Operating Systems Laboratory

Lab 8

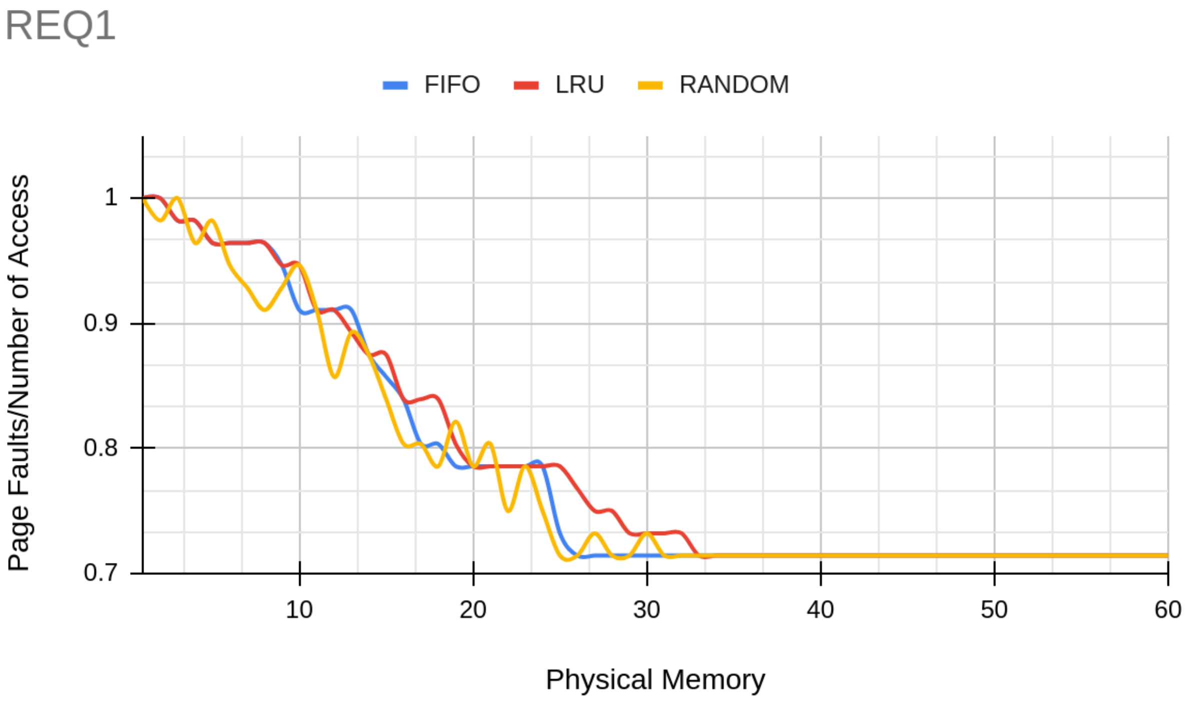
Table of Contents

All of our page access files can be found at: 🗀 Lab 08

Experiment Methodology:

- We ran multiple runs for each policy.
- Swap Space was selected so that no error is thrown out.
- Physical memory space varied from one to the maximum number of pages available for that particular page access file.
- You can take a look at run.sh file for more details.

