

운영체제 Page Replacement Simulation 보고서

미디어학과 201721107 박성범

2.1. 개요

Page Replacement 알고리즘을 시뮬레이션하는 프로그램. frame 10개에 reference string 1000개가 접근한다. FIFO, Optimal, LRU Ref Count, LRU Second Chance 총 4가지 알고리즘을 시뮬레이션하며, page fault가 발생하면 event.log 파일에 로그를 남긴다.

2.2. 프로그램 설계 및 주요 부분

2.2.1. 전체 함수

```
void initializeRefStringUni(int *refString, int max);
void initializeRefStringNorm(int *refString);
void initializeFrame(int *frame);
double generateNormalDistRand();

void printRefString(int *refString);
void printFrame(int *frame);

void testFifo(int *refString, int *frame);
void testOptimal(int *refString, int *frame);
void testLruRefCount(int *refString, int *frame);
void testSecondChance(int *refString, int *frame);
```

reference string과 frame을 초기화하는 함수와 이 둘을 출력하는 함수를 만들었고, 각 replacement 알고리즘별로 함수를 만들었다.

각 알고리즘 함수는 page replacement 알고리즘의 핵심 메커니즘을 구현했으며, 가장 먼저 전체 프레임을 순회하며 free frame이나 swap in하려는 페이지와 같은 페이지가 있는지 확인한다. 만약 없다면 page fault가 발생했다고 판단하고 해당 알고리즘이 가진 replacement 작업을 실행한다.

2.2.2. normal distribution generator

```
/**
 * Generate a random number based on normal distribution.
 * The function made by pinko_liberal.
 * https://bit.ly/2yagLC0 (Cprogramming.com, 2001)
 */
double generateNormalDistRand() {
```

보다 의미 있는 결과를 관찰하기 위해 generateNormalDistRand 함수를 통해 normal distribution에 따라 난수를 생성했다. 자체 구현이 어려워 Cprogramming.com의 사용자 ‘pinko_liberal’이 작성한 코드를 변형해 사용했다.

2.3. 프로그램 실행 조건

Windows 10에서 테스트했고, 별도의 조건은 필요하지 않았다.

2.4. 프로그램 실행 결과

```
===== 50 =====
[FIFO] Page fault occurring 952 times.
frame: [ 66 118 28 56 141 90 57 93 49 9 ]
[Optimal] Page fault occurring 865 times.
frame: [ 118 34 151 69 66 111 57 128 55 49 ]
[LRU Reference Count] Page fault occurring 950 times.
frame: [ 66 9 141 118 49 93 57 56 90 28 ]
[Second Chance] Page fault occurring 932 times.
frame: [ 152 257 28 72 23 185 73 24 26 110 ]
===== 100 =====
[FIFO] Page fault occurring 943 times.
frame: [ 118 262 150 3 61 56 51 238 15 57 ]
[Optimal] Page fault occurring 869 times.
frame: [ 150 43 28 4 31 3 57 15 42 7 ]
[LRU Reference Count] Page fault occurring 941 times.
frame: [ 150 51 3 262 15 118 56 238 61 57 ]
[Second Chance] Page fault occurring 921 times.
frame: [ 9 124 46 13 58 50 166 2 44 65 ]
===== 150 =====
[FIFO] Page fault occurring 926 times.
frame: [ 3 26 133 43 25 59 168 41 125 115 ]
[Optimal] Page fault occurring 861 times.
frame: [ 133 29 62 59 66 26 28 22 109 1 ]
[LRU Reference Count] Page fault occurring 927 times.
frame: [ 43 168 59 115 25 3 26 125 133 41 ]
[Second Chance] Page fault occurring 928 times.
frame: [ 83 131 1 81 126 180 22 33 165 117 ]
===== 200 =====
[FIFO] Page fault occurring 936 times.
frame: [ 65 2 79 34 28 192 110 97 108 128 ]
[Optimal] Page fault occurring 873 times.
frame: [ 31 43 34 65 159 97 28 17 192 2 ]
[LRU Reference Count] Page fault occurring 937 times.
frame: [ 128 192 108 79 97 2 65 28 34 110 ]
[Second Chance] Page fault occurring 922 times.
frame: [ 68 185 21 38 16 126 20 5 8 47 ]
```

reference string이 50개인 경우부터 1000인 경우까지 20개의 test case를 이용해 시뮬레이션을 구동했다. 각 알고리즘별로 page fault가 발생한 횟수가 출력되었으며, 최종 frame의 결과도

