

# 시스템 프로그래밍 프로젝트 #2

## \* 프로젝트 내용

- ControlSection 방식으로 생성된 ObjectCode(프로젝트#1의 결과물)를 실행하고 시뮬레이션할 수 있는 시뮬레이터 만들기
- 시뮬레이션 과정이 Step-by-Step으로 Visual하게 보여주는 Java GUI 프로그램
- GUI를 위한 모듈, 연산 모듈, 가상 장치(메모리, 레지스터) 모듈, 로더를 통하여 시뮬레이터 구현

## \* 과제 목적

- ControlSection 방식으로 생성된 ObjectCode를 입력으로 삼아 실제 코드가 동작하는 방식을 시뮬레이션할 수 있는 GUI Java 프로그램을 만든다.

## \* 과제 제출 마감 - 6/6(일) 오후 6:00까지 스마트 캠퍼스 과제란에 제출

(제출시간 이후 제출 시 10 point 패널티 부과, 이후부터는 매일 10 point씩 추가 패널티를 부과함)

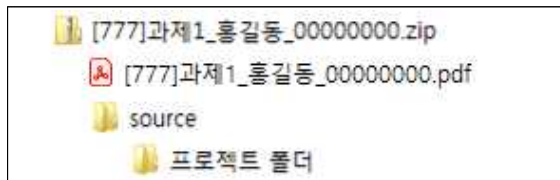
## \* 제출물 - 레포트 파일(PDF) + 프로그램 소스코드

### \* 제출 레포트(50 point)

- 요구사항 : 미니 커뮤니티 혹은 스마트 캠퍼스에 올라간 보고서 양식을 사용할 것  
(5p) (학번, 이름, 출석번호, 과제명, 수업 구분<가,나>)
- 1. 동기/목적(5p)    2. 설계/구현 아이디어(10p)    3. 수행결과(10p)
- 4. 결론 및 보충할 점(10p)    5. 소스코드(+주석)(10p)
- 소스코드는 2 column으로 출력할 것 (별도로 아래와 같이 파일로도 제출)
- 점수 평가에서 레포트의 비중이 높으므로 제출 마감 전까지 성심껏 작성하기 바랍니다.

### \* 제출 파일양식 - [출석번호]프로젝트2\_이름\_학번.zip

- 레포트 파일은 한글문서, 워드문서로 한정
- 소스코드는 프로젝트 폴더를 그대로 첨부
  - 비주얼 스튜디오의 경우 .sln 파일 반드시 첨부
- 소스코드의 "00000000"위치는 자신의 학번으로 교체



(제출 파일 구성 예시)

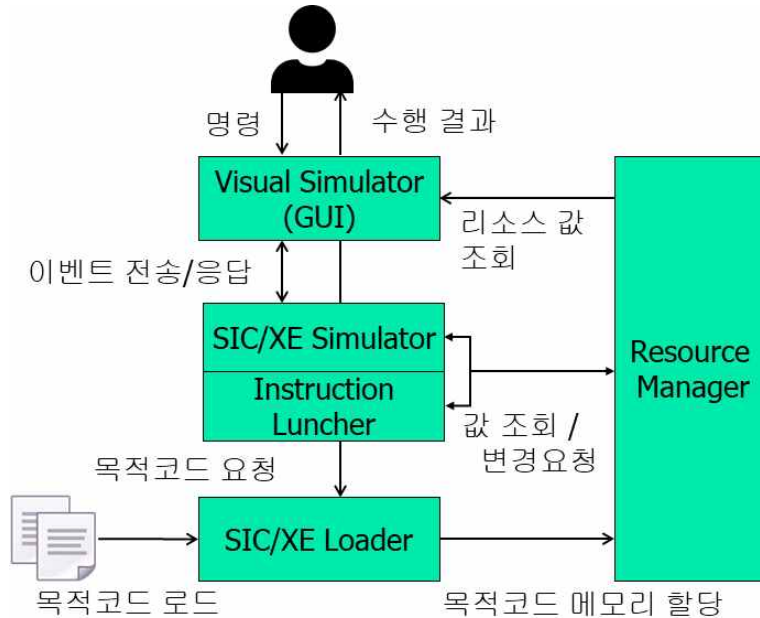
### \* 제출 파일양식을 지키지 않을 시 미제출로 간주

### \* 기간 내에 레포트 및 파일 미제출 시에 Late Penalty 부여

### \* 프로그램 Input은 프로젝트#1에서 진행한 Control Section의 목적코드 파일로 한다.

**\* 프로그램 구현 시 다음과 같은 양식을 따를 것**

- 프로그램을 구성하는 모듈은 기본적으로 다음과 같은 형태를 지닌다.



