

시스템 프로그래밍 과제 #6

* 과제내용

- SIC/XE 어셈블러를 개발하기 전에, 프로그램을 입력받을 수 있는 파싱 프로그램 작성
- 명령어에 해당하는 OPCODE를 찾아서 해당 명령어 옆에 출력
- 과제에 주어진 C 인터페이스 라이브러리를 사용

* 과제 목적

- 소스코드를 입력으로 받아 파싱하고 OPCODE를 매핑시킴으로써, 기본적인 SIC/XE 머신을 이해한다.

* 과제 제출 마감 - 4월 9일(일) 오전 11:59까지 스마트캠퍼스에 제출

(제출시간 이후 제출 시 2 point (10%) 패널티 부과,
이후부터는 매일 2 point씩 (10%) 추가 패널티를 부과함)

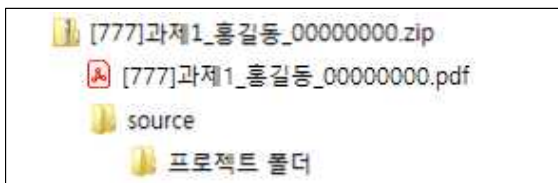
* 제출물 - 레포트 파일(PDF) + 프로그램 소스코드

* 제출 레포트 (100 point)

- 요구사항 : 보고서에 아래 내용을 포함하여 보고서를 작성할 것
(학번, 이름, 출석번호, 과제명, 수업 구분<가,나>)
 1. 동기/목적
 2. 설계/구현 아이디어
 3. 수행결과
 4. 결론 및 보충할 점
 5. 소스코드(+주석)
- 소스코드는 2단으로 출력할 것 (별도로 아래와 같이 파일로도 스마트캠퍼스에 제출)
- 점수 평가에서 레포트의 비중이 높으므로 제출 마감 전까지 성심껏 작성하기 바랍니다.

* 제출 파일양식 - [출석번호]과제6_이름_000.zip

- 레포트 파일은 PDF로 한정
- 소스코드는 프로젝트 폴더를 그대로 첨부
 - 비주얼 스튜디오의 경우 .sln 파일 반드시 첨부
- 소스코드의 “000” 은 자신의 출석번호로 교체



(제출 파일 구성 예시)

* 제출 파일양식을 지키지 않을 시 미제출로 간주

* 제출 파일은 smart-campus 과제게시판에 올릴 것

* 기간 내 레포트 및 파일 미제출 시 Late Penalty 부여

* 프로그램 Input/Output은 표시된 Input/Output 문서를 기준으로 함

*** 프로그램 구현에 사용해야할 인터페이스 내용**

- 매핑을 위한 OPCODE 테이블은 Appendix 참고
- 아래에 주어진 명세를 참고하여 과제 코드를 구현할 것

파일명 : my_assembler.h

내 용 : my_assembler.c를 위한 변수선언과 테이블 관리를 위한 구조체 정의

1. input 프로그램 관리를 위해 기술되는 구조체 정보

```
// 어셈블리 할 소스코드를 파일로부터 불러와 라인별로 관리하는 테이블 생성
char* input[MAX_INPUT_LINES];
int input_length;

// 어셈블리 할 라인 별 소스코드를 토큰 단위로 관리하기 위한 구조체 변수
typedef struct _token {
    char* label;                //명령어 라인 중 label
    char* operator;             //명령어 라인 중 operator
    char* operand[MAX_OPERAND_PER_INST]; //명령어 라인 중 operand
    char* comment;              //명령어 라인 중 comment
} token;

// 어셈블리 할 소스코드를 5000라인까지 관리하는 테이블 생성
token *tokens[MAX_INPUT_LINES];
```

2. Instruction 테이블 관리를 위한 구조체 선언

```
// Instruction set을 바탕으로 최대 256개의 instruction을 관리하는 테이블 생성
// Instruction을 관리하기 위한 구조체 변수

typedef struct _inst {
    char str[10];                //instruction의 이름
    unsigned char op;            //명령어의 OPCODE
    int format;                  //instruction의 형식
    int ops;                      //instruction의 operand 개수
} inst;

// Instruction의 정보를 가진 구조체를 관리하는 테이블 생성
inst *inst_table[MAX_INST_TABLE_LENGTH];
int inst_table_length;
```

- my_assembler.c 파일에 이미 명시되어있는 함수들을 구현하고, 추가적으로 필요한 함수 구현과 변수 생성은 자유. **명시된 함수 정의 및 선언부 수정하지 말 것!! 수정 시 감점** (기본 함수는 모두 사용해야 함, 추가적인 함수 구현 시 리포트에 목적 명시)

* 프로그램 수행에 따른 입력과 출력은 다음과 같아야 한다.

