

과제 4 - 30%

가상메모리 관리를 위해 사용되는 주요 페이지 교체 알고리즘들의 동작을 확인할 수 있는 시뮬레이터 프로그램 구현

*** 4가지 페이지 교체 알고리즘들의 동작을 확인하는 C 프로그램 구현 (100점 만점, 총 과제 점수의 30%. 50점 이상 Pass)**

- 가상메모리 관리를 위해 사용되는 4가지 페이지 교체 알고리즘 Optimal, FIFO, LRU, Second-Chance(One-handed Clock)를 C언어로 프로그래밍하여 각 페이지 교체 알고리즘에 대한 동작 결과를 출력하는 시뮬레이터 구현하기
 - Optimal: 25점, FIFO: 20점, LRU: 25점, Second-Chance: 25점
 - 구현 과정 설명, 실행 화면 스크샷, 실행 결과 분석 내용을 포함하는 보고서 작성 (5점)

구현 요구사항

- 리눅스 환경에서 GCC 컴파일러로 프로그래밍한다.
- 4가지 페이지 교체 알고리즘 Optimal, FIFO, LRU, Second-Chance(One-handed Clock) 구현 시 요구사항
 - 가상주소의 길이, 페이지(프레임) 크기, 물리 메모리의 크기는 실행 시 입력을 받을 수 있어야 함. 가상주소의 길이는 18bits ~ 20bits 까지 사용 가능하고, 페이지(프레임)의 크기는 1KB, 2KB, 4KB 중에서 하나를 사용하며, 물리 메모리의 크기는 32KB, 64KB 중에서 하나를 사용한다.
 - 참조되는(참조는 Read만 있으며 Write는 없다고 가정함) 가상주소(Virtual Address) 스트리밍을 입력받는 방식은 (1) 입력 파일 이름을 입력받아서 해당 파일을 사용하거나 (2) 랜덤 함수를 사용하여 가상주소 스트리밍을 저장하고 있는 input.in 파일을 생성한 후 이를 입력으로 사용한다.
 - 입력 파일을 통해 5,000개의 가상주소를 입력받으며, 가상주소는 10진수로 표현된다.
 - 시뮬레이터는 입력되는 가상주소에 대응하는 출력 결과를 적용된 페이지 교체 알고리즘에 따라 output.opt 또는 output.fifo 또는 output.lru 또는 output.sc 파일 중 하나의 파일에 출력하고, 파일의 마지막에는 Page Faults 발생 횟수를 출력한다. 이때, 파일에 출력되는 물리주소는 10진수로 표현한다.
 - 비어 있는 메모리 프레임이 있는 2개 이상 있는 경우에는, 가장 낮은 주소를 가지는 메모리 프레임에 페이지를 저장한다.
 - Optimal, LRU, Second-Chance 구현 시 교체 대상이 되는 메모리 프레임이 2개 이상인 경우에는 대상이 되는 메모리 프레임들 중에서 FIFO에 따라 하나를 선택한다.
 - 프로그램 실행 시 입력값에 대한 기본적인 예러 처리는 자율적으로 처리하고, 프로그램은 시뮬레이션 후 자동으로 종료되도록 구현한다.

- 실행화면 예시

```
A. Simulation에 사용할 가상주소 길이를 선택하시오 (1. 18bits 2. 19bits 3. 20bits): 1
B. Simulation에 사용할 페이지(프레임)의 크기를 선택하시오 (1. 1KB 2. 2KB 3. 4KB): 3
C. Simulation에 사용할 물리메모리의 크기를 선택하시오 (1. 32KB 2. 64KB): 1
D. Simulation에 적용할 Page Replacement 알고리즘을 선택하시오
(1. Optimal 2. FIFO 3. LRU 4. Second-Chance): 2
E. 가상주소 스트링 입력방식을 선택하시오
(1. input.in 자동 생성 2. 기존 파일 사용): 1
```

(E 단계에서 1을 선택한 경우 시뮬레이션 수행 후 출력 파일 생성하고 프로그램 종료)

(E 단계에서 2를 선택한 경우)

```
A. Simulation에 사용할 가상주소 길이를 선택하시오 (1. 18bits 2. 19bits 3. 20bits): 1
B. Simulation에 사용할 페이지(프레임)의 크기를 선택하시오 (1. 1KB 2. 2KB 3. 4KB): 3
C. Simulation에 사용할 물리메모리의 크기를 선택하시오 (1. 32KB 2. 64KB): 1
D. Simulation에 적용할 Page Replacement 알고리즘을 선택하시오
(1. Optimal 2. FIFO 3. LRU 4. Second-Chance): 2
E. 가상주소 스트링 입력방식을 선택하시오
(1. input.in 자동 생성 2. 기존 파일 사용): 2
F. 입력 파일 이름을 입력하시오:
```

(시뮬레이션 수행 후 출력 파일 생성하고 프로그램 종료)

- 입력 파일 예시 (입력 파일에는 5,000개의 가상주소가 (10진수) 한줄씩 들어 있음)

```
62142
214
8927
63589
164587
19
87953
200156
62687
98750
123658
391
135479
136
=====
4179
```

- 출력 결과 예시

(출력 파일에는 입력된 5,000개의 가상주소에 대해, 가상주소, 페이지 번호, 프레임 번호, 물리주소, Page Fault 발생 유무(H/F)를 한 줄씩 출력한 후, 파일의 마지막에는 Page Faults 발생 횟수를 출력해야 함)

No.	V.A.	Page No.	Frame No.	P.A.	Page Fault
1	62142	15	0	702	F
2	214	0	1	4310	F
3	8927	2	2	8927	F
4	63589	15	0	2149	H
5	164587	40	3	13035	F
6	19	0	1	4115	H
7	87953	21	4	18321	F
8	200156	48	5	24028	F
9	62687	15	0	1247	H
10	98750	24	6	25022	F
11	123658	30	7	29450	F
12	391	0	1	4487	H
13	135479	33	0	311	F
14	136	0	1	4232	H
=====					
5000	4179	1	6	24659	F
Total Number of Page Faults: 1,074					

▶ 제출 요구사항

- 12월 4일(월요일) 자정 전까지 제출해야 함. 기한 이후에는 과제 제출 불가
- 리눅스에서 GCC 컴파일러로 테스트할 것임
- 소스코드는 (Makefile 포함) 하나의 zip 파일로 압축하여 제출하고, 보고서는 별도의 PDF 파일로 제출할 것
- 보고서에는 본인이 수행한 작업을 (소스코드 설명 포함) 전체적으로 설명해야 하고, 과제 수행을 확인할 수 있는 실행 결과 스냅샷 및 실행 결과 분석 내용을 (각 기법 간의 Page Faults 발생 횟수에 대한 비교, 분석) 반드시 포함하여야 함
- 제출 요구사항 미준수 시에는 10% 감점 처리함