**Report of Computer Architecture (Simple\_Cache\_Simulator)**

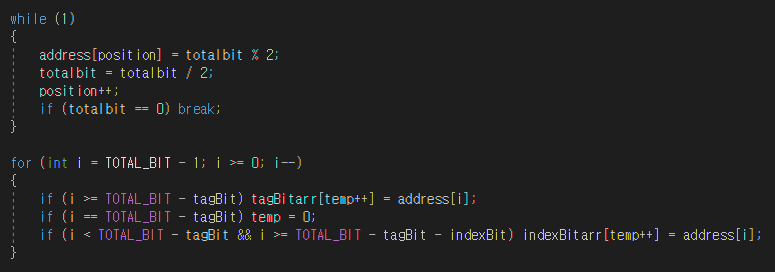
**임유택 (201514768)**

**전북대학교 컴퓨터공학부**

**j75575863@naver.com**

1. **실습 프로그램의 구성 및 동작 원리**

**trace파일에 있는 값 중 두 번 째 메모리 주소를 이진법으로 나타낸 후 tagBit, indexBit, offsetBit로 나눈다.**

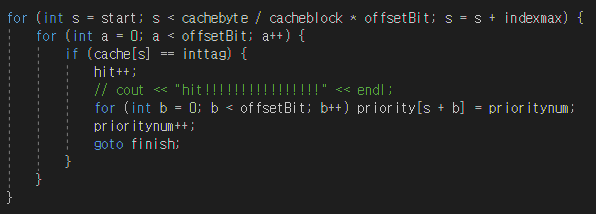


**그 후 배열에 나눠져 있는 이진수를 변수 하나에 십진법으로 나타낸다.**

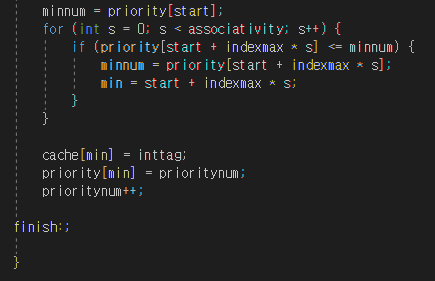


**start 변수는 associativity로 나눠진 여러 구역 중 첫 구역에서의 index로 초기화 해 놓은 변수이다.**

**그렇기 때문에 start에서 indexmax, 즉 index가 가질 수 있는 최대 값인 index gap을 더해주며 tag와 일치하는지 확인한다. 일치한다면 hit를 올리고 Least Recently Used를 구현하기 위해 만들어 놓은 priority라는 배열에 offset만큼 prioritynum를 넣어준다. 그리고 finish로 이동해 다음 tag비교로 넘어간다.**

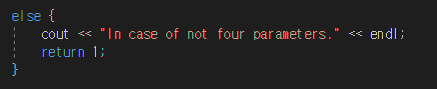


**만약 끝까지 tag와 메모리가 일치하지 않는다면 이제 index에서 Least Recently Used를 찾아야 하기 때문에 같은 index끼리 접근했던 기록을 비교한다. 같은 index중 최소의 값을 가지고 있는 index를 min으로 저장한다. 그 후 cache[min]에 새로운 tag를 넣어준다.**

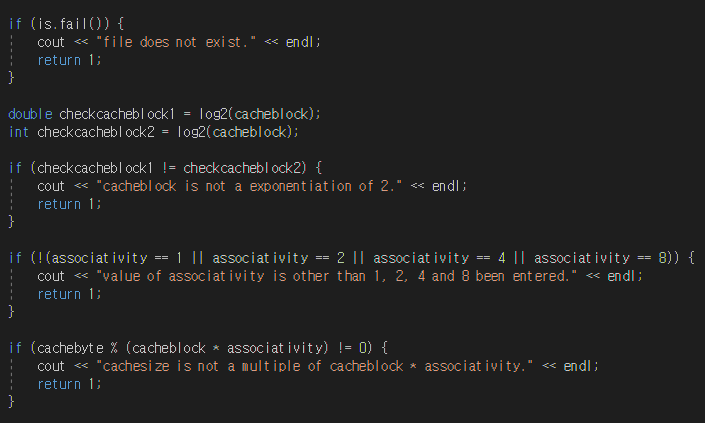


**예외처리**

**처음 입력할 때 인자의 수부터 확인한다. 만약 인자의 수가 정상적이라면 프로그램을 작동시킨다.**

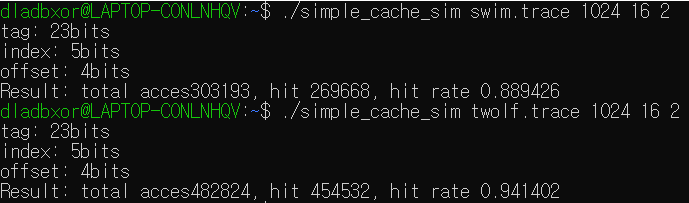


**그 외의 예외 처리는 입력 받은 cachebyte, cacheblock, associativity로 확인할 수 있다.**

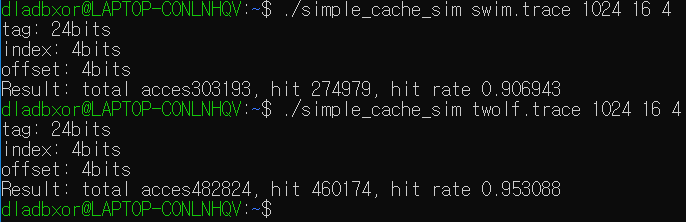


**2. 결과**

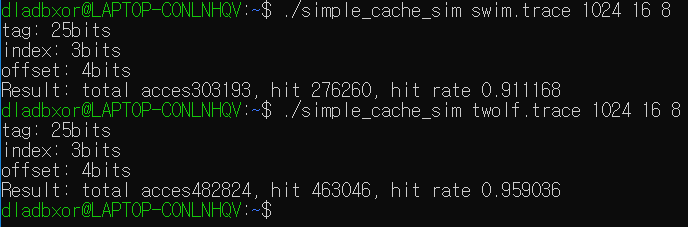
**cachebyte 1024, cacheblock 16, associativity 2 기준**

****

**cachebyte 1024, cacheblock 16, associativity 4 기준**

****

**cachebyte 1024, cacheblock 16, associativity 8 기준**

****

**error 처리**

