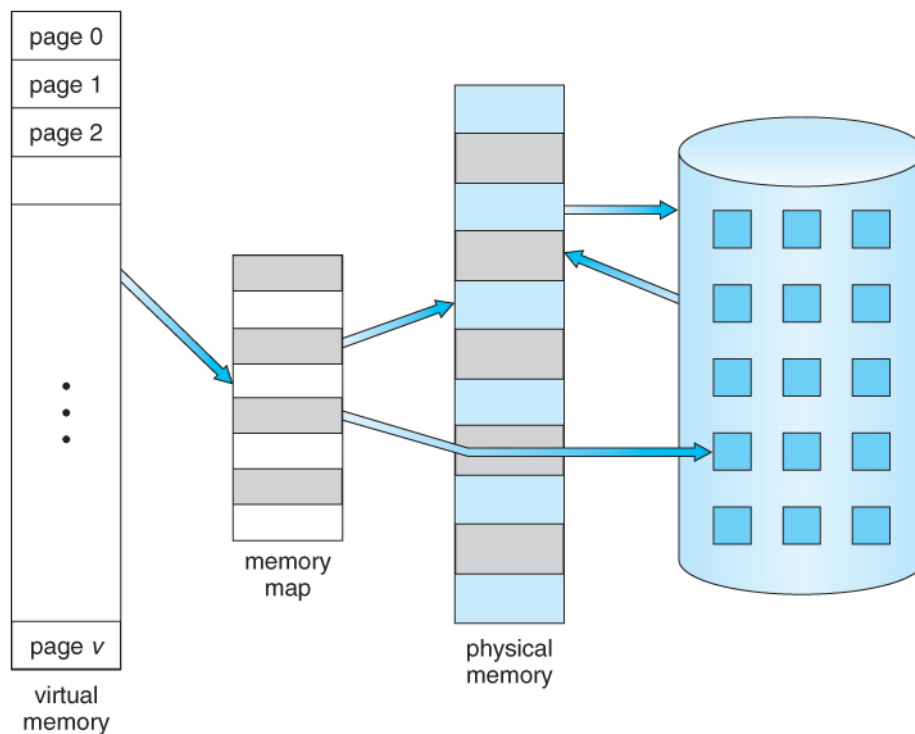
<b>Introduction</b>	<b>3</b>
<b>Analysis</b>	<b>4</b>
Focus . . . . .	4
Scan . . . . .	4
Sort . . . . .	5
<b>Conclusion</b>	<b>6</b>

## Introduction

The assignment consisted in implementing a page fault handler that manages the virtualization of physical memory in a simulation run in C, which also implements data transfer between memory and disk. This C program gives the user a choice between three programs that will run and request space in memory for allocating data, or ask to read information somewhere in memory.

Given that the number of frames in physical memory could be significantly lower than the number of pages in virtual memory, the objective to this assignment is to implement a function that manages all possible faults that may occur, fix them using one of three available algorithms and return to the program, so that it gives the user the illusion of running smoothly and returning the expected result at the end, thus achieving the illusion of infinite memory using virtualization.

It is assumed that we can give read and write bits from the beginning to every valid entry in the page table, thus a page fault will never arise because of this. It is also assumed that every available frame can be related to a page from the beginning, making it easier to manage a page fault.

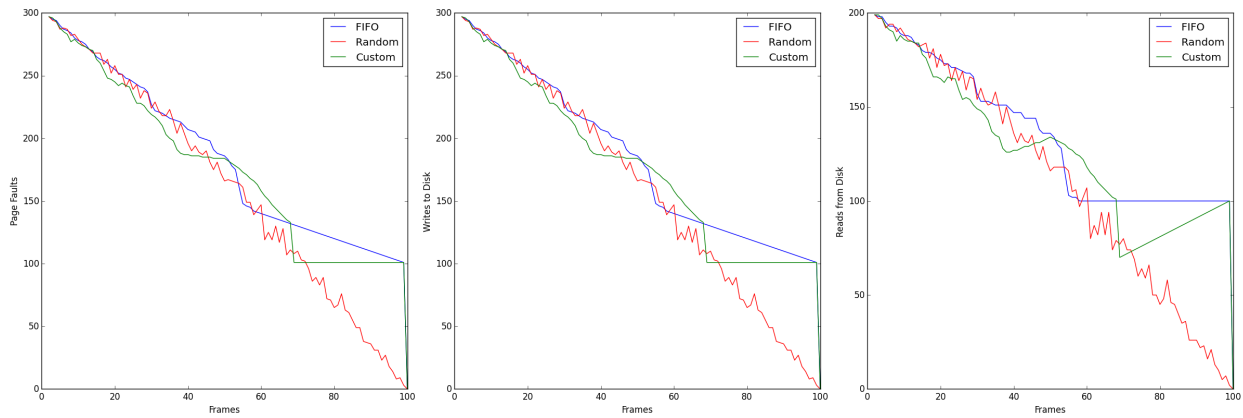


## Analysis

Next, a display of multiple graphs illustrate the behavior of each algorithm used to manage page faults in the execution of each program.

