```
 1    arr:        [handwritten: u64 arr[100]]
 2            .zero   800
 3    main:       [handwritten: int main(void){]
 4            addi    sp,sp,-32
 5            sd      ra,24(sp)
 6            sd      s0,16(sp)
 7            addi    s0,sp,32
 8    [handwritten: int i =20]
 9            li      a5,20
10            sw      a5,-20(s0)
11    [handwritten: int j=50]
12            li      a5,50
13            sw      a5,-24(s0)
14    [handwritten: int k=50]
15            li      a5,50
16            sw      a5,-28(s0)
17    [handwritten: arr[i] = 50]
18            lui     a5,%hi(arr)
19            addi    a4,a5,%lo(arr)
20            lw      a5,-20(s0)
21            slli    a5,a5,3
22            add     a5,a4,a5
23            li      a4,50
24            sd      a4,0(a5)
25            li      a5,0
26    [handwritten: }]
27            mv      a0,a5
28            ld      ra,24(sp)
29            ld      s0,16(sp)
30            addi    sp,sp,32
31            jr      ra
```
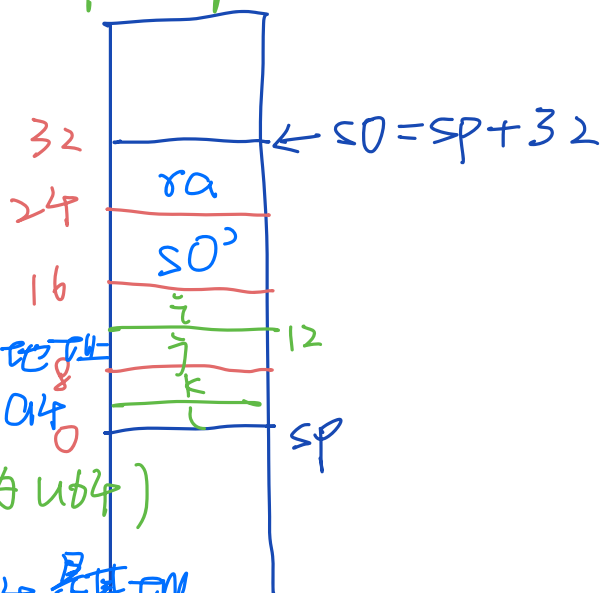
Handwritten annotations (Chinese):

SP 的大小 ⇒ 看有多少个局部变量
SP 通常是以 16B 对齐的

S0 = SP+32
32 → ra
24 → s0
16 → i
j — 12
8 → k
0 — SP

将 arr 地址放到 a4
i × 8 (因为 u64)

a4 是基址
a0 是 offset
a5 = a4 + a5 得到地址
a4 被用了, 放入 50
store a4 → 0(a5)
将 a4 放到 a5+0 位置