The purpose of this experiment is to see which page replacement algorithm is the most efficient that is tested on built-in programs. I ran the experiment on Hamachi since it is an Ubuntu environment but did the implementation of the code in a Mac machine.

Command Line arguments:

```
./virtmem 100 10 rand sort
./virtmem 100 10 fifo sort
./virtmem 100 10 lru sort

./virtmem 100 50 rand scan
./virtmem 100 50 fifo scan
./virtmem 100 50 lru scan

./virtmem 100 85 rand focus
./virtmem 100 85 fifo focus
./virtmem 100 85 lru focus
```

Number of Page Faults:

	Rand	Fifo	LRU
Page Number: 100 , Frame Number: 10	115	100	100
Page Number: 100 , Frame Number: 50	352	100	100
Page Number: 100 , Frame Number: 85	581	100	100

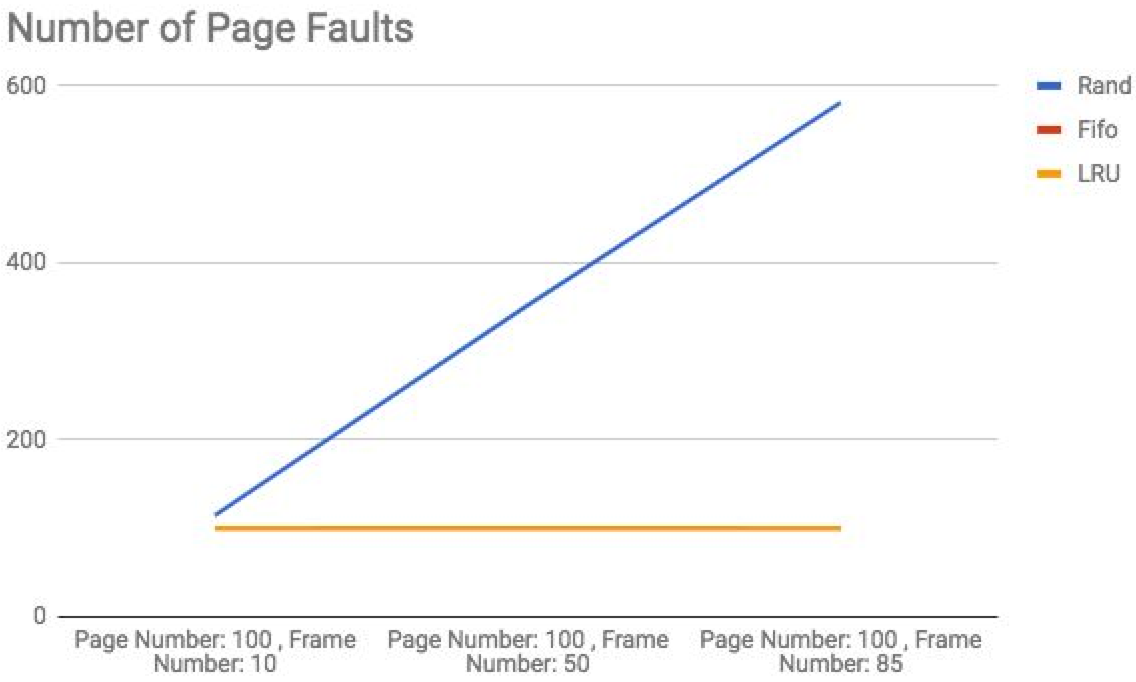
Number of Disk Reads:

