

# 운영체제 프로젝트3 결과보고서

---



수 강 과 목 : 운영체제SWE3004\_41  
담 당 교 수 : 엄영익 교수님  
학 과 : 소프트웨어학과  
학 번 : 2020315798  
이 름 : 최진우  
제 출 일 : 2022년 11월 19일

## 1. 프로젝트 개요

- 1) 목표 : Virtual Memory Management 구현
- 2) 세부 목표
  - Demand paging system을 위한 Page Replacement 기법 구현 및 검증

## 2. 프로젝트 개발 환경

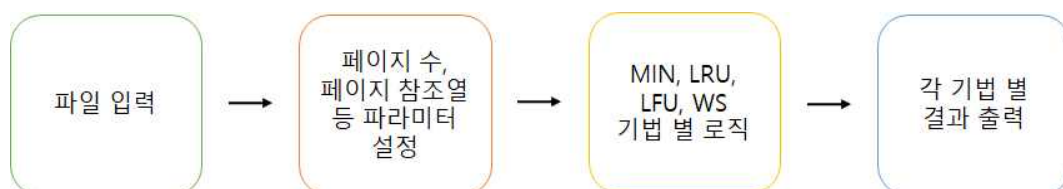
- 1) 개발 언어 : C
- 2) 개발 환경 : Ubuntu 18.04.5 LTS (성균관대학교 인의예지 클러스터)
- 3) 컴파일러 : gcc 7.5.0
- 4) 사용 프로그램 및 목적
  - Putty : 컴파일 및 vi 에디터
  - File Zilla : SFTP를 이용한 백업 및 테스트 케이스 전송
- 5) 라이브러리 및 목적
  - stdio.h : 표준 출력 및 파일 접근
  - stdlib.h : 동적 할당
  - string.h : input txt 파일의 띄어쓰기 토큰화
- 6) 컴파일/실행 방법
 

Makefile과 C파일을 동일 디렉토리에 위치한 뒤 make 명령어를 실행하여  
 'proj3' 이라는 실행 파일 생성되면 ./proj3 실행할텍스트파일.txt 명령어 실행