req1.dat

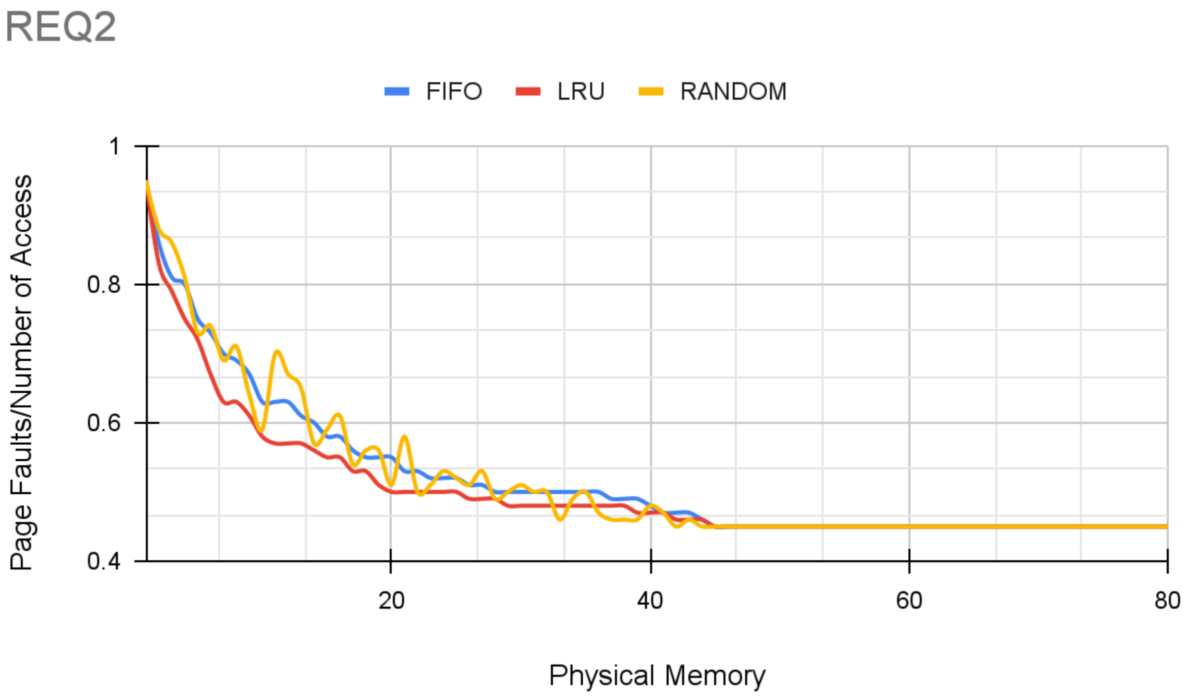


Observation:

```
Configuration for this run:
      Physical Memory Frames 1 to 60
      Request File: req1.dat
```

Here, we can observe that increasing the number of frames in general reduces the number of page faults. At around 40 the number of page faults is constant, and the ones occurring are the necessary page faults that occur at the system start. This trend is observed because as we increase the number of physical memory frames the need for putting the pages into the swap space is reduced.

