

시스템프로그래밍 2021 보고서

보고서 제출서약서

나는 숭실대학교 컴퓨터학부의 일원으로 명예를 지키면서 생활하고 있습니다. 나는 보고서를 작성하면서 다음과 같은 사항을 준수하였음을 엄숙히 서약합니다.

- 1. 나는 자력으로 보고서를 작성하였습니다.
 - 1.1. 나는 동료의 보고서를 베끼지 않았습니다.
 - 1.2. 나는 비공식적으로 얻은 해답/해설을 기초로 보고서를 작성하지 않았습니다.
- 2. 나는 보고서에서 참조한 문헌의 출처를 밝혔으며 표절하지 않았습니다. (나는 특히 인터넷에서 다운로드한 내용을 보고서에 거의 그대로 복사하여 사용하지 않았습니다.)
- 3. 나는 보고서를 제출하기 전에 동료에게 보여주지 않았습니다.
- 4. 나는 보고서의 내용을 조작하거나 날조하지 않았습니다.

과목	시스템프로그래밍 2021	
과제명	프로젝트2	
담당교수	최 재 영 교 수	
제출인	컴퓨터학부 20192698 심원준 (출석번호 124번)	
제출일	2021년 6월 6일	

- 1장 프로젝트 동기/목적
- 2장 설계/구현 아이디어
- 3장 수행결과(구현 화면 포함)
- 4장 결론 및 보충할 점
- 5장 디버깅
- 6장 소스코드(+주석)

1.프로젝트 동기/목적

이 프로젝트는 SIC/XE Object Code를 실행하며 이 과정을 보여줄 수 있는 로더와 시뮬레이터를 포함한 JAVA GUI프로그램을 제작함을 목적으로 한다.

2.설계/구현 아이디어

SIC Loader/ResourceManager/InstLuncher/SicSimulator/SymbolTable/VisualSimulator의 6개 .java(class)와 Instluncher와 SicSimulator에서 명령어를 분석하기위한 instruction class로 구성되어있다.

가.SicLoader

입력된 objectcode를 memory에 올리는 것을 목표로한다.

objectcode 각 줄의 첫 글자를 읽어 'H','T','D','R','M'에 따라 분류하여 동작을 달리한다. 첫 pass에서는 'D'의 줄을 읽어와 symbolTable 클래스의 인스턴스에 extdef 심볼을 저장한다.

두 번째 pass에서는 'T' 줄을 읽어와 메모리에 올리고, 첫 글자가 'M'인 modification code 에 정의된 위치에 symbolTable에서 찾아온 주소를 더하거나 뺀다.

이 과정을 마치면 objectcode가 memory에 올라가게된다.

나.ResourceManager

ResourceManager은 SicLoader가 objectcode를 올릴 메모리가 char[]배열로 정의되어있으며, register들 또한 정의되어있다. 또한 Memory에 저장하고, 찾아올 수 있도록 하는 getMemory, setMemory메소드가 정의되어있다. Register에 저장하고 찾아올 수 있는 setRegister, getRegister 또한 정의되어있다. WD,TD,RD작업에 이용될 디바이스를 관리하는 deviceManager arraylist가 정의되어있으며 이들 메소드들 또한 정의되어있다.

다.InstLuncher

InstLuncher에는 각 명령어에 따른 동작이 정의되어있다. SIC/XE Appendix에 나와있는 명령어들을 정의해 두었으며, 이들이 하는 동작에 따라 resourceManager의 memory/register/device가 수정 및 동작한다. 또한 Instruction class 또한 정의되어있는데, 각 명령어가 작동될 때 받은 명령어와 주소값을 받아서 flag/대상 주소/레지스터 값을 분석하는 역할을 한다.

라.SIC Simulator

Sic Simulator는 visual Simulator로부터 받은 event에 따라서 동작을 실행하게 하는 역할을 한다. visual Simulator로부터 요청을 받아 instLuncher의 명령어 메소드를 실행하게 하며,

이 결과를 다시 visualSimulator로 보내도록 한다.

마. Visual Simulator

visual Simulator는 GUI프로그램의 frame을 정의하며, 버튼입력등의 동작(event)를 관리할 eventHandler 또한 정의되어있어 이에 따라 SIC SImulator로 요청과 응답을 주고받는다.

3.실행결과

visual simulator를 구성하지 못하여 디버깅 및 RUN 을 하지 못하였다.

4. 결론 및 보충할 점

SicSimulator와 visualSimulator의 두 클래스를 완전히 구현하지 못하여 프로그램을 완성하지 못하였다. Java의 Swing을 이용해 GUI를 구현하는 데에 알지 못하여 이에 어려움을 겪었다. 또한 Loader와 linker의 동작에 대해서 명확하게 이해를 하지 못하여 구현할 아이디어(세부적 로직)을 구성하는 데에 어려움을 겪어 이에 대한 추가적 학습이 필요하다.

5.디버깅

visual simulator를 구성하지 못하여 디버깅을 하지 못하였다.

