**과제 2: 순차읽기 및 랜덤읽기의 디스크 I/O 비용 측정**

프로그램 구현 시 지켜야 할 조건들뿐만 아니라 개발환경(컴파일 등)과 과제 파일의 제출 시 요구되는 조건들을 지키지 않아서 과제 점수에 감점을 받을 수 있으므로 모든 조건들을 잘 숙지하고 반드시 지켜야 합니다. 본인 잘못에 의한 과제 점수의 감점은 전혀 참작의 고려 대상이 아닙니다. 과제 수행 중 스스로 해결하지 못한 궁금한 점들은 스마트캠퍼스의 문의게시판에 꼭 질문을 하여 해결난 후 과제 파일을 제출하기 바랍니다.

3장에서 배운 레코드 파일의 순차읽기와 랜덤읽기에 대한 프로그램을 구현하며, 다음과 같은 제약 사항들을 준수한다.

- 파일 I/O 연산은 system call 또는 C 라이브러리만을 사용한다.

(1) 학생 레코드 파일 생성: create\_records.c 작성

주어진 <record\_file\_name> 파일에 학생 레코드를 하나씩 반복적으로 <#records> 수만큼 저장한다. 학생 레코드의 크기는 200 바이트이며 레코드에는 굳이 의미있는 데이터를 채울 필요가 없으며 심지어 모든 학생 레코드의 데이터가 동일해도 상관없다. 단, 각 학생 레코드의 크기는 반드시 200 바이트가 되어야 한다. 명령의 실행 후 화면에 출력값은 없다.

a.out <#records> <record\_file\_name>

(2) 순차읽기: seq\_read.c 작성

주어진 <record\_file\_name> 파일에 저장되어 있는 모든 레코드를 순차적으로(sequential) 하나씩 사용자 프로그램 상으로 읽어 들이며, 이 과정에서 발생하는 전체 시간 비용을 출력한다 (gettimeofday() 함수 등을 사용). 아래 예시와 같이, 명령의 실행 후 화면에 그 시간 비용을 usec 단위의 정수값으로 출력한다.

a.out <record\_file\_name>

<예시>

$ a.out student.dat

3200 usec

(3) 랜덤읽기: rand\_read.c 작성

주어진 <record\_file\_name> 파일에 저장되어 있는 모든 레코드를 랜덤(random)하게 하나씩 사용자 프로그램 상으로 읽어 들이며, 이 과정에서 발생하는 전체 시간 비용을 출력한다. 아래 예시와 같이, 명령의 실행 후 화면에 그 시간 비용을 usec 단위의 정수값으로 출력한다.

a.out <record\_file\_name>

<예시>

$ a.out student.dat

120300 usec

**2. 개발 환경**

- OS: Linux 우분투 버전 18.04

- 컴파일러: gcc 7.5

\* 반드시 이 환경을 준수해야 하며, 이를 따르지 않아서 발생하는 불이익은 본인이 책임져야 함

**3. 제출물**

- 완성된 세 개의 C 소스파일들을 하위폴더 없이(최상위 위치에) zip파일로 압축하여 스마트캠퍼스의 과제 게시판에 제출함 (모든 제출 파일들의 파일명은 제공된 파일명과 반드시 일치해야 함)

- 압축한 파일은 반드시 학번\_2.zip (예시 20201084\_2.zip)과 같이 작성하며, 여기서 2는 두 번째 과제임을 의미함

\* 채점은 채점 프로그램을 통해 자동으로 처리하기 때문에 위의 사항들을 준수하지 않는 경우 채점 점수가 0이 될 수 도 있기 때문에 반드시 준수하기 바라며, 이를 따르지 않아서 발생하는 불이익은 본인이 책임져야 함