

과제 5: 레코드 저장, 삭제 및 검색

1. 개요

주어진 'student.h'에 정의되어 있는 'STUDENT'에 대해 학생 레코드를 저장하고 삭제할 수 있는 프로그램을 작성한다.

■ 레코드 파일

- 레코드 파일은 헤더(header)와 레코드의 집합으로 구성된다.

■ 헤더(Header)

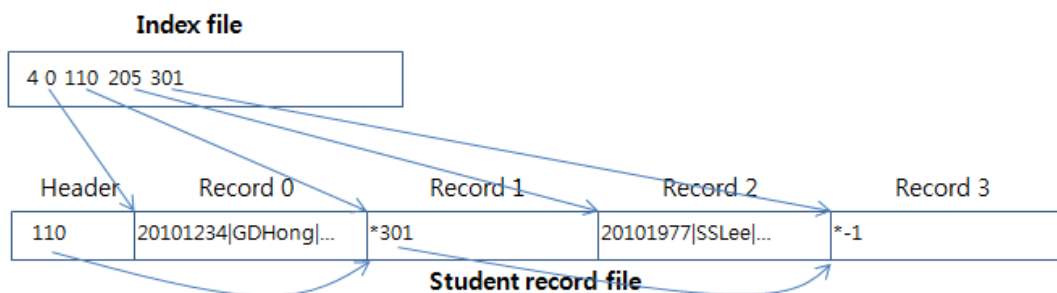
- 삭제 레코드를 관리하기 위한 linked list의 헤드(head), 즉 가장 최근에 삭제된 레코드의 바이트오프셋(byte offset)을 유지하며, head의 초기값은 -1이다. head를 저장하기 위한 공간의 크기는 2바이트이며, 따라서 헤더의 크기는 2바이트이다.

■ 레코드

- 레코드는 가변길이 레코드(variable-length-record) 방식으로 저장되며, 필드는 delimiter(=|) 방식으로 구분된다.
- 레코드를 저장할 때 최대 127바이트의 공간을 할당한다 (student.h 참조). 여기에는 7개의 delimiter도 포함된다.

■ 인덱스 파일

- 삭제 레코드를 포함한 전체 레코드의 수(#records)와 각 레코드의 바이트오프셋을 저장한다.
- 레코드의 수와 바이트오프셋을 저장하기 위한 공간의 크기는 각각 2바이트이다.



2. 프로그램 기능

■ 레코드 저장(add)

- 터미널에서 학생 정보를 사용자로부터 입력 받고, 이를 학생 레코드 파일에 저장한다. 만약, 삭제 레코드가 존재하면 이 공간을 재사용한다.
- 저장과 동시에 인덱스 파일에도 관련 정보를 업데이트해 준다.
- 사용자로부터 레코드 파일 이름과 필드값 리스트를 입력 받는다. 학생 레코드는 7개의 필드로 구성되어 있으므로 동일한 수의 필드값이 필요하다. 필드값은 반드시 ID,

NAME, DEPT, YEAR, ADDR, PHONE, EMAIL 순서대로 주어져야 하며, 필드값은 영문자와 숫자로만 구성되어 있다고 가정한다. 이 명령어를 수행하고 난 후의 출력은 없다.

```
$ a.out -a "field_value1" "field_value2" ...
```

- 실행파일과 같은 디렉토리에 있는 'student.dat' 파일에 대해서 새로운 학생 레코드를 저장하는 예는 다음과 같다.

```
$ a.out -a "20101234" "Gildong Hong" "Computer Science" "3"
"Dongjak-Gu,Seoul" "010-828-0567" "gdhong@ssu.ac.kr"
```

- 레코드 저장과 동시에 'student.idx' 인덱스 파일을 갱신한다.
- 삭제 레코드를 재사용할 때 first-fit 전략을 사용한다.
- 삭제 레코드를 재사용하고 난 후 남는 공간은 그대로 둔다. 즉, 별도의 삭제 레코드로 바꾸어서 삭제 레코드 리스트에 저장하여 재사용하지 않는다.

■ 레코드 삭제(delete)

- 사용자로부터 입력 받은 학번 키값과 일치하는 레코드를 찾아서 삭제한다. 삭제 레코드 관리 및 재사용은 "linked list" 기법을 적용한다.
- 사용자로부터 레코드 파일 이름과 학번 키값을 입력 받는다. 주어진 키값을 갖는 학생 레코드를 찾아서 그 레코드를 삭제한다. 여기서 삭제는 해당 레코드의 맨 앞에 delete mark(=|)와 바로 직전에 삭제된 레코드의 바이트오프셋을 저장하는 것을 의미한다. 수행 결과의 출력은 없다.

```
$ a.out -d "field_value1"
```

- 학생 레코드 파일에 대해 "20101234"를 만족하는 삭제의 예는 다음과 같다.

```
$ a.out -d "20101234"
```

■ 레코드 검색(search)

- 사용자로부터 입력 받은 학번 키값과 일치하는 레코드를 검색하고 출력한다. 검색 시 삭제 레코드는 검색 대상에서 제외한다.
- 사용자로부터 레코드 파일 이름과 학번 키값을 입력 받는다. 주어진 키값과 일치하는 학생 레코드를 찾아서 화면에 출력한다. 이때 출력은 반드시 student.c에 정의되어 있는 레코드 출력함수를 사용한다.

```
$ a.out -s "field_value1"
```

- 학생 레코드 파일에 대해 “20101234”를 만족하는 검색 결과의 예는 다음과 같다.

```
$ a.out -s “20101234”
```

20071234|Gildong Hong|Computer Science|3|Dongjak-Gu,
Seoul|010-828-0567| |gdhong@ssu.ac.kr

3. 주의 사항

- 위의 add, delete, search에서 프로그램의 실행인자로 학생 레코드 파일명을 줄 필요가 없으며, 대신 student.h에 정의되어 있는 학생 레코드 파일명을 사용해야 함
- student.c 파일만을 완성해서 제출하면 됨

3. 개발 환경

- OS : Linux 우분투 버전 16이상
- 컴파일러 : gcc
- 반드시 이 환경을 준수해야 하며, 이를 따르지 않아서 발생하는 불이익은 학생이 책임져야 함!!

4. 제출물

- student.c를 하위폴더 없이(최상위 위치에) zip파일로 압축하여 myclass.ssu.ac.kr 과제 게시판에 제출 (소스파일, 헤더파일, zip파일은 반드시 소문자로 작성)
- 압축한 파일은 반드시 학번_5.zip (예시 20061084_5.zip)과 같이 작성하며, 여기서 5는 다섯 번째 과제임을 의미함
- 채점 프로그램상 오류가 날 수 있으니 꼭 위 사항을 준수!