6.소스코드

package sp21_simulator; * @param rMgr */ import java.io.BufferedReader; public void setResourceManager(ResourceManager import java.io.File; resourceManager) { import java.io.FileNotFoundException; import java.io.FileReader; this.rMar import iava.io.IOException; resourceManager; } /** * SicLoader는 프로그램을 해석해서 메모 리에 올리는 역할을 수행한다. 이 과정에서 * object code를 읽어서 load과정 linker의 역할 또한 수행한다. 수행한다. load한 데이터는 resourceManager가 관리하는 메모리에 올 * SicLoader가 수행하는 일을 예를 들면 라가도록 다음과 같다. - program code를 메모리에 * 한다. load과정에서 만들어진 적재시키기 - 주어진 공간만큼 메모리에 빈 symbol table 등 자료구조 역시 * 공간 할당하기 - 과정에서 발생하는 resourceManager에 전달한다. symbol. 프로그램 시작주소, section 등 실행을 위한 정보 생성 및 관리 * @param objectCode 읽어들인 */ 파일 public class SicLoader { * @throws IOException */ ResourceManager rMgr; int sttadd=0; public SicLoader(ResourceManager public void load(File objectCode) resourceManager) { throws IOException { // 필요하다면 초기화 FileReader finput=new FileReader(objectCode); setResourceManager(resourceManager); BufferedReader } bufread=new BufferedReader(finput); String line=new String(); sttadd=0;//starting int * Loader와 프로그램을 적재할 메 address of program 모리를 연결시킨다. int length=0;

```
while((line=bufread.readLine())!=null) {
                                               if(line.charAt(0)=='H')//Header
                                                                      {
if(line.charAt(0)=='D')
                      {
                                                if(count==0)//첫 프로그램의 경우
                                                                              {
                              for (int
i=1;i<line.length();i+=12)
                              {
                                                length=Integer.parseInt(line.substring(13),16
rMgr.symtabList.putSymbol(line.substring(i,i
                                                );
rMgr.proglength+Integer.parseInt(line.substri
                                                sttadd=Integer.parseInt(line.substring(7,13),
ng(i+6,i+12),16));
                                                16);
                              }
                      }
                                                rMgr.sttadd=sttadd;
                      е
if(line.charAt(0)=='H')
                                                length=Integer.parseInt(line.substring(13),16
                                                );
                                                rMgr.symtabList.putSymbol(line.substring(1,
rMgr.proglength+=length;
                                                7), sttadd);
length=Integer.parseInt(line.substring(13),16
);
                                                count+=1;
                      }
                                                                              }
               }
                                                                              else
               bufread.close();
                                                                              {
               finput.close();
               FileReader
                               finput2=new
                                                rMgr.proglength+=length;
FileReader(objectCode);
               BufferedReader
                                                length=Integer.parseInt(line.substring(13),16
bufread2=new BufferedReader(finput2);
                                               );
               length=0;
               rMgr.proglength=0;
                                                rMgr.symtabList.putSymbol(line.substring(1,
               int count=0;
                                                7),rMgr.proglength );
                                                                              }
                                                                       }
while((line=bufread2.readLine())!=null) {
                                                                       е
                                                                                          е
```

```
if(line.charAt(0)=='M')
                                                                                   i n t
                      {
                                              temperary=0;
                             float
fixadd=rMgr.proglength+(float)Integer.parsel
                                              temperary=Integer.parseInt(String.valueOf(r
nt(line.substring(1,7),16);
                                              Mgr.getMemory(fixadd, numf)),16);
                             i
                                         t
numf=Integer.parseInt(line.substring(7,9));
                                              String refsym=new String();
if(numf%2==1)//홀수일경우
                            0.5더해주어야
                                              refsym=line.substring(10);
함 주소값
                                                                                   i n t
                             {
                                              toadd=rMgr.symtabList.search(refsym);
fixadd+=0.5;
                                              temperary-=toadd;
                             }
                                              rMgr.setMemory(fixadd,Integer.toHexString(t
if(line.charAt(9)=='+')
                                              emperary).toCharArray(), numf);
                             {
                                                                     }
                                     i n t
temperary=0;
                                              if(line.charAt(0)=='T')
                                                                     {
temperary=Integer.parseInt(String.valueOf(r
Mgr.getMemory(fixadd, numf)),16);
                                              stad=Integer.parseInt(line.substring(1,7),16)
String refsym=new String();
                                                                                  n
                                              lenofline=0;
refsym=line.substring(10);
                                     i n t
toadd=rMgr.symtabList.search(refsym);
                                              lenofline=Integer.parseInt(line.substring(7,9)
                                               ,16);
temperary+=toadd;
                                              r Mgr. set Memory (stad,
rMgr.setMemory(fixadd,Integer.toHexString(t
                                              line.substring(9).toCharArray(), lenofline);
emperary).toCharArray(), numf);
                                                                     }
                             }
                                                             }
                             else
                                                             rMgr.proglength+=length;
```

```
rMgr.setRegister(2,rMgr.proglength);
                                                    int addr(int curr.int[] flags) {
                                                            int returningadd=0;
                                                            if(flags[4]==1)//pc
       };
                                             relative
                                                                   curr+=format;
}
                                                                   String
                                             temp=this.instruction.toString();
package sp21_simulator;
                                             returningadd=Integer.parseInt(temp.sub
                                             string(3),16);
class Instruction{
       char[]instruction;
                                             returningadd+=curr;
       int format=0;
       Instruction(char[linstruction){
                                                            else//ref
this.instruction=instruction.clone();
                                                                   String
                                             temp=this.instruction.toString();
       int[] getflag(){
              int []flags=new int[6];
                                             returningadd=Integer.parseInt(temp.sub
              format=3;
                                             string(3),16);
              for(int i=0; i<6; i++)
                                                            return returningadd;
                     flags[i]=0;
                                                    int [] registers() {
                                                           int []temp=new int[2];
if((this.instruction[1]\&2)==2)//n
                                             temp[0]=Character.getNumericValue(in
                     flags[0]=1;
                                             struction[2]);
                                             temp[1]=Character.getNumericValue(in
if((this.instruction[1]&1)==1)//i
                                             struction[3]);
                                                            format=2;
                     flags[1]=1;
                                                            return temp;
                                                    int format()
if((this.instruction[2]\&8)==8)//x
                                                            return format;
                     flags[2]=1;
if((this.instruction[2]&4)==4)//b
                                             package sp21_simulator;
                                             import java.awt.*;
                     flags[3]=1;
                                             import java.awt.event.ActionListener;
                                             import javax.swing.*;
if((this.instruction[2]\&2)==2)//p
                                             import java.io.File;
                                             import java.io.IOException;
                     flags[4]=1;
                                             * VisualSimulator는 사용자와의
상호작용을 담당한다. 즉, 버튼 클릭등의
이벤트를 전달하고 그에 따른 결과값을
화면에 업데이트
if((this.instruction[2]\&1)==1)//e
                     flags[5]=1;
                     format=4;
                                              * 하는 역할을 수행한다.
              return flags;
                                              * 실제적인 작업은 SicSimulator에서
```

```
수행하도록 구현한다.
                                      fr.setDefaultCloseOperation(JFrame.EX
                                      IT ON CLOSE);
@SuppressWarnings("serial")
                                                   Container
public class VisualSimulator extends
                                       contentpane=fr.getContentPane();
JFrame {
      ResourceManager
resourceManager = new
ResourceManager();
      SicLoader sicLoader = new
                                      package sp21_simulator;
SicLoader(resourceManager);
      SicSimulator sicSimulator =
                                      import java.util.ArrayList;
new SicSimulator(resourceManager);
      static JFrame fr;
                                       * symbol과 관련된 데이터와 연산을
                                       소유한다. section 별로 하나씩
       * 프로그램 로드 명령을
                                       인스턴스를 할당한다.
전달한다.
       * @throws IOException
                                      public class SymbolTable {
                                             ArrayList<String> symbolList;
                                             ArrayList<Integer> addressList;
      public void load(File program)
                                       // 기타 literal, external 선언 및
처리방법을 구현한다.
throws IOException {
            sicLoader.load(program);
                                              * 새로운 Symbol을 table에
sicSimulator.load(program);
                                       추가한다.
      };
                                              * @param symbol : 새로
      * 하나의 명령어만 수행할 것을
                                       추가되는 symbol의 label
SicSimulator에 요청한다.
                                              * @param address : 해당
                                       symbol이 가지는 주소값 <br>
                                                              <br>
      public void oneStep() {
                                                              주의 : 만약
                                       중복된 symbol이 putSymbol을 통해서
입력된다면 이는 프로그램 코드에 문제가
      };
      /**
                                       있음을
      * 남아있는 모든 명령어를
                                                             나타낸다.
수행할 것을 SicSimulator에 요청한다.
                                       매칭되는 주소값의 변경은
                                       modifySymbol()을 통해서 이루어져야
                                       하다.
      public void allStep() {
      };
                                             public void putSymbol(String
                                       symbol, int address) {
      /**
      * 화면을 최신값으로 갱신하는
                                      this.symbolList.add(symbol);
       */
                                       this.addressList.add(address);
      public void update() {
      };
                                       * 기존에 존재하는 symbol 값에
대해서 가리키는 주소값을 변경한다.
      public static void main(String[]
args)
            fr=new JFrame("SP21
                                       * @param symbol
원하는 symbol의 label
                                                                   : 변경을
pro2 20192698");
            fr.setSize(700,300);
                                              * @param newaddress : 새로
                                       바꾸고자 하는 주소값
            fr.setVisible(true);
```

