

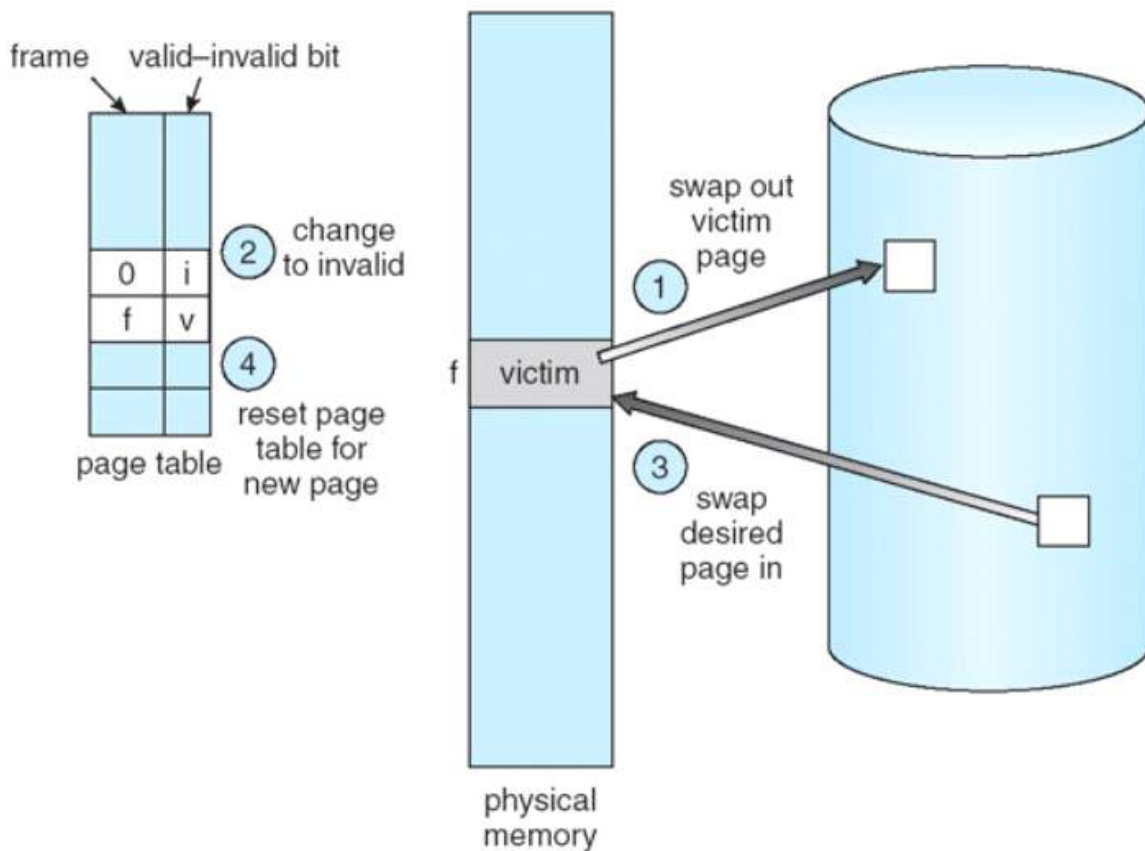
과목명	운영체제	분반	X	담당교수	김화성 교수님
학과	전자통신공학과	학번	2016707079	이름	하상천
과제명: H/W 11 - Page Replacement 알고리즘					

