

Cpre 308

Deeksha Juneja

Lab 9

4/6/2016

Introduction

In this lab, we look at different page replacement algorithms. Some of the systems are fifo, lru, clock and second chance. For the purpose of this lab, first come first serve was implemented as an example. The implementation task for this lab was to make lru which is similar to the fifo and also clock.

Fifo in comparison to lru

Basically fifo is first come and first remove algorithm. So the page that comes in the first, is removed the first. Whereas, lru is least recently used. So, the page that has been least recently used, is removed. The difference is that fifo looks at the placement time, that is the time that the page came in and lru looks at the access time which is the last time the page was used.

My assumption on performance

My assumption is that lru should work better than fifo because we are looking at the access time not the placement time. Hence, if a page came in first but was used recently, it shouldn't be removed as it has more chances of being used again.

Observation

Fifo-seqt vs lru-seqt

Both have a hit rate of 0. Hence, they both appear to be same.

Fifo-rand vs lru-rand

Fifo has a hit rate of 0.261000 vs lru which has a hit rate of 0.258000

Looking at this, it seems that fifo is better than lru due to the higher hit rate.

Fifo-spatl vs lru-spatl

Fifo hit rate = 0.830000 and lru hit rate = 0.839000. Hence, the hit rate of lru is marginally better in this case.

From this observation, it is hard to say which one is better because they have similar performances. It can be said that maybe fifo is better. It would also highly depend on the OS and way pages come in.

Fifo and lru vs Clock

Clock page replacement had a very different implementation. I was rather stuck on how to use the clock hand to accurately point towards the desired page. I had to reset the clock hand in a lot of places. The algorithm given on blackboard was rather helpful. Accessing the hit rate was similar to that of the hit in

fifo but the miss needed a couple of cases to be dealt with. The clock hand points to a page and replaces the page depending on the value of ref. Hence, it seems to go in a circle just like a real clock.

My assumption

Given the more complicated implementation and higher thought process in designing clock, I think that it will have better performance than lru and clock.

Observation

Clock-sept hit rate – 0 which is same as the hit rate of lru and fifo.

Clock-rand has a hit rate of 0.263000 which is marginally better than fifo, making clock a superior algorithm.

Clock-seqt has a hit rate of 0.840000 which is better than lru, making clock again a superior algorithm.

Hence, by observation it seems that clock is better than lru and fifo.

Hence, I think out of these three, clock is the best.

For Extra credit

I experimented with two different file numbers name 6, 12 and 18.

I did my testing on the rand file.

The hit rate for number 6:

1. Fifo 0.206000
2. Lru 0.202000
3. Clock 0.204000

By reducing the number of pages, the hit rate seems to be effected rather negatively. However, Fifo emerges as the winner in terms of performance. Although, there is not a major difference in their performance.

The hit rate for 12 pages:

1. Fifo 0.382000
2. Lru 0.391000
3. Clock 0.391000

By increasing the number of pages, the hit rate seems to jump up. Lru and clock tie on performance. I think increased page number shows the true usefulness of lru in comparison to fifo.

I next increased the page number to 18.

That gave me the winner as Clock which had a hit rate of 0.570000 whereas lru had a hit rate of 0.567000 and fifo had 0.562000.

The hit rate will continue to increase with the number of pages because more number of pages get to stay in the memory and don't need to be replaced. However, this is bad because a lot of memory gets occupied and remains without being used.

It is hard to say which would be optimal irrespective of the algorithm. Though 8 or 10 seem to be a pretty good bet.

Conclusion

In conclusion, this was a pretty interesting and low on stress lab. I learnt a lot about the different types of page replacement algorithms. There are some pretty unique page replacements out there which I got to read about.