

컴퓨터 구조 Project 1 보고서

201711174 조준규

0. 컴파일/실행 방법, 환경

코드는 Windows 10 운영체제에서 Visual Studio 2019 IDE 를 사용하여 C++로 작성하였다. 테스트는 WSL(Windows Subsystem for Linux)의 Ubuntu 18.04 환경에서 진행하였다. 컴파일러는 g++ 7.5.0 버전을 사용하였고 옵션으로 -std=c++14 를 추가하여 컴파일하였다.

컴파일과 실행은 다음과 같은 command 를 통해 진행할 수 있다.

```
$ g++ -c main.cpp -std=c++14
```

```
$ g++ -c format.h format.cpp -std=c++14
```

```
$ g++ -o runfile main.o format.o -std=c++14
```

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

1. format.h format.cpp

Format 이라는 class 를 생성하여 각 명령어의 형식에 따라 다른 함수가 적용될 수 있도록 하였다.

string rform2bin(string inst, string op, string rs, string rt, string rd, string shamt, string funct)

<매개변수>

- string inst 는 instruction 의 이름을 저장하는 매개변수이다.
- string op 는 opcode 를 저장하는 매개변수이다.
- string rs 는 rs 레지스터의 번호를 저장하는 매개변수이다.
- string rt 는 rt 레지스터의 번호를 저장하는 매개변수이다.
- string shamt 는 shift 연산을 할 때 자리이동 량을 저장하는 매개변수이다.
- string funct 는 연산의 종류를 나타내는 값을 저장하는 매개변수이다.

<함수설명>

32bit 2 진수를 저장할 strInst 라는 문자열을 선언한다. inst 를 제외한 rform2bin 의 각 인자에 bitset<n>(stoi(k)).to_string() 함수를 적용하여 더하고 strInst 에 저장한다. 이는 문자열 k 를 정수로 만들고 n bit 2 진수로 바꾼 후 다시 문자열로 변환하는 함수다. 변환을 마치면 strInst 를 반환한다.

string iform2bin(string inst, string op, string rs, string rt, string immediate)

<매개변수>

- string inst 는 instruction 의 이름을 저장하는 매개변수이다.
- string op 는 opcode 를 저장하는 매개변수이다.
- string rs 는 rs 레지스터의 이름을 저장하는 매개변수이다.
- string rt 는 rt 레지스터의 이름을 저장하는 매개변수이다.
- string immediate 는 입력된 상수 혹은 주소를 저장하는 매개변수이다.

<함수설명>

rform2bin 과 동일한 방식으로 진행된다. 단, immediate 에 16 진수와 10 진수 모두 올 수 있기 때문에 16 진수를 받는 경우 stoi(immediate, nullptr, 16)에서와 같이 16 이라는 인자를 추가하여 변환시킬 문자열이 16 진수로 표현되어 있는 수라는 것을 알려주었다.

string jform2bin(string inst, string op, string jump_target)

<매개변수>

- string inst 는 instruction 의 이름을 저장하는 매개변수이다.
- string op 는 opcode 를 저장하는 매개변수이다.
- string jump_target 은 연산의 종류를 나타내는 값을 저장하는 매개변수이다.

<함수설명>

rform2bin 과 동일한 방식으로 진행된다. 단, jump_target 는 16 진수만 받기 때문에 경우에 상관없이 16 이라는 인자를 추가하여 변환시킬 문자열이 16 진수로 표현되어 있는 수라는 것을 알려주었다.

2. main.cpp

Main 함수에서는 크게 파일을 읽어 word 의 데이터와 label 을 저장하는 과정과 파일을 읽어 instruction 을 binary 코드로 변환하는 과정으로 나뉘어져 있다.

2.0 파일명 확인

fstream 으로 파일을 불러와 파일명을 확인한다. 파일명이 틀리면 재입력해달라는 문구와 함께 main 함수를 return 한다.

2.1 word 데이터와 label 저장

파일을 읽을 때는 한 단어씩 읽게 된다. 만약 ".data"를 읽게 된다면 data section 안에 있는 label 의 이름과 index 를 저장한다. ".text"를 읽게 되면 루프를 빠져나온다.

데이터 저장을 마친 후에는 text section 의 label 을 찾아 읽게 된다. 명령어와 label 의 이름 및 index 를 저장하며 파일 끝까지 읽는다. 이후엔 파일을 닫는다.

2.2 instruction 을 binary 로 저장

파일을 읽을 때는 한 단어씩 읽게 된다. 만약 ".data"를 읽게 된다면 data section 안에 있는 word 의 value 를 저장한다. ".text"를 읽게 되면 루프를 빠져나온다.

데이터 저장을 마친 후에는 text section 의 명령어를 찾아 읽게 된다. 각 명령어에 해당하는 함수와 인자를 입력하여 실행하게 된다.

2.3 파일 저장

입력된 파일 이름을 가진 output 파일에 데이터들을 출력한다. 각 값은 16 진수로 출력된다.