req2.dat

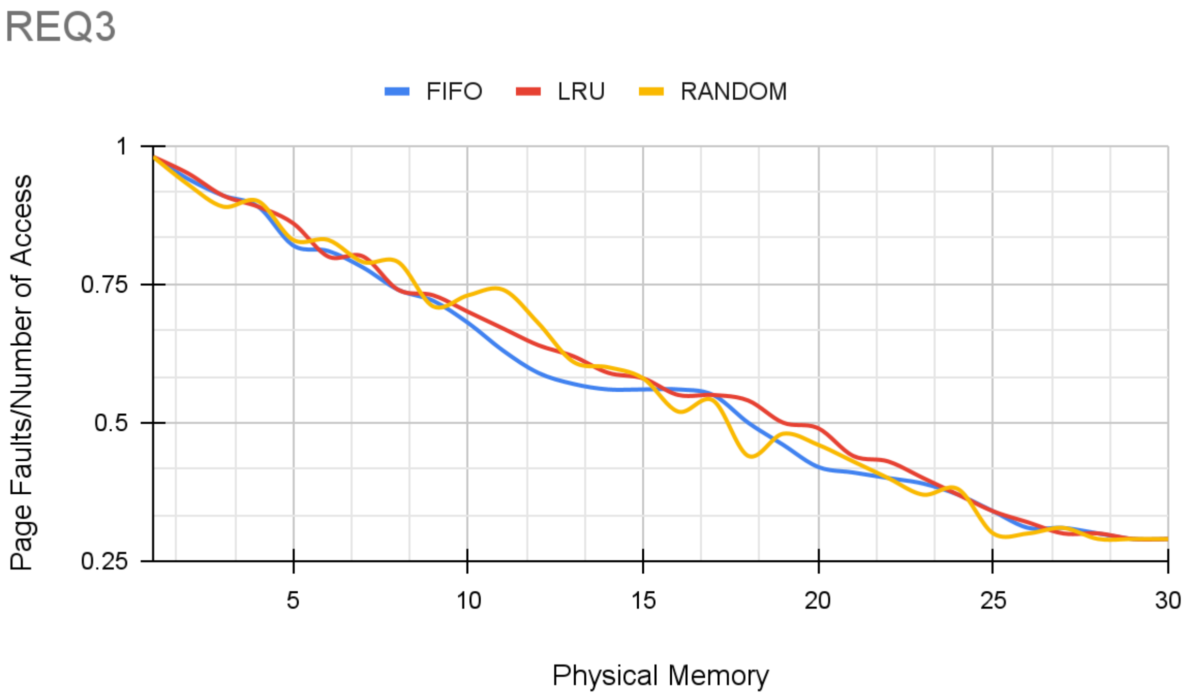


Observation:

```
Configuration for this run:
      Physical Memory Frames 1 to 80
      Request File: req2.dat
```

In this request scenario, we frequently request the previously used pages.
Also, as we increase the number of frames in the memory, the page faults generally reduce as more pages can be stored, thus reducing the need for swap space.
Here, we see that LRU performs well because it stores the frequently accessed pages in the memory.
Random access also performs good here, but its performance is unreliable.

req3.dat



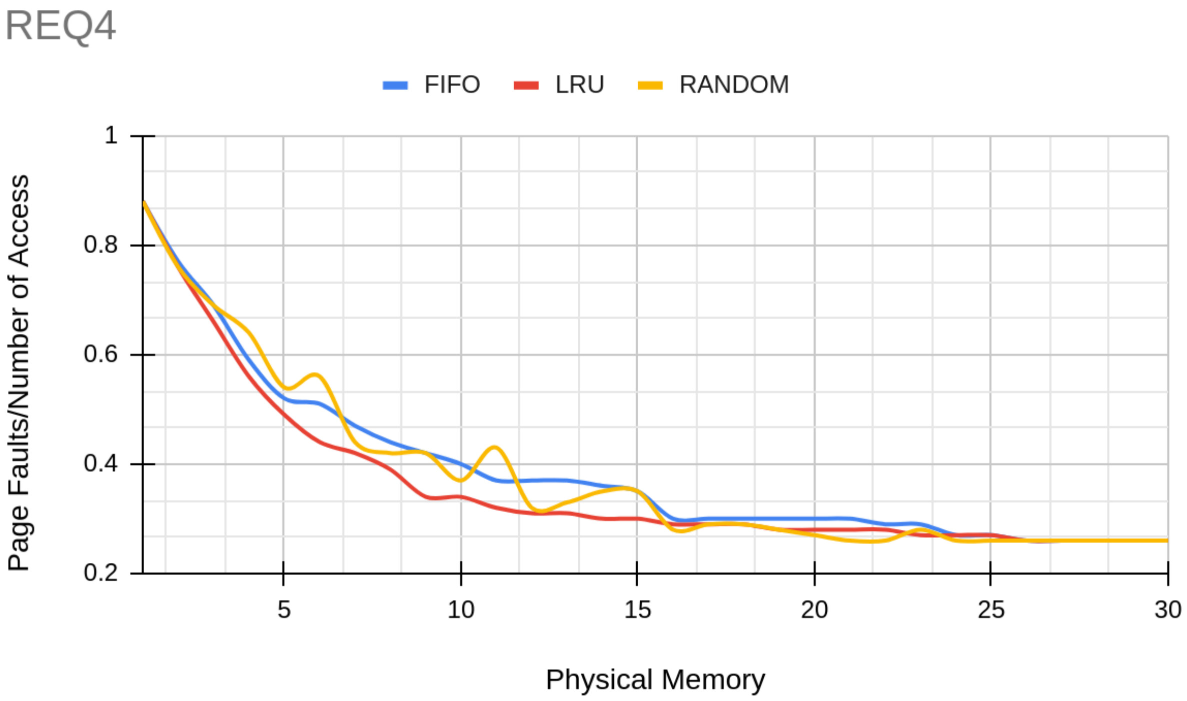
Observation:

```
Configuration for this run:
      Physical Memory Frames 1 to 30
      Request File: req3.dat
```

In this request file the requests are generated completely randomly.