Input

COPY	START	0	COPY FILE FROM IN TO OUTPUT
	EXTDEF	BUFFER,BUFEND,LENGTH	
	EXTREF	RDREC,WRREC	
FIRST	STL	RETADR	SAVE RETURN ADDRESS
CLOOP	+JSUB	RDREC	READ INPUT RECORD
	LDA	LENGTH	TEST FOR EOF (LENGTH = 0)
	COMP	#0	
	JEQ	ENDFIL	EXIT IF EOF FOUND
	+JSUB	WRREC	WRITE OUTPUT RECORD
	J	CLOOP	LOOP
ENDFIL	LDA	=C'EOF'	INSERT END OF FILE MARKER
	STA	BUFFER	
	LDA	#3	SET LENGTH = 3
	STA	LENGTH	
	+JSUB	WRREC	WRITE EOF
	J	@RETADR	RETURN TO CALLER
RETADR	RESW	1	
LENGTH	RESW	1	LENGTH OF RECORD
	LTORG		
BUFFER	RESB	4096	4096-BYTE BUFFER AREA
BUFEND	EQU	*	
MAXLEN	EQU	BUFEND-BUFFER	MAXIMUM RECORD LENGTH
RDREC	CSECT		
.			
.			SUBROUTINE TO READ RECORD INTO BUFFER
.			
	EXTREF	BUFFER,LENGTH,BUFEND	
	CLEAR	X	CLEAR LOOP COUNTER
	CLEAR	A	CLEAR A TO ZERO
	CLEAR	S	CLEAR S TO ZERO
	LDT	MAXLEN	
RLOOP	TD	INPUT	TEST INPUT DEVICE
	JEQ	RLOOP	LOOP UNTIL READY
	RD	INPUT	READ CHARACTER INTO REGISTER A
	COMPR	A,S	TEST FOR END OF RECORD (X'00')
	JEQ	EXIT	EXIT LOOP IF EOR
	+STCH	BUFFER,X	STORE CHARACTER IN BUFFER
	TIXR	T	LOOP UNLESS MAX LENGTH
	JLT	RLOOP	HAS BEEN REACHED
EXIT	+STX	LENGTH	SAVE RECORD LENGTH
	RSUB		RETURN TO CALLER
INPUT	BYTE	X'F1'	CODE FOR INPUT DEVICE
MAXLEN	WORD	BUFEND-BUFFER	
WRREC	CSECT		
.			
.			SUBROUTINE TO WRITE RECORD FROM BUFFER
.			
	EXTREF	LENGTH,BUFFER	

	CLEAR	X	CLEAR LOOP COUNTER
	+LDT	LENGTH	
WLOOP	TD	=X'05'	TEST OUTPUT DEVICE
	JEQ	WLOOP	LOOP UNTIL READY
	+LDCH	BUFFER,X	GET CHARACTER FROM BUFFER
	WD	=X'05'	WRITE CHARACTER
	TIXR	T	LOOP UNTIL ALL CHARACTERS
	JLT	WLOOP	HAVE BEEN WRITTEN
	RSUB		RETURN TO CALLER
	END	FIRST	

Output

COPY	START		0
	EXTDEF	BUFFER,BUFEND,LENGTH	
	EXTREF	RDREC,WRREC	
FIRST	STL	RETADR	14
CLOOP	+JSUB	RDREC	48
	LDA	LENGTH	00
	COMP	#0	28
	JEQ	ENDFIL	30
	+JSUB	WRREC	48
	J	CLOOP	3C
ENDFIL	LDA	=C'EOF'	00
	STA	BUFFER	0C
	LDA	#3	00
	STA	LENGTH	0C
	+JSUB	WRREC	48
	J	@RETADR	3C
RETADR	RESW	1	
LENGTH	RESW	1	
		LTORG	
BUFFER	RESB	4096	
BUFEND	EQU	*	
MAXLEN	EQU	BUFEND-BUFFER	
RDREC	CSECT		
.			
.		SUBROUTINE TO READ RECORD INTO BUFFER	
.			
	EXTREF	BUFFER,LENGTH,BUFEND	
	CLEAR	X	B4
	CLEAR	A	B4
	CLEAR	S	B4
	LDT	MAXLEN	74
RLOOP	TD	INPUT	E0

	JEQ	RLOOP	30
	RD	INPUT	D8
	COMPR	A,S	A0
	JEQ	EXIT	30
	+STCH	BUFFER,X	54
	TIXR	T	B8
	JLT	RLOOP	38
EXIT	+STX	LENGTH	10
	RSUB		4C
INPUT	BYTE	X'F1'	
MAXLEN	WORD	BUFEND-BUFFER	
WRREC	CSECT		
.			
.		SUBROUTINE TO WRITE RECORD FROM BUFFER	
.			
	EXTREF	LENGTH,BUFFER	
	CLEAR	X	B4
	+LDT	LENGTH	74
WLOOP	TD	=X'05'	E0
	JEQ	WLOOP	30
	+LDCH	BUFFER,X	50
	WD	=X'05'	DC
	TIXR	T	B8
	JLT	WLOOP	38
	RSUB		4C
	END	FIRST	

* ‘.’ 으로 시작하는 주석 Line은 output에 출력하지 않아도 된다.(선택사항)