Zhenyu Yan

# Project5 report

In this project, I implement a virtual memory simulator in order to understand the behavior of a page table and the page replacement algorithms. Virtual memory allows the execution of processes that are not completely in memory. This is achieved by page table and page replacement techniques. A page table stores the mapping between virtual addresses and physical addresses. A page fault mechanism by which the memory management unit (MMU) can ask the operating system (OS) to bring in a page from the disk. The system we are simulating is 16-bit machine. Below I will talk about the replacement algorithms, efficiency, performance and comparison.

# Implementation

For implementation, since we have a straight forward boiler plate, we need to implement page and TLB in the page table class and replacement FIFO, LRU and CLOCK method.

For FIFO:

```
int next_page;
int clockTime;
```

## FIFO

```
int fifo(){
  return next_page ++%NUM_PAGE;
}
```

## LRU

```
int lru(){
  int used = -1;
  for(int i = 0; i < NUM_FRAME; i++){
    if(frame_table[i].used == 1){
      used = i;
      break;
    }
  }
  if(used == -1){
    used = 0;
  }
  return used;
}
```

## CLOCK

```
int clock(){
   while(frame_table[clockTime].used){
      frame_table[clockTime].used = 0;
      clockTime = clockTime++%NUM_FRAME;
   }
   return clockTime;
}
```

I add an boolean “used” to the frame structure. When the TLB table is hit, I set used = 1, so in the LRU replacement, I can check if it is used or not. If both are not used, it will replace the first one.

This used Boolean also help me to implemented the clock method because if it is used, I can use it to determine If the current frame is used and it will go to the next index of the clock.

# Efficiency/Performance

**RANDOM**

Replacement Policy: 0 - RANDOM

```
[pid 0, W] TLB:miss,        Page:miss,        0x4a8     -> 0xa8
[pid 0, W] TLB:miss,        Page:miss,        0x7ae     -> 0x1ae
[pid 1, W] TLB:miss,        Page:miss,        0x303     -> 0x203
[pid 1, W] TLB:miss,        Page:miss,        0x42e     -> 0x32e
[pid 1, R] TLB:miss,        Page:miss,        0x6ec     -> 0x4ec
[pid 1, R] TLB:hit,         Page:hit,         0x439     -> 0x339
[pid 1, W] TLB:miss,        Page:miss,        0x7d0     -> 0x5d0
[pid 1, W] TLB:miss,        Page:miss,        0x16      -> 0x616
[pid 0, R] TLB:hit,         Page:hit,         0x742     -> 0x142
[pid 0, W] TLB:miss,        Page:miss,        0x6e3     -> 0x7e3
[pid 1, W] TLB:miss,        Page:miss,        0x5eb     -> 0x3eb
[pid 1, R] TLB:miss,        Page:hit,         0x38e     -> 0x28e
[pid 1, R] TLB:miss,        Page:hit,         0x7bc     -> 0x5bc
[pid 1, R] TLB:miss,        Page:miss,        0x11e     -> 0x1e
[pid 0, W] TLB:miss,        Page:miss,        0x3e5     -> 0x6e5
[pid 1, W] TLB:miss,        Page:miss,        0xac      -> 0xac
[pid 1, W] TLB:miss,        Page:hit,         0x52e     -> 0x32e
[pid 0, W] TLB:miss,        Page:miss,        0x471     -> 0x71
[pid 1, W] TLB:miss,        Page:hit,         0x6a6     -> 0x4a6
[pid 1, R] TLB:miss,        Page:miss,        0xd9      -> 0x2d9
====================================
Request: 20
Page Hit: 6 (30.00%)
```

Page Miss: 14 (70.00%)
TLB Hit: 2 (10.00%)
TLB Miss: 18 (90.00%)
Disk read: 14
Disk write: 5

