

## 1. 간략한 설명

: MIPS 클래스를 통해 <map> type의 register\_file과 memory\_data, memory\_text, option(m,d,n)을 멤버변수로 관리하여 emulator 실행과 결과 출력을 진행한다.

- 처음 main에서 Init() 전역함수를 실행하여 map type의 R과 I에 key 값을 넣는다. R type의 경우, function field의 6bit가 I type의 경우, opcode 6bit가 각각의 instruction을 대표하여 key값으로 저장된다. 저장된 key값은 후에 R\_emulate, I\_emulate 멤버함수 안에서 해당하는 수행코드로 가기 위한 switch문에서 사용되게 된다. J type의 경우, 2개라서 처리함수(emulate()) 내에서 처리된다.

- 이후 MIPS 클래스의 객체로 mips를 생성하여 클래스 생성자를 통해 register\_file을 초기화 하고 option 정보도 초기화 한다.

- 이후 main에서 실행 명령어의 option(m,d,n) 정보를 받아 MIPS 클래스의 멤버변수 option에 각각 저장한다. d option의 경우, 뒤에 별다른 숫자가 명시되지 않기에 d가 들어왔다는 표시로 option["d"]에는 1을 저장하여 처리한다.

- 이후 main에서 mips.read\_file 멤버함수를 통해 .o 파일을 읽어오며, 메모리 영역 text, data에 각각 instruction binary code, 16진수 data를 넣는다.

- main에서 emulator 멤버함수를 실행하여 주어진 option 조건에 맞춰 emulate()멤버 함수를 호출하여 각 instruction을 수행하게 된다. emulate()함수 내에서는 J, Jal을 따로 처리하며, 그 외의 R, I type의 instruction의 경우, R\_emulate, I\_emulate 멤버함수를 통해 수행하게 만들어 각 instruction 별 처리를 담당하게 하였다

- 마지막으로, emulator 멤버함수에서는 각 실행마다 option(m,n,d) 조건을 확인하여 output() 멤버 함수를 통해 자동 출력한다.

## 2. 과제의 컴파일 방법 및 컴파일 환경(사용한 OS, 컴파일러 버전)

VMware workstation 환경에서 g++ (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0 버전을 이용하여 다음 명령어로 컴파일 하였다.

```
g++ -o main main.cpp
```

## 3. 과제의 실행 방법 및 실행 환경

과제 안내 pdf예제 명령어들과 유사하게 다음과 같은 명령어들로 실행하였으며,

```
./main -m 0x400000:0x400004 -n 20 -d sample.o
```

.o파일이 들어있는 파일경로에서 위의 명령어를 이용하여 실행하였다.