

컴퓨터 구조 Project2 보고서

202011125 유선우

1. 컴파일/실행 방법, 환경

코딩과 테스트는 docker를 이용한 ubuntu 20.04에서 진행하였다.

해당 환경에서 g++의 버전은 9.4.0을 사용하였고,

다음 command를 통해서 컴파일과 실행을 할 수 있었다.

```
$ g++ emulator.cpp -o runfile
```

```
$ ./runfile [-m addr1:addr2] [-d] [-n num_instruction] <object file>
```

2. 코드 작동 Flow

Emulator의 동작을 하기 위해서는

- 1) Input command parsing
- 2) Input file parsing
- 3) Input file의 data, text를 memory에 할당
- 4) Instruction을 line by line으로 실행하며 1)에서 확인한 method에 맞춰 정보 출력

의 순서를 거쳐야 한다.

- 1) Input command parsing

Input command에 -m이 들어오면 printMem 함수의 실행여부 파라미터를 활성화하여 print가 될 때마다 printMem함수를 실행하여 addr1 ~ addr2 사이에 할당된 값들을 프린트한다.

-d가 들어오면 printEveryTime이라는 변수를 활성화하여 매 iteration마다 instruction이 실행된 이후 printReg함수를 통해 register의 값과 -m도 들어왔다면

printMem을 함께 실행한다.

-n이 들어오지 않으면 모든 text가 실행될 때까지 작동되는데, -n이 들어오면 해당 iteration을 text가 끝날 때까지가 아니라 횟수를 할당해주어서 횟수가 끝나면 프로그램이 작동이 멈추도록 한다.

2) Input file parsing

Object file의 모든 line을 한 줄씩 읽으며, 10진수 값으로 변환하여 vector에 push_back해 저장한다.

3) Input file의 data, text를 memory에 할당

Object file의 첫번째, 두번째 줄을 확인하여 text와 data값을 word단위로 메모리에 할당해준다.

4) Instruction을 line by line으로 실행하며 1)에서 확인한 method에 맞춰 정보 출력

While(1)문을 사용하여 반복적으로 instruction을 실행하고, PC를 매번 PC+4로 이동시킨다. Input command의 -n값이 할당되어 있거나 text가 할당된 메모리 주소를 벗어날 경우를 while문의 탈출 조건으로 걸어주었다.

해당 방식으로 반복적으로 instruction을 실행하여 register값과 PC값을 update해주며 input command에 맞춰 출력을 해주도록 구현하였다.