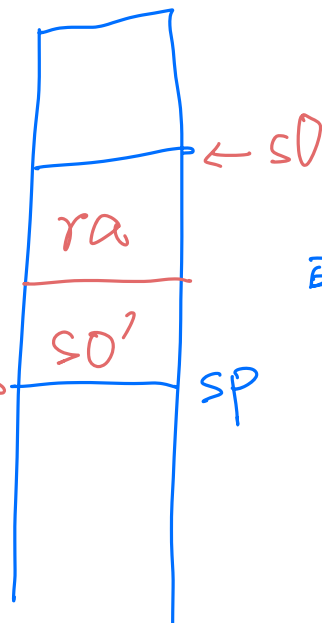


```

1  x:
2      .zero    4
3  y:
4      .zero    4
5  addtwonum:
6      addi     sp,sp,-16
7      sd      ra,8(sp)
8      sd      s0,0(sp)
9      addi     s0,sp,16
10  extern x=1
11      lui     a5,%hi(x)
12      li      a4,1
13      sw      a4,%lo(x)(a5)
14  extern y=1
15      lui     a5,%hi(y)
16      li      a4,2
17      sw      a4,%lo(y)(a5)
18  return x+y
19      lui     a5,%hi(x)
20      lw      a4,%lo(x)(a5)
21      lui     a5,%hi(y)
22      lw      a4,%lo(y)(a5)
23      addw    a5,a4,a5
24      sext.w  a5,a5
25      mv      a0,a5
26      ld      ra,8(sp)
27      ld      s0,0(sp)
28      addi     sp,sp,16
29      jr      ra
30
31  .LC0:
32      .string  "result \344\270\272: %d"
33  main:
34      addi     sp,sp,-32
35      sd      ra,24(sp)
36      sd      s0,16(sp)
37      addi     s0,sp,32
38      int result = addTwoNum
39      call     addtwonum
40      mv      a5,a0
41      sw      a5,-20(s0)
42      lw      a5,-20(s0)
43      mv      a1,a5
44      lui     a5,%hi(.LC0)
45      addi     a0,a5,%lo(.LC0)
46      call     printf
47      li      a5,0
48      mv      a0,a5
49      ld      ra,24(sp)
50      ld      s0,16(sp)
51      addi     sp,sp,32
52      jr      ra

```

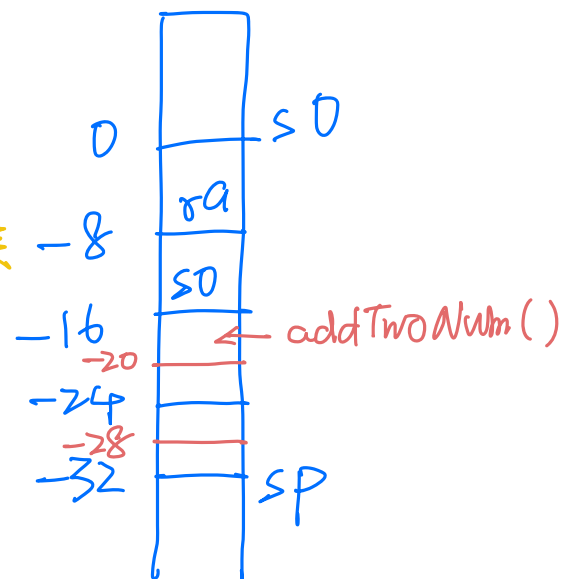


可以看到
x,y 表明为
extern 的时候
没有入栈

这其实是立即数

设置返回值

(退栈)



退栈

发现 gcc 特别喜欢使用 a5, a4

(首先 a0, a1 是不好的, 因为 a0, a1 通常
用于参数传递, 因此要保护)