- SIC/XE Loader

시뮬레이터에서 가장 먼저 수행하는 모듈로써 어셈블러로 만들어진 목적 코드를 읽어, Resource Manager에 변수로 지정되어 있는 가상의 메모리 영역에 로드해주는 역할을 수행한다.

- Resource Manager

SIC/XE 머신은 실제 물리적인 하드웨어가 아니기 때문에 시뮬레이터를 구동시키기 위해서는 가상의 하드웨어 장치를 가정하여야 한다. 리소스 매니저에서는 목적 코드를 올리기 위한 메모리 영역, 프로세서가 연산에 사용하는 레지스터영역 등을 각각의 변수로 배정하여 사용한다.

- SIC/XE Simulator

로더를 이용하여 리소스 매니저에 목적코드를 로드시키고 로드된 코드를 실행하여 실질적으로 SIC/XE 머신이 목적코드를 수행하는 것과 유사하게 동작을 수행시키는 시뮬레이터 모듈이다.

- Instruction Luncher

SIC/XE 머신이 명령어에 따라 수행할 동작을 정의한 모듈이다. SIC/XE Simulator에서 Instruction Luncher를 통해 목적코드를 수행한다.

- Visual Simulator

시뮬레이터의 동작을 GUI 방식으로 보여주는 모듈이다. 실질적으로 시뮬레이터가 수행하는 작업은 SIC/XE 시뮬레이터 모듈에게 전달시키며, SIC/XE 시뮬레이터가 동작시킨 이후 리소스 매니저 내의 값들을 읽어 사용자에게 보여준다.

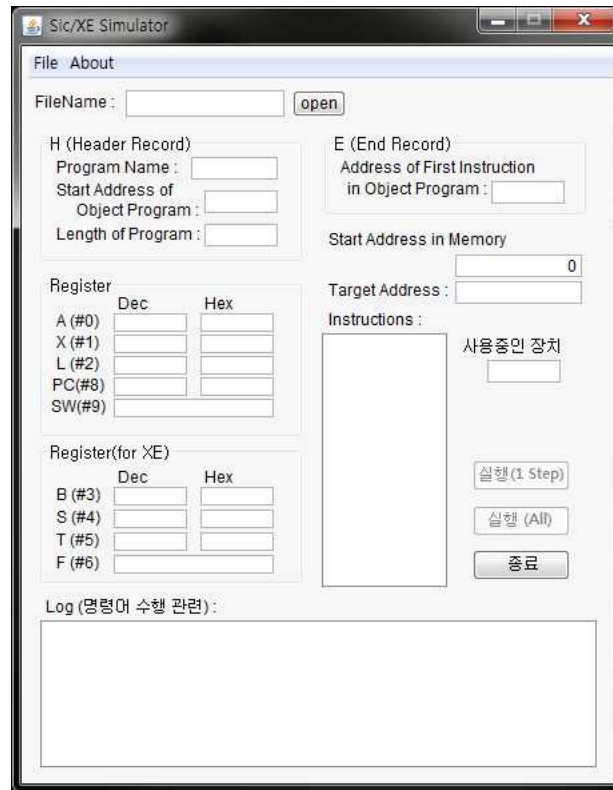
**\* 위 모듈 구성은 기본적인 구성이고, 각 모듈에 의존적인 하부 모듈을 추가로 만드는 것은 자유이다. (단, 그 모듈은 해당 상위 모듈에서만 사용하는 하부 모듈이어야 한다.)**

**\* GUI 구현시 Swing 등을 사용하여야 하기 때문에 JFrame 등을 extend하여야 한다.**

**\* 구현하여야 하는 GUI에서의 기능**

- 프로그램 종료 버튼
- 파일 오픈기능(파일 오픈 다이얼로그 창 이용)
- 레지스터 영역(SIC 및 SIC/XE머신의 레지스터를 모두 포함)
- 메모리 영역(다음과 같은 두 가지 표현 방식 중 한가지 방식으로 보여주어야 한다)
  1. 가상으로 설정한 메모리를 직접 보여주고, 현재 수행되고 있는 Instruction의 주소를 포인팅 해준다(가능하면 해당 Instruction을 영역지정까지)
  2. 메모리에 올라간 코드를 파싱하여 명령어 목록을 만든다. 그리고 해당 리스트를 표시 해준다. 현재 수행되고 있는 명령어를 리스트에서 선택하여 표시해준다.
- 프로그램 정보  
(프로그램 길이, 현재 포인팅 주소, 사용중인 장치, 현재 수행되고 있는 명령어 정보 등)
- 1 step 및 all step 기능

\* 초기화면(예시)



\* 파일 오픈 시(파일 다이얼로그 창)

