실습 10주차 과제 12161633 이영주

1) 코드 설명

OPCODE가 ‘START’ 인 OPERAND를 찾아 출력하기 전에 우선, 이 프로그램이 실행될 pycharm에프로그램 파라미터를 미리 입력한다. 함수가 하나의 입력파라미터만을 가지고 있는지 확인한다. 이 기능은 checkArgv()라는 함수에서 구현한다. 만약 sys.argv의 길이가 1이라면 어셈블러 파일의 이름이 지정이 안된 것이므로 잘못된 형식이다. “Please set asm filename”이라는 문구를 출력한다. 만약 길이가 3 이상이라면 어셈블러 파일에 하나 이상의 입력파라미터가 주어진 것이므로 이것 또한 잘 못 된 것이다. “Please only set 1 argument for asm filename”이라는 문장을 출력하여 수정할 수 있도록 한다. 그 후, readAsmFile(filepath)라는 함수를 통해 경로로부터 파일을 읽어올 수 있다. 먼저, 파일을 읽고 파일에 저장된 것들을 문장 단위로 저장한 후, 반환한다. 문장을 읽어 입력된 것을 구분해주는 readAsmLine(line)을 구현한다. split() 함수를 통해 읽어온 문장을 나눠주고 그 값을 line\_split에 저장한다. label, opcode, operand를 뜻하는 line\_label, line\_opcode, line\_operand를 초기화 시켜준다. 각 문장에는 2가지 경우가 존재한다. 처음에 label이 있는 문장과 없는 문장으로 나뉜다. 만약 line\_split의 전체 길이가 2라면 label이 없는 것을 뜻한다. 그러므로 나뉜 2가지의 값을 각각 line\_opcode, line\_operand에 저장해준다. 만약 길이가 3이라면 label이 있는 것이므로 line\_label, line\_opcode, line\_operand에 저장한다. 이러한 함수들을 토대로 main() 함수에서 입력파라미터가 하나가 맞는지를 확인해주는 checkArgv()를 호출한다. 또한, LOCCTR이라는 값을 0으로 초기화시켜준다. 그 후, readAsmFile(sys.argv[1])을 호출하여 입력파라미터에 저장된 문장들을 lines에 반환해준다. 여러 개의 문장으로 이루어질 수 있기 때문에 반복문인 for문을 사용하여 처음문장부터 마지막 문장까지 확인해준다. 각 문장마다 readAsmLine(line)을 이용하여 line\_label, line\_opcode, line\_operand값을 정해준다. 만약 line\_opcode가 ‘START’를 나타내는 byte값이라면 그 주소를 나타내는 line\_operand를 int형태로 변환하여 start\_addr에 저장한다. 0으로 초기화된 LOCCTR에 start\_addr을 저장한다. 그 다음 print(‘Start address is {:d}.formant(LOCCTR))를 통해 “Start address is 3000”이라는 문장을 출력해준다.

2) 출력 결과

