

오픈소스 자료구조 및 알고리즘 분석_C Project 과제

2020년 1차 과정 (2020년 4월 6일 ~ 5월 4일)

◆ 과제 배점
60점

◆ 과제 설명
각 기능의 구현에 따라 점수가 부여 됩니다.

◆ 제출 방법
"영문이름_사번" 으로 폴더 만들고 , 폴더 안에
과제설명.txt를 다음과 같이

- (1) 어느정도 기능을 구현 했는지 적어 주세요.
- (2) 각 함수의 기능 설명을 적어 주세요.
- (3) 과제 구현 시 느낀 점을 적어 주세요.

실행 파일은 필요 없습니다.

소스만 포함해 주세요.

폴더를 통째로 압축해서 LGE MOOC "Project 과제"
"select file" 버튼을 눌러서 압축파일을 올려 주세요.

압축파일명은 반드시 영문이어야 합니다.

시스템 특성상 파일명은 한글을 사용할 수 없습니다.

◆ 제출 기한
2020년 5월 4일(월요일) 17:00 까지

Project 과제

LRU(Least Recently Used) 알고리즘 구현

Introduction

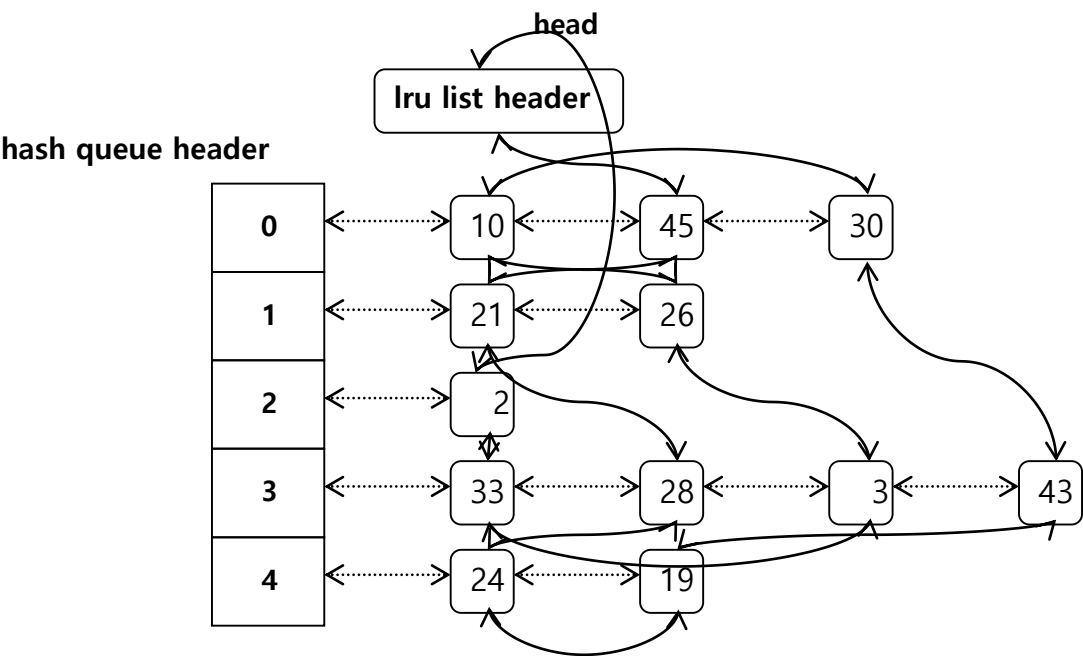
메모리 재배치 전략인 LRU 시스템의 시뮬레이터를 직접 구현함으로써
메모리 관리 시 교육에서 배운 자료구조의 실제 사용법을 이해한다.
LRU는 물리 메모리 공간이 꽉 찬 경우 퇴출 해야 될 page를 결정하는 알고리즘으로
가장 오랫동안 접근이 안된 page를 퇴출한다.

Objectives

이번 과제를 수행하면 다음과 같은 Skill들을 이해하고 적용할 수 있다.

- page number를 이용한 hash 구조 분리
- hash를 이용한 빠른 page 검색(hash 함수는 배운 내용으로 구현)
- LRU list는 Generic Linked List로 구현
- page 접근 시 동작이해
- 하나의 page가 여러 자료구조에 포함되는 구조 이해
- LRU 알고리즘의 동작 방식 이해

전체 구조



Requirements

LRU 알고리즘의 동작을 그대로 시뮬레이터에 재현 한다.

<전제 조건>

1. LRU List는 Generic Lined List를 사용하여 구현한다.
2. Page의 검색은 Generic Hash를 사용하여 구현한다.
3. 전체 페이지는 0번 ~ 39번의 범위를 갖는다.
4. hash에 할당할 수 있는 페이지는 20개 이다.
5. hash에 버킷의 수는 8개이다.

<page 구조체 설계>

```
struct page
{
    int pfn; // page frame number
    int data;
    struct hlist_node hnode;
    struct list_head list;
};
```

< 프로그램 동작 요구사항 >

1. 0~39번 page는 pages 구조체 배열로 스택에 미리 할당한다.
 2. page의 data에는 page number + 1000을 초기화 한다.
 3. 사용자가 엔터를 입력하면 0~39번 page를 랜덤하게 접근한다.(rand함수)
 4. 접근된 페이지가 hash에 있는 경우 page내의 data를 출력 하고,
LRU List에서는 해당 page를 맨 앞쪽으로 옮긴다. (최근 사용 page 이동)
 5. 접근할 page가 hash에 없는 경우 pages 배열에서 해당 page를 찾아
hash에 삽입하고 LRU list의 맨 앞에도 삽입한다.
 6. hash에 20개의 page가 꽉 차면 LRU list의 가장 끝에 있는 노드가 LRU와 hash에서
제거 된다.
 7. 실행 파일의 이름은 lru로 작성한다.
 8. 제공된 sample run을 참조 하여 작성하되 출력은 다양하게 바꿀 수 있다.
-

Functions

함수명은 자유롭게 작성하되 의미 있는 이름으로 부여해야 한다.

run & result

```
# ./sample_run_1

pfn = 36 접근
page 꽂참 , pfn=2 퇴출
새로운 page, 해쉬에 추가, 현재 page 수 = 20
[HASH TABLE]
[0]<->[34]<->[ 5]<->[18]<->[39]<->[ 0]
[1]<->[36]<->[31]
[2]<->[15]
[3]<->[ 4]<->[38]
[4]<->[30]<->[17]<->[ 9]
[5]<->[ 6]<->[14]<->[35]
[6]<->[32]<->[ 3]
[7]<->[16]<->[ 8]

[LRU LIST]
[1ru]<->[36]<->[ 4]<->[15]<->[34]<->[ 6]<->[30]<->[17]<->[ 3]<->[31]<->[32]
<->[16]<->[38]<->[35]<->[18]<->[ 5]<->[ 9]<->[ 8]<->[14]<->[39]<->[ 0]
```