

과제 1: 파일 I/O 연산 프로그램 작성

1. 개요

아래의 기능들을 각각 수행하는 프로그램을 구현한다. 구현할 때 다음과 같은 제약 사항들을 따른다.

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

(1) 복사: copy.c 작성

<원본파일명>과 <복사본파일명>을 표준입력으로 줄 때 새로운 복사본파일을 생성하고 원본파일로부터 10바이트 단위로 데이터를 읽어 복사본파일에 저장한다. 처음 또는 마지막 남은 데이터가 10바이트가 되지 않는 경우 그 크기의 데이터를 읽어 복사본파일에 저장한다. 명령어를 실행한 후 복사본파일의 내용은 항상 원본파일의 것과 동일해야 한다.

a.out <원본파일명> <복사본파일명>

(2) 읽기: read.c 작성

주어진 파일에서 <오프셋(offset= 0, 1, 2, ...)>을 포함하여 오른쪽에 존재하는 <읽기 바이트 수>만큼의 데이터를 읽어서 화면에 출력한다. 만약 <읽기 바이트 수>만큼의 데이터가 존재하지 않으면 오프셋부터 파일의 맨마지막 바이트(EOF 제외)까지 읽어서 출력한다. 예를 들면, <오프셋>이 7이고 <읽기 바이트 수>가 10이면 오프셋 7부터 16까지 데이터를 읽어서 출력한다.

a.out <파일명> <오프셋> <읽기 바이트 수>

(3) 병합하기(merge): merge.c 작성

주어진 두 개의 파일, 파일2와 파일3을 병합한 파일1을 생성한다. 병합은 파일2와 파일3의 순서를 따른다.

a.out <파일명1> <파일명2> <파일명3>

(4) 덮어쓰기(overwrite): overwrite.c 작성

주어진 파일에서 <오프셋>에서부터 <데이터>를 덮어쓴다. 덮어쓰기를 할 때 파일의 EOF를 만나면 중단하지 않고 그대로 쓰기를 진행한다.

a.out <파일명> <오프셋> <데이터>

(5) 끼워넣기(insert): insert.c 작성

주어진 파일에서 <오프셋>과 <오프셋+ 1> 사이에 <데이터>를 끼워 넣는다.

a.out <파일명> <오프셋> <데이터>

(6) 삭제하기(delete): delete.c 작성

주어진 파일에서 <오프셋>을 포함하여 오른쪽에 존재하는 <삭제 바이트 수>만큼의 데이터를 삭제한다. 만약 <삭제 바이트 수>만큼의 데이터가 존재하지 않는 경우 파일의 마지막 바이트까지 삭제한다. 삭제 후 삭제 데이터를 기준으로 전후 데이터가 병합되어야 한다. 즉, 삭제 데이터의 공간은 파일에서 사라져야 한다.

a.out <파일명> <오프셋> <삭제 바이트 수>

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

- 6 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_1.zip (예시 20161084_1.zip)과 같이 작성하며, 여기서 1은 첫 번째 과제임을 의미함

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