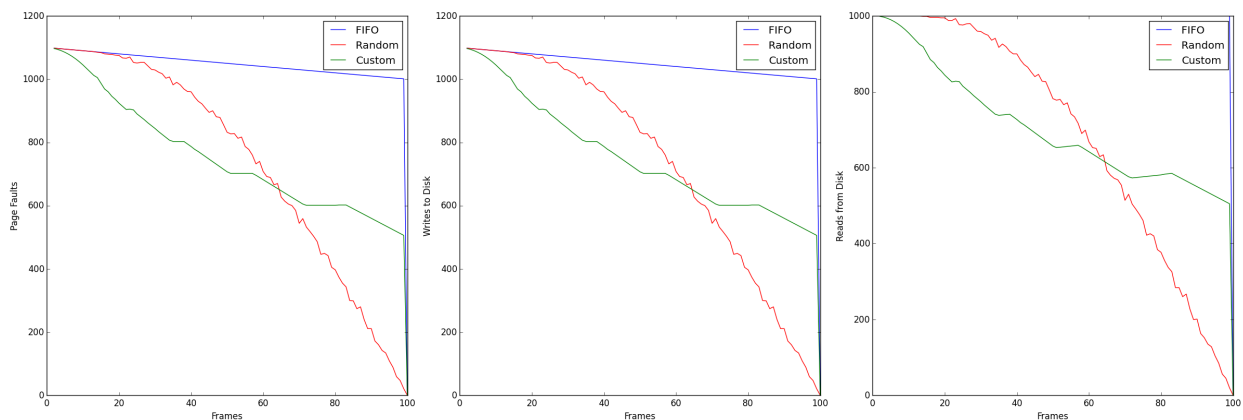
### Focus

Because this program uses random number to choose the data, the three algorithms will behave more regular than in other programs, making the random algorithm the less regular.



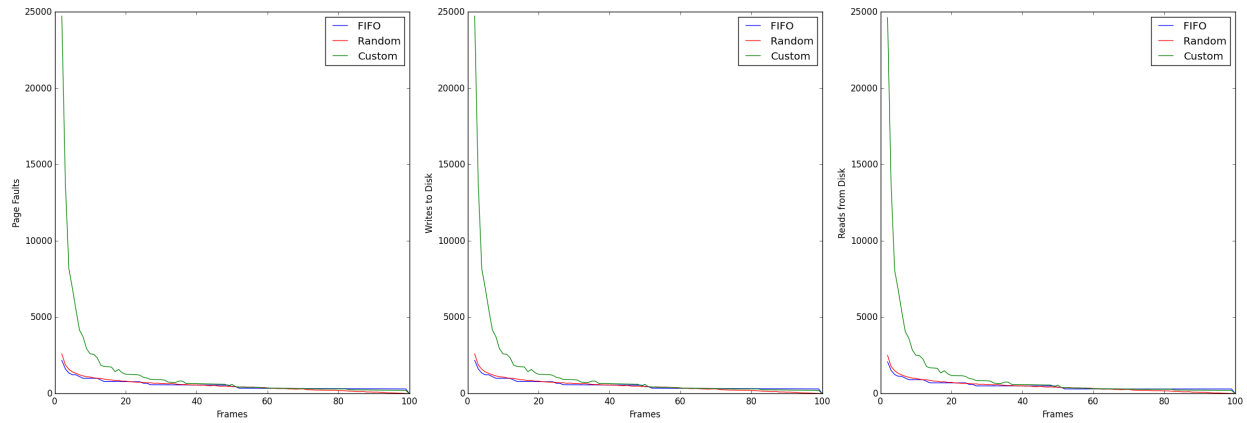
### Scan

The scan program goes through the data in a linear manner, thus making the FIFO algorithm the worst for this case, because it sends to the disk the frame that will be used next. The custom algorithm will send to the disk the first frame that meets the requirements, and the random algorithm will be the most regular.



## Sort

The random algorithm, as in the scan program, will be the most regular. The custom algorithm will behave badly when the difference between the pages and the frames is too high. When this difference is low, the three algorithms will be almost the same, being the random the best of them.



## Conclusion

From the analysis of the different graphs presented above, it's quite noticeable that the "random" algorithm presents the most regular and predictable behavior when facing any of the three processes implemented in this simulation. Thus, it can be concluded that this page handler method is the most reliable in the sense that it's outcome can be easily predicted and developers can model their software or hardware around it.