public void modifySymbol(String symbol, int newaddress) {	,	
modifySymbol(String symbol, int newaddress) { int tindex=this.symbolList.indexOf(symbol); int tindex=this.symbolList.indexOf(symbol); int addressList.set(tindex, newaddress); ** ** *** ** ** *** * ** ** ** ** ** ** * ** ** * ** * ** ** * ** * ** * ** * ** * ** * ** * * ** *	*/	{ Instruction tomi=now
int tindex=this.symbolList.indexOf(symbol); addressList.set(tindex, newaddress); /** ** 연자로 전달된 symbol의 이번 수소를 지하는지 알려준다. ** @param symbol : 검색을 원하는 symbol의 label ** @return symbol의 가지고 있는 주소값 해당 symbol의 입을 경우 -1 리턴 */ public int search(String symbol); return address: } package sp21_simulator; import java.io.iOException; import java.util.ArrayList; // instruction에 따라 통작을 수행하는 데소드를 정의하는 글레스 public class InstLuncher { ResourceManager resourceManager resourceManager resourceManager resourceManager resourceManager resourceManager resourceManager resourceManager int []flags=new int[6]; flags=temi.getflag(); int add=temi.addr(curr,flags); int curpridgr.getRegister(0); flags=temi.getflag(); int int []flags=new int[6]; flags=temi.getflag(); int int []flags=new int [6]; flags=temi.getflag(); int int []flags=new int [6]; flags=temi.getflag(); int int []flags=new int [6]; flags=te		
int tindex=this.symbolList.indexOf(symbol); ; addressList.set(tindex, newaddress):		
int r2=rMgr.getRegister(regs[1]): rMgr.setRegister(2, r2+r1); rMgr.setRegister(2, r2+r1); rMgr.setRegister(2, r2+r1); rMgr.setRegister(2, r2+r1); rMgr.setRegister(3, rMgr.setRegister(4, r2+r1); rMgr.setRegister(8, curr+temi.format(0); } public void CLEAR(int curr,char[]instruction) ** @param symbol : 검색을 bymbol of 없을 경우 class int address = this.symbolList.indexOf(symbol): return address; } public void complete int scarch(String symbol) int llregs=temi.register(8, curr+temi.format(0)); rMgr.setRegister(regs[0], curr+temi.format(0); int llregs=temi.register(0); rMgr.setRegister(regs[0], curr+temi.format(0); lint llregs=temi.register(0); rMgr.setRegister(regs[0], curr+temi.format(0); lint llregs=temi.register(0); lint llregs=temi.register(0); lint llregs=temi.register(0); lint add=temi.addr(curr,flags); if(add==rMgr.getRegister(0)); lint add=temi.addr(curr,flags); int curr-char[]instruction temi=new lnstruction(instruction); lint llregs=temi.register(0); lint add=temi.dedr(curr,flags); int curr-char[]instruction temi=new lnstruction(instruction); lint llregs=temi.register(0); lint add=temi.dedr(curr,flags); int curr-char[]instruction(instruction); lint llregs=temi.register(0); lint add=temi.dedr(curr,flags); lint add=temi.dedr(curr,flags); lint add=temi.dedr(curr,flags); lint add=te		int
rMgr.setRegister(r2, r2+r1): rMgr.setRegiste	tindex=this.symbolList.indexOf(symbol)	rl=rMgr.getRegister(regs[0]);
rMgr.setRegister(r2, r2+r1): rMgr.setRegiste	address list set(tindex	r2=rMgr getRegister(regs[1]).
### * 인자로 전단된 symbol의 어떤 주소를 지칭하는지 알려준다. # @param symbol : 검색은 원하는 symbol의 label # @return symbol의 가지고 있는 주소값, 해당 symbol의 없을 경우 그리턴 # public int search(String symbol) # int address = this.symbolList.indexOf(symbol): return address: } package sp21_simulator: import java.uitl.ArrayList: # // instruction에 따라 동작을 수행하는 메소드를 정의하는 클래스 public class InstLuncher { ResourceManager resourceManager resourceManager (bits.rMgr = resourceManager; } public class InstLuncher(ResourceManager resourceManager) {		
# 인자로 전달된 symbol의 어떤 주소를 지칭하는지 알려준다. # @param symbol : 검색을	}	rMgr.setRegister(8,
* 인자로 전달된 symbol의 어떤 주소를 지칭하는지 알려준다. * @param symbol : 검색을 원하는 symbol의 label * @return symbol의 가지고 있는 주소설: 해당 symbol의 없을 경우 -1 라틴 */ public int search(String symbol):	,	<pre>curr+ temi.format());</pre>
주소를 지칭하는지 알려준다. ** @param symbol : 검색을 원하는 symbol의 label ** @return symbol이 가지고 있는 주소값. 해당 symbol이 없을 경우 -1 리턴 int address = this.symbollst.indexOf(symbol); return address; } package sp21_simulator; import java.io.IOException; import java.util.ArrayList; // instruction에 따라 중작을 수행하는 메소드를 정의하는 클래스 public class InstLuncher { ResourceManager resourceManage	/	}
* @param symbol : 검색을	* 현작도 선물선 Symbol의 의원 주소를 지칭하는지 알려준다	
원하는 symbol의 label 있는 주소값. 해당 symbol이 있을 경우 -1 라틴 */ public int search(String symbol) { int address = this.symbolList.indexOf(symbol); return address; } package sp21_simulator; import java.io.IOException: import java.util.ArrayList; // instruction에 따라 동작을 수행하는 메소트를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public Instruction(mstruction); int []flags=new int[6]; public void add(int curr,char[]instruction) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []regs=temi.registers(0); rMgr.setRegister(8, curr+ temi.format()); int []flags=new int[6]; int add=temi.addr(curr,flags); int []flags=new int[6]; flags=temi.getflag(); int add=temi.addr(curr,flags); int mpp=rMgr.getRegister(0); rMgr.setRegister(9, 1); rMgr.setRegister({
* @return symbol 이 가지고 있는 주소값. 해당 symbol 이 없을 경우 - 1 라인 */ public int search(String symbol) { int address = this.symbolList.indexOf(symbol); return address; } } package sp21_simulator; import java.io.IOException; import java.util.ArrayList;		
지는 주소값. 해당 symbol의 없을 경우 -1 런턴 // public int search(String symbol) { int address = this.symbolList.indexOf(symbol); return address; } package sp21_simulator; import java.io.IOException: import java.util.ArrayList; // instruction에 따라 동작을 수행하는 메소트를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int [Iflags=new int[6]: flags=temi.getRegister(0) {		
1 리턴 */ public int search(String symbol) { int address = this.symbolList.indexOf(symbol); return address; } } package sp21_simulator;		
#/ public int search(String symbol) { int address = this.symbolList.indexOf(symbol); return address; } package sp21_simulator; import java.io.IOException; import java.util.ArrayList; // instruction에 따라 동작을 수행하는 테소트를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public list.rMgr = resourceManager; } public void add(int curr,char[jinstruction) int []flags=new int[6]; flags=temi.getflag(); int add=temi.addr(curr,flags); if(add=rmi.addr(curr,flags); if(add=rmi.addr(curr,flags); int add=temi.addr(curr,flags); int add=temi.addr(curr,flags); int [] rMgr.setRegister(9, 0); } public void cOMP(int curr,char[]instruction) { rMgr.setRegister(8, curr+temi.format()); int add=temi.addr(curr,flags); int add=temi.addr(curr,flags); int add=temi.addr(curr,flags); int [] rMgr.setRegister(9, 1); } public void add(int curr,char[]instruction) { rMgr.setRegister(8, curr+temi.format()); int add=temi.addr(curr,flags); int add=temi.addr(curr,flags); int [] rMgr.setRegister(9, 0); } public void COMP(int curr,char[]instruction) { rMgr.setRegister(8, curr+temi.format()); int add=temi.addr(curr,flags);	-1 리턴	