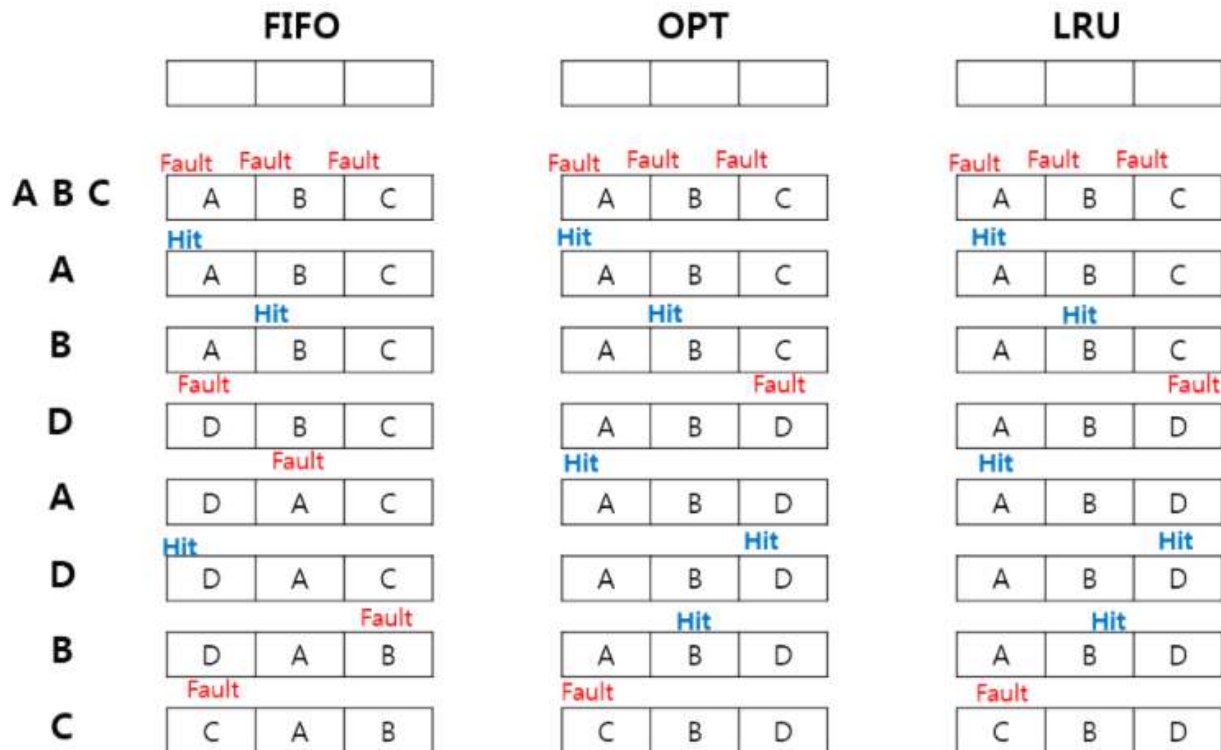
1. 과제설명 (사용자 요구사항 기술: 과제에 대한 설명 및 목표)

Demand paging 시스템에서 FIFO, LRU, Optimal page replacement 알고리즘을 구현하고 성능을 시험하라.

1. page reference string은 각 digit이 random generation을 통하여 획득한 0~9 사이의 숫자로 표현된 최소 15 digit 이상의 reference를 사용할 것.
2. FIFO, LRU, Optimal page replacement 각각에 대하여 page reference string을 적용하여 page fault의 숫자를 구하라.
3. 프로세스에게 할당된 frame의 갯수가 2개부터 7개까지 할당되는 각각의 경우에 성능 평가를 실시함.

2. 알고리즘 및 자료구조 설계 내용





3. 소스코드 설명 (직접 작성한 소스코드중에 핵심 부분을 발췌하여 설명)

```
#include<stdio.h>
#include<stdlib.h>
#define TRUE 1
#define FALSE 0
#define FRAME_NUMBERS 1
#define FIFO 2
#define LRU 3
#define OPTIMAL 4
#define EXIT 5
#define SEED 6
#define PAGE_REFERENCE_NUM 32
int count_frame;
int page_ref[PAGE_REFERENCE_NUM];
int p[7];
int count_fault = 0;
```

