

1. 배경지식

1) Pseudo instruction

Pseudo instruction은 assembler에게 전달하는 명령어이다. assembler는 Pseudo instruction을 어셈블리 명령어로 변환하여 동작한다. Pseudo instruction은 하나 이상의 어셈블리 명령어로 변환될 수 있다. LDR, ADR 등의 명령어가 pseudo instruction에 해당한다.

2) strcpy

strcpy는 문자열을 복사하는 함수이다. C언어에서 strcpy는 문자열의 끝을 의미하는 NULL까지 복사하고 종료된다. 이번 과제에서 구현한 strcpy는 문자열의 끝을 cr EQU 0x0d로 정의한 cr을 이용한다.

2. 코드내용

프로그램의 첫 번째 줄에서 0x0d를 cr로 정의했다. cr은 프로그램에서 문자열의 마지막을 표시하는 것에 이용된다. 프로그램이 실행되면, LDR, ADR명령어를 이용해 복사할 문자열이 저장된 주소와 복사결과를 저장할 주소를 r0과 r1에 각각 저장한다. Loop label아래에서 r0에서 한 byte를 읽어 r1에 복사한다. 이후 해당 값이 문자열의 끝인 cr인지 확인하고, 맞는 경우 프로그램 종료, 아닌 경우 다음 1byte를 읽어 복사하기 위해 Loop로 이동한다.

Disassembly에 나타난 결과는 다음과 같다. 붉은 색으로 나타난 부분이 직접 작성한 코드이고, 검은 색으로 나타난 부분이 실행된 assembly code이다. ADR은 ADD명령어로 바뀌어 실행되었고, #1로 표현된 정수는 더 정확한 표현인 #0x0001로 바뀌었다. 명령 반복에 사용된 label인 Loop와 Done은 0x0000001C와 같은 값으로 바뀐 것을 확인할 수 있다. 또한 equ를 이용하여 정의했던 cr도 해당 값으로 치환되었다.

```

Disassembly
6:      ADR r0, Table      ;r0 = address of string
0x00000000 E28F001C ADD     R0,PC,#0x0000001C
7:      LDR r1, TEMPADDR1 ;to save strcpy result
8:
9: Loop
0x00000004 E59F1014 LDR     R1,[PC,#0x0014]
10:     LDRB r2, [r0], #1   ; read one byte
0x00000008 E4D02001 LDRB     R2,[R0],#0x0001
11:     STRB r2, [r1], #1   ; copy 1 byte
0x0000000C E4C12001 STRB     R2,[R1],#0x0001
12:     CMP r2, #cr        ; string already copied
0x00000010 E352000D CMP     R2,#0x0000000D
13:     BEQ Done           ; r2==cr : break
0x00000014 0A000000 BEQ     0x0000001C
14:     B Loop            ; to read next byte
15:
16: Done
0x00000018 EAF0FFFA B       0x00000008
17:     MOV pc,#0
0x0000001C E3A0F000 MOV     PC,#0x00000000

```

메모리의 주소값을 레지스터로 불러오는 `ADR`, `LDR` 명령어의 동작을 비교하면 다음과 같다. `ADR`의 경우 `ADD`로 바뀌었고, `R0`에 `PC`의 값과 `#0x0000001C`를 더한 것을 확인할 수 있다. 반면 `LDR`은 다른 명령어로 바뀌어 실행되지는 않았고, `TEMPADDR1`이 [`PC`, `#0x0014`]로 바뀌었다. 이것으로 보아 `ADR`와 `LDR`은 같은 동작을 하지만, 동작을 수행하는 방법이 내부적으로 다르다는 것을 알 수 있다. 또한 `label`을 이용하여 저장한 주소값은 두 경우 모두에서 `PC`와 하나의 수로 바뀌었다. `ADR`과 `LDR` 연산에서 주소값은 `PC`에 대한 상대적인 위치를 이용하여 계산한다고 생각할 수 있다.

```

6:          ADR r0, Table      ;r0 = address of string
0x00000000 E28F001C ADD      R0,PC,#0x0000001C
7:          LDR r1, TEMPADDR1 ;to save strcpy result
0x00000004 E59F1014 LDR      R1,[PC,#0x0014]

```

3. 결과

복사할 문자열

[illegible]

실행과정

Register	Value
Current	
R0	0x00000029
R1	0x40000000
R2	0x00000057
R3	0x00000000

실행 완료 후 복사된 문자열

[illegible]

4. 고찰

LDR과 STR을 한 번 이용하여 문자열을 복사하는 방법으로 기능을 구현했었다. 그러나 복사하려는 문자열의 길이가 긴 경우 문자열을 모두 복사할 수 없다는 것을 확인하고, 예제문제인 strlen에 이용된 것을 참고하여 문자열의 마지막에 특정 값을 삽입하는 방법을 사용하는 것으로 수정하였다. 이 방법을 사용하면 B 명령어를 이용한 서브루틴을 생성하여 실행해야하는 명령어의 수가 많아지지만, 원하는 문자열 전체를 복사할 수 있었다.