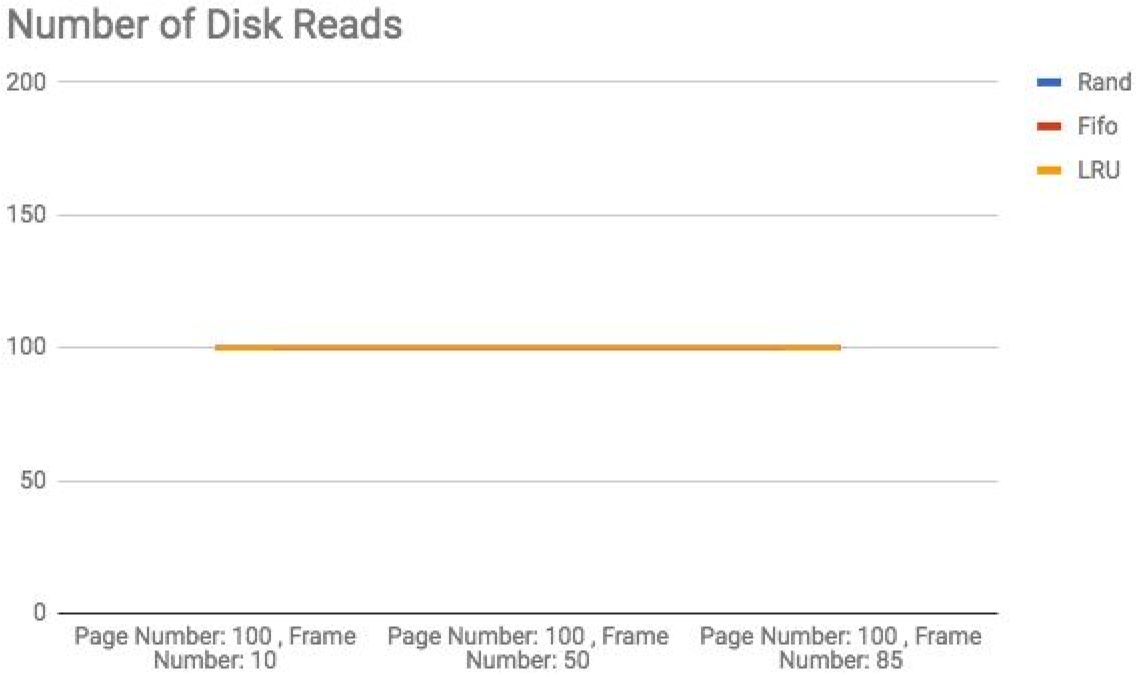
	Rand	Fifo	LRU
Page Number: 100 , Frame Number: 10	100	100	100
Page Number: 100 , Frame Number: 50	100	100	100
Page Number: 100 , Frame Number: 85	100	100	100

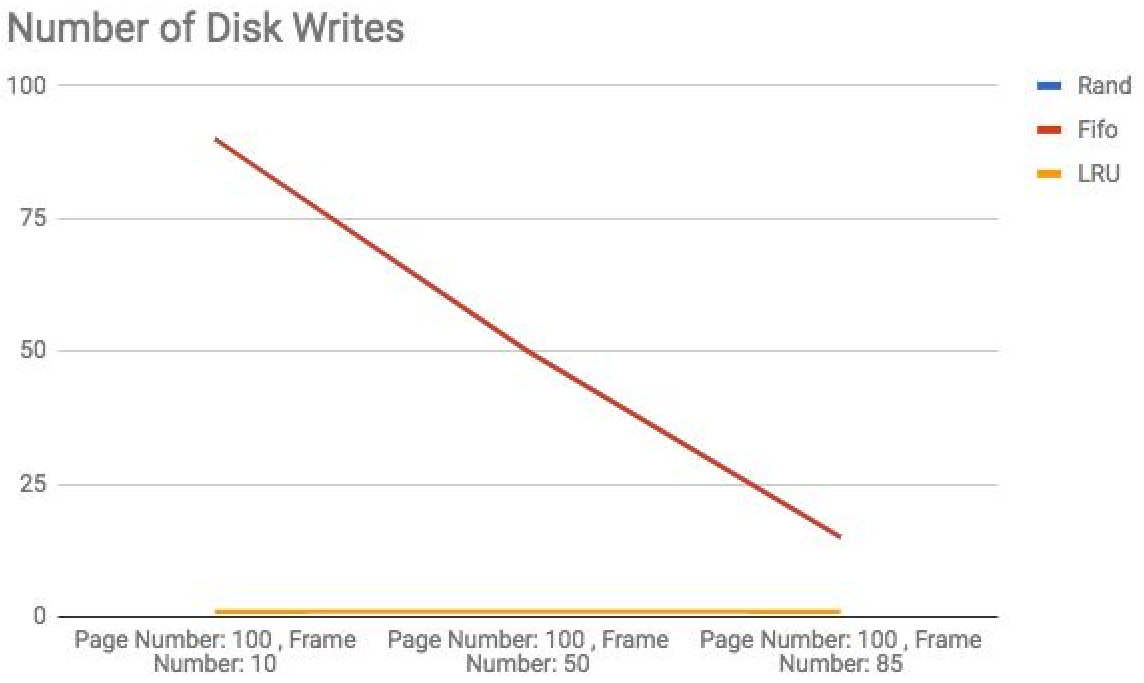
Number of Disk Writes:

	Rand	Fifo	LRU
Page Number: 100 , Frame Number: 10	90	90	1
Page Number: 100 , Frame Number: 50	50	50	1
Page Number: 100 , Frame Number: 85	15	15	1

LRU has pretty bad performance compared to other algorithms but the reason for these performance is because of bad/wrong implementation of the algorithm which in theory LRU is suppose to be better than Random replacement algorithm.






Number of Disk Writes
100
75
50
25
0
Rand
Fifo
LRU
Page Number: 100 , Frame Number: 10
Page Number: 100 , Frame Number: 50
Page Number: 100 , Frame Number: 85