define을 통해서 TRUE, FALSE를 1, 0으로 치환하였고, UI에 사용되는 if문에서 가독성을 높이기 위해 번호를 알맞은 설명으로 치환하였다. 또한 PAGE_REFERENCE_NUM을 32로 치환하여 32 digit reference를 사용하였다. frame 수와 page fault 개수를 세는 count_fault, page reference string 배열, frame 배열을 전역변수로 선언하였다.

```

void initialize()
{
count_fault = 0; //전역변수 초기화
for (int i = 0; i < count_frame; i++)
p[i] = 9999; //프레임 초기화
printf("page reference string :"); // print page reference string
for (int i = 0; i < PAGE_REFERENCE_NUM i++) {
printf(" %d", page_ref[i]);
}
printf("\n");
}

int isHit(int data) // data가 p배열에 있으면 hit !
{
int hit = FALSE;
for (int j = 0; j < count_frame; j++)
{
if (p[j] == data)
{
hit = TRUE;
break
}
}
return hit;
}

void print_frames() //현재까지의page frame 정보를 출력
{
for (int k = 0; k < count_frame; k++)
{
if (p[k] != 9999)
printf(" %d", p[k]);
}
}

void print_page_faults() //전체page fault 개수를 출력
{
printf("\nTotal the number of page faults:%d\n\n", count_fault);
}

```

initialize 함수를 통해서 count_fault 전역변수를 매번 0으로 초기화 해주고, frame 배열은 9999로 모두 초기화해주고, page reference string을 출력해준다. isHit 함수를 통해서 frame 배열안에 data가 있는지 확인하고 있으면 TRUE를 반환해준다. print_frames 함수는 9999가 아닌 다른 값이 있다면 data가 들어와 있는 것이기 때문에 현재까지의 page frame 정보를 출력해준다. print_page_faults 함수는 전체 page fault 개수를 출력해준다.

```

int main()
{
int choice;
do {
printf("Page Replacement Algorithm\n
1.Enter the Data\n2.FIFO\n3.LRU\n4.Optimal\n5.Exit\nEnter your choice: ");
scanf("%d", &choice);
if (choice == FRAME_NUMBERS) {
getData();
}
else if (choice == FIFO) {
fifo();
}
else if (choice == LRU) {
lru();
}
else if (choice == OPTIMAL) {
optimal();
}
else if (choice == EXIT) {
break;
}
else {
continue;
}
} while (TRUE);

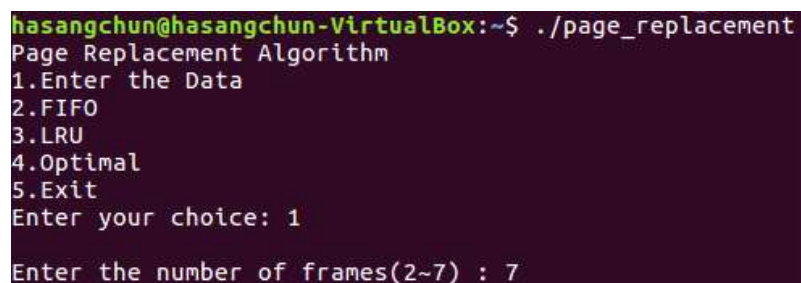
```

main함수에서는 do while문을 통해 UI를 구현하였다. 1에서 5사이의 값이 아니면 continue를 통해 다시 입력하도록 하였고, 5번 EXIT를 눌러야만 반복문이 종료되도록 하였다. 1번을 누르면 frame 개수를 입력받는 화면으로 넘어가고, 2번을 누르면 fifo, 3번을 누르면 lru, 4번을 누르면 optimal 알고리즘을 사용하여 계산된 전체 page fault 값 등이 출력하도록 하였고, 5번을 누르면 반복문을 탈출하여 프로그램이 종료된다.

