

과제 2: 디스크 I/O 비용 측정

1. 개요

강의자료 Chap3에서 배운 순차읽기와 랜덤읽기에 대한 세 가지 프로그램을 구현하며, 다음과 같은 제약을 따른다.

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

- (1) 레코드의 크기가 250바이트라고 가정하고 아래 (2)과 (3)에서 사용할 학생 레코드 파일을 생성하는 프로그램을 구현한다. (즉, create_file.c를 완성).
 - 고정길이레코드(fixed length record) 방식으로 250바이트짜리 레코드를 레코드 파일에 저장함
 - 레코드 파일 맨앞에 4바이트짜리 헤더 레코드(head record)를 두며, 여기에 레코드 파일에 저장되어 있는 레코드의 수를 저장함. 레코드 수를 저장할 때 반드시 binary integer 모드로 저장함.
 - 헤더 레코드 다음 첫 번째 레코드를 저장해야 함
 - 레코드에 저장되는 데이터는 임의의 문자를 저장하여도 무방함
- (2) 주어진 레코드 파일에 저장되어 있는 모든 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

- create_file.c: 표준입력으로 레코드 수와 레코드 파일명을 읽어 들이고 (예: creat 1000 student1.dat), 주어진 레코드 수만큼의 레코드를 주어진 레코드 파일에 저장한다. 반드시 레코드 파일의 헤더 레코드에는 레코드 수가 저장되어야 한다.
- read_seq.c: 표준입력으로 레코드 파일명을 읽어 들이고 (예: sread student1.dat), 이 파일에 저장되어 있는 모든 레코드를 순차적으로 읽고 이때 발생하는 전체 시간적 비용을 계산하여 출력한다.
- read_rand.c: 표준입력으로 레코드 파일명을 읽어 들이고 (예: rread student2.dat), 이 파일에 저장되어 있는 모든 레코드를 무작위로 읽고 이때 발생하는 전체 시간적 비용을 계산하여 출력한다.

<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, create_file.c의 경우 출력 데이터가 없으며, read_seq.c와 read_rand.c의 경우 프로그램에서 읽어 들인 전체 레코드의 개수와 이때 걸리는 시간을 다음과 같은 예시로 출력하여야 한다 (아래는 예시일 뿐 실제적으로는 다른 값이 출력될 수 있음). 아래에서 #records는 전체 레코드의 수를, elapsed_time은 전체 레코드를 읽을 때 걸리는 시간을 나타낸다.

```
$ ./sread student1.dat
#records: 10000 elapsed_time: 2300 us
$ ./rread student2.dat
#records: 10000 elapsed_time: 9300000 us
```

<주의사항>

- 시간 단위는 반드시 'us'를 사용해야 하며, 모든 출력값은 '정수'이어야 함
- 명령어 수행 후 데이터를 출력할 때 개행문자를 넣어야 함
- 출력 포맷을 반드시 지켜야 하며, 그렇지 않을 경우 0점 처리됨

4. 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

5. 제출물

- 세 개의 C 소스파일들을 하위폴더 없이(최상위 위치에) zip파일로 압축하여 myclass.ssu.ac.kr 과제 게시판에 제출 (모든 제출 파일들의 파일명은 반드시 소문자로 작성)
- 압축한 파일은 반드시 학번_2.zip (예시 20161084_2.zip)과 같이 작성하며, 여기서 2는 두 번째 프로젝트를 의미함

** 채점 프로그램상 오류가 날 수 있으니 꼭 위 사항을 준수하기 바라며, 이를 따르지 않아서 발생하는 불이익은 본인이 책임져야 합니다.

** 수많은 학생들의 과제를 채점하기 위해서는 채점 프로그램의 자동화율을 높여야 하며, 따라서 준수해야 할 규칙을 만들어 사용할 수밖에 없는 점을 이해해 주기 바랍니다.