CSE305 Project 2: Building a Simple MIPS Emulator

201811011 곽지혁

- Ubuntu 20.04.5 LTS 환경에서 gcc 9.4.0 버전을 이용하여 다음 명령어로 컴파일하였다.

  gcc -o <runfile> project2.c -lm

- 실행 명령어는 다음과 같다.

  $ ./runfile [-m addr1:addr2] [-d] [-n num_instruction] input file

- 명령어 Option에 관한 설명은 다음과 같다.

- m: 프로그램이 종료될 때 메모리 주소 범위 (addr1 ~ addr2)에 있는 내용들을 출력한다. 해당 주소가 data section의 주소(0x10000000에서 시작) 혹은 text section의 주소(0x400000에서 시작)를 벗어난 범위를 가리키고 있을 경우, 해당주소에는 값이 할당되지 않았으므로 0x0을 출력한다. 디폴트 값은 없기 때문에 본 플래그가 있을 때는 항상 주소 값을 입력해주어야 한다.

- d: 한 인스트럭션의 실행이 끝날 때마다 모든 레지스터(R0-R31, PC)의 내용을 출력한다. –m 옵션으로 출력할 메모리 주소 범위가 지정되어 있다면 지정된 메모리 범위의 내용도 출력한다.

- n: 수행될 명령어의 개수를 지정한다. 디폴트 값은 없기 때문에 본 플래그가 있을 때는 항상 개수를 지정해주어야 한다. -d 옵션이 지정되지 않았을 경우에는 프로그램이 종료되는 시점에 모든 레지스터(R0-R31, PC)의 내용을 출력한다. 만약 -m 옵션을 통해 출력할 메모리가 지정되었다면 프로그램이 종료되는 시점에 지정된 메모리의 내용 또한 출력해준다.

- 전체적인 Flow는 다음과 같다.

1. 기본적인 세팅을 한다. (register array 생성 및 초기화, pc 생성 및 0x400000로 초기화)
2. 인자로 받은 옵션들을 저장한다.
3. 인자로 받은 파일을 열어 buf에 저장해준다.

   2-1. 만일 이때 파일의 사이즈가 0이라면, option d, m에 맞추어 레지스터와 메모리를 출력하고 종료한다.

4. buf에서 Text Segment와 Data Segment의 size를 읽어와 각각의 array를 생성하고, 해당하는 Text값과 Data값들을 저장해준다.
5. While 문을 돌면서 해당하는 pc의 값에 맞추어 intruction을 읽어온다. 반복문이 돌때마다 n의 값을 1씩 줄여 n이 0이 된다면 해당 반복문을 멈춘다. 만일 option n을 주지 않았다면 n의 값은 음수로 초기화되어 pc값에 해당하는 instruction이 없을 때까지 반복하게 된다.
6. Pc를 업데이트 해준다.(pc = pc + 4)
7. 앞의 6bit를 읽고 op를 구하고 이를 기반으로 format을 구분한다.
8. Format에 맞추어 작동한다.

   8-1. R일 경우 rs, rt, rd, shamt, funct로 구분해 각각을 구하고 funct의 값에 맞추어 계산한다.

   8-2. J일 경우 target을 구하고 ra 즉 register의 31번째 값을 pc로 바꾸어 준 후에 target에 맞추어 pc를 업데이트 해준다.

   8-3. I일 경우 rs, rt를 구하고 imm를 zero_extension한 uimm이랑, sign_extionsion한 simm을 따로 구한다. Op에 맞게 계산해준다.

9. 반복문이 종료되었다면 option d와 m에 맞추어 출력해준다.
10. 종료.