4. 실행결과 및 설명 (실행 결과를 캡처하여 첨부한 후 설명)

(그림을 문서에 포함, 글자처럼 취급 옵션, 잉크 절약과 잘 보이게 하기위해 그림 반전)

프레임 개수 선택



```

hasangchun@hasangchun-VirtualBox:~$ ./page_replacement
Page Replacement Algorithm
1.Enter the Data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice: 1
Enter the number of frames(2~7) : 7

```

프레임 개수가 2일 때 FIFO, LRU, Optimal 순서대로 결과 사진

```
Enter your choice: 2
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7
5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 4 2
For 9 : 2 9
For 1 : 9 1
For 3 : 1 3
For 2 : 3 2
For 1 : 2 1
For 7 : 1 7
For 3 : 7 3
For 7 : No page fault
For 1 : 3 1
For 6 : 1 6
For 7 : 6 7
For 5 : 7 5
For 4 : 5 4
For 0 : 4 0
For 8 : 0 8
For 5 : 8 5
For 7 : 5 7
For 5 : No page fault
For 8 : 7 8
For 7 : No page fault
For 2 : 8 2
For 0 : 2 0
For 7 : 0 7
For 4 : 7 4
For 7 : No page fault
For 5 : 4 5
For 3 : 5 3
For 3 : No page fault
For 2 : 3 2
Total the number of page faults:27
```

```
Enter your choice: 3
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7
5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 2 4
For 9 : 2 9
For 1 : 1 9
For 3 : 1 3
For 2 : 2 3
For 1 : 2 1
For 7 : 7 1
For 3 : 7 3
For 7 : No page fault
For 1 : 7 1
For 6 : 6 1
For 7 : 6 7
For 5 : 5 7
For 4 : 5 4
For 0 : 0 4
For 8 : 0 8
For 5 : 5 8
For 7 : 5 7
For 5 : No page fault
For 8 : 5 8
For 7 : 7 8
For 2 : 7 2
For 0 : 0 2
For 7 : 0 7
For 4 : 4 7
For 7 : No page fault
For 5 : 5 7
For 3 : 5 3
For 3 : No page fault
For 2 : 2 3
Total the number of page faults:28
```

```

Enter your choice: 4
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 2
For 9 : 9 2
For 1 : 1 2
For 3 : 3 2
For 2 : No page fault
For 1 : 3 1
For 7 : 3 7
For 3 : No page fault
For 7 : No page fault
For 1 : 1 7
For 6 : 6 7
For 7 : No page fault
For 5 : 5 7
For 4 : 5 4
For 0 : 5 0
For 8 : 5 8
For 5 : No page fault
For 7 : 5 7
For 5 : No page fault
For 8 : 8 7
For 7 : No page fault
For 2 : 2 7
For 0 : 0 7
For 7 : No page fault
For 4 : 4 7
For 7 : No page fault
For 5 : 5 7
For 3 : 3 7
For 3 : No page fault
For 2 : 2 7
Total the number of page faults:22

```

프레임 개수가 3일 때 FIFO, LRU, Optimal 순서대로 결과 사진

```

Enter your choice: 2
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 4 2 9
For 1 : 2 9 1
For 3 : 9 1 3
For 2 : 1 3 2
For 1 : No page fault
For 7 : 3 2 7
For 3 : No page fault
For 7 : No page fault
For 1 : 2 7 1
For 6 : 7 1 6
For 7 : No page fault
For 5 : 1 6 5
For 4 : 6 5 4
For 0 : 5 4 0
For 8 : 4 0 8
For 5 : 0 8 5
For 7 : 8 5 7
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 5 7 2
For 0 : 7 2 0
For 7 : No page fault
For 4 : 2 0 4
For 7 : 0 4 7
For 5 : 4 7 5
For 3 : 7 5 3
For 3 : No page fault
For 2 : 5 3 2
Total the number of page faults:23

```

```

Enter your choice: 3
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 9 4 2
For 1 : 9 1 2
For 3 : 9 1 3
For 2 : 2 1 3
For 1 : No page fault
For 7 : 2 1 7
For 3 : 3 1 7
For 7 : No page fault
For 1 : No page fault
For 6 : 6 1 7
For 7 : No page fault
For 5 : 6 5 7
For 4 : 4 5 7
For 0 : 4 5 0
For 8 : 4 8 0
For 5 : 5 8 0
For 7 : 5 8 7
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 2 8 7
For 0 : 2 0 7
For 7 : No page fault
For 4 : 4 0 7
For 7 : No page fault
For 5 : 4 5 7
For 3 : 3 5 7
For 3 : No page fault
For 2 : 3 5 2
Total the number of page faults:22