Section of the public void ADDR(int curr, char[] instruction) Section of this. symbolList.indexOf(symbol); return address; Public void COMP(int curr, char[] instruction) Section of the public void add(int curr, char[] instruction) Section of this.rMgr = resourceManager resou	*/	rMgr.setRegister(8,
int address = this.symbolList.indexOf(symbol); return address; } creturn address; } lints.symbolList.indexOf(symbol); return address; } lints.symbolList.indexOf(symbol); return address; } lints.symbolList.indexOf(symbol); return address; } lints.coid COMP(int curr,char[]instruction)		curr+ temi.format());
this.symbolList.indexOf(symbol); return address; } curr.char[]instruction Instruction temi=new Instruction temi=new Instruction(instruction); int []flags=temi.getflag(); int add=temi.addr(curr,flags); int add=temi		public void COMP(int
return address; } Instruction temi=new Instruction(instruction); int []flags=new int[6]; flags=temi.getflag(); int add=temi.addr(curr,flags); if(add==rMgr.getRegister(0)) { rMgr.setRegister(9, 0); } else if(rMgr.getRegister(0) < add) { rMgr.setRegister(9, -1); } else rMgr.setRegister(9, 1); } public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag(); int add=temi.getflag();		
Instruction(instruction); package sp21_simulator; import java.io.IOException; import java.util.ArrayList; // instruction에 따라 동작을 수행하는 메소드를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr.char[]instruction) { Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int madd=temi.addr(curr,flags); int []flags=temi.getflag(); int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(equilibrian) {		{
int []flags=new int[6]; flags=temi.getflag(); int add=temi.addr(curr,flags); if(add==rMgr.getRegister(0)) import java.util.ArrayList; // instruction에 따라 동작을 수행하는 메소드를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(9, 1); } public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction temi=new Instruction temi=new Instruction(instruction); int []regs=temi.register(0])==rMgr.get Register(regs[0])==rMgr.get Register(regs[1])) } public void ADDR(int	}	
package sp21_simulator; import java.io.IOException; import java.util.ArrayList; // instruction에 따라 동작을 수행하는 메소트를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(end); int []regs=temi.registers(end); int []regs=temi.registers(end); int []regs=temi.registers(end); int []regs=temi.register(end); int []regs=temi.getflag(); int add=temi.addr(curr,flags); if(rMgr.getRegister(9, 0); if(rMgr.getRegister(9, 0); if(rMgr.getRegister(gol))==rMgr.get Register(end); if(rMgr.getRegister(end)); int []regs=temi.register(end); int []	1	
import java.io.IOException; import java.util.ArrayList; import java.util.ArrayList; if(add==rMgr.getRegister(0)) import java.util.ArrayList; if(add==rMgr.getRegister(0)) if(add==rMgr.getRegister(0)) if(add==rMgr.getRegister(0)); if(add==rMgr.getRegister(0), add if(array-getRegister(0), array-getRegister(0), array-getRegister(0), array-getRegister(0, array-getRegister(0), array-getRegister(0, array-getRegister(0), array-getRegister(0, array-getRegister(0), array-getRegister(0, array-getRegister(0), array-getRegister(0, array-getRegister(0, array-getRegister(0), array-getRegister(0, array-getRegister(0), array-getRegister(0, array-g	package sp21 simulator:	
import java.io.IOException; import java.util.ArrayList;	package op21_omalator,	
rMgr.setRegister(9, 0); // instruction에 따라 동작을 수행하는 메소드를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(9, 1); } rMgr.setRegister(9, 1); } rMgr.setRegister(9, 1); } rMgr.setRegister(8, curr+ temi.format()); } public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction temi=new Instruction(instruction); int []regs=temi.registers(); int []regs=temi.registers(); if(rMgr.getRegister(regs[0])==rMgr.get Register(regs[1])) { rMgr.setRegister(9, 0);		if(add==rMgr.getRegister(0))
// instruction에 따라 동작을 수행하는 메소드를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(9, 1); } public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction temi=new Instruction(instruction); int []flags=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(regs[0])==rMgr.get Register(regs[1])) { rMgr.setRegister(9,0); int []regs=temi.registers(0); register(regs[1])) { rMgr.setRegister(9,0); int []regs=temi.register(0); if(rMgr.getRegister(9,0); int []regs=temi.registers(0); int []regs=temi.register(0); if(rMgr.getRegister(9,0);	import java.util.ArrayList;	{ rMar antPagiator(0, 0):
메소드를 정의하는 클래스 public class InstLuncher { ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(9, -1); } rMgr.setRegister(9, 1); rMgr.setRegister(8, curr+ temi.format()); } public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction temi=new Instruction(instruction); int emp=rMgr.getRegister(0); int temp=rMgr.getRegister(0); rMgr.setRegister(9, 1); } public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []regs=temi.registers(); if(rMgr.getRegister(9, -1); } rMgr.setRegister(9, -1); rMgr.setRegister	// instruction에 따라 동작을 수행하는	I wigi . Settlegister (9, 0),
ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(9, 1); rMgr.setRegister(8, curr+ temi.format()); public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction temi=new Instruction(instruction); int []regs=temi.registers(); int []regs=temi.registers(); if(rMgr.getRegister(regs[0])==rMgr.get		else if(rMgr.getRegister(0) <add)< td=""></add)<>
ResourceManager rMgr; public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } public void add(int curr,char[]instruction) Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(9, 1); rMgr.setRegister(8, curr+ temi.format()); public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction temi=new Instruction(instruction); int []regs=temi.registers(); int []regs=temi.registers(); if(rMgr.getRegister(regs[0])==rMgr.get		{
<pre>public InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } rMgr.setRegister(9, 1); this.rMgr = resourceManager; } rMgr.setRegister(8, curr+ temi.format()); public void add(int { curr, char[]instruction) {</pre>		rMgr.setRegister(9, -1);
