

시스템 프로그래밍 프로젝트 2

1. 프로젝트 문제 및 목표

프로젝트1에서 구현한 셸(shell)에 assemble 기능을 추가하는 프로그램. SIC/XE의 assembly program source 파일을 입력 받아서 object파일을 생성하고, 어셈블리 과정 중 생성된 symbol table과 결과물인 object 파일을 볼 수 있는 기능을 제공해야 함. 교재의 2.2까지 설명된 SIC/XE 어셈블러의 기능을 구현함을 원칙으로 한다.

2. 요구사항

2.1 프로젝트 목표 설정

이미 제출한 프로젝트1에 아래의 기능들을 추가해야 합니다.

- ① Shell 관련 명령어들
- ② assembler (assemble)
- ③ assemble 관련 명령어 (symbol)

2.2 합성

프로젝트1에서 구현한 셸(shell)에 assemble 기능을 추가하는 프로그램을 작성하는 프로젝트로, SIC/XE machine의 assembly program source 파일을 입력 받아서 object파일을 생성하고, 어셈블리 과정 중 생성된 symbol table과 결과물인 object 파일을 볼 수 있는 기능을 제공해야 한다. 이와 같은 기능을 제공하는 프로그램을 작성하기 위해 필요한 자료구조와 알고리즘을 구상하여 전체적인 프로그램을 설계한다.

2.3 제작

- 1) Shell 관련 명령어

① help

아래와 같이 Shell에서 실행 가능한 모든 명령어들의 리스트를 화면에 출력해준다.

```
sicsim> help
h[elp]
d[ir]
q[uit]
hi[story]
du[mp] [start, end]
e[dit] address, value
f[ill] start, end, value
reset
opcode mnemonic
opodelist
assemble filename
type filename
symbol
```

② type filename

filename에 해당하는 파일을 현재 디렉터리에서 읽어서 화면에 출력한다.

현재 디렉터리에 해당 파일이 존재하지 않으면 에러 메시지를 출력한다.

filename이 디렉토리인 경우는 고려하지 않는다.

시스템 콜을 사용하지 않는다.

```
sicsim> type copy.obj
HCOPY 000000001077
T00000001D17202D69202D4B1010360320262900003320074B10105D3F2FEC032010
T00001D130F20160100030F200D4B10105D3E2003454F46
T0010361DB410B400B44075101000E32019332FFADB2013A00433200857C003B850
T0010531D3B2FEA1340004F0000F1B410774000E32011332FFA53C003DF2008B850
T001070073B2FEF4F000005
```

2) SIC/XE 어셈블러 명령

① assemble filename

filename에 해당하는 소스 파일을 읽어서 object파일과 리스팅 파일을 만든다.

소스 파일의 확장자는 .asm이다.

리스팅 파일의 파일명은 소스파일과 동일하고 확장자는 .lst이다

object 파일의 파일명은 소스 파일과 동일하고 확장자는 .obj이다

lst 파일과 .obj 파일은 출력형식을 반드시 지킬 것

소스파일에 에러가 존재할 경우, 리스팅 파일과 object파일을 생성하지 않고 에러 내용을 화면에 출력합니다. 에러 발생시 바로 명령을 종료한다.

에러의 내용은 디버깅을 위해 어떤 라인에서 에러가 발생했는지 출력한다.

라인번호는 5의 배수 단위로 지정하여 출력한다.

각 문자열의 길이는 30자 이내로 가정한다.

label은 영문, 숫자를 input으로 가정하며, 그 외 특수문자는 고려하지 않는다.

```
sicsim> assemble 2_5.asm
Successfully assemble 2_5.asm.
sicsim> type 2_5.obj
HCOPY 000000001077
T0000001D17202D69202D4B1010360320262900003320074B10105D3F2FEC032010
T00001D130F20160100030F200D4B10105D3E2003454F46
T0010361DB410B400B44075101000E32019332FFADB2013A00433200857C003B850
T0010531D3B2FEA1340004F0000F1B410774000E32011332FFA53C003DF2008B850
T001070073B2FEF4F000005
M00000705
M00001405
M00002705
E0000000
```

```
sicsim> type 2_5.lst
5      0000      COPY      START      0
10     0000      FIRST    STL      RETADR      17202D
15     0003              LDB      #LENGTH      69202D
20              BASE      LENGTH
25     0006      CLOOP    +JSUB    RDREC      4B101036
30     000A              LDA      LENGTH      032026
35     000D              COMP     #0      290000
40     0010              JEQ      ENDFIL      332007
45     0013              +JSUB    WRREC      4B10105D
50     0017              J        CLOOP      3F2FEC
55     001A      ENDFIL    LDA      EOF      032010
60     001D              STA      BUFFER      0F2016
65     0020              LDA      #3      010003
70     0023              STA      LENGTH      0F200D
75     0026              +JSUB    WRREC      4B10105D
80     002A              J        @RETADR      3E2003
85     002D      EOF      BYTE     C 'EOF '      454F46
```

