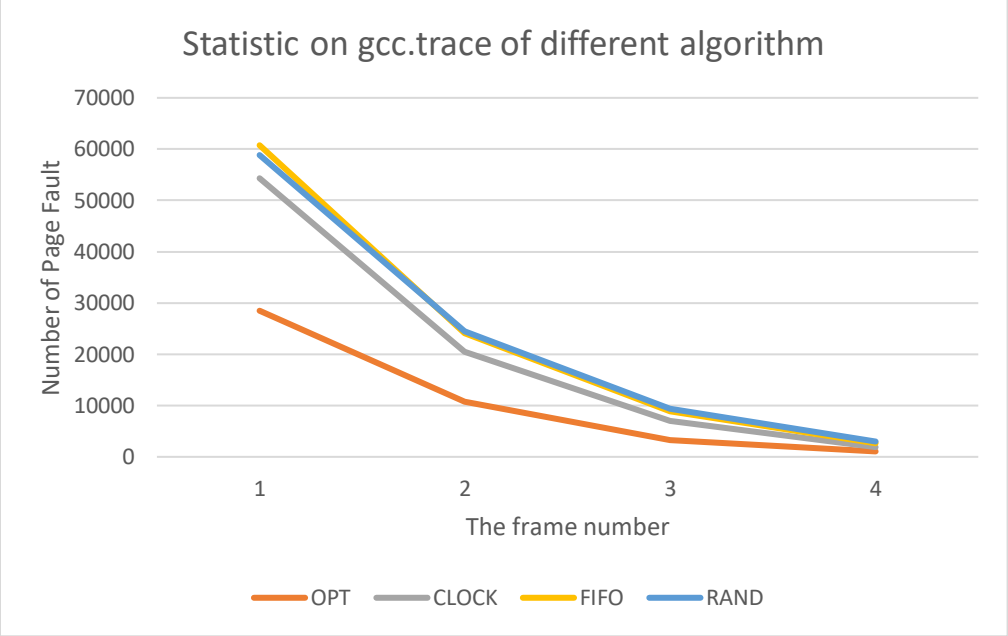
Write up for Project 4: Virtualization
Author: Quan Nguyen
Class: CSC452

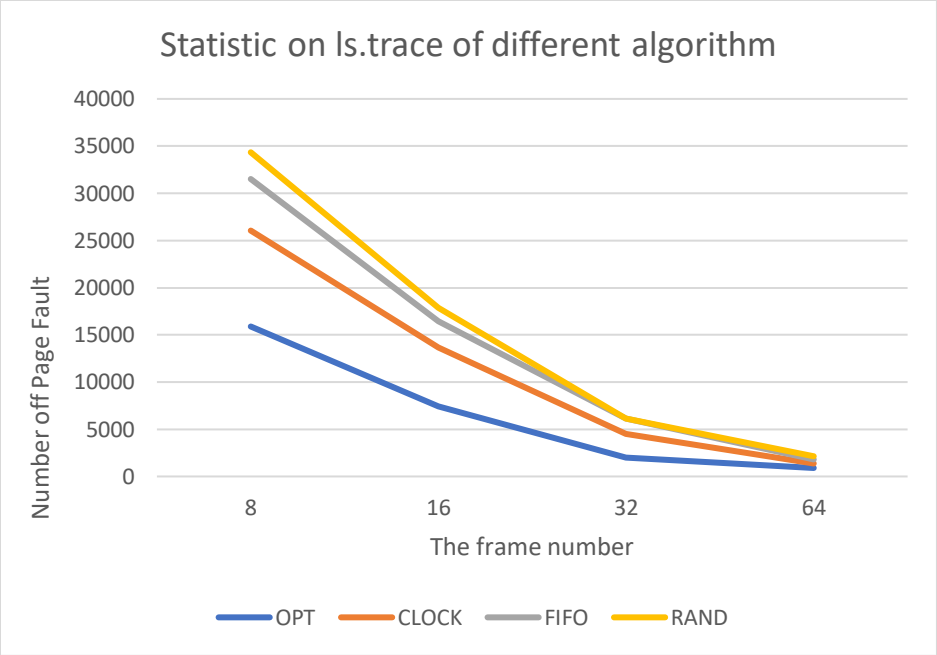
1/ For four of the algorithms, this is the table of the number of page faults for the algorithms with different frames number from 8, 16, 32, and 64 in the trace file of gcc.trace and ls.trace. As the base line of comparison as OPT, we can see that the most appropriate to use in an actual operating system is the CLOCK algorithm which use the better implementation of the second-chance algorithm of the circular buffer since it is superior to the other two algorithms by a large margin on every number of frames. On the other note, if CLOCK implementation is not possible somehow, RAND and FIFO are very close with each other, but FIFO beats RAND most of the time.

In conclusion, CLOCK algorithm would be the best for real world operating system implementations.



Frames	OPT	CLOCK	FIFO	RAND
8	28492	54319	60754	58841
16	10776	20458	24075	24442
32	3309	7022	8917	9416
64	1063	1903	2657	2989

Table1: Stat for gcc.trace



Frames	OPT	CLOCK	FIFO	RAND
8	15888	26050	31514	34343
16	7406	13653	16449	17847
32	2025	4521	6167	6139
64	896	1370	1804	2138

Table 2: Stat for ls.trace

2/ From the table above, and through individual testing on the file gcc.trace of the frame number from 1 to 16, I couldn't find any Belady's anomaly behavior, which is the page faults number increase.