COMP304: Project 3

Owner	Altun Hasanli
Last edited time	@June 3, 2023 2:15 AM

Project completed by Altun Hasanli and Batuhan Altınyollar together.

Part 1

- TLB is implemented as a circular buffer with FIFO replacement.
- Since virtual memory is the same size as the physical memory, no special replacement policy is implemented.

Part 2

```
#define PAGE_FRAMES 256

struct frameentry {
    int logical;
    int references; // for LRU
    int reference_bit; // for Second Chance
};

// FIFO circular buffer
struct frameentry frames[PAGE_FRAMES];
```

- If page faults are less than the page frames, we use one of the free pages.
- Otherwise, for `-p 0`, we use **Second Chance** replacement policy:
  - Initially all reference bits are 0.
  - When a page is referenced, its reference bit is set to 1.
  - When a page needs to be replaced, we start at the current frame index and iterate through the frames until we find a page with a reference bit of 0.
    - If the reference bit is 1, we give it a second chance, set it to 0 and continue iterating.

        - If the reference bit is 0, we evict the page.
- for `-p 1`, we use **LRU** replacement policy:
    - Initially all ages are set to 0.
    - When a page is referenced, its age is set to the current `total_addresses` counter.
    - When a page needs to be replaced, we iterate through the pages and find the page with the lowest age.
    - If two pages have the same age, we evict the first one we find.

Policy Comparison

Since the addresses in the input file provided seldom reference memory locations within the same page, that's why TLB hits are relatively low, and do not vary across replacement policies.