

# Computer Architecture Assignment

박준호 (201510756)  
전북대학교 컴퓨터공학부  
Krs0123@naver.com

## 요약

MIPS assembly를 이용하여 주어진 C코드를 MIPS 어셈블리 코드로 구현한다.

첫 번째 코드는 1부터 주어진 숫자까지의 합을 구하는 코드이다.

두 번째 코드는 피보나치 수열 중 주어진 숫자인 n 번째의 값은 무엇인지 찾는 코드이다.

## 1. 실습 프로그램의 구성 및 동작 원리

첫번째 과제.

```
12 #put your code here
13 sum:
14 addi $sp, $sp, -8
15 sw $ra, 0($sp)
16 sw $a0, 4($sp)
17
18 slti $t0, $a0, 1
19 beq $t0, $0, recur
20
21 add $v0, $0, $0
22 addi $sp, $sp, 8
23 jr $ra
24
25 recur:
26 addi $a0, $a0, -1
27 jal sum
28
29 lw $ra, 0($sp)
30 lw $a0, 4($sp)
31 addi $sp, $sp, 8
32
33 add $v0, $a0, $v0
34 jr $ra
```

함수를 재귀 하기 위해서 recur라는 라벨을 따로 만들어 재귀하게 만들었다. 재귀 할 때마다 스택을 8만큼씩 비우고, 재귀가 끝나고 돌아올 때 다시 8만큼씩 채워줬다. \$a0의 값이 0이 되면 recur 로 가지 않게 되고 \$v0를 0으로 초기화 하며, jr을 통해 sum 밑으로 돌아가 \$v0에 \$a0의 값을 계속해서 저장하게 된다 이때 비워 뒀던 스택을 다시 채워준다.

처음에 재귀함수를 구현해 내는 것이 가장 어려웠으며, 8만큼 비워 뒀던 스택을 다시 채워주지 않아 무한루프 도는 실수도 있었다. 재귀함수를 돌면서 스택에 백업 하는 것도 0, 4, 8, 12 ... 이런 식으로 쌓았는데 이렇게 하지 않아도 된다는 것을 알게 되었다.

처음에 주어진 코드에 추가하는 것이 없도록 구현하였고, 메모리 누수가 없도록 다시 다 채워줬다.

## 두번째 과제.

```

12 #put your code here
13 fib:
14 addi $sp, $sp, -8
15 sw $ra, 0($sp)
16 sw $a0, 4($sp)
17 addi $t2, $a0, 0
18 addi $t0, $0, 1
19
20 beq $a0, $t0, one
21 slli $t1, $a0, 2
22 beq $t1, $0, recur
23 jr $ra
24
25 recur:
26 addi $a0, $t2, -1
27 jal fib
28
29 lw $a0, 4($sp)
30 addi $t2, $a0, 0
31
32 addi $a0, $t2, -2
33 jal fib
34
35 lw $ra, 0($sp)
36 addi $sp, $sp, 8
37 jr $ra
38
39 one:
40 lw $ra, 0($sp)
41 addi $v0, $v0, 1
42 jr $ra

```

기본 틀은 첫번째 과제와 비슷하게 만들었다. fib함수에서 recur로 들어가 재귀하고, 돌아오면서 스택을 다시 채우며 값이 나오는 형식이다.

처음 값이 2이라고 하면, recur로 가게 되고 값을 1만큼 감소시킨 후 첫번째 fib를 재귀호출 하게 된다. 그럼 값은 1이므로 one으로 가 \$v0에 1을 더해 넣고, \$ra로 다시 돌아온다. 그러면 \$a0의 값이 변질 되어있으므로 백업해 두었던 \$a0 값을 다시 불러오고 2만큼 감소시킨 값으로 두번째 fib를 부른다.

이때는 값이 0이므로 아무것도 하지 않은 채 다시 돌아오게 되고, recur에서 \$ra로 돌아 가게 된다. 그러면 2번째 피보나치 수열의 값은 one에 들렸던 수의 합이 되므로 1이 된다.

이 코드를 짜면서 무한루프가 끊이질 않았다. 스택을 다시 되돌려주는 게 미숙했던 것 같다. 첫번째 과제에서 재귀적으로 돌아가는 알고리즘이 익숙해져 있어서 피보나치를 구현해 내는 것에만 신경 썼다.

