

## 과제 2: 디스크 I/O 비용 측정

### 1. 개요

강의자료 Chap3의 15쪽에서 배운 순차 읽기와 랜덤 읽기에 대한 두 가지 프로그램을 구현하며, 다음과 같은 제약 사항을 따른다.

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

(1) ‘학생’ 레코드가 100바이트라고 가정할 때 아래 (2)과 (3)에서 사용할 학생 레코드 파일을 준비한다. 레코드의 데이터는 굳이 신경쓰지 않아도 되며, 레코드 파일에는  $n$  개의 학생 레코드가 저장되며  $n$ 은 비교적 큰 수( $>1000$ )이며, 자유롭게 선택해서 사용한다 (당연히 레코드 파일의 크기는  $n*100$ 바이트가 되어야 함).

(2) 주어진 레코드 파일에 저장되어 있는  $n$  개의 레코드를 순차적으로(sequential) 사용자 프로그램 상으로 읽어 들이는 프로그램을 구현한다 (즉, read\_seq.c를 완성). 또한, 이때 걸리는 시간을 측정하는 코드를 추가한다 (예를 들면, gettimeofday() 함수 등을 사용).

(3) 주어진 레코드 파일에 저장되어 있는 모든 레코드를 랜덤(random)하게 읽어 들이는 프로그램을 구현한다 (즉, read\_random.c를 완성). 또한, 이때 걸리는 시간을 측정하는 코드를 프로그램에 추가한다.

### 2. 입력

- read\_seq.c: 레코드 파일명을 표준입력으로 읽어 들이고(예: sread record\_file1), 이 파일에 저장되어 있는 레코드의 개수를 간단히 계산하고 그 개수를 순차적 레코드 읽기에 사용한다. 이때 레코드 파일에는 100바이트짜리 학생 레코드가 정확히 정수 개만큼 저장되어 있어야 한다.

- read\_rand.c: 레코드 파일명을 표준입력으로 읽어 들이고(예: rread record\_file2), 이 파일에 저장되어 있는 레코드의 개수를 간단히 계산하고 그 개수를 무작위 레코드 읽기에 사용한다. 이때 레코드 파일에는 100바이트짜리 학생 레코드가 정확히 정수 개만큼 저장되어 있어야 한다.

<주의사항>

- 명령어에 인자로 주어지는 레코드 파일은 실행 디렉토리에 존재해야 함

- 입력 포맷을 반드시 지켜야 하며, 그렇지 않을 경우 채점 프로그램에 오류가 발생하며 0점 처리됨

### 3. 출력

출력은 표준출력으로 하며, 구현한 명령어를 수행시킬 때 주어진 레코드 파일에 저장되어

있는 전체 레코드의 개수와 전체 레코드를 각각의 방식으로 모두 읽을 때 걸리는 시간을 다음과 같은 예시로 출력하여야 한다 (아래는 예시일 뿐 실제로는 다른 값이 출력될 수 있음). 아래에서 #records는 전체 레코드의 수를, timecost는 전체 레코드를 읽을 때 걸리는 시간을 나타낸다.

```
$ ./sread s1.dat
#records: 1000 timecost: 2300 us
$ ./rread s2.dat
#records: 1000 timecost: 9300000 us
```

<주의사항>

- 시간 단위는 반드시 'us'를 사용해야 하며, 모든 출력값은 '정수'이어야 함
- 출력 포맷을 반드시 지켜야 하며, 그렇지 않을 경우 0점 처리됨

#### 4. 개발 환경

- OS: Linux 우분투 버전 18.0.4
- 컴파일러: gcc 7.5

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

#### 5. 제출물

- 두 개의 소스파일(read\_seq.c와 read\_rand.c)을 하위폴더 없이(최상위 위치에) zip파일로 압축하여 myclass.ssu.ac.kr 과제 게시판에 제출한다 (모든 제출 파일들의 파일명은 반드시 소문자로 작성).
- 압축한 파일은 반드시 학번\_2.zip (예시 20061084\_2.zip)과 같이 작성하며, 여기서 2는 두 번째 과제임을 의미함
- \*\* 채점 프로그램상 오류가 날 수 있으니 꼭 위 사항을 준수하기 바라며, 이를 따르지 않아서 발행하는 불이익은 본인이 책임져야 함