```

```

Enter your choice: 4
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 9 2
For 1 : 6 1 2
For 3 : 3 1 2
For 2 : No page fault
For 1 : No page fault
For 7 : 3 1 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : 3 6 7
For 7 : No page fault
For 5 : 3 5 7
For 4 : 4 5 7
For 0 : 0 5 7
For 8 : 8 5 7
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 2 5 7
For 0 : 0 5 7
For 7 : No page fault
For 4 : 4 5 7
For 7 : No page fault
For 5 : No page fault
For 3 : 3 5 7
For 3 : No page fault
For 2 : 2 5 7
Total the number of page faults:17

```


프레임 개수가 4일 때 FIFO, LRU, Optimal 순서대로 결과 사진

```
1.Enter the Data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice: 2
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 4 2 9 1
For 3 : 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 9 1 3 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : 1 3 7 6
For 7 : No page fault
For 5 : 3 7 6 5
For 4 : 7 6 5 4
For 0 : 6 5 4 0
For 8 : 5 4 0 8
For 5 : No page fault
For 7 : 4 0 8 7
For 5 : 0 8 7 5
For 8 : No page fault
For 7 : No page fault
For 2 : 8 7 5 2
For 0 : 7 5 2 0
For 7 : No page fault
For 4 : 5 2 0 4
For 7 : 2 0 4 7
For 5 : 0 4 7 5
For 3 : 4 7 5 3
For 3 : No page fault
For 2 : 7 5 3 2
Total the number of page faults:21
```

```
Enter your choice: 3
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 1 4 2 9
For 3 : 1 3 2 9
For 2 : No page fault
For 1 : No page fault
For 7 : 1 3 2 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : 1 3 6 7
For 7 : No page fault
For 5 : 1 5 6 7
For 4 : 4 5 6 7
For 0 : 4 5 0 7
For 8 : 4 5 0 8
For 5 : No page fault
For 7 : 7 5 0 8
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 7 5 2 8
For 0 : 7 0 2 8
For 7 : No page fault
For 4 : 7 0 2 4
For 7 : No page fault
For 5 : 7 0 5 4
For 3 : 7 3 5 4
For 3 : No page fault
For 2 : 7 3 5 2
Total the number of page faults:19
```



```

Enter your choice: 4
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 1
For 3 : 6 3 2 1
For 2 : No page fault
For 1 : No page fault
For 7 : 6 3 7 1
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : No page fault
For 7 : No page fault
For 5 : 5 3 7 1
For 4 : 5 3 7 4
For 0 : 5 0 7 4
For 8 : 5 0 7 8
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 5 0 7 2
For 0 : No page fault
For 7 : No page fault
For 4 : 5 4 7 2
For 7 : No page fault
For 5 : No page fault
For 3 : 3 4 7 2
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:14

```

프레임 개수가 5일 때 FIFO, LRU, Optimal 순서대로 결과 사진

```

Enter your choice: 2
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 4 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 2 9 1 3 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : 9 1 3 7 6
For 7 : No page fault
For 5 : 1 3 7 6 5
For 4 : 3 7 6 5 4
For 0 : 7 6 5 4 0
For 8 : 6 5 4 0 8
For 5 : No page fault
For 7 : 5 4 0 8 7
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 4 0 8 7 2
For 0 : No page fault
For 7 : No page fault
For 4 : No page fault
For 7 : No page fault
For 5 : 0 8 7 2 5
For 3 : 8 7 2 5 3
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:16

```

```

Enter your choice: 3
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 3 4 2 9 1
For 2 : No page fault
For 1 : No page fault
For 7 : 3 7 2 9 1
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : 3 7 2 6 1
For 7 : No page fault
For 5 : 3 7 5 6 1
For 4 : 4 7 5 6 1
For 0 : 4 7 5 6 0
For 8 : 4 7 5 8 0
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 2 7 5 8 0
For 0 : No page fault
For 7 : No page fault
For 4 : 2 7 4 8 0
For 7 : No page fault
For 5 : 2 7 4 5 0
For 3 : 3 7 4 5 0
For 3 : No page fault
For 2 : 3 7 4 5 2
Total the number of page faults:17

