

# 컴퓨터 구조 과제 1 보고서

## 1. 간단한 Flow

1. main()에서 input file을 string으로 입력받고 getline(inputFile, line)을 이용하여 line별로 메모리 할당을 시작합니다.
2. input file의 형식을 파악하여 data/instruction/label을 각각 memory\_data(주소, 값) / memory\_text(주소, instruction)/ memory\_label(라벨명, 주소)에 할당합니다. (할당할 때 마다 address가 4씩 증가하도록 만듭니다.)
3. text section을 읽는 과정에서 LA instruction이 들어오는 예외항목을 만듭니다. load하고자 하는 data의 주소를 memory\_data에서 불러옵니다. memory\_text LA instruction을 바탕으로 새로 만든 lui instruction을 할당합니다. datad의 주소가 0x\*\*\*\*0000이 아닌 경우, memory\_text에 새로 만든 ori instruction을 할당합니다.
4. binary\_code라는 vector를 출력하는 convertInstructionsToBinary 함수를 실행합니다. memory\_text에 오름차순으로 정렬되어있는 instruction들을 MIPS CODE를 바탕으로 binary code로 변환 후 vector에 push\_back하여 저장합니다.
5. input file이름을 바탕으로 output 파일명을 생성하고 작성을 시작합니다. memory\_text, memory\_data의 size에 4를 곱한 값, binary\_code에 저장된 코드, 그리고 memory\_data에 저장된 값들을 16진수로 변환하여 저장합니다.

## 2. 컴파일/실행방법, 환경

g++9.4.0으로 compile을 진행했습니다.

```
zeeho@zeeho-VirtualBox:~/project1$ g++ --version
g++ (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

컴파일을 실행한 ubuntu는 20.04 버전입니다.



리눅스 콘솔창에서 아래와 같이 assembler를 실행했습니다. 제가 작성한 cpp 파일명은 assembler4.cpp이고 컴파일이 성공적으로 이뤄진 것을 확인할 수 있습니다. sample.s과 sample2.s실행 후 출력물 sample.o, sample2.o가 같은 디렉토리에 저장되었음을 알 수 있습니다.

```
zeeho@zeeho-VirtualBox:~/project1$ ls
assembler4.cpp sample.s sample2.s
zeeho@zeeho-VirtualBox:~/project1$ g++ -std=c++17 -o assembler4 assembler4.cpp
zeeho@zeeho-VirtualBox:~/project1$ ./assembler4 sample.s
zeeho@zeeho-VirtualBox:~/project1$ ./assembler4 sample2.s
zeeho@zeeho-VirtualBox:~/project1$ ls
assembler4 assembler4.cpp sample.o sample.s sample2.o sample2.s
```

```
1 0x50
2 0x14
3 0x24020400
4 0x421821
5 0x622025
6 0x240504d2
7 0x53400
8 0x24c7270f
9 0xe24023
10 0x834827
11 0x344a00ff
12 0x65942
13 0x66102
14 0x3c041000
15 0x3484000c
16 0x80820001
17 0xa0820006
18 0x1656824
19 0x308e0064
20 0xa7823
21 0x3c110064
22 0x2402000a
23 0x3
24 0x123
25 0x4346
26 0x12345678
27 0xffffffff
```

그림 5 sample.o

```
1 0x30
2 0x4
3 0x3c081000
4 0x8d090000
5 0x91021
6 0xc100005
7 0x810000c
8 0x2c410001
9 0x14200003
10 0x621821
11 0x2442ffff
12 0x8100005
13 0x602021
14 0x3e00008
15 0x5
```

그림 4 sample2.o