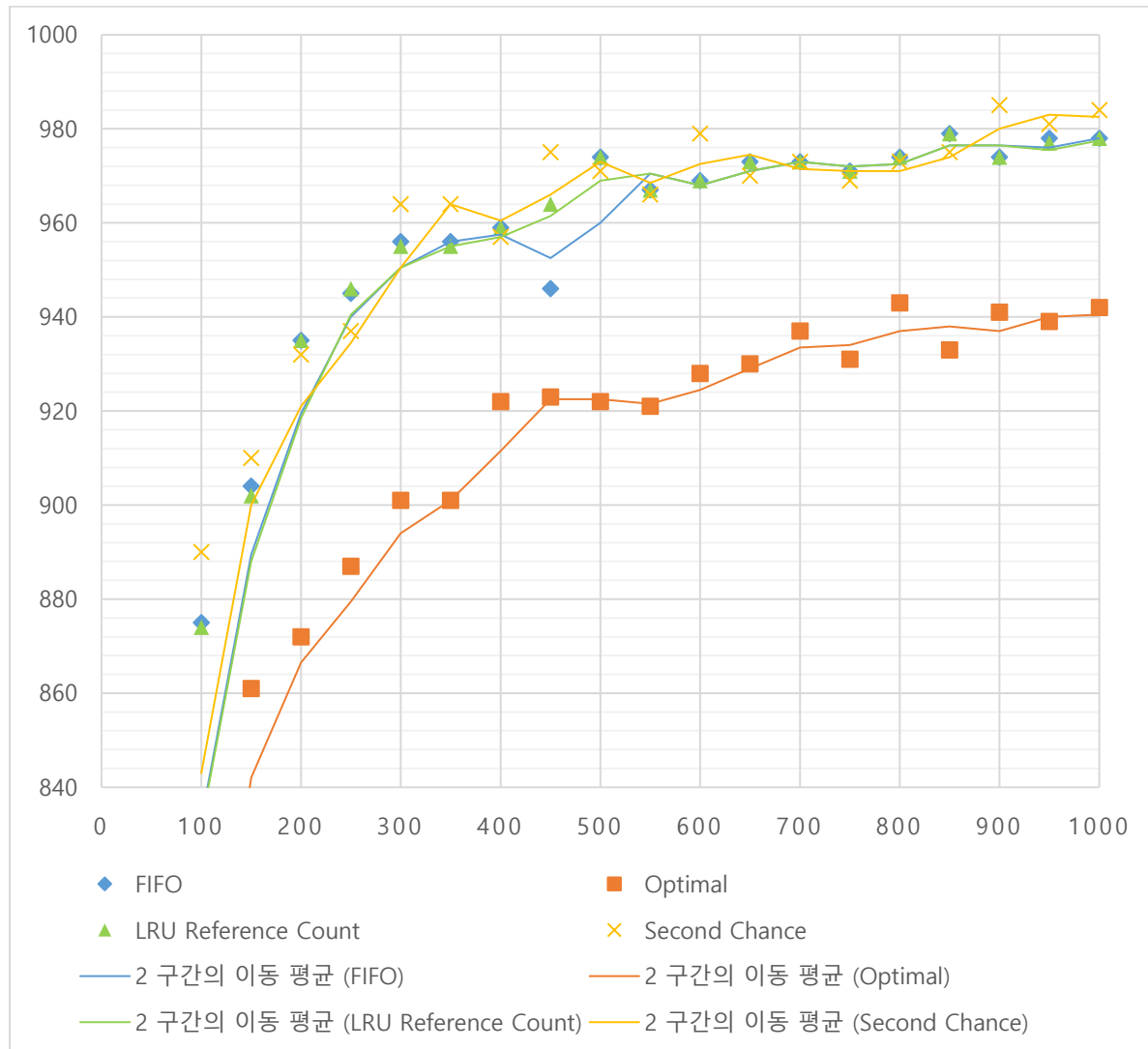
함께 출력했다.

```
[FIFO] Page fault has occurred at 10
[FIFO] Page fault has occurred at 11
[FIFO] Page fault has occurred at 12
[FIFO] Page fault has occurred at 13
[FIFO] Page fault has occurred at 15
[FIFO] Page fault has occurred at 16
[FIFO] Page fault has occurred at 17
[FIFO] Page fault has occurred at 18
[FIFO] Page fault has occurred at 19
[FIFO] Page fault has occurred at 20
[FIFO] Page fault has occurred at 21
[FIFO] Page fault has occurred at 22
[FIFO] Page fault has occurred at 23
[FIFO] Page fault has occurred at 24
[FIFO] Page fault has occurred at 25
[FIFO] Page fault has occurred at 26
[FIFO] Page fault has occurred at 27
[FIFO] Page fault has occurred at 28
[FIFO] Page fault has occurred at 29
[FIFO] Page fault has occurred at 30
[FIFO] Page fault has occurred at 31
[FIFO] Page fault has occurred at 32
[FIFO] Page fault has occurred at 33
[FIFO] Page fault has occurred at 34
[FIFO] Page fault has occurred at 35
[FIFO] Page fault has occurred at 36
[FIFO] Page fault has occurred at 37
```

