

mips 设计文档

李昊宸 计算机科学与技术 2017K8009929044

```
int main()
{
    int i = 1;
    int sum = 0;
    do
    {
        sum = sum + i;
        i = i + 1;
    } while (i <= 100);
    printf("sum = %d\n",sum);
    return 0;
}
```

将以上代码修改为 MIPS 汇编实现

```
.section .data
    output: .ascii "sum = "
.section .text
    .globl __start
__start:
    addiu    $v0, $0, 4004
    addiu    $t0, $0, 1
    addiu    $t1, $0, 1
    addiu    $t2, $0, 0
    addiu    $t3, $0, 101
    addiu    $t4, $0, 10
    addiu    $t6, $0, 4

    L1:
    add      $t2, $t2, $t1
    addiu    $t1, $t1, 1
    bne      $t1, $t3, L1
    nop

    la       $a1, output
    addiu    $a1, $a1, 9

    L2:
    divu     $t2, $t4
```

#输出结果的文字段

#写文件调用

#t1 为每次循环中增量的部分
#t2 存放部分和
#t3 寄存器存放循环次数
#t4 存放立即数 10，用于将
部分和转换成 ascii 码的形式
#L2 循环次数

#sum = sum + i
#i = i + 1
#i < 101?

#sum = 占 5 个字节
#数字 5050 占 4 个字节

#mod 运算，逐位将 sum 变为

```
mfhi    $t5
mflo    $t2
addiu   $t5, $t5, 48
```

#ascii 形式
#[HI] = GPR[t2] mod GPR[t4]
#[LO] = GPR[t2]/GPR[t4]
#ascii 化

```
sb      $t5, ($a1)
addiu   $a1, $a1, -1
addiu   $t6, $t6, -1
bne     $t6, $0, L2
nop
```

#存到 output 后面
#地址向前移动

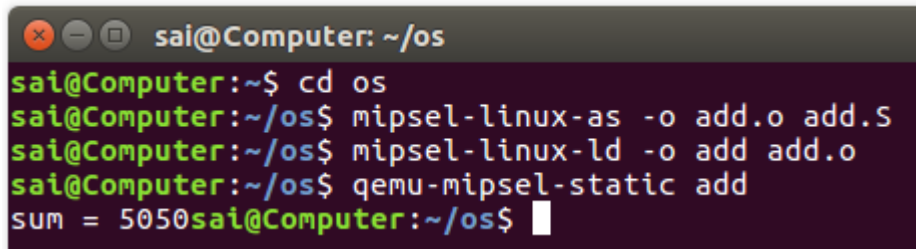
```
addiu   $a1, $a1, -5
addiu   $a2, $0, 9
syscall
```

#地址回归
#输出字节数

```
addiu   $v0, $0, 4001
addiu   $a0, $0, 0
syscall
```

#退出

运行结果



```
sai@Computer: ~/os
sai@Computer:~$ cd os
sai@Computer:~/os$ mipsel-linux-as -o add.o add.S
sai@Computer:~/os$ mipsel-linux-ld -o add add.o
sai@Computer:~/os$ qemu-mipsel-static add
sum = 5050sai@Computer:~/os$
```