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ls
input.txt Makefile min1.txt OS41_2022-2_2020315798_ChoiJinWoo_P03.c wsl.txt
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ make
gcc -w -c OS41_2022-2_2020315798_ChoiJinWoo_P03.c
gcc -w -o proj3 OS41_2022-2_2020315798_ChoiJinWoo_P03.o
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ls
input.txt Makefile min1.txt OS41_2022-2_2020315798_ChoiJinWoo_P03.c OS41_2022-2_2020315798_ChoiJinWoo_P03.o proj3 wsl.txt
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./proj3 input.txt
+-----+
| MIN algorithm |
+-----+
+-----+
| Time | Ref. str | Page fault |
+-----+
| 1 | 1 | 1 | 1 |
```

## 3. 프로젝트 설계 및 알고리즘

- 1) 설계 흐름도



## 2) 설계/알고리즘 설명

강의를 통해 접한 Fixed allocation based replacement 중 MIN, LRU, LFU 기법 및 Variable allocation based replacement 중 WS 기법에 대한 구현입니다. 입력 테스트 파일을 argument로 받아 첫 행에서 페이지 개수, 페이지 프레임 개수, 윈도우 사이즈, 페이지 참조열 길이를 각각 설정합니다. 페이지 참조열의 길이만큼 동적 할당을 받아 두 번째 행으로부터 페이지 참조열을 설정합니다.

### [MIN, LRU, LFU 공통]

초기 페이지 프레임 배열을 -1로 초기화하고 현재 시각 기준, 신규 페이지가 이미 할당되어 있는지 확인합니다. 이미 할당된 경우는 넘어갑니다. 하지만 할당되지 않았을 경우, 빈 프레임이 있을 때는 낮은 인덱스부터 순서대로 할당합니다. 빈 프레임이 없는 경우엔 각 기법의 Scheme대로 진행합니다. 두 경우 모두 결과적으로 할당을 하게 되므로 Page fault를 1로 설정합니다.

### [MIN]

앞으로 사용 안 할 경우를 대비해 향후 예정된 시간을 최대 시간+1로 초기화합니다. 페이지 참조열을 통해 현재 각 페이지 프레임에 할당된 페이지의 예정된 사용 시간을 설정하고 비교하여 현재 시각 대비 가장 먼 페이지를 Victim으로 설정합니다. 그리고 해당 페이지 프레임에 신규 페이지를 할당합니다. Tie break의 경우 강의 내용과 같이 페이지 프레임의 낮은 인덱스 우선 할당합니다.

### [LRU]

페이지 참조열을 통해 현재 각 페이지 프레임에 할당된 페이지의 과거 사용 시간을 설정하고 비교하여 현재 시각 대비 가장 먼 페이지를 Victim으로 설정합니다. 그리고 해당 페이지 프레임에 신규 페이지를 할당합니다.

### [LFU]

참조 횟수 계산 배열을 통해 현재 페이지 프레임에 할당된 페이지의 각 참조 횟수를 누적합니다. 각 참조 횟수를 비교하여 제일 작은 값을 구하고 해당 최저 참조 횟수를 가진 페이지 프레임의 Tie break 체크 배열에 1을 할당합니다. 한 가지면 바로 할당하고 여러 개면 해당 프레임들에 대해 LRU를 진행하여 Victim으로 설정합니다. 신규 페이지가 할당된 순간 참조 횟수는 1로 초기화됩니다.

### [WS]

윈도우 사이즈 만큼의 페이지 할당 상태를 저장하기 위해 0, 1의 값을 갖는 페이지 상태 배열을 선언합니다. 처음(1초)부터 설정된 윈도우 사이즈 만큼은 누락되는 페이지 없이 기할당 여부만 체크하며 할당을 진행합니다. 이후 윈도우 사이즈보다 시간이 지나면 본격적으로 직전 페이지의 탈락 여부, 신규 페이지의 Page fault 여부를 범위별로 중복 체크를 진행하며 설정합니다.

## 4. 프로젝트 결과 정책

### 1) input.txt

N	M	W	K	
$r_1$	$r_2$	$r_3$	$r_4$	$r_5 \cdots r_K$

6	3	3	15	
0	1	2	3	2
3	4	5	4	1
3	4	3	4	5

- N은 process가 갖는 page 개수 (최대 100)
- M은 할당 page frame 개수 (최대 20, WS 기법에서는 비사용)
- W는 window size (최대 100, WS 기법에서만 사용)
- K는 page reference string 길이 (최대 1,000)
- " $r_1 r_2 r_3 r_4 r_5 \cdots r_K$ "는 page reference string
  - Page 번호( $r_i$ )는 0번부터 시작

2022-2 운영체제 과제안내서3

### 2) 콘솔 결과

#### - MIN, LRU, LFU 기법

MIN algorithm			
Time	Ref. str	Page fault	
1	4	F	4 - -
2	3	F	4 3 -
3	0	F	4 3 0
4	2	F	4 3 2
5	2		4 3 2
6	3		4 3 2
7	1	F	4 1 2
8	2		4 1 2
9	4		4 1 2
10	2		4 1 2
11	4		4 1 2
12	0	F	0 1 2
13	3	F	3 1 2
# of Page fault :			7

- 기법 종류 출력
- 시간별 각 페이지 참조열, Page fault 발생 여부, Residence Set(표 외부 우측) 출력
- 하단 Page fault 발생 횟수 출력

#### - WS 기법

WS algorithm					
Time	Ref. str	Page fault	Pws	Qws	alloc frame cnt
1	4	F	4		1
2	3	F	3		2
3	0	F	0		3
4	2	F	2		4
5	2			4	3
6	3				3
7	1	F	1	0	3
8	2				3
9	4	F	4		4
10	2			3	3
11	4			1	2
12	0	F	0		3
13	3	F	3		4
8 page fault with 2.92 page frames allocated in average					

- 기법 종류 출력
- 시간별 페이지 참조열, Page fault 발생 여부, Pws, Qws, 할당된 프레임 수, Residence Set(표 외부 우측) 출력
- 하단 Page fault 발생 횟수 및 평균 할당 프레임 수 출력

## 5. 프로젝트 결과 케이스 분석

5-1.

1) 입력 값