## 2. 결과

### 첫번째 과제

		Registers		Coproc 1	Coproc 0
		Name	Number	Value	
4: lb \$a0, n		\$zero	0	0x00000000	
		\$at	1	0x10010000	
		\$v0	2	0x0000000a	
5: jal sum		\$v1	3	0x0000000f	
6: add \$v1, \$v1, \$v0		\$a0	4	0x00000005	
9: li \$v0, 10		\$a1	5	0x00000000	
10: syscall		\$a2	6	0x00000000	
14: addi \$sp, \$sp, -8		\$a3	7	0x00000000	
15: sw \$ra, 0(\$sp)		\$t0	8	0x00000001	
16: sw \$a0, 4(\$sp)		\$t1	9	0x00000000	
18: slli \$t0, \$a0, 1		\$t2	10	0x00000000	
19: beq \$t0, \$0, recur		\$t3	11	0x00000000	
21: add \$v0, \$0, \$0		\$t4	12	0x00000000	
22: addi \$sp, \$sp, 8		\$t5	13	0x00000000	
23: jr \$ra		\$t6	14	0x00000000	
26: addi \$a0, \$a0, -1		\$t7	15	0x00000000	
27: jal sum		\$a0	16	0x00000000	
29: lw \$ra, 0(\$sp)		\$s1	17	0x00000000	
30: lw \$a0, 4(\$sp)		\$s2	18	0x00000000	
31: addi \$sp, \$sp, 8		\$s3	19	0x00000000	
33: add \$v0, \$a0, \$v0		\$s4	20	0x00000000	
34: jr \$ra		\$s5	21	0x00000000	
		\$s6	22	0x00000000	

n이 5일 때 결과값 = f = 15

1~5 까지의 합은 15 이므로 실행결과 문제없음.

## 두번째 과제

		Registers	Coproc 1	Coproc 0
		Name	Number	Value
		\$zero	0	0x00000000
		\$at	1	0x10010000
		\$v0	2	0x00000000
		<b>\$v1</b>	<b>3</b>	<b>0x00000005</b>
		\$a0	4	0x00000001
		\$a1	5	0x00000000
		\$a2	6	0x00000000
		\$a3	7	0x00000000
		\$t0	8	0x00000001
		\$t1	9	0x00000001
		\$t2	10	0x00000001
		\$t3	11	0x00000000
		\$t4	12	0x00000000
		\$t5	13	0x00000000
		\$t6	14	0x00000000
		\$t7	15	0x00000000
		\$s0	16	0x00000000
		\$s1	17	0x00000000
		\$s2	18	0x00000000
		\$s3	19	0x00000000
		\$s4	20	0x00000000
		\$s5	21	0x00000000
		\$s6	22	0x00000000
		\$s7	23	0x00000000
		\$t8	24	0x00000000
		\$t9	25	0x00000000
		\$k0	26	0x00000000
		\$k1	27	0x00000000
		\$sp	28	0x10008000
		\$fp	29	0x7fffffc0
		\$gp	30	0x00000000
		\$ra	31	0x00400000
		pc		0x00400018
		hi		0x00000000
		lo		0x00000000

n이 5일 때 = 5번째 피보나치 수열의 값 = 5

		Registers	Coproc 1	Coproc 0
		Name	Number	Value
		\$zero	0	0x00000000
		\$at	1	0x10010000
		\$v0	2	0x00000000
		<b>\$v1</b>	<b>3</b>	<b>0x00000037</b>
		\$a0	4	0x00000000
		\$a1	5	0x00000000
		\$a2	6	0x00000000
		\$a3	7	0x00000000
		\$t0	8	0x00000001
		\$t1	9	0x00000001
		\$t2	10	0x00000000
		\$t3	11	0x00000000
		\$t4	12	0x00000000
		\$t5	13	0x00000000
		\$t6	14	0x00000000
		\$t7	15	0x00000000
		\$s0	16	0x00000000
		\$s1	17	0x00000000
		\$s2	18	0x00000000
		\$s3	19	0x00000000
		\$s4	20	0x00000000
		\$s5	21	0x00000000
		\$s6	22	0x00000000
		\$s7	23	0x00000000
		\$t8	24	0x00000000
		\$t9	25	0x00000000
		\$k0	26	0x00000000
		\$k1	27	0x00000000
		\$sp	28	0x10008000
		\$fp	29	0x7fffffc0
		\$gp	30	0x00000000
		\$ra	31	0x00400000
		pc		0x00400018
		hi		0x00000000
		lo		0x00000000

N이 10일 때 = 10번째 피보나치 수열의 값 = 0x37 = 55.

## 3. 결론

재귀함수로 코드를 짜는데 많은 어려움이 있었고, 스택을 할당하고 다시 되돌려주는 알고리즘도 이해가 안돼서 시행착오도 많이 겪었다. ppt도 많이 찾아봤고, 인터넷 검색도 하며 이해하고 코드를 작성했다. 그 중에서도 특히 재귀함수를 돌릴 때 무한루프가 자주 돌아갔는데 그 때마다 어디에 문제가 있는지 찾는데 가장 고생한 것 같았다. 코드 알고리즘 내에 문제가 있었을 때도 있었고, 스택에 문제가 있을 때도 있었다. 결국은 코드를 다 짰으니까 이젠 문제 없다.