

컴퓨터 구조 Project 1, Simple MIPS assembler: Report

201911039 김태연

1. 수행 과제에 대한 간략한 설명

A. 과제 내용

이 과제는 일부 instruction set을 사용하는 MIPS assembler 파일을 받아 데이터와 instruction을 binary file로 변환하여 출력하는 기능을 하는 프로그램을 작성하는 과제이다. 출력 파일에는 instruction수와 data수, instruction과 data가 16진수 형태로 한 줄씩 출력되어 있다.

B. 과제 파일(.cpp, .h)

proj1.cpp, modInst.cpp, modInst.h, modNum.cpp, modNum.h, modFile.cpp, modFile.h 파일을 작성하여 과제를 해결하였다. proj1.cpp는 main함수를 포함하고 있는 파일이며, 모든 헤더 파일을 include한다. modInst.cpp는 진법에 따른 숫자 변환을 위해 modNum.h를 함께 include하고 있다. 각 파일들의 성격과 내용은 다음과 같다.

파일이름	성격	내용
proj1.cpp	실행 파일	main함수 포함
proj1_modFile.cpp	파일 입출력과 관련	파일 이름을 인자로 받아 파일의 내용을 string list에 넣어 반환하는 함수
proj1_modNum.cpp	숫자 형변환 관련	Decimal에서 hexadecimal, decimal에서 binary, 2's complement 구현 함수 등.
proj1_modInst.cpp	Instruction 변환 관련	R,I,J 타입에 따른 각 명령어들과 operation code의 map 이를 매칭해 주고 instruction을 형태에 맞게 binary code 변환한 뒤 16진수 text로 반환하는 함수. Pseudo instruction인 la를 실제 instruction으로 변환하여 반환하는 함수 등.
헤더파일들	각 cpp file의 함수 선언	이름이 동일한 cpp file의 함수 및 사용 라이브러리 include부를 담고 있다. modInst.h의 경우 전역변수인 Lable map을 선언하고 있다.

C. 수행 원리

어셈블리 코드가 바이너리 코드로 변환되는 과정은 다음의 단계를 거친다.

1) 실행 시 샘플 파일이름을 입력받아, 파일 내용을 한 줄씩 list<string>에 저장하고 파일을 닫는다.

2) 파일이 담겨있는 리스트에서 먼저 .data 하위의 정보를 읽어온다. Label과 그에 대응되는 주소를 저장할 컨테이너를 생성하고, 한 줄씩 읽으며 label이 등장하면 컨테이너에 주소와 함께 저장한다. data의 값은 data를 저장할 선언해둔 queue에 넣는다. data를 읽을 때마다 주소는 0x10000000에서 +4씩 증가한다.

3) .text하위의 instruction 정보를 읽어 오기 전에, la instruction code가 있는지 검사하고 있다면 실제 instruction(lui 혹은 lui, ori)으로 변경하여 리스트에 삽입한다. 기존의 la instruction code는 삭제한다.

4) .text하위의 instruction 정보를 읽어온다. label이 있다면 label과 주소를 저장하는 컨테이너에 저장하고, instruction은 hexadecimal code로 변환하여 instruction queue에 넣는다. instruction을 읽을 때마다 주소는 0x00400000에서 4씩 증가한다.

4-1) instruction을 변환할 때, 만일 label이름으로 된 target이 있다면 Label과 주소를 저장해둔 map에서 찾아와 주소로 바꾸어 사용한다. BEQ와 BNE의 경우, 주소를 받아와 offset을 역으로 계산하여 사용하였다.

5) 입력 파일과 동명의(타입은 .o) 파일을 생성하고, 한 줄씩 읽으며 세어 둔 data와 instruction의 size를 출력하고, 차례대로 instruction과 data를 출력한다.

2. 컴파일 및 실행 방법과 환경

Ubuntu 20.04 환경에서 **g++ 버전 9.3.0**으로 컴파일하여 실행하였다.

다음과 같이 컴파일하여 runfile인 a.out을 생성한 뒤 명령어로 실행하였다.

컴파일 : **g++ proj1.cpp modInst.cpp modFile.cpp modNum.cpp**

실행: **./a.out sample.s**