```
7 4 4 14
1 2 6 1 4 5 1 2 1 4 5 6 4 5
```

2) 출력 값

[MIN]

MIN algorithm			
Time	Ref. str	Page fault	
1	1	F	1 - - -
2	2	F	1 2 - -
3	6	F	1 2 6 -
4	1		1 2 6 -
5	4	F	1 2 6 4
6	5	F	1 2 5 4
7	1		1 2 5 4
8	2		1 2 5 4
9	1		1 2 5 4
10	4		1 2 5 4
11	5		1 2 5 4
12	6	F	6 2 5 4
13	4		6 2 5 4
14	5		6 2 5 4
# of Page fault :		6	

[LRU]

LRU algorithm			
Time	Ref. str	Page fault	
1	1	F	1 - - -
2	2	F	1 2 - -
3	6	F	1 2 6 -
4	1		1 2 6 -
5	4	F	1 2 6 4
6	5	F	1 5 6 4
7	1		1 5 6 4
8	2	F	1 5 2 4
9	1		1 5 2 4
10	4		1 5 2 4
11	5		1 5 2 4
12	6	F	1 5 6 4
13	4		1 5 6 4
14	5		1 5 6 4
# of Page fault :		7	

[LFU]

LFU algorithm			
Time	Ref. str	Page fault	
1	1	F	1 - - -
2	2	F	1 2 - -
3	6	F	1 2 6 -
4	1		1 2 6 -
5	4	F	1 2 6 4
6	5	F	1 5 6 4
7	1		1 5 6 4
8	2	F	1 5 2 4
9	1		1 5 2 4
10	4		1 5 2 4
11	5		1 5 2 4
12	6	F	1 5 6 4
13	4		1 5 6 4
14	5		1 5 6 4
# of Page fault :			7

[WS]

WS algorithm						
Time	Ref. str	Page fault	Pws	Qws	alloc frame cnt	
1	1	F	1		1	- 1 - - - - -
2	2	F	2		2	- 1 2 - - - -
3	6	F	6		3	- 1 2 - - - 6
4	1				3	- 1 2 - - - 6
5	4	F	4		4	- 1 2 - 4 - 6
6	5	F	5		5	- 1 2 - 4 5 6
7	1			2	4	- 1 - - 4 5 6
8	2	F	2	6	4	- 1 2 - 4 5 -
9	1				4	- 1 2 - 4 5 -
10	4				4	- 1 2 - 4 5 -
11	5				4	- 1 2 - 4 5 -
12	6	F	6		5	- 1 2 - 4 5 6
13	4			2	4	- 1 - - 4 5 6
14	5			1	3	- - - - 4 5 6
7 page fault with 3.57 page frames allocated in average						

운영체제 강의자료 OS10 중 Fixed Allocation에 전반적으로 걸쳐서 나오는 예시입니다. MIN, LRU, LFU의 결과가 강의자료와 같이 나오는 것을 확인할 수 있었습니다. 또한, 강의 내용에서 배운 것과 같이 MIN 기법이 가장 Optimal 한 기법답게 Page fault의 발생 횟수가 가장 낮았습니다. WS 기법에선 같은 페이지 참조열을 사용하되 임의로 Window Size = 4로 진행했습니다. 14초 동안 평균적으로 3.57 페이지 프레임 수와 7번의 Page fault가 발생하는 것을 확인했습니다.

5-2.

1) 입력 값

