汇编

\_add\_a\_and\_b:

push %ebx

mov %eax, [%esp+8]

mov %ebx, [%esp+12]

add %eax, %ebx

pop %ebx

ret

\_main:

push 3

push 2

call \_add\_a\_and\_b

add %esp,8

ret

一、cpu指令

1行、push 指令

push 3

\_main开始执行，并且从堆上面为main建立一帧，且指向的地址写入esp寄存器。push 3 将运算子3放入堆，将3写入main帧

push存在一个前置操作，先取出esp里面地址，将其减去4字节，将新地址写入esp寄存器，使用减法是因为堆从高到底发展，4个字节是因为3类型是int，占用4字节，得到新地址，3就会写入这个地址开始的4个字节。

2行、

push 2

第2行一样，将2写入main帧，位置紧贴着1行的3，esp再减去4个字节。（esp-8）

3行、call指令

call \_add\_a\_and\_b

表示 调用\_add\_a\_and\_b函数，程序会去找标签，并且建立帧

4、\_add\_a\_and\_b的第二行

push %ebx

表示将ebx写入\_add\_a\_and\_b帧，push再将esp-4地址。

5、\_add\_a\_and\_b的第三行

mov指令

mov指令用于将一个值写入某个寄存器

**mov %eax, [%esp+8]**

表示esp寄存器地址加上8个字节，得到新地址，在这个地址堆取出数据，