1. Because it is random, every time is different.
2. Compare to the output it provided, it is pretty much similar.

**FIFO**

Replacement Policy: 1 - FIFO
[pid 0, W] TLB:miss, Page:miss, 0x4a8 -> 0xa8
[pid 0, W] TLB:miss, Page:miss, 0x7ae -> 0x1ae
[pid 1, W] TLB:miss, Page:miss, 0x303 -> 0x203
[pid 1, W] TLB:miss, Page:miss, 0x42e -> 0x32e
[pid 1, R] TLB:miss, Page:miss, 0x6ec -> 0x4ec
[pid 1, R] TLB:hit, Page:hit, 0x439 -> 0x339
[pid 1, W] TLB:miss, Page:miss, 0x7d0 -> 0x5d0
[pid 1, W] TLB:miss, Page:miss, 0x16 -> 0x616
[pid 0, R] TLB:hit, Page:hit, 0x742 -> 0x142
[pid 0, W] TLB:miss, Page:miss, 0x6e3 -> 0x7e3
[pid 1, W] TLB:miss, Page:miss, 0x5eb -> 0xeb
[pid 1, R] TLB:miss, Page:hit, 0x38e -> 0x28e
[pid 1, R] TLB:miss, Page:hit, 0x7bc -> 0x5bc
[pid 1, R] TLB:miss, Page:miss, 0x11e -> 0x11e
[pid 0, W] TLB:miss, Page:miss, 0x3e5 -> 0x2e5
[pid 1, W] TLB:hit, Page:hit, 0xac -> 0x6ac
[pid 1, W] TLB:miss, Page:hit, 0x52e -> 0x2e
[pid 0, W] TLB:miss, Page:miss, 0x471 -> 0x371
[pid 1, W] TLB:miss, Page:hit, 0x6a6 -> 0x4a6
[pid 1, R] TLB:hit, Page:hit, 0xd9 -> 0x6d9
==========
Request: 20
Page Hit: 8 (40.00%)
Page Miss: 12 (60.00%)
TLB Hit: 4 (20.00%)
TLB Miss: 16 (80.00%)
Disk read: 12
Disk write: 4

1. As first in first out, the output remain same every time.
2. Compare to the output it provided, it is the same

## LRU

Replacement Policy: 2 - LRU

```
[pid 0, W] TLB:miss,        Page:miss,        0x4a8     -> 0xa8
[pid 0, W] TLB:miss,        Page:miss,        0x7ae     -> 0x1ae
[pid 1, W] TLB:miss,        Page:miss,        0x303     -> 0x203
[pid 1, W] TLB:miss,        Page:miss,        0x42e     -> 0x32e
[pid 1, R] TLB:miss,        Page:miss,        0x6ec     -> 0x4ec
[pid 1, R] TLB:hit,         Page:hit,         0x439     -> 0x339
[pid 1, W] TLB:miss,        Page:miss,        0x7d0     -> 0x5d0
[pid 1, W] TLB:miss,        Page:miss,        0x16      -> 0x616
[pid 0, R] TLB:hit,         Page:hit,         0x742     -> 0x142
[pid 0, W] TLB:miss,        Page:miss,        0x6e3     -> 0x7e3
[pid 1, W] TLB:miss,        Page:miss,        0x5eb     -> 0x1eb
[pid 1, R] TLB:miss,        Page:hit,         0x38e     -> 0x28e
[pid 1, R] TLB:miss,        Page:hit,         0x7bc     -> 0x5bc
[pid 1, R] TLB:miss,        Page:miss,        0x11e     -> 0x31e
[pid 0, W] TLB:miss,        Page:miss,        0x3e5     -> 0xe5
[pid 1, W] TLB:hit,         Page:hit,         0xac      -> 0x6ac
[pid 1, W] TLB:miss,        Page:hit,         0x52e     -> 0x12e
[pid 0, W] TLB:miss,        Page:miss,        0x471     -> 0x671
[pid 1, W] TLB:miss,        Page:hit,         0x6a6     -> 0x4a6
[pid 1, R] TLB:miss,        Page:miss,        0xd9      -> 0xd9
====================================
Request: 20
Page Hit: 7 (35.00%)
Page Miss: 13 (65.00%)
TLB Hit: 3 (15.00%)
TLB Miss: 17 (85.00%)
Disk read: 13
Disk write: 5
```

1. Compare to the outputted it provided, my page hit has one more, so the page miss is one less.
2. My extra hit on page is 0x6a6.

