

## 1. instructions를 번역하는 방법

- 입력한 명령어는 token 단위로 번역이 되므로, token[0]이 어떤 MIPS 명령어를 갖는지에 따라 구분
  - 구분 기준은 R-format, I-format, S-format
    - I format의 lw, sw의 경우 다른 format과 다른 instruction 기준을 가지므로 따로 구분

## 2. Token 및 immediate value를 기계어로 번역하는 방법 및 Register name -> register number로 번역하는 방법

- 각각의 token을 전부 문자열 형태의 2진수로 저장하고, 마지막에 16진수로 return 하는 것으로 결정
- 따라서 register name과 number가 대응되도록 MIPS\_Registers에 저장해두고, token값과 같은 문자열을 가지는 레지스터를 찾아서 return 할 수 있도록 함
  - 이때, 모든 리턴값은 문자열 기반
- R format
  - Opcode, shamt= 0
  - Rs, rt, rd는 token에 해당하는 레지스터의 값을 찾아서 레지스터 값을 리턴함
  - Funct 역시 각 명령어에 따라 구분하여 대입
  - 위의 과정으로 opcode, rs, rt, rd, shamt, funct의 모든 값을 2진수 형태의 문자열을 가지고 있기 때문에 sprintf로 모든 문자열을 합치고, strtol을 이용해 정수형 2진수로 변환하여 리턴함
- S format
  - R format과 같은 방법으로 rs, rt 값을 각각 지정
  - Rs, opcode는 0이므로 0으로 세팅
  - S format의 경우 shamt값을 10진수로도, 16진수로도 입력받음
    - 그렇기 때문에 tokens[3][1]이 x인 경우 (==16진수인 경우)에 대해 strtol로 16진수 정수형으로 대입받도록 함
    - 아닌 경우에 대해서는 10진수 정수형으로 대입받도록 함
    - 정수형으로 대입받아 변수 shamtInt에 저장
    - 결국 2진수로 만들어주어야하기 때문에, shift 연산과 논리연산을 이용하여 shamtInt값을 2진수 형태로 바꾸어 shamt 변수에 저장함
  - 위의 과정으로 opcode, rs, rt, rd, shamt, funct를 구하고, R format과 동일한 과정으로 sprintf로 모든 문자열을 합치고, strtol을 이용해 정수형 2진수로 변환하여 리턴하도록 함

- I format
  - Lw, sw의 경우
    - 앞과 같은 방법으로 opcode, rt, rs를 세팅함
    - Lw, sw의 경우, 입력받은 token에서 rt, rs를 세팅하는 순서가 나머지와 다르기 때문에 lw, sw와 나머지를 구분함
    - Lw, sw의 경우 tokens[2]에서 offset을 입력 받으므로, S format의 경우와 반대로 tokens[2][1]이 x인경우에 대해 16진수 변환 수행
    - 마지막에는 2진수 문자열 형태로 바꾸어주어야하므로, offset에 대해 16bit의 2진수로 변환과정 수행
    - 위의 과정으로 opcode, rs, rt, offset을 모두 구하고, R, I format과 동일한 과정으로 문자형 2진수를 정수형 2진수로 변환하여 리턴하도록함
  - Lw, sw를 제외한 나머지 I format
    - 앞과 동일한 방법으로 opcode, rt, rs를 세팅함
    - token[3]이 음수인지, 16진수인지, 10진수인지에 대해 구분이 필요하기 때문에 tokens[3][1].tokens[3][0]값에 대해 비교를 수행하여 구분
    - 마지막 16bit의 constant, address부분은 문자열인 token의 값을 해당 진수에 맞는 정수형으로 변환하여, 비트 연산을 통해 정수형 2진수로 변환
      - 이때, 16bit의 2진수로 만들도록 함
    - 위의 과정으로 opcode, rs, rt, rd, constant or address를 구하고, 위의 과정과 동일하게 문자열 2진수를 정수형 2진수로 변환하여 리턴하도록함

### 3. 음수값을 번역하는 방법 (예시값 포함)

이번 과제에서는 -0x10, -0x4bad 등의 형태로 2진수값을 받게 된다. pa1.c의 경우, 기본적인 매커니즘이 명령어에 해당하는 모든 값을 2진수 형태를 가지는 문자열 자료형을 가지기 때문에, 음수값을 처리하기 위해 다음과 같은 매커니즘을 따른다.

-0x10을 10진수로 변환 -> -16 (10진수) -> 16을 2의 보수로 변환하여 2진수로 음수값을 표현하도록함

Ex) -0x10 -> -16(10) -> 1111111111110000(2)