과목 명: 시스템프로그래밍

담당 교수 명: 김 지 환

<<Assignment 3>>

**서강대학교 컴퓨터공학과**

**[학번 : 20141602]**

**[이름 : 황기덕]**

목 차

1. 프로그램 개요 3

2. 프로그램 설명 3

2.1 프로그램 흐름도 3

2.1.1 loader 4

2.1.2 run 5

3. 모듈 정의

3.1 모듈 이름 : int cmd\_bp(int); 5

3.1.1 기능 5

3.1.2 사용 변수 5

3.2 모듈 이름: void cmd\_bpclear(); 6

3.2.1 기능 6

3.2.2 사용 변수 6

3.3 모듈이름: void cmd\_bpprint(); 6

3.3.1 기능 6

3.3.2 사용변수 6

3.4 모듈이름: int cmd\_loader(char\*, char\*, char\*); 6

3.4.1 기능 6

3.4.2 사용변수 6

3.5 모듈이름: int hextodex(char\*, int); 7

3.5.1 기능 7

3.5.2 사용변수 7

3.6 모듈이름: int findetab(char\*, int, int\*); 7

3.6.1 기능 7

3.6.2 사용변수 7

3.7 모듈이름: void error\_loader(FILE\*,FILE\*,FILE\*,int, char\*); 8

3.7.1 기능 8

3.7.2 사용변수 8

3.8 모듈이름: void cmd\_run(); 8

3.8.1 기능 8

3.8.2 사용변수 8

3.9 모듈이름: void disassemble(int,char,int,int,unsigned int\*, unsigned int\*); 8

3.9.1 기능 8

3.9.2 사용변수 8

4. 전역 변수 정의 9

4.1 int program\_address; 9

4.2 int CSADDR; 9

4.3 breakpoint \*first\_breakpoint, \*cur\_bp; 9

4.4 ESTAB \*etable[3]; 9

4.5 int run\_start\_addr; 9

4.6 int run\_cur\_addr; 9

4.7 int prog\_end\_addr; 9

4.8 int stop\_prog\_addr 9

4.9 unsigned int reg[16]; 9

4.10 int prog\_run\_flag 9

# 프로그램 개요

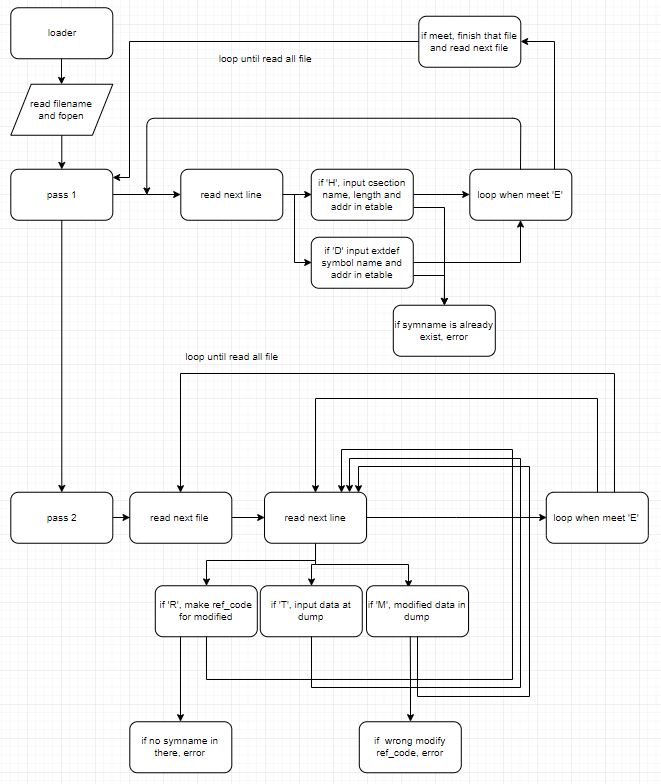
프로젝트 #2에서 구현한 쉘에 linking loader, run, bp, progaddr등의 기능을 추가한다.

SIC/XE의 assembly program source 파일을 입력 받아서 생성된 object file들을 linking 시킨 후 메모리에 load 시키며, 그 후 실제로 프로그램을 실행시켜 본다.