## CLOCK

Replacement Policy: 3 - CLOCK

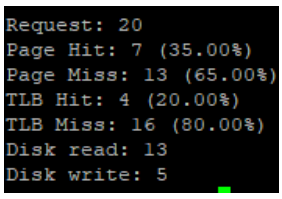
```
[pid 0, W] TLB:miss,        Page:miss,        0x4a8     -> 0xa8
[pid 0, W] TLB:miss,        Page:miss,        0x7ae     -> 0x1ae
[pid 1, W] TLB:miss,        Page:miss,        0x303     -> 0x203
[pid 1, W] TLB:miss,        Page:miss,        0x42e     -> 0x32e
[pid 1, R] TLB:miss,        Page:miss,        0x6ec     -> 0x4ec
[pid 1, R] TLB:hit,         Page:hit,         0x439     -> 0x339
[pid 1, W] TLB:miss,        Page:miss,        0x7d0     -> 0x5d0
[pid 1, W] TLB:miss,        Page:miss,        0x16      -> 0x616
```

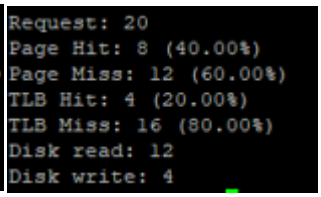
[pid 0, R] TLB:hit, Page:hit, 0x742 -> 0x142
[pid 0, W] TLB:miss, Page:miss, 0x6e3 -> 0x7e3
[pid 1, W] TLB:miss, Page:miss, 0x5eb -> 0xeb
[pid 1, R] TLB:miss, Page:hit, 0x38e -> 0x28e
[pid 1, R] TLB:miss, Page:hit, 0x7bc -> 0x5bc
[pid 1, R] TLB:miss, Page:miss, 0x11e -> 0x1e
[pid 0, W] TLB:miss, Page:miss, 0x3e5 -> 0xe5
[pid 1, W] TLB:hit, Page:hit, 0xac -> 0x6ac
[pid 1, W] TLB:miss, Page:miss, 0x52e -> 0x2e
[pid 0, W] TLB:miss, Page:miss, 0x471 -> 0x71
[pid 1, W] TLB:miss, Page:hit, 0x6a6 -> 0x4a6
[pid 1, R] TLB:hit, Page:hit, 0xd9 -> 0x6d9
==========

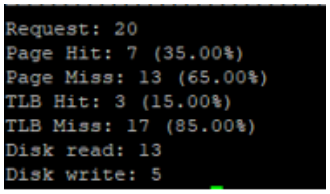
Request: 20
Page Hit: 7 (35.00%)
Page Miss: 13 (65.00%)
TLB Hit: 4 (20.00%)
TLB Miss: 16 (80.00%)
Disk read: 13
Disk write: 4

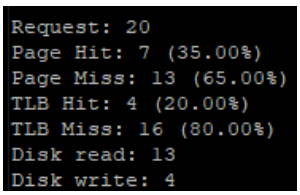
1. Since we do not have an output for clock method, I could not compare.

# COMPARESION

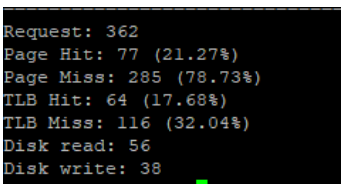


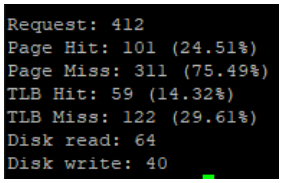



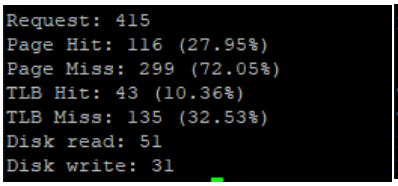



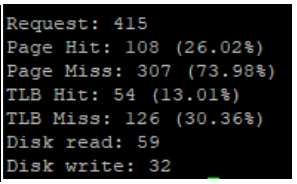




1. As compare, they are all similar in the small input, let's try the large input.




1. As we can see, the LRU and CLOCK require significantly 25% less disk write.
2. LRU disk read is 17% less than FIFO.
3. All the other are very close.

# Conclusion

In conclusion, LRU and clock is better than fifo, but we did not see much different in the result. We might need to generate a bigger data to test out.

Video Link:
https://youtu.be/5GsbSJKChp4 (also in README.txt)