

어셈블리프로그래밍설계 및 실습 보고서

실험제목: pseudo_instruction

실험일자: 2023년 10월 31일 (화)

제출일자: 2023년 11월 14일 (화)

학 과: 컴퓨터공학과

담당교수: 이준환 교수님

실습분반: 화 6,7 목 5

학 번: 2020202037

성 명: 엄정호

1. 제목 및 목적 (3%)

A. 제목

pseudo_instruction

B. 목적

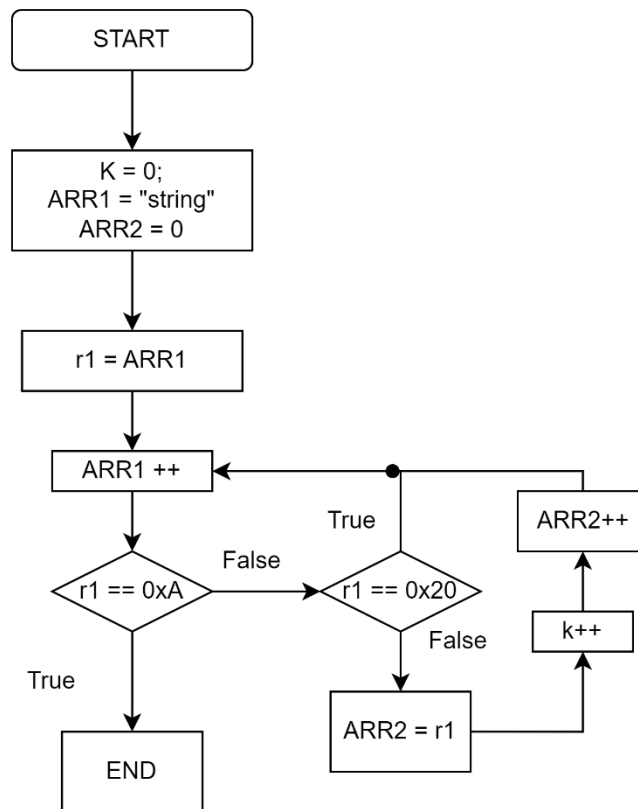
Pseudo instruction이 어셈블러에 의해 어떻게 컴퓨터 인식하는 instruction으로 변환되는지 직접 확인하고 이해하며, Label로 선언한 이름이 실제 주소로 어떻게 이용되는지 확인한다.

2. 설계 (Design) (50%)

A. Pseudo code

```
Main{  
  Call data memory arr1  
    Call data memory arr2  
  k = 0;  
  Call loop  
  
  END  
}  
LOOP{  
  If arr1[k] == end of line  
  ➔ Return k;  
  If arr1[k] == ' '  
  ➔ Arr1 ++;  
  Else arr1 == arr2;  
  K++  
  Arr2  
}
```

B. Flow chart 작성



C. Result

First Data - arr1

```
0x00000003D: 48 65 6C 6C 6F 20 57 6F 72 6C 64 0A
```

Copied data -arr2

```
0x00000004C: 48 65 6C 6C 6F 57 6F 72 6C 64
```

RESULT K;

```
0x00000003C: 0A (
```

D. Performance

Code Size : 96

State : 159

3. 고찰 및 결론

A. 고찰 (35%)

프로그램 메모리 관련해서 문제가 발생했다.

기존에 Label에 데이터를 저장하고 이를 출력해 다른 memory주소에 저장 하는 것을 진행 했을 때는 에러가 발생하지 않았지만 이번엔 RO와 RW?가 같다는 오류가 발생했다.

작성한 코드 중 AREA 이 부분을 제외한다면 문제가 발생하지 않았지만 이 코드를 쓰기 위해서는 에러의 원인을 찾아야 했다.

내가 찾아낸 결론은 해당 프로젝트의 linker부분에서 R/O의 주소 값과 R/W의 주소 값이 같아 문제가 발생한 것으로 데이터의 범위를 지정해 주는 dataArray,DATA를 넣기 위해서는 다르게 해야 할 필요가 있었다.

또한 R/W의 데이터를 내 임의대로 지정하는 것이 아닌 컴파일 오류에서 지정한 위치로 변경을 해주어야 문자열의 데이터 위치를 찾아 내는 것이 가능했다.

```
Disassembly
5:      LDR r1, =Arr1 ; array 1
0x00000000 E59F1034 LDR      R1,[PC,#0x0034]
6:      LDR r2, =Arr2 ; array 2
0x00000004 E59F2034 LDR      R2,[PC,#0x0034]
7:      LDR r5, =K ; length of array2
0x00000008 E59F5034 LDR      R5,[PC,#0x0034]
8:      mov r4,#0 ; reg of
9:
10:
0x0000000C E3A04000 MOV      R4,#0x00000000
11:      BL LOOP; ; loop for cpoy
12:
0x00000010 EB000001 BL      0x0000001C
13:      STR r4,[r5] ; store length
14:
15:
0x00000014 F5854000 STB      B4 [B5]
```

또한 내가 작성한 Pseudo instruction이 어떻게 컴퓨터에서 작용하는지 확인할 수 있었다.

B. 결론 (10%)

이번 실험에서는 메모리에 문자열을 저장하고 저장된 문자열의 주소를 받아와 복사해 다른 메모리에 넣어주는 기능을 구현했다. ARR1이라는 Label또한 어딘가의 주소 값을 가리킨다는 것을 이해했고, 해당 주소를 관리하기 위해서는 읽고 쓰는 linker의 데이터를 다르게 조작할 필요가 있다는 것을 알게 되었다.

4. 참고문헌 (2%)

이준환/어셈블리프로그래밍설계및실습/광운대학교/2023