<pre>InstLuncher(ResourceManager resourceManager) { this.rMgr = resourceManager; } rMgr.setRegister(9, 1); rMgr.setRegister(8, curr+ temi.format()); public void add(int { curr,char[]instruction) { Instruction temi=new int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(0); int temp=rMgr.getRegister(0); rMgr.setRegister(regs[0])==rMgr.get Register(regs[1])) } public void ADDR(int</pre>) else
<pre>resourceManager) { this.rMgr = resourceManager; } rMgr.setRegister(9, 1); rMgr.setRegister(8,</pre>		{
rMgr.setRegister(8, curr+temi.format()); public void add(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(0, temp+add); } public void COMPR(int curr,char[]instruction) { Instruction temi=new Instruction(instruction); int []regs=temi.registers(); if(rMgr.getRegister(regs[0])==rMgr.get Register(regs[1])) } public void ADDR(int rMgr.setRegister(9,0);	resourceManager) {	rMgr.setRegister(9, 1);
<pre>curr, char[]instruction)</pre>	this.rMgr = resourceManager;	}
<pre>public void add(int curr,char[]instruction)</pre>	}	rMgr.setKegister(8,
<pre>curr,char[]instruction)</pre>	public void add(int	}
<pre>Instruction temi=new Instruction(instruction); int []flags=new int[6]; int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(0, temp+ add); } public_void_ADDR(int Instruction temi=new Instruction(instruction); int []regs=temi.registers(); if(rMgr.getRegister(regs[0])==rMgr.get Register(regs[1])) { rMgr.setRegister(9,0); } }</pre>		public void COMPR(int
<pre>Instruction(instruction);</pre>	{	curr,char[]instruction)
<pre>int []flags=new int[6]; flags=temi.getflag(); int add=temi.addr(curr,flags); int temp=rMgr.getRegister(0); rMgr.setRegister(0, temp+ add); } public void ADDR(int Instruction(instruction); int []regs=temi.registers(); if(rMgr.getRegister(regs[0])==rMgr.get Register(regs[1])) { rMgr.setRegister(9,0); }</pre>		Instruction tomi=now
<pre>flags=temi.getflag();</pre>		
<pre>int temp=rMgr.getRegister(0); rMgr.setRegister(0, temp+ add); } public void ADDR(int</pre>		
rMgr.setRegister(0, temp+add); Register(regs[1])) } public_void_ADDR(int rMgr.setRegister(9,0);		:(/ M
<pre>public void ADDR(int</pre>		II(rMgr.getKegister(regs[U])==rMgr.get
	mgr.sethegister(0, temp=add), 	{ IveRivier(LeRv[1]))
	public void ADDR(int	rMgr.setRegister(9,0);
curr, enar (jinistraction)	curr,char[]instruction)	}

```
int add=temi.addr(currloc,flags);
       else
if(rMgr.register[regs[0]]<rMgr.registe
                                                     if(rMgr.getRegister(9)==0)
r[regs[1]]
                                                             rMgr.setRegister(8,add);
              rMgr.setRegister(9, -1);
                                                     else
       else
                                                             rMgr.setRegister(8,
              rMgr.setRegister(9, 1);
                                              currloc+ temi.format());
       rMgr.setRegister(8,
curr+temi.format());
                                                  public void JLT(int
                                              currloc, char[]instruction)
    public void DIV(int
curr,char[]instruction)
                                                     Instruction temi=new
                                              Instruction(instruction);
                                                     int []flags=new int[6];
       Instruction temi=new
                                                     flags=temi.getflag();
Instruction(instruction);
       int []flags=new int[6];
                                                     int add=temi.addr(currloc,flags);
       flags=temi.getflag();
                                                     if(rMgr.getRegister(9)<0)
       int add=temi.addr(curr,flags);
                                                             rMgr.setRegister(8,add);
rMgr.setRegister(0,rMgr.getRegister(0)
/add);
                                                     else
       rMgr.setRegister(8,
curr+ temi.format());
                                              rMgr.setRegister(8,currloc+temi.forma
    public void DIVR(int
                                              t());
curr,char[]instruction)
                                              public void JSUB(int
curr,char[]instruction)
       Instruction temi=new
Instruction(instruction);
       int []regs=temi.registers();
       rMgr.setRegister(regs[1],
                                                     Instruction temi=new
rMgr.getRegister(regs[1])/rMgr.getReg
                                              Instruction(instruction);
ister(regs[0]));
                                                     int []flags=new int[6];
                                                     flags=temi.getflag();
       rMgr.setRegister(8,
curr+temi.format());
                                                     int add=temi.addr(curr,flags);
                                                     rMgr.setRegister(2,
public void J(int
curr,char[]instruction)
                                              rMgr.getRegister(8));
                                                     rMgr.setRegister(8, add);
       Instruction temi=new
                                                  public void LDA(int
Instruction(instruction);
                                              curr,char[]instruction)
       int []flags=new int[6];
       flags=temi.getflag();
                                                     Instruction temi=new
       int add=temi.addr(curr,flags);
                                              Instruction(instruction);
rMgr.setRegister(8, add);
//pc register에 대상
address저장
                                                     int []flags=new int[6];
                                                     flags=temi.getflag();
                                                     int add=temi.addr(curr,flags);
                                                     rMgr.setRegister(0, add);
                                                     rMgr.setRegister(8,
    public void JEQ(int
currloc,char[]instruction)
                                              curr+ temi.format());
                                              public void LDCH(int
curr,char[]instruction)
       Instruction temi=new
Instruction(instruction);
       int []flags=new int[6];
       flags=temi.getflag();
                                                     Instruction temi=new
```