```
5 3 3 13
4 3 0 2 2 3 1 2 4 2 4 0 3
```

2) 출력 값

[MIN]

MIN algorithm			
Time	Ref. str	Page fault	
1	4	F	4 - -
2	3	F	4 3 -
3	0	F	4 3 0
4	2	F	4 3 2
5	2		4 3 2
6	3		4 3 2
7	1	F	4 1 2
8	2		4 1 2
9	4		4 1 2
10	2		4 1 2
11	4		4 1 2
12	0	F	0 1 2
13	3	F	3 1 2
# of Page fault :			7

[LRU]

LRU algorithm			
Time	Ref. str	Page fault	
1	4	F	4 - -
2	3	F	4 3 -
3	0	F	4 3 0
4	2	F	2 3 0
5	2		2 3 0
6	3		2 3 0
7	1	F	2 3 1
8	2		2 3 1
9	4	F	2 4 1
10	2		2 4 1
11	4		2 4 1
12	0	F	2 4 0
13	3	F	3 4 0



[LFU]

LFU algorithm								
Time	Ref. str	Page fault						
1	4	F	4	-	-			
2	3	F	4	3	-			
3	0	F	4	3	0			
4	2	F	2	3	0			
5	2		2	3	0			
6	3		2	3	0			
7	1	F	2	3	1			
8	2		2	3	1			
9	4	F	2	3	4			
10	2		2	3	4			
11	4		2	3	4			
12	0	F	2	0	4			
13	3	F	2	3	4			
# of Page fault :			8					

[WS]

WS algorithm											
Time	Ref. str	Page fault	Pws	Qws	alloc frame cnt						
1	4	F	4		1	-	-	-	-	4	
2	3	F	3		2	-	-	-	3	4	
3	0	F	0		3	0	-	-	3	4	
4	2	F	2		4	0	-	2	3	4	
5	2			4	3	0	-	2	3	-	
6	3				3	0	-	2	3	-	
7	1	F	1	0	3	-	1	2	3	-	
8	2				3	-	1	2	3	-	
9	4	F	4		4	-	1	2	3	4	
10	2			3	3	-	1	2	-	4	
11	4			1	2	-	-	2	-	4	
12	0	F	0		3	0	-	2	-	4	
13	3	F	3		4	0	-	2	3	4	
8 page fault with 2.92 page frames allocated in average											

운영체제 강의자료 OS10-S66-74 WS 기법의 예시로 나온 페이지 참조열과 윈도우 사이즈로 진행한 결과입니다. 페이지 프레임은 3개로 설정하고 MIN, LRU, LFU를 진행한 결과 역시나 MIN 기법의 Page fault가 가장 낮은 것을 확인할 수 있었습니다.

WS 기법의 경우 강의자료와 같은 결과를 확인할 수 있었지만, 강의자료와 다르게 1초부터 계산한 결과 총 8번의 Page fault 및 2.92 평균 페이지 프레임을 기록했습니다.



[illegible]

[WS]

961	7				61
962	61			46	60
963	7				60
964	18	F	18	22	60
965	58	F	58		61
966	31			67	60
967	2	F	2		61
968	93				61
969	11				61
970	75				61
971	22	F	22		62
972	43	F	43		63
973	94	F	94	0	63
974	9	F	9	76	63
975	80				63
976	15				63
977	88				63
978	69				63
979	39				63
980	60	F	60		64
981	48	F	48	55	64
982	79			64	63
983	22				63
984	54				63
985	45				63
986	27				63
987	96	F	96		64
988	28	F	28		65
989	39			87	64
990	85				64
991	29	F	29		65
992	99	F	99		66
993	46	F	46		67
994	88				67
995	17	F	17		68
996	56	F	56		69
997	72			16	68
998	71	F	71		69
999	1				69
1000	35			36	68
393 page fault with 61.38 page frames allocated in average					

WS의 경우 우측 정보는 길어서 생략했습니다.

모든 파라미터가 최대고 랜덤으로 값을 넣어서 실행해본 결과 역시 MIN기법의 Page fault가 가장 적은 것을 알 수 있었습니다.

994	22				67
995	63				67
996	27			66	66
997	9				66
998	12	F	12		67
999	49			32	66
1000	99			51	65
389 page fault with 60.66 page frames allocated in average					
real 0m0.076s					
user 0m0.023s					
sys 0m0.012s					

또한, 이 경우 진행 시간은 time 명령어로 확인해본 결과 0.076초가량이 소요되었습니다.

## 6. 프로젝트 오류 대응

### 1) 파일이 없는 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input1.txt
Failed to open input file.
```

### 2) 페이지 개수가 0보다 작거나 같고 과제 조건(최대 100)보다 큰 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
-1 3 3 10
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page number
```

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
101 3 3 10
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page number
```

### 3) 페이지 프레임 개수가 0보다 작거나 같고 과제 조건(최대 20)보다 큰 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 -1 3 10
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page frame number
```

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 21 3 10
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page frame number
```

### 4) 윈도우 사이즈가 0보다 작거나 과제 조건(최대 100)보다 큰 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 3 -1 10
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Window Size
```

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 3 101 10
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Window Size
```



5) 페이지 참조열 길이가 0보다 작거나 과제 조건(최대 1000)보다 큰 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 3 3 -1
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page Ref Str number

2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 3 3 1001
1 2 3 4 5 4 9 8 3 6
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page Ref Str number
```

6) 페이지 참조열의 페이지가 설정된 페이지 개수의 인덱스 범위(0~갯수-1)를 벗어난 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 3 3 10
5 2 3 4 2 4 2 0 3 4
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page Ref Str
```

7) 선언한 페이지 참조열의 길이보다 페이지 참조열 내용(2행에서 설정)이 짧은 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 1 3 5
4 2 3
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ ./m input.txt
Invalid input : Page Ref Str mismatch
```

8) 선언한 페이지 참조열의 길이보다 페이지 참조열 내용(2행에서 설정)이 긴 경우

```
2020315798@swui:/home/2020315798/2022-2/OS/proj3$ cat input.txt
5 1 3 1
4 2
```

코드 내에서 선언한 길이 만큼만 자동으로 할당