```

```

Enter your choice: 4
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 6 4 2 3 1
For 2 : No page fault
For 1 : No page fault
For 7 : 6 4 7 3 1
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : No page fault
For 7 : No page fault
For 5 : 5 4 7 3 1
For 4 : No page fault
For 0 : 5 4 7 3 0
For 8 : 5 4 7 8 0
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 5 4 7 2 0
For 0 : No page fault
For 7 : No page fault
For 4 : No page fault
For 7 : No page fault
For 5 : No page fault
For 3 : 3 4 7 2 0
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:12

```

프레임 개수가 6일 때 FIFO, LRU, Optimal 순서대로 결과 사진

```
Enter your choice: 2
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 6 4 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 4 2 9 1 3 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : 2 9 1 3 7 6
For 7 : No page fault
For 5 : 9 1 3 7 6 5
For 4 : 1 3 7 6 5 4
For 0 : 3 7 6 5 4 0
For 8 : 7 6 5 4 0 8
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 6 5 4 0 8 2
For 0 : No page fault
For 7 : 5 4 0 8 2 7
For 4 : No page fault
For 7 : No page fault
For 5 : No page fault
For 3 : 4 0 8 2 7 3
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:15
```

```
Enter your choice: 3
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 6 4 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 7 4 2 9 1 3
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : 7 6 2 9 1 3
For 7 : No page fault
For 5 : 7 6 2 5 1 3
For 4 : 7 6 4 5 1 3
For 0 : 7 6 4 5 1 0
For 8 : 7 6 4 5 8 0
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 7 2 4 5 8 0
For 0 : No page fault
For 7 : No page fault
For 4 : No page fault
For 7 : No page fault
For 5 : No page fault
For 3 : 7 2 4 5 3 0
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:14
```

```

Enter your choice: 4
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 6 4 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 6 4 2 7 1 3
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : No page fault
For 7 : No page fault
For 5 : 5 4 2 7 1 3
For 4 : No page fault
For 0 : 5 4 2 7 0 3
For 8 : 5 4 2 7 0 8
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : No page fault
For 0 : No page fault
For 7 : No page fault
For 4 : No page fault
For 7 : No page fault
For 5 : No page fault
For 3 : 3 4 2 7 0 8
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:11

```

프레임 개수가 7일 때 FIFO, LRU, Optimal 순서대로 결과 사진

```

Enter your choice: 2
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 6 4 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 6 4 2 9 1 3 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : No page fault
For 7 : No page fault
For 5 : 4 2 9 1 3 7 5
For 4 : No page fault
For 0 : 2 9 1 3 7 5 0
For 8 : 9 1 3 7 5 0 8
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 1 3 7 5 0 8 2
For 0 : No page fault
For 7 : No page fault
For 4 : 3 7 5 0 8 2 4
For 7 : No page fault
For 5 : No page fault
For 3 : No page fault
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:12

```



```

Enter your choice: 3
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 6 4 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 6 4 2 9 1 3 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : No page fault
For 7 : No page fault
For 5 : 6 5 2 9 1 3 7
For 4 : 6 5 2 4 1 3 7
For 0 : 6 5 0 4 1 3 7
For 8 : 6 5 0 4 1 8 7
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : 6 5 0 4 2 8 7
For 0 : No page fault
For 7 : No page fault
For 4 : No page fault
For 7 : No page fault
For 5 : No page fault
For 3 : 3 5 0 4 2 8 7
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:13

