파일구조 과제 3 **2025. 5. 19**

아래의 기능을 만족하는 프로그램을 C++로 작성하여라.

1. 다음과 같은 두 데이터 파일을 Binary로 서로 다른 파일(student.dat와 professor.dat)에 각각 저장한다.

   Student(unsigned ID, char name[20], float score, unsigned advisorID)

   Professor(char name[20], unsigned ID, char dept[10])

   여기서 한 학생은 advisorID 로 참조되는 한 명의 지도교수에 지정된다. 참고로 위의 name[20]이나 dept[10]에 end of string character 는 포함되지 않는다.

2. 다음과 같은 조건을 만족하여야 한다.
   1) 하나의 record가 두개의 block에 걸쳐지면 안된다.
   2) Record는 binary로 저장된다.
   3) 다음의 기능을 만족하여야 한다.

      (1) Record 입력:

      boolean insertStudent(string* name, unsigned ID, float score, unsigned advisorID)

      boolean insertProfessor(string* name, unsigned ID, string* dept)

      return: true – 입력이 성공, false – 입력이 실패

      입력되는 데이터는 각 parameter로 지정하며, ID가 중복되는 것이 있으면 false로 return

      만일 처음 입력되는 record이면 student.dat (또는 professor.dat) 파일을 생성

      (2) n-번째 Record의 검색:

      boolean readStudent(unsigned rank, unsigned *ID, string* name, float *score, unsigned* advisorID)

      return: true – 검색이 성공, false – 검색이 실패

      검색된 데이터는 나머지 parameter에 지정

      위에서 rank는 앞에서 몇 번째인가를 나타내는 수이며, 첫번째 student의 record는 0번째로 간주한다.

      (3) ID로 검색:

      boolean searchStudent(unsigned ID, string* name, float *score, , unsigned* advisorID)

      return: true – 검색이 성공, false – 검색이 실패

      검색한 데이터는 나머지 parameter로 반환

      (4) Join: 두 파일을 아래와 같은 함수로 join

      bool join(string* pName, string* sName, unsigned n)

      지도학생의 수가 n 명인 교수의 이름과 지도학생의 이름을 pName과 sName으로 반환.

      만일 여러 개의 결과가 있으면, 제일 작은 Prof.ID의 교수와 이 교수가 지도하는 학생 중 제일 작은 Student.ID를 가지는 학생의 이름을 각각 pName과 sName으로 반환

      return: 검색결과가 없으면 false, 한 경우라도 있으면 true

4) 위의 함수는 main 프로그램에서 호출되며, 숙제와 함께 주어진다. 필요에 따라 주어지는 main 프로그램을 수정할 수는 있으나, 위 함수 interface는 반드시 지켜져야 하며, 시간을 측정하는 장소는 변경하면 안된다.
5) (2), (3), (4)번의 함수는 (1)번으로 생성된 binary 파일을 이용하여 수행되어야 한다.
6) 데이터는 student, professor의 데이터를 위하여 각각 1개의 block (=4K bytes), 출력을 위한 1개의 block에 해당하는 main memory를 사용하여야 한다. 개별 값을 위한 변수는 사용 가능하다. 즉, 사용 가능한 main memory는 개별적으로 선언된 변수와 세 개의 block 뿐이다.

3. 제출방법:
- 위의 함수 프로그램 소스코드 (C++)를 main.cpp와 결합하여 Plato에 upload. main.cpp는 주어진 파일을 사용
- 테스트는 C++11을 기준으로 컴파일하여 결과를 확인할 예정이다.
- 제출마감: 6월 2일 오후 4시

4. Data는 text file로 student.txt, professor.txt로, 검사를 위하여 IDQuery.txt, rankQuery.txt로 아래와 같은 네 개의 파일이 주어진다.

student.txt: name ID score advisorID (the first line indicates the number of students)
professor.txt: name ID department (the first line indicates the number of professors)
IDQuery.txt: ID (the first line indicates the number of IDs)
rankQuery.txt: rank (the first line indicates the number of ranks)

5. 다음의 사항을 반드시 고려하여 프로그램을 작성한다.
1) 위 함수가 호출될 때마다 파일에 직접 데이터를 읽어 처리하다. 미리 main memory에 데이터를 올리고 처리할 경우, 0 점으로 처리된다.
2) Binary로 파일이 저장되지 않을 경우, 0점으로 처리된다.
3) 수행시간 조사: 수행시간을 조사하여 1등은 총점의 20%, 2등(2명)은 총점의 10%가 가산된다.

6. 참고: 여러 분이 작성한 프로그램은 다음 프로그램 (Extensible Dynamic Hash와 B+-tree 프로그램) 과제에 그대로 사용될 예정이다.