## (1)定位耗费CPU的进程

top -c，就可以显示进程列表，然后输入P，按照cpu使用率排序。

## (2)定位耗费cpu的线程

top -Hp xxx进程Id，就是输入那个进程id，然后输入P,按照cpu使用率排序。

## (3)定位哪段代码导致的cpu过高

printf“"%6x\n" 16872,把线程pid 转换成16进制，比如41e8

jstack 43987 | grep '0x41e8" -C5 --color

这个就是用jstack打印进程的堆栈信息，而且通过grep那个线程的16进制的pid,找到那个线程相关的东西,这个时候就可以在打印出的代码里，看到是哪个类的哪个方法导致的这个cpu 100%的问题

printf "%6x\n" 30063

jstack 26873|grep 756f -C5 --color=auto