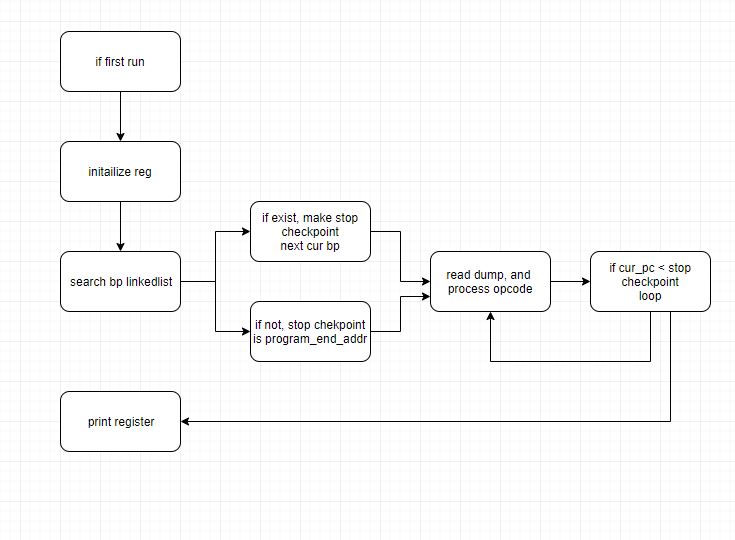
# 프로그램 설명

## 2. 1 프로그램 흐름도

2. 1. 1 loader



2.1.2 run



# 모듈 정의

## int cmd\_bp(int);

## 기능

command bp 의 기능을 수행해준다

3.1.2 사용 변수

### breakpoint \*tmp, \*step, \*bstep; // linked list 연결을 위한 변수

## void cmd\_bpclear();

## 기능

## bp의 내용을 지워준다

## 사용 변수

### breakpoint \*step, \*bstep; // bp 삭제를 위한 step 변수

### 

## void cmd\_bpprint();

## 기능

## bp의 내용들을 print해준다

## 사용 변수

breakpoint \*step; // bp 출력을 위한 변수

### 

## int cmd\_loader(char\*, char\*, char\*);

## 기능

## command loader의 기능을 수행해준다. linking, loader 기능을 수행해준다

## 사용 변수

### int all\_filenum = 0; // 몇 개의 file이 들어왔는가

### int csaddr, execsaddr; //csaddr과 execsaddr 저장

### int i, k, l; //for loop

### int cslength; //section 의 길이 저장

### int returnaddr; // return addr 저장

### int T\_start\_addr; //T 줄에서, 시작 addr

### int T\_len; //T 줄에서, 그 줄의 길이

### int M\_addr, M\_mod, M\_div; //M 줄에서, 그 addr과 modi 수행시 필요한 변수

### unsigned int tmp\_dump; // dump에 저장시 필요

### int total\_len = 0; //program의 총 길이

### int \*ref\_code; //ref code, modi 수행시 필요

### FILE \*fp[3]; //file 포인터

### char input\_string[100]; // file에서 입력받기 위하여

### char tmp\_symname[10]; //symbol name 저장

### char tmp\_addr[10]; //tmp addr 저장

### ESTAB \*etmp, \*estep; //etable을 사용하기 위하여

### 

## int hextodec(char\*, int);

## 기능

## 16진수가 저장되어있는 string을 받아 10진수로 return해준다

## 사용 변수

int l; // loop 변수

int tmpnum = 0; // 임시저장 후, 이 값을 돌려준다.

## int findetab(char\*, int, int\*);

## 기능

## etable을 탐색해 같은 것이 있나 확인해준다

## 사용 변수

### int i; //loop 변수

### ESTAB \*step; //원하는 값을 찾기 위한 step

### 

## void error\_loader(FILE\*, FILE\*, FILE\*, int, char\*);

## 기능

## loader 중 에러가 났을 시 연 파일을 닫고 에러 메세지를 출력해준다

## 사용 변수

### 없음

## void cmd\_run();

## 기능

## command run 기능을 수행해준다

## 사용 변수

### unsigned char tmp\_opcode;

### int ni; // for ni

### int i; //for loop

### int cur\_obj\_code; // cur code

### unsigned int address, value; // address와 value(첫번째, 두번째 parameter 저장)

### unsigned char a,b; //for divide objcode

### char opsym[10]; //for op format

### int formatnum; //format 형식 저장(1,2,3,4)

### int bpflag = 0; //bp가 있는지 flag

### int cc = -2; // for current stat. sw

### 

## void disassemble(int, char, int, int, unsigned int\*, unsigned int\*);

## 기능

## obj code를 disassemble하여 원하는 주소를 되돌려준다.

## 사용 변수

### unsigned int target\_address = 0; // target address 저장

### int disp = 0, forni2 = 0; // disp와 포멧 2를 위한 disp

### char xbpe; // xbpe 부분을 저장하기 위한 변수

# 전역 변수 정의

# int program\_address

시작 program address를 저장한다

# int CSADDR

loader 중 현재 addr을 저장한다.

# breakpoint \*first\_breakpoint, \*cur\_bp

bp의 시작 주소와 현재 bp 포인터를 가르킨다.(run 이용시)

# ESTAB \*etable[3]

loader 수행 중, 외부 symbol table 을 저장한다.

# int run\_start\_addr

run에서 시작 prog addr을 저장한다.

# int run\_cur\_addr

run에서 현재 addr을 저장한다(== pc)

# int prog\_end\_addr

run에서 끝 addr을 저장한다

# int stop\_prog\_addr

run에서 중지할 addr을 저장한다(bp)

# unsigned int reg[16]

각 번호에 맞는 reg를 저장한다. ex) reg[0] == a register...

# int prog\_run\_flag

이 program이 처음 run 했는지 안했는지의 정보를 저장한다.