Instruction(instruction);	<pre>int add=temi.addr(curr,flags);</pre>
int []flags=new int[6];	int add teim, addi (edi i , nago),
flags=temi.getflag();	rMgr.setMemory(add,Integer.toHexStri
int add=temi.addr(curr,flags);	ng(rMgr.getRegister(3)).toCharArray(),
int (1)/0):	1);
target=add+ (rMgr.getRegister(1)/2);	rMgr.setRegister(8,
char []	curr+ temi.format());
temp=rMgr.getMemory(target, 1);	}
rMgr.setRegister(0,temp);	public void STCH(int
rMgr.setRegister(8,	curr,char[]instruction)
curr+ temi.format());	{
}	Instruction temi=new
public_void RD(int	Instruction(instruction);
curr,char[]instruction) throws	int []flags=new int[6];
IOException	flags=temi.getflag();
{	int add=temi.addr(curr,flags);
Instruction temi=new	char[]temp=new char[1];
Instruction(instruction);	·
int []flags=new int[6];	temp[0]=(char)rMgr.getRegister(0);
flags=temi.getflag();	rMgr.setMemory(add, temp, 1);
int add=temi.addr(curr,flags);	rMgr.setRegister(8,
String	curr+ temi.format());
devname=String.valueOf(rMgr.getMem	}
ory(add,2));	public void STL(int
01 (((((((((((((((((((curr,char[]instruction)
rMgr.setRegister(0,rMgr.readDevice(de	{
vname, 1));	Instruction temi=new
rMgr.setRegister(8,	Instruction(instruction);
curr+ temi.format());	int []flags=new int[6];
teminormat()),	flags=temi.getflag();
public void RSUB(int	int add=temi.addr(curr,flags);
curr,char[]instruction)	char[]temp=new char[1];
curr, char []iiiStr uction/	char[]temp=new char[1],
rMgr.setRegister(8,	temp[0]=(char)rMgr.getRegister(2);
rMgr.getRegister(2));	rMgr.setMemory(add, temp, 1);
I Mgr.gethegrster(2));	
public void STA(int	rMgr.setRegister(8,
	curr+ temi.format());
curr,char[]instruction)	public resid CTV(int
Instruction tomi-news	public void STX(int
Instruction temi=new	curr,char[]instruction)
Instruction(instruction);	Instruction to mi-
int []flags=new int[6];	Instruction temi=new
flags=temi.getflag();	Instruction(instruction);
<pre>int add=temi.addr(curr,flags);</pre>	int []flags=new int[6];
	flags=temi.getflag();
rMgr.setMemory(add,Integer.toHexStri	int add=temi.addr(curr,flags);
ng(rMgr.getRegister(0)).toCharArray(),	char[]temp=new char[1];
1);	
rMgr.setRegister(8,	temp[0]=(char)rMgr.getRegister(1);
curr+ temi.format());	rMgr.setMemory(add, temp, 1);
}	rMgr.setRegister(8,
public void STB(int	curr+ temi.format());
curr,char[]instruction)	}
{	public int TD(int
Instruction temi=new	curr,char[]instruction) throws
Instruction(instruction);	IOException
int []flags=new int[6];	{
flags=temi.getflag();	Instruction temi=new

```
Instruction(instruction);
       int []flags=new int[6];
                                               package sp21 simulator;
       flags=temi.getflag();
       int add=temi.addr(curr,flags);
                                               import java.io.File;
       char[]temp=new_char[2];
                                               import java.io.IOException;
       rMgr.getMemory(add, 2);
                                               * 시뮬레이터로서의 작업을 담당한다.
VisualSimulator에서 사용자의 요청을
res=rMgr.testDevice(String.valueOf(te
mp));
       rMgr.setRegister(8,
                                               받으면 이에 따라 ResourceManager에
                                               접근하여
curr+temi.format());
                                                * 작업을 수행한다.
       return res;
                                               * 작성중의 유의사항 : 1) 새로운
클래스, 새로운 변수, 새로운 함수 선언은
얼마든지 허용됨. 단, 기존의 변수와
함수들을 삭제하거나
    public_void TIXR(int
curr,char[]instruction)
       Instruction temi=new
                                               * 완전히 대체하는 것은 지양할 것. 2)
필요에 따라 예외처리, 인터페이스 또는
상속 사용 또한 허용됨. 3) 모든 void
타입의 리턴값은
Instruction(instruction);
       int []regs=temi.registers();
       rMgr.setRegister(1,
rMgr.getRegister(\bar{1})+1);
                                               * 유저의 필요에 따라 다른 리턴
       int x=rMgr.getRegister(1);
                                               타입으로 변경 가능. 4) 파일, 또는
콘솔창에 한글을 출력시키지 말 것.
(채점상의 이유. 주석에 포함된
* 한글은 상관 없음)
       if(x<rMgr.getRegister(regs[0]))</pre>
              rMgr.setRegister(2,-1);
       else
if(x==rMgr.getRegister(regs[0]))
                                               * + 제공하는 프로그램 구조의
개선방법을 제안하고 싶은 분들은
보고서의 결론 뒷부분에 첨부 바랍니다.
내용에 따라 가산점이 있을 수
              rMgr.setRegister(2, 0);
       else
                                                * 있습니다.
*/
              rMgr.setRegister(2, 1);
       rMgr.setRegister(8,
                                               public class SicSimulator {
curr+temi.format());
                                                      ResourceManager rMgr;
                                                      SicLoader load;
public void WD(int
curr,char[]instruction) throws
                                                      InstLuncher lunch;
                                                      int currloc;
                                                      public
IOException
                                               SicSimulator(ResourceManager
       Instruction temi=new
                                               resourceManager) {
Instruction(instruction);
                                                              // 필요하다면 초기화 과정
       int []flags=new int[6];
                                               추가
       flags=temi.getflag();
                                                              this.rMgr =
       int add=temi.addr(curr,flags);
                                               resourceManager;
       String
                                                              this.lunch=new
devname=String.valueOf(rMgr.getMem
                                               InstLuncher(resourceManager);
ory(add,2));
       char[]temp=new char[1];
                                                      /**
       temp[0]=(char)
                                                       * 레지스터, 메모리 초기화 등
rMgr.getRegister(0);
                                               프로그램 load와 관련된 작업 수행. 단, object code의 메모리 적재 및 해석은
rMgr.writeDevice(devname, temp, 1);
                                                       * SicLoader에서 수행하도록
    }
                                               한다.
                                                        * @throws IOException
```

```
lunch.ADDR(currloc,
      public void load(File program)
                                          rMgr.getMemory(currloc,2));
throws IOException {
             load=new
                                          currloc=rMgr.getRegister(8);//PC레지스
                                          터의 주소로 이동.
SicLoader(rMgr);
             load.load(program);
             currloc=rMgr.sttadd;
                                                       else
                                          if(opc.equals("B4"))//clear
      }
      /**
       * 1개의 instruction이 수행된
                                          lunch.CLEAR(currloc,
모습을 보인다.
                                          rMgr.getMemory(currloc,2));
       * @throws IOException
                                          currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동.
      public void oneStep() throws
IOException {
                                                       else
char[]op=rMgr.getMemory(currloc, 1);
                                          if(opc.equals("2B"))//COMP
             String
opc=String.valueOf(op);
                                                              if(e==true)
             char
[]eflag=rMgr.getMemory(currloc+1,1);
                                          lunch.COMP(currloc,
             String
ef=String.valueOf(eflag);
                                          rMgr.getMemory(currloc,4));
efla=Integer.parseInt(ef.substring(0,1),
                                                              else
16);
             boolean e;
             if((efla\&1)==1)
                                          lunch.COMP(currloc,
                                          rMgr.getMemory(currloc,3));
                    e=true;
                                          currloc=rMgr.getRegister(8);//PC레지스
             else
                                          터의 주소로 이동.
                    e=false;
                                                       else
             if(opc.equals("1B"))//ADD
                                          if(opc.equals("A0"))//COMPR
                    if(e==true)
                                          lunch.COMPR(currloc,
                                          rMgr.getMemory(currloc,2));
lunch.add(currloc,
                                          currloc=rMgr.getRegister(8);//PC레지스
rMgr.getMemory(currloc,4));
                                          터의 주소로 이동.
                    else
                                                       else
                                          if(opc.equals("27"))//DIV
lunch.add(currloc,
rMgr.getMemory(currloc,3));
                                                              if(e==true)
currloc=rMgr.getRegister(8);//PC레지스
                                          lunch.DIV(currloc,
터의 주소로 이동.
                                          rMgr.getMemory(currloc,4));
             else
                                                              else
if(opc.equals(90))//ADDR
                                          lunch.DIV(currloc,
```

```
rMgr.getMemory(currloc,3));
                                                                {
                                           lunch.JLT(currloc,
currloc=rMgr.getRegister(8);//PC레지스
                                           rMgr.getMemory(currloc,4));
터의 주소로 이동
                                                                else
             else
if(opc.equals("9C"))//DIVR
                                           lunch.JLT(currloc,
                                           rMgr.getMemory(currloc,3));
lunch.DIVR(currloc,
rMgr.getMemory(currloc,2));
                                           currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동.
                                                         else
                                           if(opc.equals("4B"))//JSUB
             else
if(opc.equals("3F"))//J
                                                                if(e==true)
                    if(e==true)
                                           lunch.JSUB(currloc,
                                           rMgr.getMemory(currloc,4));
lunch.J(currloc,
rMgr.getMemory(currloc,4));
                                                                else
                    else
                                           lunch.JSUB(currloc,
lunch.J(currloc,
                                           rMgr.getMemory(currloc,3));
rMgr.getMemory(currloc,3));
