

1. 작성한 과제에 대한 간략한 설명

파일을 읽어내려가며 각 글 토크를 token으로 만들었다. 예를 들어 명령어 토큰인 `addiu` %2, \$0, 1024는 'addiu', '\$s', '\$0' 그리고 '1024' 이렇게 4개의 토큰으로 만들어진다. 토큰은 크게 3가지 라벨, data 그리고 instruction으로 나뉘어 처리한다. 먼저 파일을 한 번 읽어 'main:'과 같은 라벨을 LabelTable에 기록하고 주소를 저장한다. 그리고 파일을 두번째로 읽으며 여러 조건문으로 각 토큰에 따라 맞는 작업을 수행했다. 예를 들어 'addiu' 토큰은 앞으로 R 명령어를 완성해야함을 기록하고 OPcode, Function Field 그리고 Shamt의 값을 Buffer에 넣는다. 'j exit'의 'exit'은 라벨로 이미 처리가 됐으므로 그 주소를 Buffer에 채우고 완성한다. Buffer는 이처럼 각 토큰들로 완성된 후 데이터라면 DataValue에, 명령어라면 TextValue에 입력된다. 마지막으로 명령어의 갯수와 데이터의 갯수를 파일에 쓰고, TextValue (변환한 명령어들이 들어있음)와 DataValue(변환한 데이터들이 들어있음)를 sample.o에 저장한다.

2. 과제의 컴파일 방법 및 컴파일 환경/ 과제의 실행 방법 및 실행 환경

C언어를 사용하여 코딩했고 Ubuntu 18.04 터미널에서 프로젝트 파일에 들어간 뒤 gcc 7.5.0 버전을 이용하여 main.c이 있는 창에서 다음과 같이 컴파일 하였다.

```
gcc -o runfile main.c
```

터미널에서 다음과 같이 실행하였다.

```
./runfile sample.s
```

그리고 sample.o 파일을 확인하였다. 그리고 다시 sample2.s를 실행할 때에는 다음과 같이 했다.

```
./runfile sample2.s
```

또한 sample.o 파일을 확인했다.