event.log 파일에는 page fault가 발생한 모든 시점에 로그를 남겼다.

2.5. 알고리즘 비교 분석

2.5.1. Uniform Distribution

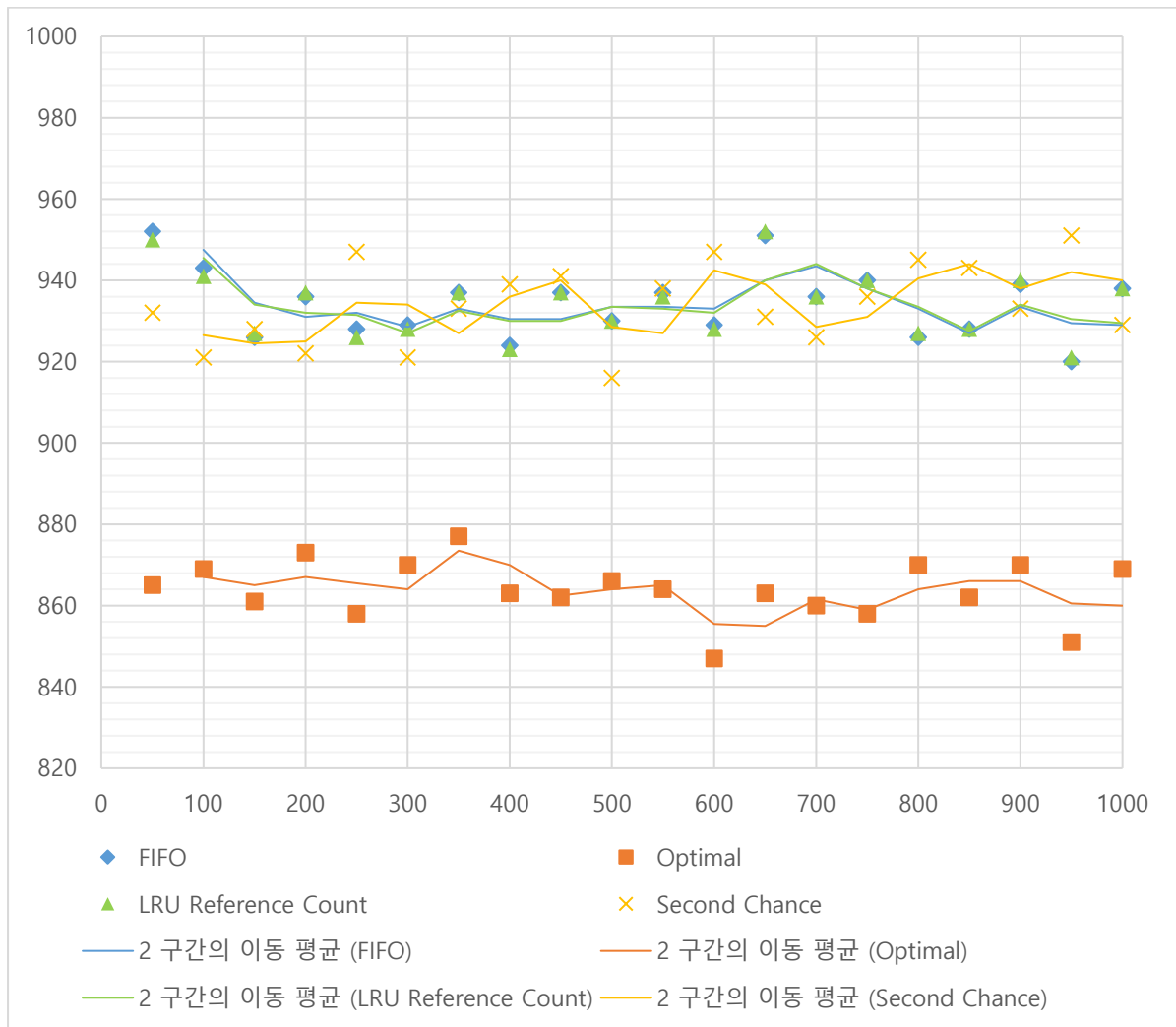


X축은 reference string 값의 범위 0~n이며, Y축은 page fault의 발생 횟수를 의미한다. reference string의 범위를 0~50부터 0~1000까지로 설정해 총 20회 테스트했다.