                                           currloc=rMgr.getRegister(8);//PC레지스
currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
                                           터의 주소로 이동
                                                         else
                                           if(opc.equals("03")||opc.equals("01"))//
             else
if(opc.equals("33"))//JEQ
                                           LDĀ
                    if(e==true)
                                                                if(e==true)
lunch.JEQ(currloc,
                                           lunch.LDA(currloc,
rMgr.getMemory(currloc,4));
                                           rMgr.getMemory(currloc,4));
                    else
                                                                else
lunch.JEQ(currloc,
                                           lunch.LDA(currloc,
rMgr.getMemory(currloc,3));
                                           rMgr.getMemory(currloc,3));
currloc=rMgr.getRegister(8);//PC레지스
                                           currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
                                           터의 주소로
             else
                                                         else
                                           if(opc.equals("53"))//LDCH
if(opc.equals("3B"))//JLT
                    if(e==true)
                                                                if(e==true)
```

```
{
                                           lunch.STA(currloc,
lunch.LDCH(currloc,
                                           rMgr.getMemory(currloc,4));
rMgr.getMemory(currloc,4));
                                                                }
                                                                else
                    else
                                           lunch.STA(currloc,
lunch.LDCH(currloc,
                                           rMgr.getMemory(currloc,3));
rMgr.getMemory(currloc,3));
                                           currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
                                                         else
                                           if(opc.equals("7B"))//STB
if(opc.equals("DB"))//RD
                                                                if(e==true)
                    if(e==true)
                                           lunch.STB(currloc,
lunch.RD(currloc,
rMgr.getMemory(currloc,4));
                                           rMgr.getMemory(currloc,4));
                                                                else
                    else
                                           lunch.STB(currloc,
                                           rMgr.getMemory(currloc,3));
lunch.RD(currloc,
rMgr.getMemory(currloc,3));
                                           currloc=rMgr.getRegister(8);//PC레지스
currloc=rMgr.getRegister(8);//PC레지스
                                           터의 주소로 이동
터의 주소로 이동
                                                         else
                                           if(opc.equals("57"))//STCH {
             else
if(opc.equals("4F"))//RSUB
                                                                if(e==true)
                    if(e==true)
                                           lunch.STCH(currloc,
lunch.RSUB(currloc,
                                           rMgr.getMemory(currloc,4));
rMgr.getMemory(currloc,4));
                                                                else
                    else
                                           lunch.STCH(currloc,
                                           rMgr.getMemory(currloc,3));
lunch.RSUB(currloc,
rMgr.getMemory(currloc,3));
                                           currloc=rMgr.getRegister(8);//PC레지스
currloc=rMgr.getRegister(8);//PC레지스
                                           터의 주소로 이동
터의 주소로 이동
                                           else if(opc.equals("17"))//STL
             else
if(opc.equals("0F"))//STA
                                                                if(e==true)
                    if(e==true)
```

```
lunch.STL(currloc,
                                         currloc=rMgr.getRegister(8);//PC레지스
                                          터의 주소로 이동.
rMgr.getMemory(currloc,4));
                   else
                                                       else
                                         if(opc.equals("DF"))//WD
lunch.STL(currloc,
                                                             if(e==true)
rMgr.getMemory(currloc,3));
                                         lunch.WD(currloc
currloc=rMgr.getRegister(8);//PC레지스
                                         rMgr.getMemory(currloc,4));
터의 주소로 이동
                                                             else
             else
if(opc.equals("13"))//STX
                                         lunch.WD(currloc,
                   if(e==true)
                                         rMgr.getMemory(currloc,3));
lunch.STX(currloc,
                                         currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
rMgr.getMemory(currloc,4));
                   else
                                                }
lunch.STX(currloc,
                                                * 남은 모든 instruction이
모습을 보인다.
rMgr.getMemory(currloc,3));
                                                 * @throws IOException
currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
                                                public void allStep() throws
                                         IOException {
             else
                                                       while(true)
if(opc.equals("E3"))//TD
                   if(e==true)
                                         if(currloc==rMgr.proglength)
                                                                    break;
lunch.TD(currloc,
rMgr.getMemory(currloc,4));
                                                             else
                   else
                                                                    oneStep();
                                                       }
                                                }
lunch.TD(currloc,
rMgr.getMemory(currloc,3));
                                                /**
                                                 * 각 단계를 수행할 때 마다
                                          관련된 기록을 남기도록 한다.
currloc=rMgr.getRegister(8);//PC레지스
터의 주소로 이동
                                                public void addLog(String log)
                                          {
             else
if(opc.equals("B8"))//TIXR
                                                }
                                          }
lunch.TIXR(currloc,
rMgr.getMemory(currloc,2));
                                         package sp21_simulator;
```

```
import java.io.BufferedReader;
                                             구현해서 사용해도 괜찮습니다.
                                                     */
import java.io.File;
import java.io.FileNotFoundException;
                                                    HashMap<String, Integer>
                                             deviceManager = new
import java.io.FileReader;
import java.io.FileWriter;
                                             HashMap<String,Integer>();
                                             char[] memory; // String으로
수정해서 산용하여도 무방함.
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
                                                    int[] register;
                                                    double register_F;
import java.util.HashMap;
                                                    int proglength=0;
/**
                                                    int sttadd;
* ResourceManager는 컴퓨터의 가상
리소스들을 선언하고 관리하는
클래스이다. 크게 네가지의 가상 자원
공간을 선언하고, 이를
                                                    SymbolTable symtabList;
// 이외에도 필요한 변수 선언해서
                                             사용할 것.
 * 관리할 수 있는 함수들을 제공한다.
                                                     * 메모리, 레지스터등 가상
                                             리소스들을 초기화한다.
* 1) 입출력을 위한 외부 장치 또는
device 2) 프로그램 로드 및 실행을 위한
메모리 공간. 여기서는 64KB를
                                                    public void initializeResource()
최대값으로 잡는다.
* 3) 연산을 수행하는데 사용하는 레지스터 공간. 4) SYMTAB 등 simulator의 실행 과정에서 사용되는
                                                           memory= new
                                             char[65536];
                                                           register = new int[10];
데이터들을 위한 변수들.
                                                    /**
                                             * deviceManager가 관리하고
있는 파일 입출력 stream들을 전부
종료시키는 역할. 프로그램을 종료하거나
* 2번은 simulator위에서 실행되는
프로그램을 위한 메모리공간인 반면
4번은 simulator의 실행을 위한 메모리
공간이라는 점에서
* 차이가 있다.
                                             연결을 끊을 때
                                                     * 호출한다.
                                                     * @throws IOException
public class ResourceManager {
       /**
                                                    public void closeDevice()
* 디바이스는 원래 입출력
장치들을 의미 하지만 여기서는 파일로
                                             throws IOException {
                                                           reader.close();
디바이스를 대체한다. 즉, 'F1'이라는
디바이스는 'F1'이라는 이름의
                                                           writer.close();
* 파일을 의미한다.
deviceManager는 디바이스의 이름을
입력받았을 때 해당 이름의 파일 입출력
관리 클래스를 리턴하는 역할을 한다.
* 예를 들어, 'A1'이라는
디바이스에서 파일을 read모드로 열었을
                                                    /**
                                             * 디바이스를 사용할 수 있는
상황인지 체크. TD명령어를 사용했을 때
                                             호출되는 함수. 입출력 stream을 열고
                                             deviceManager를
경우, hashMap에 <"A1", scanner(A1)>
                                                     * 통해 관리시킨다.
        * 넣음으로서 이를 관리할 수
                                                     * @param devName
                                             확인하고자 하는 디바이스의 번호,또는 이름
있다.
        * 변형된 형태로 사용하는 것
                                                     */
역시 허용한다. 예를 들면 key값으로
String대신 Integer를 사용할 수 있다.
                                                    @SuppressWarnings("resource")
파일 입출력을 위해
                                                    public int testDevice(String
* 사용하는 stream 역시
자유로이 선택, 구현한다.
                                             devName) throws IOException {
                                                           FileReader ft=new
                                             FileReader (devName);
        * 이것도 복잡하면 알아서
```

