

# Project 1. Simple MIPS Assembler Report

201611057 김준우

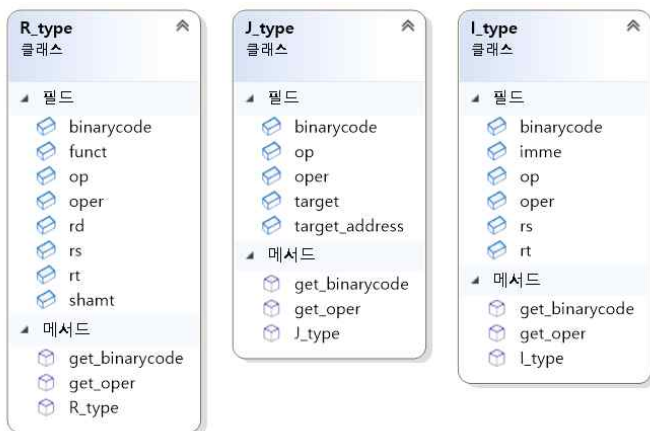
## 1. Introduction

이 프로그램은 MIPS(Big-endian) 어셈블리 코드를 바이너리 코드로 변환하는 MIPS ISA Assembler를 구현한 것이다.

## 2. Environment

VMWare Workstation 16을 사용하여 Ubuntu 20.04.4를 구동하였으며, Ubuntu 내부에서는 VSCode로 코딩을 한 후, 터미널을 통해 구동을 확인하였다. 사용한 언어는 C++이다.

## 3. Explanation



해당 프로그램의 코드는 3개의 클래스와 .text 영역을 해석하는 text\_reading(string input\_str) 함수, la operator의 동작을 결정해주는 la\_define(string op, int var1, int var2) 함수와 메인함수로 구성되어 있다. 그리고 클래스는 MIPS의 3가지 타입인 R, I, J 타입으로 구성되어 있으며 각각의 인자들의 변수와 2진수로 변환하여 16진수로 값을 반환하는 함수, operator를 반환하는 함수가 포함되어 있다.

사용한 STL은 [iostream, fstream, vector, bitset, string, algorithm, vector, sstream, stdlib.h] 이다.

메인함수에서는 우선 외부 어셈블리 파일을 읽어들이고 문자열과 숫자들을 분석하여 string type으로 .data영역과 .text영역을 각각 vector 구조로 저장한다. (문자열과 숫자가 섞여있어, 관리를 용이하게 하기 위해 string타입을 사용했고, 자료구조의 쉬운 사용을 위해 vector를 사용했다.) 그 후, data영역의 vector에서 각각 인자를 가져와 바이너리로 변환시키고, 마찬가지로 이어서 text영역도 바이너리로 변환시켜 또 다른 벡터에 저장을 한다. 위 바이너리의 변환과정이 끝나고 나면, 바이너리 파일로 출력이 되도록 출력과정이 실행된다.

프로그램은 우분투의 Terminal 콘솔창에서 c++ -o runfile project1.cpp로 컴파일 후 실행이 가능했으며, \$./runfile sample.s 와같은 명령어로 실행할 수 있다. 이때 입력되는 어셈블리 파일의 이름명과 동일한 .o 파일이 출력된다.

## 4. Results

프로그램은 정상적으로 실행되었으며, 출력된 바이너리 파일도 정상적으로 출력되었다.