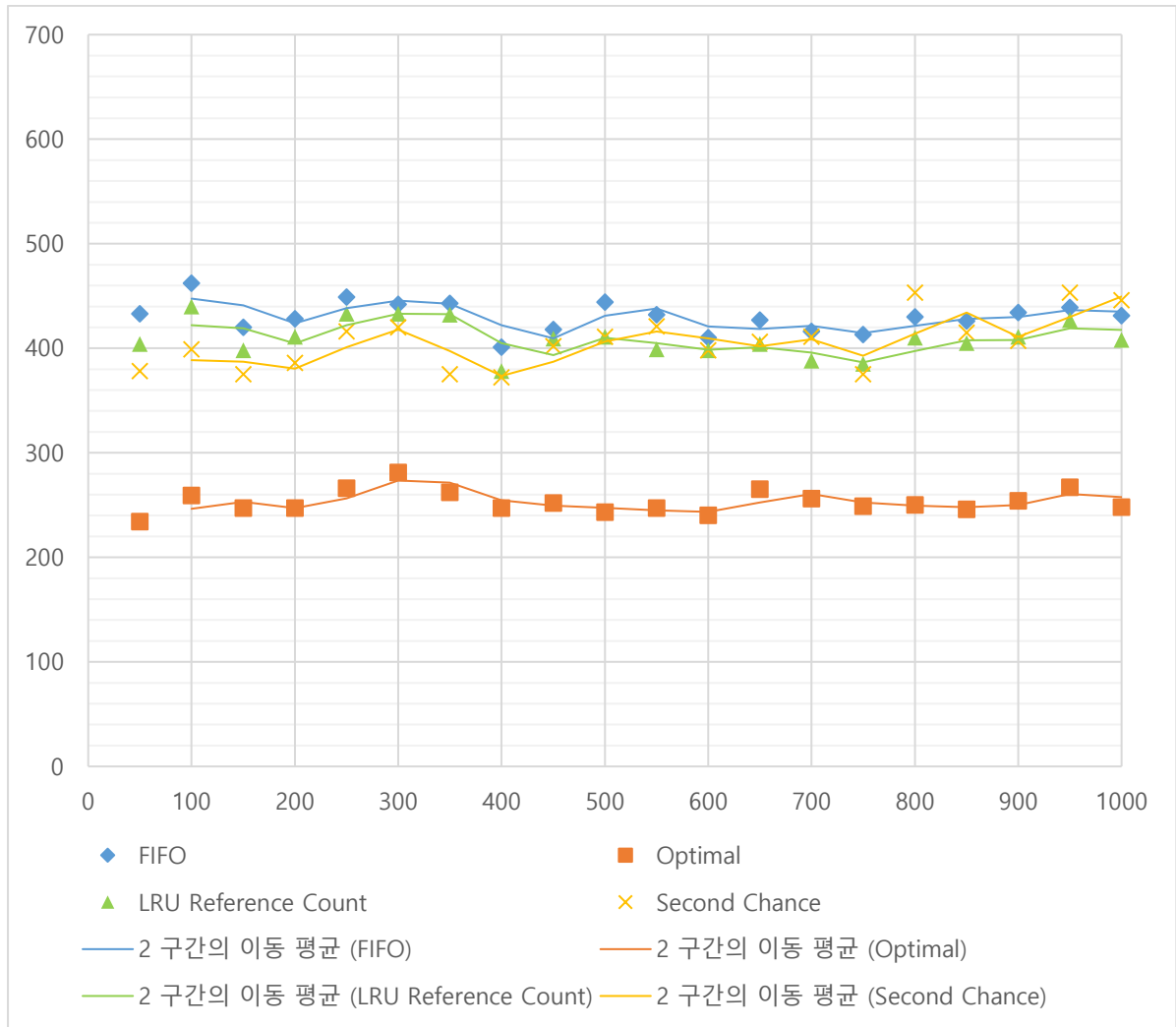
Optimal은 모든 경우에서 page fault 발생 횟수가 가장 낮았다. 다만 실제로는 미래에 사용되지 않을 page를 찾는 데 큰 어려움이 있을 것이며, 시뮬레이션의 경우 매번 reference string을 순회해 빈도가 낮은 page를 찾아야 하므로 시간적 비용을 무시할 수 없을 것으로 보인다.

FIFO와 Second Chance, LRU Reference Count은 모두 비슷한 결과를 보였으며, Second Chance와 LRU Reference Count는 전체 페이지를 순회하는 상황이 발생할 수 있어 FIFO보다 많은 시간과 공간을 요구할 것이라고 생각된다.

2.5.2. Normal Distribution



normal distribution의 경우 알고리즘별로 보다 큰 차이가 생길 것으로 예상했으나 그 차이는 크지 않았다. LRU Reference Count의 경우 FIFO와 거의 같은 결과값을 냈고, Second Chance는 뚜렷한 일관성을 보이지 않았다. 차후 실제 reference string 데이터를 이용해 시뮬레이션해 보면 또 다른 흐름의 결과가 나올 수 있을 것 같다.



frame의 수를 100으로 늘려 시뮬레이션을 다시 수행했다. 이번에는 FIFO, LRU Reference Count, Second Chance, Optimal 순으로 뚜렷하게 page fault의 발생 횟수가 적음을 확인할 수 있었다. 이후 reference string의 수가 늘어날수록 Second Chance 알고리즘의 page fault 발생 빈도가 높아졌고, 반면 LRU Reference Count의 발생 빈도는 낮아지는 경향이 보였다.

frame이 10개인 경우의 시뮬레이션과 비교해 Belady's Anomaly 현상을 관찰하기 위한 시도였으나 기대와는 다른 결과나 나왔다. 보다 다양한 test case를 만들어 여러 차례 시뮬레이션을 수행하면 관찰이 가능하지 않을까 싶다.