Here we see that all the page access policies perform almost the same. Also, the trend of reducing page faults as we increase the number of available pages continues, irrespective of the page access policy.

req4.dat

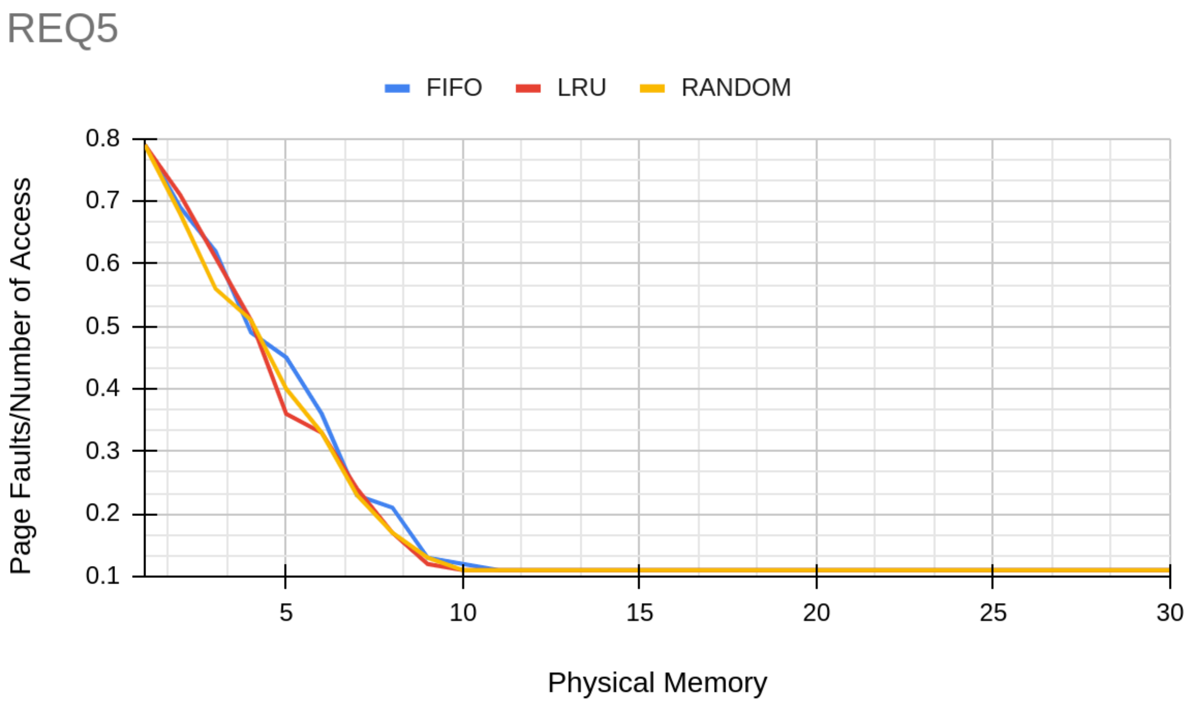


Observation:

```
Configuration for this run:
      Physical Memory Frames 1 to 30
      Request File: req4.dat
```

Some pages are accessed much more frequently in this request sequence than others.
Here we observe that LRU again performs well, as the pages are accessed more frequently. Again random performs well, too, but with extremely unreliable performance. Regardless, the number of page faults becomes constant as we keep on increasing the number of pages that can be kept in the memory.

req5.dat



Observation:

```
Configuration for this run:
      Physical Memory Frames 1 to 30
      Request File: req5.dat
```

Not many new pages are included in this request file. This simulates the situation when the same data is being requested over and over again.

Here we see that almost every policy performs the same, and as not many new pages are accessed, the page faults become constant very soon. At around 10, the physical memory is capable of storing almost every page, and the faults occur due to the empty memory at the system start. Regardless of the policy, they perform almost similarly.
similarly