

## 과제 1: 파일 I/O 연산 실습

### 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

원본파일명과 복사본파일명을 표준입력으로 줄 때 복사본파일을 생성하고 원본파일로부터 100바이트 단위로 데이터를 읽어 복사본파일에 저장한다.

a.out <원본파일명> <복사본파일명>

#### (2) 읽기: read.c 작성

파일에서 주어진 오프셋(offset = 0, 1, 2, ...)으로부터 오른쪽에 존재하는, 주어진 <읽기 바이트 수>만큼의 데이터를 읽어서 화면에 출력한다. 만약 주어진 <읽기 바이트 수>만큼의 데이터가 존재하지 않으면 오프셋부터 파일의 맨마지막 바이트(EOF 제외)까지 읽어서 출력한다.

a.out <파일명> <오프셋> <읽기 바이트 수>

#### (3) 병합하기(merge): merge.c 작성

주어진 두 개의 파일들을 병합하며, 병합하는 순서는 주어진 파일 순서와 동일하다.

a.out <파일명1> <파일명2>

#### (4) 덮어쓰기(overwrite): overwrite.c 작성

주어진 오프셋 위치에서부터 데이터를 덮어쓴다. 덮어쓰기를 할 때 파일의 EOF를 만나면 중단하지 않고 그대로 쓰기를 진행한다. 즉 주어진 데이터가 오프셋에서부터 그대로 저장되어야 한다.

a.out <파일명> <오프셋> <데이터>

#### (5) 끼워넣기(insert): insert.c 작성

파일에서 주어진 오프셋에 데이터를 끼워 넣는다. 만약 주어진 오프셋이 5라면 끼워넣는 데이터의 첫 번째 바이트는 오프셋 5에 저장된다.

a.out <파일명> <오프셋> <데이터>

(6) 삭제하기(delete): delete.c 작성

주어진 오프셋 위치에서부터 오른쪽에 존재하는, 주어진 <삭제 바이트 수>만큼의 데이터를 파일에서 삭제한다. 만약 <삭제 바이트 수>만큼의 데이터가 존재하지 않는 경우 파일의 마지막 바이트까지 삭제한다. 삭제하고 난 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 18.0.4
- 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 20061084\_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