데이터구조 활용 리포트 주제 분석

2021년2학기

학과		학년		학번		이름		
주제명	페이지 교체 알고리즘	(LRU) 및 캐	시 구현 (배	열, 큐)	제출일	2021년	월	일
정의 및 개념	포털 사이트의 검색 기능은 클라이언트가 서버에 자료의 한 부분을 요청하고, 서버는 데이터베이스나 다른 서버에서 해당 자료의 특정 부분을 조회하는 과정으로 이루어지는데, 데이터베이스는 크기가 크면 조회 시간이 길어진다는 단점이 있다. 이는 고객 서비스경험의 불편함을 가져온다. 조회 시간이 길어지는 문제를 완화하기 위해 데이터를 임시로 저장할 가상의 저장공간인 '캐시'를 도입하며 이 캐시의 빠른 접근 시간을 토대로 어플리케이션은 빠른 처리성능을 확보할수 있다. 캐시는 가장 최근에 사용된, 또는 가장 빈번하게 사용되는 데이터를 저장하는 역할을 가진다. 효율적인 동작을 위해 캐시(배열)의 크기는 일정한 크기로 정해져 있기 때문에 (가장최근에 사용된 값을 저장하는 캐시에서는) 꽉 찬 캐시에 요소를 새로 넣을 때에는 캐시의 가장오래된 요소를 삭제하거나 캐시에 이미 존재하는 해당 요소를 삭제하고 재입력하는 과정이 필요하다. 이러한 페이지 교체 알고리즘을 LRU(Least Recently Used)라고 부르는데, 이 알고리즘을 큐로 이해하고 구현할 수 있다.							
개발하 려는 내용을 설명하 시오.	LRU 알고리즘을 큐(혹은 FIFO)로 이해하는 방법은 다음과 같다. 캐시(배열)에 저장되어 있지 않은 새로운 요소를 추가할 때 큐와 같이 FIFO 방식을 사용해 캐시의 가장 오래된 요소를 버린다. 반대로 캐시에 이미 저장되어 있는 요소를 다시 추가할 때는 해당 요소를 배열의 맨 뒤로 가져오는데, 이는 배열에서 해당 요소를 삭제(큐의 get과는 달리 배열의 어느 위치에서나 원하는 요소를 삭제할 수 있다)하고, 다시 해당 요소를 배열에서 입력 (put)하는 과정과 같다. LRU 알고리즘의 성능에는 여러 요인들이 작용하지만 과제에서 알고리즘을 구현하면서 가장유의해 살펴볼 요인은 캐시의 크기이다. DB캐시의 경우 CPU에서 메인 메모리의 자료를 조회하고 싶을 때 메인 메모리의 최근 값들을 가지는 DB캐시부터 확인하게 되는데, 이 때 메모리에서 조회하려는 값이 캐시에 존재하는지 아닌지를 확인하는 과정을 가진다. 메인 메모리에 조회							
	위와 같은 전제를 이용 을 구현 한다. 캐시 엔.							

개 요소로 가정하며, 해당 요소가 가지는 데이터는 문자열 값이다. 문자열은 일종의 키워드로, 사용자가 어플리케이션의 검색 기능을 사용할 때 어플리케이션의 검색 기능이 메인 메모리가 아닌 DB캐시에 먼저 접근하는 예시를 캐시로 구현하기 위해 키워드에서 대문자와 소문자, 띄 어쓰기는 구분하지 않는다.

위에서 언급한 대로, '캐시 힛'과 '캐시 미스'의 실행시간을 각각 1과 5로 하여 캐시의 길이 별로 캐시의 실행 시간이 얼마나 걸리는지를 비교하며 이에 따라서 적합한 캐시의 길이를 구할 수 있는지, 그렇다면 캐시의 적합한 길이는 얼마가 되는지를 확인해보는 것을 알고리즘 구현의 최종 목표로 한다(알고리즘 자체에 대한 테스트는 갖지 않는다).

사용하 려는 개발도

1) 도구 : **Visual Stuido(O)**, Replit(), 기타()

2) 언어: C언어(), **Python(O),** 기타()