

시스템 프로그래밍 2024 Project #1b

* 과제내용

- ControlSection 방식의 SIC/XE 소스를 Object Program Code로 바꾸는 어셈블러 개발
- 과제에 주어진 자바 프로젝트 파일 사용

* 과제 목적

- SIC/XE 소스를 Object Program code 로 변환하여 봄으로써 SIC/XE 어셈블러의 동작을 이해한다.
- 주어진 자바파일을 이용하여 SIC/XE 소스를 Object Program Code로 변환하는 과정을 복습하고 C와 비교한다.

* 과제 제출 마감 - 5월 13일(월) 오전 11:59까지 스마트 캠퍼스 과제란에 제출

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

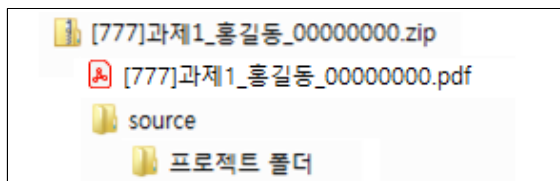
* 제출물 - 레포트 파일(PDF) + 프로그램 소스코드(.java 파일) + 입력 파일(input.txt 및 inst_table.txt 파일) + 출력 결과 파일(output_syntab.txt, output_littab.txt, output_objectcode.txt 파일)

* 제출 레포트 (50 point)

- 요구사항 : 표지 반드시 넣을 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 제출)
- 점수 평가에서 레포트의 비중이 높으므로 제출 마감 전까지 성심껏 작성하기 바랍니다.

* 제출 파일양식 (50 point) - **[출석번호]프로젝트1b_이름_00000000.zip**

- 레포트 파일은 PDF로 한정
- 소스코드는 프로젝트 폴더를 그대로 첨부
 - 프로젝트 폴더 내에 소스코드, 입력 파일, 출력 결과 파일이 존재해야 함
- 소스코드의 "00000000"은 자신의 학번으로 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 구현 중 유의사항**

- 주어진 java 파일을 참고하여 과제 코드를 구현할 것
- ControlSection.java 파일에서 각 control section에 해당하는 소스 코드가 독립적으로 컴파일됨. 독립적인 컴파일을 보장하기 위하여, Assembler.java에서 각 control section 소스 코드에 END directive를 추가하였음
- Exception Handling을 통해 SIC/XE 소스 코드의 문법적 오류를 탐지할 수 있도록 구현할 것. 최소한 exception handling을 통해 정의되지 않은 심볼 사용 오류를 감지하여야 하며, 다른 오류 감지를 추가적으로 수행할 경우 가산점이 주어질 수 있음
- EXTREF에서 선언된 심볼 또한 심볼 테이블에서 출력할 것
- Assembler.java 파일은 변경하지 말 것
- Assembler.java를 제외한 파일은 필드와 메소드를 삭제 혹은 대체하지 않는 선에서 변경하여도 됨. 기존 작성된 코드는 가이드라인을 제공하기 위해 작성된 것이기에, 해당 형식을 따르는 것을 권장함.
- 제공된 코드의 개선점을 제안하고 싶은 경우, 보고서의 결론 뒷부분에 첨부 바람. 내용에 따라 가산점이 주어질 수 있음

* 프로그램 수행에 따른 입력과 출력은 다음과 같아야 한다.

input.txt (개행 문자는 CRLF 혹은 LF 중 하나로 통일할 것)

```

COPY  START      0          COPY FILE FROM IN TO OUTPUT
      EXTDEF     BUFFER,BUFEND,LENGTH
      EXTREF     RREC,WRREC
FIRST STL      RETADR    SAVE RETURN ADDRESS
CLOOP +JSUB     RREC      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
RREC 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
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
END            FIRST

```

output_syntab.txt (패스 1 종료 후 출력. 포맷은 중요하지 않음)

- 3열의 COPY, RDREC, WRREC은 1열의 심볼이 해당 심볼의 상대 주소임을 의미함.
- 2열의 REF는 1열의 심볼이 EXTREF로 생성된 심볼임을 의미함.

| | | |
|--------|--------|-------|
| COPY | 0x0000 | |
| FIRST | 0x0000 | COPY |
| LOOP | 0x0003 | COPY |
| ENDFIL | 0x0017 | COPY |
| RETADR | 0x002A | COPY |
| LENGTH | 0x002D | COPY |
| BUFFER | 0x0033 | COPY |
| BUFEND | 0x1033 | COPY |
| MAXLEN | 0x1000 | |
| RDREC | REF | |
| WRREC | REF | |
| RDREC | 0x0000 | |
| RL00P | 0x0009 | RDREC |
| EXIT | 0x0020 | RDREC |
| INPUT | 0x0027 | RDREC |
| MAXLEN | 0x0028 | RDREC |
| BUFFER | REF | |
| LENGTH | REF | |
| BUFEND | REF | |
| WRREC | 0x0000 | |
| WLOOP | 0x0006 | WRREC |
| LENGTH | REF | |
| BUFFER | REF | |

output_littab.txt (패스 1 종료 후 출력. 포맷은 중요하지 않음)

| | |
|---------|----|
| =C'EOF' | 30 |
| =X'05' | 1B |

output_objectcode.txt (패스 2 종료 후 출력. 출력 결과는 아래 예시와 반드시 동일해야 함. 각 라인 뒤쪽의 공백 문자, 개행 문자의 차이, 각 control section 사이의 개행 문자 개수 차이는 허용함)

| | |
|---|---------------------------------|
| HCOPY | 000000001033 |
| DBUFFER | 0000033BUFEND001033LENGTH00002D |
| RRDREC | WRREC |
| T0000001D1720274B1000000320232900003320074B1000003F2FEC0320160F2016 | |
| T000010D0D0100030F200A4B1000003E2000 | |
| T00003003454F46 | |
| M00000405+RDREC | |
| M00001105+WRREC | |
| M00002405+WRREC | |
| E000000 | |
| HRDREC | 00000000002B |
| RBUFFERLENGTHBUFEND | |
| T0000001DB410B400B44077201FE3201B332FFADB2015A00433200957900000B850 | |
| T000010D0E3B2FE9131000004F0000F1000000 | |
| M00001805+BUFFER | |
| M00002105+LENGTH | |
| M00002806+BUFEND | |
| M00002806-BUFFER | |
| E | |
| HWRREC | 00000000001C |
| RLLENGTHBUFFER | |
| T0000001CB41077100000E32012332FFA53900000DF2008B8503B2FEE4F000005 | |
| M00000305+LENGTH | |
| M00000D05+BUFFER | |
| E | |

input2_error.txt (9번째 줄이 변경되었음)

```

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   WAAAA     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

```

실행 결과 (stdout 출력 결과. 출력 문구는 동일하지 않아도 됨. Exception 처리를 수행했다면 충분함)

Error : missing symbol definition

(no label) + JSUB WAAAA WRITE OUTPUT RECORD