요약

캐쉬 메모리에 대한 시뮬레이션을 만들기 위해 시작했으며, 개발을 위해서 비주얼 스튜디오 C++를 이용했다.

1.실습 프로그램의 구성 및 동작 원리

```
typedef struct set_Associative
{
    string Tag;
    string Data = { '₩0' };
    bool v = NULL;
    int LRU = 0;
};
```

캐쉬를 위한 구조체를 선언했습니다.

```
if (argc != 5)
{
    cout << "인수의 갯수가 잘못됐습니다.₩n";
    pirnt();
}


Memory_Trace = argv[1];
Cache_Byte = atoi(argv[2]);
Cache_Block_Byte = atoi(argv[3]);
Associativity = atoi(argv[4]);
```

실행시 인자값을 받고 갯수가 맞지않으면 에러출력

```
set_Associative** sa;

sa = new set_Associative*[Associativity];
for (int i = 0; i < Associativity; i++)
{
    sa[i] = new set_Associative[Cache_Byte / Cache_Block_Byte];
}
```

인자값을 이용하여 캐쉬메모리 생성(2차원 동적 배열)

```
if (sa[N][i].v == NULL)
{
    sa[N][i].v = true;
    sa[N][i].LRU = access;
    sa[N][i].Tag = tc;
    sa[N][i].Data = oc;
    break;
```

블럭이 비어있다면 데이터 삽입.

```
else if (sa[N][i].v == true)
{
    if (sa[N][i].Tag == tc)
    {
        if (sa[N][i].Data == oc)
        {
            hit_count++;
            break;
```

블럭에 모든 데이터가 일치하면 히트카운트 증가

```
        if (minLRU >= sa[N][i].LRU)
        {
            minLRU = sa[N][i].LRU;
            mini = i;
        }
    }
    if (i== (Cache_Byte / Cache_Block_Byte)-1)
    {
        sa[N][mini].LRU = access;
        sa[N][mini].Tag = tc;
        sa[N][mini].Data = oc;
        break;
```

블럭을 끝까지 확인했지만 원하는 데이터가 없는경우 LRU를 이용하여 제일 오래된 블럭 교체

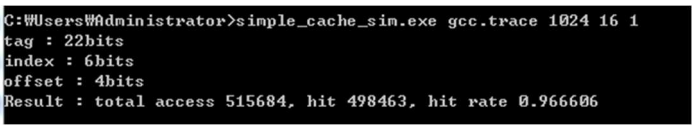
```
offset = log2(Cache_Block_Byte);
index = log2(Cache_Byte / Cache_Block_Byte / Associativity);
tag = 32 - offset - index;
```

인자값을 이용하여 출력변수 계산

실습중에 프로그램을 실행하면서 변수를 넣는 방법과 파일 오픈하는 방법을 몰라서 고생했었고 오류출력과 더불어 시행착오를 많이 겪었습니다. 그리고 2차원 배열을 생성 할때 변수로 생성 하지 못해 동적 할당을 받는데 어려움이 있었습니다. LRU를 어떤식으로 구현할지 몰라서 나름대로 구현했고 시행착오도 많이 겪었습니다.

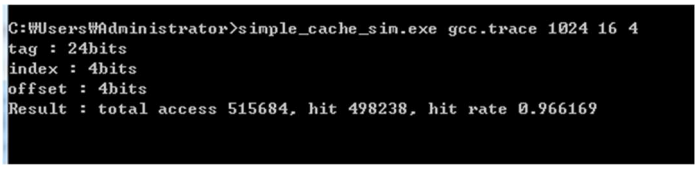
2. 결과

화면 캡쳐 등을 이용한 실행 결과를 보이고 설명.

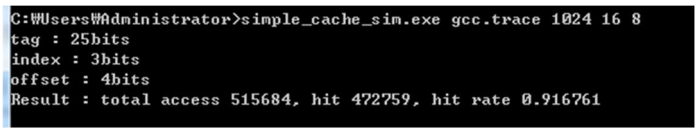


1way일때 결과값입니다.

2way일때 결과값입니다.



4way일때 결과값입니다.



8way일때 결과값입니다.

2way일때 가장높은 hit rate를 보였습니다.

3. 결론

코드를 구현 할 때 오류의 상황에 대해서 모두 잡고 싶었지만 int형 변수에 char를 넣을 때 생기는 무한반복루프를 없애는 방법을 찾을 수 없었습니다.
또한 교수님이 보내주신 PDF결과와 값이 다르게 나왔으므로 완벽하지 못한 코드인 것을 인지했습니다.

실행방법 : 미리컴파일된 헤더를 사용안함으로 체크한 상태에서 컴파일을 하면됩니다.