Random == Grey, NRU == Yellow, Clock == Blue, Optimal == Red

Before starting the discussion about which is the best page replacement algorithm, let’s just go ahead and disqualify random from the running because it sucks. Optimal is obviously not a choice for our operating system because it’s impossible, so we’re kind of left to choose between clock and NRU. In terms of page faults, it can be seen that clock performs better than NRU in both tracefiles. However, NRU overwhelmingly outpaces clock in terms of disk writes, which can be clearly seen in the swim.trace graph for disk writes.

So, which do we choose? My conclusion is NRU. Since disk writes are super expensive I/O bound operations, I’d argue that using NRU would allow us to utilize the CPU’s time more efficiently. Furthermore, although clock beats NRU in terms of page faults, the difference is relatively negligible for processes that are allocated more than 8 frames for their frame table.

Next, let’s discuss how I chose the refresh periods for NRU.

|  |  |  |
| --- | --- | --- |
| Tracefile | Frames | Period |
| gcc.trace | 8 | 20 |
|  | 16 | 90 |
|  | 32 | 180 |
|  | 64 | 640 |
| swim.trace | 8 | 40 |
|  | 16 | 140 |
|  | 32 | 1000 |
|  | 64 | 2500 |

I picked the refresh period by messing around with different values until I noticed a decrease in page faults around a specific range of values in relation to other values I tested. I started homing in on the central value based off this. The only noteworthy conclusion I came to was that you need to increase the refresh period exponentially (don’t know the actual value of the exponent though lol) based off the number of frames.

And that’s pretty much it! Hope you aren’t wanting to die because of all these projects, I commented a ton to make your life a lot easier ☺ hopefully that consideration is reflected in my grade haha. Just kidding lol but ya hope you’re doing ok and thanks for doing all this because I know nobody wants to and I really appreciate you for biting the bullet (and I’m sure Misurda does too). Have a nice 2018 and cheers to the end of 2017!