(생략)

```
210    105D      WRREC    CLEAR    X      B410
215    105F              LDT      LENGTH      774000
220    1062      WLOOP    TD        OUTPUT      E32011
225    1065              JEQ      WLOOP      332FFA
230    1068              LDCH     BUFFER, X      53C003
235    106B              WD        OUTPUT      DF2008
240    106E              TIXR      T      B850
245    1070              JLT      WLOOP      3B2FEF
250    1073              RSUB      4F0000
255    1076      OUTPUT  BYTE     X '05 '      05
260              END      FIRST
```

② symbol

assemble 과정 중에 생성된 symbol table을 화면에 출력합니다. Symbol table은 각자 설정을 하고, 출력은 아래와 같이 한다.

가장 최근에 assemble 한 파일의 symbol table을 출력합니다.

symbol의 출력은 symbol을 기준으로 알파벳 내림차순으로 정렬이 되어야 합니다.

출력형식을 반드시 지킬 것

```
sicsim> symbol
      BUFFER  0036
      CLOOP   0006
      ENDFIL   001A
      EOF      002D
      EXIT     1056
      FIRST    0000
      INPUT    105C
      LENGTH   0033
      OUTPUT   1076
      RDREC    1036
      RETADR    0030
      RLOOP    1040
      WLOOP    1062
      WRREC    105D
```

2.4 시험

교재의 2.2까지 설명된 SIC/XE 어셈블러의 기능을 구현함을 원칙으로 한다. 원래의 SIC/XE machine은 standard machine에 하위 호환 되어야 하지만 이번 SIC/XE 어셈블러에서는 체크하지 않는다. Compile 해야 되는 기본 소스파일은 p55 Figure 2.5을 사용한다.

3. 환경

반드시 gcc만을 이용해서 C언어로 프로그램 한다. 특히 C언어가 아닌 C++ 등 다른 언어를 사용하거나, 도스 및 윈도우에서만 동작 가능한 경우 0점 처리한다.

컴파일 시, make 파일에 gcc -Wall 옵션을 사용하여 warning 을 철저히 확인한다. (warning 발생시 감점)

4. 수행 기간

4월 8일(수) 00:00 ~ 4월 21일(화) 23:59시까지 제출

5. 제출물

- 1) 프로그램 소스 및 헤더파일
 - ① main 함수가 포함된 파일명을 학번.c 으로 고정.
 - ② 최소 1개 이상의 헤더파일을 반드시 작성.
- 2) Makefile
 - ① Makefile이 정의가 안되어 있을시 프로젝트 0점
 - ② 반드시 clean 옵션을 구현. 없을 시 감점
 - ③ 실행파일은 학번.out 으로 고정
- 3) 프로그램 리포트: 이번에는 XE 소스 assemble이 주된 기능이니 만큼 이에 대한 프로그램 흐름이나 알고리즘 설명 (어떻게 구현하였는지)을 꼭 넣어주시기 바랍니다.
- 4) 프로그램의 컴파일 방법 및 실행방법에 대한 간단한 내용을 적은 README파일
- 5) 기타 수행에 필요한 파일 (ex) opcode.txt, 2_5.asm 등)

6. 제출 방법

sp학번_proj2 이름의 디렉토리를 만들고, 이 디렉토리에 5에서 언급한 제출물을 넣어서 디렉토리를 tar로 압축하여 한 파일로 만든 후 과제란에 과제를 제출 해주시기 바랍니다.

tar 파일로 압축할 때 지난 프로젝트와 동일하게 -z옵션을 사용하지 않습니다.

제출주소: 사이버캠퍼스 과제란

파일형식: sp학번_proj2.tar (ex. sp20181234_proj2.tar)

7. 주의사항

제출할 파일이 잘 작성되었는지 확인하고 보내시기 바랍니다.

압축파일 형식이 .tar가 아닐 경우 10% 감점

바이너리 파일 제출시 10% 감점

제출형식(메일제목, file 이름 형식, 내용물)이 잘못되었을 시, 감점 10%

Late는 받지 않습니다.

8. 채점 관련

- 1) Compile error로 실행이 불가능한 경우 0점
- 2) Segmentation fault로 인한 프로그램 실행 불가 시 0점
- 3) Segmentation fault로 인한 명령 수행 불가 시 해당 부분 0점
- 4) 주석이 없거나 타인이 알아볼 수 없는 경우 감점
- 5) Copy 시 0점

본 프로젝트는 완벽한 어셈블러를 구현하는 것이 아닙니다. 기본 테스트 케이스(2_5.asm)과 기본 테스트 케이스를 조금 변형한 테스트 케이스를 사용할 것이기 때문에 예외처리가 아닌 기능 구현에 중점을 두시기 바랍니다.