```

```

Enter your choice: 4
page reference string : 6 4 2 9 1 3 2 1 7 3 7 1 6 7 5 4 0 8 5 7 5 8 7 2 0 7 4 7 5 3 3 2

For 6 : 6
For 4 : 6 4
For 2 : 6 4 2
For 9 : 6 4 2 9
For 1 : 6 4 2 9 1
For 3 : 6 4 2 9 1 3
For 2 : No page fault
For 1 : No page fault
For 7 : 6 4 2 9 1 3 7
For 3 : No page fault
For 7 : No page fault
For 1 : No page fault
For 6 : No page fault
For 7 : No page fault
For 5 : 5 4 2 9 1 3 7
For 4 : No page fault
For 0 : 5 4 2 0 1 3 7
For 8 : 5 4 2 0 8 3 7
For 5 : No page fault
For 7 : No page fault
For 5 : No page fault
For 8 : No page fault
For 7 : No page fault
For 2 : No page fault
For 0 : No page fault
For 7 : No page fault
For 4 : No page fault
For 7 : No page fault
For 5 : No page fault
For 3 : No page fault
For 3 : No page fault
For 2 : No page fault
Total the number of page faults:10

```

```
Page Replacement Algorithm
1.Enter the Data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice: 99
Page Replacement Algorithm
1.Enter the Data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice: 5
hasangchun@hasangchun-VirtualBox:~$
```

1에서 5사이의 값만 입력하도록 하고, 종료하는 결과 사진

프레임 수	2	3	4	5	6	7
FIFO	27	23	21	16	15	12
LRU	28	22	19	17	14	13
Optimal	22	17	14	12	11	10

전체 page fault 개수

5. 고찰 (과제를 진행하면서 배운점 이나, 시행 착오 내용, 기타 느낀점)

이번 과제는 demand paging system에서 FIFO, LRU, Optimal algorithm을 구현하고 성능을 시험하는 것이었다. page reference string은 random generation을 통해 진행하였는데 동일한 string을 사용하기 위해 seed 값을 정해주었다. 처음에는 15 digit reference를 사용하였는데 성능 차이가 얼마 나지 않아 32 digit reference를 사용하였다. 위의 전체 page fault 개수를 확인해보면, frame 수가 많아지면 많아질수록 page fault 개수가 줄어드는 것을 확인 할 수 있었다. 또한 확실히 FIFO와 LRU 알고리즘 보다는 Optimal 알고리즘의 page fault 개수가 적은 것을 확인 할 수 있었다. seed 값을 바꿀 때마다 page reference string 값이 바뀌기 때문에 성능 차이가 조금씩 있었지만, FIFO랑 LRU는 거의 비슷하게 나왔다. frame수도 한정적이고, reference 수도 제한적이기 때문에 둘의 성능이 비슷하게 나온 것 같다. 이번 과제를 통해 page replacement algorithm의 내용을 공부하는 것과 코드로 구현하는 것은 또 다른 것이라는 것을 알게 되었다. 처음에는 쉽게 구현할 수 있을 줄 알았는데, 막상 구현 해보려고 하니 생각이 잘 나지 않아 여러 가지 상황들을 적어가며 구현하였다. 이번 겨울방학 때 여러 가지 알고리즘들을 코딩으로 구현해보는 연습을 해봐야겠다.

6. 전체 소스코드 (글자크기 9에 줄간격을 120%로 유지하고 한 줄이 너무 길지 않게 작성)

```
#include<stdio.h>
#include<stdlib.h>
#define TRUE 1
#define FALSE 0
#define FRAME_NUMBERS 1
#define FIFO 2
```



```

#define LRU 3
#define OPTIMAL 4
#define EXIT 5
#define SEED 6
#define PAGE_REFERENCE_NUM 32

int count_frame;
int page_ref[PAGE_REFERENCE_NUM];
int p[7];
int count_fault = 0;

void getData()
{
    srand(SEED);
    for (int i = 0; i < PAGE_REFERENCE_NUM i++) {
        page_ref[i] = rand() % 10;
    }
    printf("\nEnter the number of frames(2~7) : ");
    scanf("%d", &count_frame);
}

void initialize()
{
    count_fault = 0; //전역변수 초기화
    for (int i = 0; i < count_frame; i++)
        p[i] = 9999; //프레임 초기화
    printf("page reference string :"); // print page reference string
    for (int i = 0; i < PAGE_REFERENCE_NUM i++) {
        printf(" %d", page_ref[i]);
    }
    printf("\n");
}

int isHit(int data) // data가 p 배열에 있으면 hit !
{
    int hit = FALSE;
    for (int j = 0; j < count_frame; j++)
    {
        if (p[j] == data)
        {
            hit = TRUE;
            break;
        }
    }
}

