

Lab4

Exercise 1: Debugging `mega1istmanips.s`

给了一个函数 `map`

```
1 void map(struct node *head, int (*f)(int)) {
2     if (!head) { return; }
3     for (int i = 0; i < head->size; i++) {
4         head->arr[i] = f(head->arr[i]);
5     }
6     map(head->next, f);
7 }
```

我们需要找出其汇编中的五处bug

本任务的难点在于，我们需要明白在调用函数前我们需要把哪些东西放到栈上面去。

- 错误1：修改 `maploop` 的第一句话 `add t1, s0, x0` 为 `lw t1, 0(s0)`
- 错误2：在调用 `jalr s1` 之前，将必要的寄存器入栈(应该只存t1就可以了)

```
1     addi sp, sp, -12
2     sw t0, 0(sp)
3     sw t1, 4(sp)
4     sw t2, 8(sp)
```

```
1     lw t0, 0(sp)
2     lw t1, 4(sp)
3     lw t2, 8(sp)
4     addi sp, sp, 12
```

- 错误3：
一个 `int` 为四个 `byte`

```
1 添加 addi t1, t1, 4
```

- 错误4：

```
1  la a0, 8(s0) 改为 lw a0, 8(s0)
```

- 错误5：

```
1  lw a1, 0(s1) 改为 add a1, s1, x0
```

注意 `lw` 和 `add` 的区别

- **Load Address (la)**
 - `la dst, label`
 - Loads address of specified label into `dst`
 - translates to: `auipc dst, <offset to label>`

Exercise 2: Write a function without branches

我们要写出一个函数 `f`, 使得其满足

```
1 f(-3) = 6
2 f(-2) = 61
3 f(-1) = 17
4 f(0) = -38
5 f(1) = 19
6 f(2) = 42
7 f(3) = 5
```

但是我们不能用分支。

解决办法: 每个自变量加3就是它在数组中的下标。

```
1 f:
2     addi t0,a0,3
3     addi t1,x0,4
4     mul t0,t0,t1
5     add t1,t0,a1
6     lw t2,0(t1)
7     mv a0,t2
8     jr ra                # Always remember to jr ra after your function!
```