



# Assignment #4

## Implementing Natural Join





## Implementing Natural Join

---

- 이번 과제는 Assignment 3에서 다룬 disk-based B+ tree 바탕으로 join operation을 구현하는 과제입니다.



## Implementing Natural Join

- Assignment 3에서 각각의 레코드들은 .db 파일에 저장되었습니다.
- .db 파일 하나에는 하나의 테이블이 저장되어 있다고 가정하겠습니다.
- 즉, {key(int, 8byte), value(string, 최대 120byte)} 형식의 튜플이 .db 파일에 저장되어있습니다.



## Implementing Natural Join

- 이번 과제에서는 `table1(key, value1)`, `table2(key, value2)`가 각각 `table1.db`, `table2.db`에 저장되어있다고 가정합니다.
- 최종적으로 `table1 NATURAL JOIN table2` 를 구현하는 과제입니다.



## Implementing Natural Join

---

- 이번 과제에서 수행해야 할 내용은 아래와 같습니다.
  1. Assignment 3에서 구현한 프로그램의 기능 중 일부를 .db 파일 2개를 open한 상태에서도 사용할 수 있도록 수정
  2. natural join 구현



## Implementing Natural Join

- 이번 과제에서 메모리 최대 사용량은 1GiB( $2^{30}$  byte)로 제한합니다.
- 특정 testcase에서 메모리 사용량이 1GiB를 초과할 경우 해당 testcase에 대한 점수는 부여되지 않습니다.



## Implementing Natural Join

1. Assignment 3에서 구현한 프로그램을 .db 파일 2개를 open 할 수 있도록 수정
  - Assignment 3에서 제공해드린 코드를 보면 open\_table 이라는 함수를 통해 'test.db' 파일 하나를 open 한 후 각종 연산을 수행했습니다.
  - 'table1.db', 'table2.db' 두 파일을 open 하도록 수정합니다.
  - find, insert, delete 등 B+ tree의 각종 기능 중 join에 필요한 기능 또한 두개의 파일이 open된 상태에서 작동할 수 있도록 수정해야 합니다.
    - join과 무관하다면 수정하지 않아도 괜찮습니다.
  - 코드를 수정할 때 추가적인 변수를 선언하거나 함수를 정의하여도 괜찮습니다.

```
int main(){
    int64_t input;
    char instruction;
    char buf[120];
    char *result;
    open_table("test.db");
```



# Implementing Natural Join

## 2. natural join 수행

- make 결과 생성된 실행파일을 실행한 후 j를 입력할 시 table1과 table2를 natural join한 결과를 출력
  - table1에 {1,“aaa”}, {2,“bbb”}, {3,“ccc”}가 저장되어 있고  
table2에 {2,“111”}, {3,“222”}, {4,“333”}이 저장되어 있다면

2,bbb,111

3,ccc,222

를 출력



# Implementing Natural Join

## 2. natural join 수행

- main.c 파일의 switch-case문을 오른쪽과 같이 수정합니다.
- db\_join 함수의 argument를 추가해도 괜찮습니다.

```
switch(instruction){
    case 'i':
        scanf("%ld %s", &input, buf);
        db_insert(input, buf);
        break;
    case 'f':
        scanf("%ld", &input);
        result = db_find(input);
        if (result) {
            printf("Key: %ld, Value: %s\n", input, result);
        }
        else
            printf("Not Exists\n");

        fflush(stdout);
        break;
    case 'd':
        scanf("%ld", &input);
        db_delete(input);
        break;

    // Assignment 4
    case 'j':
        db_join();
        break;

    //////////////////////////////////////

    case 'q':
        while (getchar() != (int)'\n');
        return EXIT_SUCCESS;
        break;
}
```



## Implementing Natural Join

- join 성공 여부와 join 수행 시간만 고려하여 채점합니다.
  - 즉, testcase를 준비한 후 프로그램을 실행한 뒤 j를 입력하였을 때 나오는 결과와 시간을 바탕으로 점수를 부여합니다.
  - 마지막 testcase에서 실행 시간에 따라 가산점이 주어집니다.
    - 기준 1 보다 짧은 실행시간 시 10점
    - 기준 2 보다 짧은 실행시간 시 5점
- 각각의 testcase마다 제한시간이 존재하므로 굉장히 비효율적인 구현의 경우 점수가 부여되지 않을 수 있습니다.



## Implementing Natural Join

- 이번 과제를 수행할 때 Assignment 3에서 본인이 구현한 코드나 이번에 제공된 B+ tree코드 중 아무거나 사용해도 괜찮습니다.
- 채점은 제공된 B+ tree 코드를 통해 생성된 .db 파일을 바탕으로 진행됩니다.
- 제공된 파일은 아래와 같은 구조를 가지고 있습니다.
  - ass\_4\_files
    - └ include
    - └ lib
    - └ src
      - └ bpt.c
      - └ main.c
    - └ Makefile



# Judging System

- 이번 프로젝트의 채점 환경(**Default Version**)은 아래와 같습니다

```
gcc (Ubuntu 11.4.0-1ubuntu1~22.04.1) 11.4.0  
GNU Make 4.3
```

- ❖ Version차이로 인하여 생기는 문제는 조교가 책임지지 않으며 **빌드와 실행이 안 될 경우 점수는 0점 입니다**



## ❖ Code

- **Completeness** : 명세의 요구 조건을 모두 올바르게 구현해야 합니다
- **Defensiveness** : 발생할 수 있는 예외 상황에 대처할 수 있어야 합니다
- **Comment** : 코드에는 반드시 주석이 있어야 합니다

## ❖ Wiki

- **Design** : 명세에서 요구하는 조건에 대해서 어떻게 구현할 계획인지, 어떤 자료구조와 알고리즘이 필요한지, 수업시간에 배운 이론이 어떻게 적용되어야 하는지 등 자신만의 디자인을 서술합니다
- **Implement** : 코드를 그대로 복사하여 문서에 붙여놓지 마시고 본인이 새롭게 구현하거나 수정한 부분에 대해서 무엇이 기존과 다른지, 해당 코드가 무엇을 목적으로 하는지에 대한 설명을 구체적으로 서술합니다
- **Result** : 해당 명세에서 요구한 부분이 정상적으로 동작하는 실행 결과를 첨부하고, 이에 대한 동작 과정에 대해 설명합니다
- **Trouble shooting** : 과제를 수행하면서 마주하였던 문제와 이에 대한 해결 과정을 서술합니다. 혹여 문제를 해결하지 못하였다면 어떤 문제였고 어떻게 해결하려 하였는지에 대해서 서술합니다.



- ❖ 아래와 같이 디렉토리를 구성한 후 압축하여 제출하시길 바랍니다.
  - ass\_4\_[이름]\_[학번] (예시: ass\_4\_주정현\_2022123123)
  - └ include
  - └ lib
  - └ src
    - └ bpt.c
    - └ main.c
  - └ Makefile
  - └ Wiki(ass\_4\_wiki\_[이름]\_[학번].pdf)
- 제출 양식을 엄수하고 의미 없는 파일 및 디렉토리는 반드시 제거하여 제출하시길 바랍니다.
- 제출 기한: 2024년 12월 19일 23시 59분
- 추가 제출 기한: 없음
- 이번 과제는 추가 제출기한이 없으니 반드시 기한 내에 제출하시길 바랍니다.