

# 시스템프로그래밍(F049-1)

## 과제1.5 레포트

Assembler & Linker Practice

아주대학교 소프트웨어학과

202220775

박민정

제출일: 2025.04.17.

## 가)프로그램 구조 및 동작설명

QUIZ.asm은 입력 문자열을 읽어서 외부에서 정의된 함수를 호출하고, 변형된 결과를 출력하는 메인 흐름을 담당한다.

먼저 +JSUB INIT 명령어를 통해 LIBS.asm 파일에 정의된 INIT 함수를 호출하여, 문자 저장에 사용할 포인터 PTR을 초기화한다. 다음 LDX =X'000000' 명령어는 X 레지스터를 0으로 설정하는 명령으로, 이후 반복문에서 문자열의 인덱스 역할을 하게 된다.

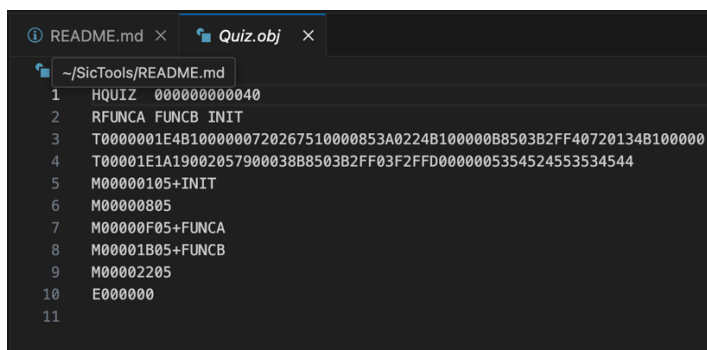
ILOOP 루프에서는 문자열 STRIN의 각 문자를 LDCH로 하나씩 읽어오고, 이 문자를 +JSUB FUNCA를 통해 버퍼에 저장한다. 포인터는 문자가 하나 저장될 때마다 1씩 증가하며, 이 과정을 문자열의 길이만큼 반복한다. LIBS.asm의 FUNCA는 현재 문자를 포인터가 가리키는 위치에 저장하고, 포인터를 증가시키는 방식으로 작동하는데, 이는 마치 스택에 값을 push하는 구조와 유사하다.

이후 OLOOP 루프에서는 +JSUB FUNCB를 통해 포인터를 1 감소시키고, 해당 위치의 문자를 다시 읽어온다. 이 문자에 ADD #32 연산을 수행하여 대문자 아스키 코드 값을 소문자로 변환한 뒤, +STCH STROUT, X 명령어로 출력 버퍼에 저장한다. FUNCB는 포인터를 감소시킨 뒤 해당 위치에서 문자를 불러오는 구조로, 이는 스택에서 값을 꺼내는 pop 동작과 유사하다. 결과적으로 입력 문자열을 거꾸로 뒤집은 소문자 형태가 출력된다.

## 나)실행과정

우선 작업 폴더에 QUIZ.asm과 LIBS.asm 파일을 작성하였다.

이후 "java -cp SicTools/out/make/sictools.jar sic.Asm QUIZ.asm", "java -cp SicTools/out/make/sictools.jar sic.Asm LIBS.asm" 명령어를 이용하여 각각의 어셈블리 파일을 .Obj파일로 변환하였다.



```
1 HQUIZ 00000000040
2 RFUNCA FUNCB INIT
3 T0000001E4B100000720267510000853A0224B100000B8503B2FF40720134B100000
4 T00001E1A19002057900038B8503B2FF03F2FFD0000005354524553534544
5 M00000105+INIT
6 M00000805
7 M00000F05+FUNCA
8 M00001B05+FUNCB
9 M00002205
10 E000000
11
```

```

1  HLIBS  000000000159
2  DFUNCA 000009FUNCB 000018INIT 000000
3  T0000001E0120270F20214F000056201B0320181900010F20124F000003200C1F2138
4  T00001E090F20065220034F0000
5  T00015603000001
6  E000000
7

```

생성된 QUIZ.obj에서 Format4 Instruction을(+JSUB) 사용한 부분에 대해 Modification Record가 생성되어있음을 확인할 수 있었다.

마지막으로 "java -cp SicTools/out/make/sictools.jar sic.Link -o ANSWER.obj QUIZ.obj LIBS.obj" 명령어를 통하여 QUIZ.OBJ와 LIBS.obj를 링커로 결합하였다.

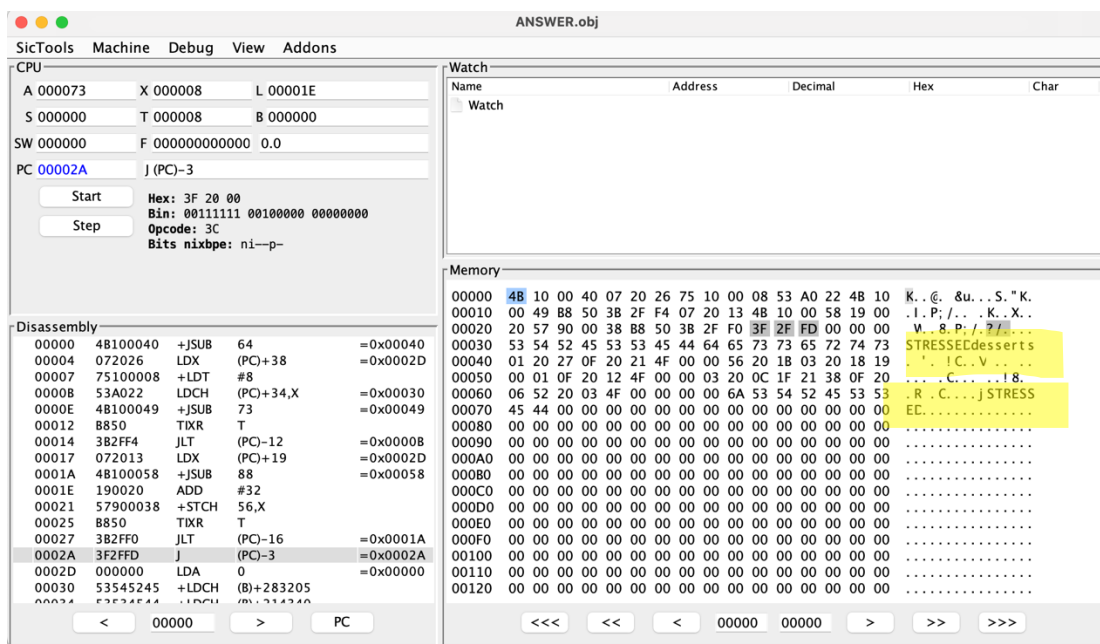
최종적으로 링커가 EXTDEF, EXTREF, Modification Record를 기반으로 하여 하나의 실행 가능한 ANSWER.obj를 생성하였다.

```

1  HANSWER000000000199
2  T0000001E4B100040072026751000053A0224B100049B8503B2FF40720134B100058
3  T00001E1A19002057900038B8503B2FF03F2FD0000005354524553534544
4  T0000401E0120270F20214F000056201B0320181900010F20124F000003200C1F2138
5  T00005E090F20065220034F0000
6  T00019603000001
7  M00000105
8  M00000005
9  M00000F05
10 M00001B05
11 M00002205
12 E000000
13

```

## 다)실행결과



SicTools을 통해 확인한 결과, 입력된 문자열 "STRESSED"가 버퍼에 저장된 뒤, 이를 역순으로 읽어 STROUT에 "desserts"라는 결과가 저장되는 것을 확인할 수 있었다.