if(ft.ready())

```
* @throws IOException
                                             */
if(deviceManager.get(devName)==null)
                                            FileWriter writer;
                                            public void writeDevice(String
                                      devName, char[] data, int num)
                                      throws IOException {
deviceManager.put(devName, 0);
                        return 1;
                                      if(deviceManager.get(devName)==0)
            }
            else
                                      deviceManager.replace(devName, 1);
                  return 0;
                                                        writer=new
                                      FileWriter(devName);
            ft.close();
            return 1;
                                                  for(int i=0;i < num;i++)
      }
                                      writer.write(data[i]);
       * 디바이스로부터 원하는
개수만큼의 글자를 읽어들인다.
RD명령어를 사용했을 때 호출되는 함수.
                                            /**
                                             * 메모리의 특정 위치에서
       * @param devName
디바이스의 이름
                                      원하는 개수만큼의 글자를 가져온다.
                         가져오는
       * @param num
글자의 개수
                                             * @param location 메모리 접근
       * @return 가져온 데이터
                                      위치 인덱스
       * @throws IOException
                                                                데이터
                                             * @param num
                                      개수
      FileReader reader;
                                             * @return 가져오는 데이터
      public char[] readDevice(String
devName, int num) throws
                                            public char[] getMemory(float
                                      location, int num) {
IOException {
                                                  int loc=(int)location*2;
                                                  char[]retch=new
if(deviceManager.get(devName)==0)
                                      char[num*2];
deviceManager.replace(devName, 1);
                                      retch=Arrays.copyOfRange(memory,
                  reader=new
                                      loc, loc+(num*2));
FileReader(devName);
                                                  return retch;
                                            }
            char[]temp=new
char[num];
            reader.read(temp,0,num);
                                             * 메모리의 특정 위치에 원하는
            return temp;
      }
                                      개수만큼의 데이터를 저장하다.
      /**
                                             * @param locate 접근 위치
      * 디바이스로 원하는 개수
                                      인덱스
만큼의 글자를 출력한다. WD명령어를
사용했을 때 호출되는 함수.
                                             * @param data
                                                              저장하려는
                                      데이터
                                             * @param num
                                                              저장하는
                                      데이터의 개수
       * @param devName
디바이스의 이름
       * @param data
                        보내는
                                      public void setMemory(int
locate, char[] data, int num) {
데이터
                         보내는
                                                  int loc=locate*2;
       * @param num
글자의 개수
                                                  for(int
```

```
i=loc;i<loc+(num*2);i++)
                                             */
                                            public void setRegister(int
                                      regNum, int value) {
memory[i]=data[i-loc];
                                                  register[regNum]=value;
                                      public void setRegister(int
regNum,char[]value) {
      }
      public void setMemory(float
                                      register[regNum]=(int)value[0];
locate,char[]data,int num) {
            int loc=(int)locate*2;
            for(int
i=loc;i<loc+(num*2);i++)
                                            /**
                                             * 주로 레지스터와 메모리간의
                                      데이터 교환에서 사용된다. int값을
memory[i]=data[i-loc];
                                      char[]형태로 변경한다.
                                             * @param data
                                             * @return
      /**
* 번호에 해당하는 레지스터가
현재 들고 있는 값을 리턴한다.
레지스터가 들고 있는 값은 문자열이
                                            public char[] intToChar(int
                                      data) {
아님에 주의하다.
                                      char[]temp=Integer.toHexString(data).t
                                      oCharArray();
       * @param regNum 레지스터
                                                  return temp;
분류번호
       * @return 레지스터가 소지한
값
                                             * 주로 레지스터와 메모리간의
                                      데이터 교환에서 사용된다. char[]값을
      public int getRegister(int
                                      int형태로 변경한다.
regNum) {
            return register[regNum];
                                             * @param data
      }
                                             * @return
      /**
                                            public int byteToInt(char[]
       * 번호에 해당하는 레지스터에
                                      data) {
새로운 값을 입력한다. 레지스터가 들고
                                                   String
있는 값은 문자열이 아님에 주의한다.
                                      temp=String.valueOf(data);
                                                  return
       * @param regNum 레지스터의
                                      Integer.parseInt(temp, 16);
      * @param value 레지스터에
집어넣는 값
```