운영체제 실습3

2018320212 컴퓨터학과 김상엽

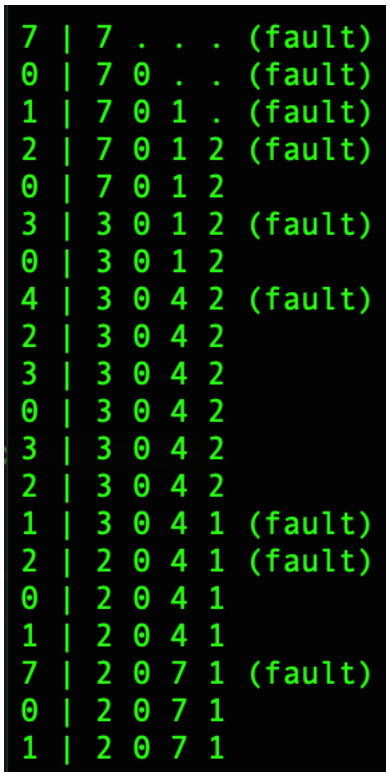
3-1. stack을 이용한 LRU Replacement

```
7 | 7 . . . (fault)              top : 7  : bottom
0 | 7 0 . . (fault)              top : 0 7  : bottom
1 | 7 0 1 . (fault)              top : 1 0 7  : bottom
2 | 7 0 1 2 (fault)              top : 2 1 0 7  : bottom
0 | 7 0 1 2                      top : 0 2 1 7  : bottom
3 | 3 0 1 2 (fault)              top : 3 0 2 1  : bottom
0 | 3 0 1 2                      top : 0 3 2 1  : bottom
4 | 3 0 4 2 (fault)              top : 4 0 3 2  : bottom
2 | 3 0 4 2                      top : 2 4 0 3  : bottom
3 | 3 0 4 2                      top : 3 2 4 0  : bottom
0 | 3 0 4 2                      top : 0 3 2 4  : bottom
3 | 3 0 4 2                      top : 3 0 2 4  : bottom
2 | 3 0 4 2                      top : 2 3 0 4  : bottom
1 | 3 0 1 2 (fault)              top : 1 2 3 0  : bottom
2 | 3 0 1 2                      top : 2 1 3 0  : bottom
0 | 3 0 1 2                      top : 0 2 1 3  : bottom
1 | 3 0 1 2                      top : 1 0 2 3  : bottom
7 | 7 0 1 2 (fault)              top : 7 1 0 2  : bottom
0 | 7 0 1 2                      top : 0 7 1 2  : bottom
1 | 7 0 1 2                      top : 1 0 7 2  : bottom
```

Doubly linked list로 스택을 구현하였다. 노드 생성, 삽입, 삭제, search 등의 기능을 구현해 선언하였다.

먼저 현재 보려는 page가 frame에 있는지부터 확인한다.(is_fault) 만약 이미 frame에 있는 경우, 참조되고 있는 그 페이지를 스택의 맨 위로 올려준다. Frame에 해당 페이지가 없는 경우, 즉 page fault가 발생한 경우에는 우선 frame에 free frame이 있는지 확인한다. Free frame이 있는 경우에는, 그 frame에 해당 페이지를 저장하고 해당 페이지는 스택의 맨위에 삽입한다. 빈칸이 없는 경우에는, frame에 속해 있으며 스택의 가장 아래에 있는 페이지(least recently used page)를 victim page로 선택한다. 그렇게 선택된 victim page를 스택에서 제거하고, 새롭게 들어온 페이지를 스택의 탑으로 push한다.

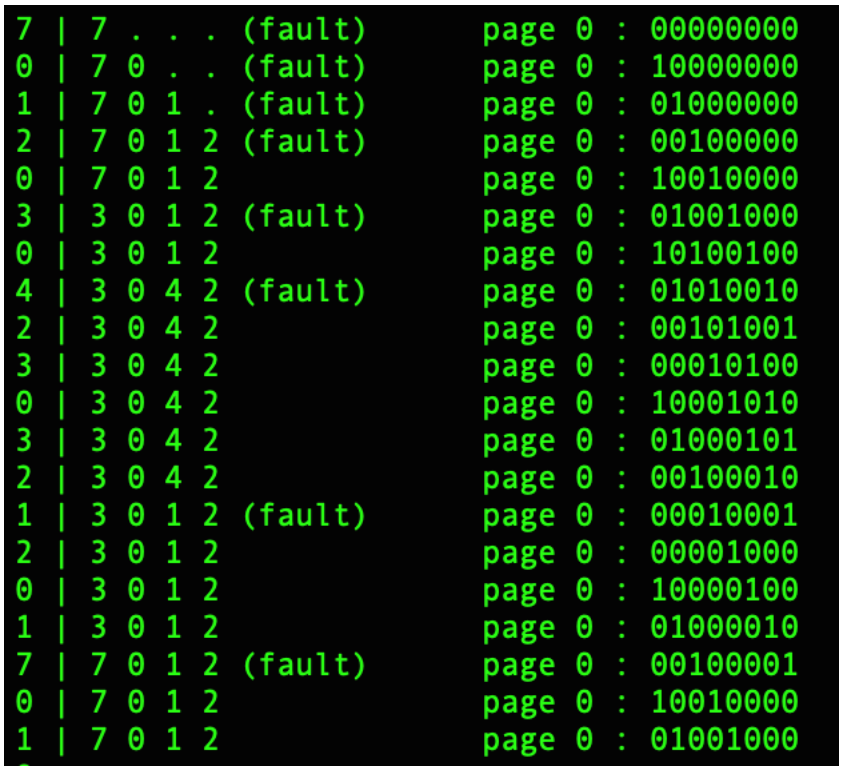
3-2. Clock Algorithm



Clock algorithm 에서는, ref_bit 배열을 선언하여 circular queue 의 역할을 하도록 했다. 해당 frame 에 있는 페이지의 reference bit 를 저장한다. Idx 라는 변수는 pointer 의 역할을 한다.

우선 현재 보려는 page 가 frame 에 있는지부터 확인한다. (is_fault) 해당 페이지가 frame 에 있을 경우, 참조되고 있는 그 페이지의 reference bit 를 1 로 바꿔준다. 해당 페이지가 없을 경우, 즉 pagefault 가 발생한 경우, reference bit 가 0 인 page 를 찾는다. 이 과정에서 reference bit 가 1 인 경우는 0 으로 바꿔주고, idx 의 값을 % frame_sz 해주어 idx 값이 순환하도록 해준다. Reference bit 가 0 인 page 를 찾으면, 그 페이지는 victim page 가 되고 그 frame 에 현재 참조된 page 로 바꿔준다. 이후 reference bit 를 1 로 바꿔주고, idx 값을 한칸 옮겨주어 다음 참조되는 페이지가 볼 idx 로 맞춰준다.

3-3. Additional Reference Bits Algorithm



Circular algorithm 과 마찬가지로 reference bits 들을 저장할 배열 ref_bit 를 선언한다.
현재 페이지를 참조하려는 경우, 일단 모든 reference bits 들을 오른쪽으로 shift 한다. 그리고 현재 참조하려는 페이지의 reference bits 의 최상위 비트를 1 로 만들어 준다. 이는 reference bits 에 이진수 1000,0000 을 or 연산을 해주는 것으로 구현을 하였다. 만약 page fault 가 발생한 경우에는, victim page 를 골라 frame 에서 바꿔줘야 한다. Victim page 는 frame 내에 있는, 비트 값이 가장 작은 page 를 선택한다.