

과제 1: 파일 I/O 프로그램 구현

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 질문을 하여 해결하고 난 후 과제 파일을 제출하기 바랍니다.

아래의 기능들을 수행하는 프로그램을 구현하며, 다음과 같은 제약 사항들을 따른다.

- 파일의 데이터는 아라비아 숫자와 영어 알파벳만으로 표현되며, 터미널 상에서 사용자 입력으로 주어지는 데이터도 이 조건을 따른다.
- 파일 I/O 연산은 system call 또는 C 라이브러리만을 사용한다.

(1) 복사: copy.c 작성

이미 존재하는 원본파일로부터 10바이트 단위로 데이터를 읽어 새로운 복사본파일에 저장한다. 마지막으로 복사할 데이터가 10바이트가 되지 않는 경우 그 크기의 데이터를 읽어 복사본파일에 저장한다. 명령어를 수행시킬 때 복사본파일은 존재하지 않으며 따라서 새로 생성해야 하며, 명령어를 실행한 후 복사본파일의 내용은 항상 원본파일의 것과 동일해야 한다.

a.out <원본파일명> <복사본파일명>

(2) 읽기: read.c 작성

이미 존재하는 파일에서 <오프셋(offset= 0, 1, 2, ...)>을 기준으로 왼쪽편 또는 오른쪽편에 존재하는 <바이트 수>의 절대값에 해당하는 크기만큼의 데이터를 읽어서 화면에 출력한다. 만약 <바이트 수>가 양수이면 오른쪽편의 데이터를, 음수이면 왼쪽편의 데이터를 읽는다. <바이트 수>가 0이면 데이터를 읽을 필요가 없으며 당연히 화면에 출력되는 데이터도 존재하지 않는다. <바이트 수>의 절대값만큼의 데이터가 왼쪽 또는 오른쪽에 존재하지 않으면 존재하는 데이터만 읽으면 된다. 예를 들면, <오프셋>이 10 이고 <바이트 수>가 -15이면 오프셋 0부터 9까지 데이터를 읽어서 출력한다.

a.out <오프셋> <바이트 수> <파일명>

(3) 병합하기(merge): merge.c 작성

이미 존재하는 두 개의 파일, <파일1>과 <파일2>의 데이터를 병합한 새로운 파일 <파일3>을 생성한다. 병합의 순서는 <파일1>과 <파일2>의 순서를 따르며, <파일3>의 크기는 입력된 두 개의 파일의 크기의 합과 정확히 일치해야 한다.

a.out <파일명1> <파일명2> <파일명3>

(4) 덮어쓰기(overwrite): overwrite.c 작성

이미 존재하는 파일에서 주어진 <오프셋(=0, 1, 2, ...)>에서부터 <데이터>를 덮어쓴다. 덮어쓰기를 할 때 파일의 EOF를 만나면 중단하지 않고 그대로 쓰기를 진행한다. <데이터>는 큰따옴표(“와 ”)로 묶어서 표현한다 (<데이터>로 “abc”라고 입력하면 abc가 덮어쓸 데이터를 의미한다). 예를 들어, <오프셋>이 5이고 <데이터>가 “abc”라고 하면 오프셋 5, 6, 7 위치에 각각 a, b, c가 저장되어야 한다.

a.out <오프셋> <데이터> <파일명>

(5) 끼워넣기(insert): insert.c 작성

이미 존재하는 파일에서 <오프셋(=0, 1, 2, ...)>과 <오프셋+1> 사이에 <데이터>를 끼워 넣는다. <오프셋>이 파일의 맨마지막(EOF제외)을 가리키면 append로 처리한다. <데이터>는 큰따옴표(“와 ”)로 묶어서 표현한다 (<데이터>로 “abc”라고 입력하면 abc가 끼워넣을 데이터를 의미한다). 예를 들어, <오프셋>이 5이고, <데이터>가 “abc”라고 하면 오프셋 5와 6 사이에 abc가 삽입되며, 기존의 오프셋 6이후 데이터는 뒤로 밀려난다.

a.out <오프셋> <데이터> <파일명>

(6) 삭제하기(delete): delete.c 작성

이미 존재하는 파일에서 <오프셋(offset= 0, 1, 2, ...)>을 기준으로 왼쪽편 또는 오른쪽편에 존재하는 <바이트 수>의 절대값에 해당하는 크기의 데이터를 삭제한다. 만약 <바이트 수>가 양수이면 오른쪽편의 데이터를, 음수이면 왼쪽편의 데이터를 삭제한다. <바이트 수>가 0이면 데이터를 삭제할 필요가 없다. <바이트 수>의 절대값만큼의 데이터가 왼쪽 또는 오른쪽에 존재하지 않으면 존재하는 데이터만 삭제하면 된다. 예를 들면, <오프셋>이 10이고 <바이트 수>가 -5이면 오프셋 5부터 9까지 데이터를 삭제한다. 삭제 후 삭제 데이터를 기준으로 전후 데이터가 병합되어야 한다. 즉, 삭제 데이터만큼의 공간은 파일에서 사라져야 한다.

a.out <오프셋> <바이트 수> <파일명>

2. 개발 환경

- OS: Linux 우분투 버전 22.04 LTS (Ubuntu 홈페이지에서 버전 확인 가능)
- 컴파일러: gcc 13.2

* 과제 채점 환경은 위와 동일하며, 따라서 프로그램 개발 환경도 위의 환경에 맞추길 권장하며 이를 따르지 않아서 발생하는 불이익은 본인이 책임져야 함

3. 제출물

- 6 개의 C 소스파일들을 하위폴더 없이(최상위 위치에) zip파일로 압축하여 스마트캠퍼스 lms.ssu.ac.kr의 과제 게시판에 제출함 (모든 제출 파일들의 파일명은 반드시 소문자로 작성)

- 압축한 파일은 반드시 학번_1.zip (예시 20201084_1.zip)과 같이 작성하며, 여기서 1은 첫 번째 과제임을 의미함

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