```

```

return hit;
}

void print_frames() //현재까지의 page frame 정보를 출력
{
for (int k = 0; k < count_frame; k++)
{
if (p[k] != 9999)
printf(" %d", p[k]);
}
}

void print_page_faults() //전체 page fault 개수를 출력
{
printf("\nTotal the number of page faults:%d\n", count_fault);
}

void fifo()
{
initialize();
int k;
for (int i = 0; i < PAGE_REFERENCE_NUM; i++)
{
printf("\nFor %d :", page_ref[i]);

if (isHit(page_ref[i]) == FALSE) //page replacement 발생
{
for (k = 0; k < count_frame - 1; k++)
p[k] = p[k + 1]; // 한칸씩 옮기기

p[k] = page_ref[i]; //배열의 마지막 위치에 값 대입
count_fault++;
print_frames();
}
else
printf(" No page fault");
}
print_page_faults();
}

void optimal()
{
initialize();

```

```

int near[7];
int flag = FALSE;
for (int i = 0; i < PAGE_REFERENCE_NUM i++)
{
    flag = FALSE;
    printf("\nFor %d :", page_ref[i]);

    if (isHit(page_ref[i]) == FALSE)//page replacement 발생
    {

        for (int j = 0; j < count_frame; j++)
        {
            int pg = p[j];
            int found = FALSE;
            for (int k = i; k < PAGE_REFERENCE_NUM k++)
            {
                if (pg == page_ref[k])
                {
                    near[j] = k;
                    found = TRUE;
                    break;
                }
            }
            else
            found = FALSE;
        }
        if (!found)
            near[j] = 9999;
    }
    int max = -9999;
    int replace_index;
    for (int j = 0; j < count_frame; j++)
    {
        for (int n = 0; n < count_frame; n++) {
            if (p[n] == 9999) {
                p[n] = page_ref[i];
                flag = TRUE;
                break;
            }
        }
        if (flag == TRUE) {
            break;
        }
        if (near[j] > max)
        {
            max = near[j];

```

```

replace_index = j;
}
}
if (flag == FALSE) {
p[replace_index] = page_ref[i];
}
count_fault++;
print_frames();
}
else
printf(" No page fault");
}
print_page_faults();
}

void lru()
{
initialize();
int least[50];
for (int i = 0; i < PAGE_REFERENCE_NUM i++)
{
printf("\nFor %d :", page_ref[i]);

if (isHit(page_ref[i]) == FALSE)//page replacement 발생
{
for (int j = 0; j < count_frame; j++)
{
int pg = p[j];
int found = FALSE;
for (int k = i - 1; k >= 0; k--)
{
if (pg == page_ref[k])
{
least[j] = k;
found = TRUE;
break
}
}
else
found = FALSE;
}
if (!found)
least[j] = -9999;
}
int min = 9999;
int replace_index;

```

```

for (int j = 0; j < count_frame; j++)
{
if (least[j] < min)
{
min = least[j];
replace_index = j;
}
}
p[replace_index] = page_ref[i];
count_fault++;
print_frames();
}
else
printf(" No page fault");
}
print_page_faults();
}

int main()
{
int choice;
do {
printf("Page Replacement Algorithm\n
1.Enter the Data\n2.FIFO\n3.LRU\n4.Optimal\n5.Exit\nEnter your choice: ");
scanf("%d", &choice);
if (choice == FRAME_NUMBERS) {
getData();
}
else if (choice == FIFO) {
fifo();
}
else if (choice == LRU) {
lru();
}
else if (choice == OPTIMAL) {
optimal();
}
else if (choice == EXIT) {
break;
}
else {
continue;
}
} while (TRUE);

```

```
return 0;  
}
```

(글자크기는 10으로 유지하고 줄간격도 160%를 유지할 것)