# CPU调度

内核config CONFIG\_PREEMPT，此config打开表示内核态支持抢占，结果会引起吞吐量变小，进程切换频繁。

服务器内核一般配置成内核态不支持抢占，服务器追求吞吐量。

手机一般配置成内核态支持抢占，因为手机要降低调度延迟。

…

工具：<https://github.com/bytedance/trace-noschedule>

用于检查业务中延迟高导致的问题。

用此工具检查出高延迟的问题后，对于可睡眠的场景加 cond\_resched 主动释放cpu；对于不可睡眠的场景，需要排查原因做优化。

nice: https://www.runoob.com/linux/linux-comm-nice.html

# 内存管理

## cache

### arm64 memset

arm64对memset有一个优化，如果memset的值为0，则会用 dc zva 来操作 cache，通过 cache来加速内存的清0。但这样会导致一个问题: 对于 device memory 是不能进行任何 dc 操作的，会引起 alignment faut.(arm64 的cache支持非对齐访问，dc zva 只对 cacheable normal memory使用)

## 栈

### 综述

简单说，所有新进程都是继承parent的寄存器上下文(内核线程除外) 多线程与多进程在用户栈处理有所区别，其他的都差不多。

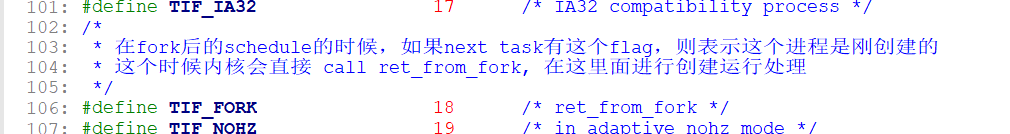
### arm64

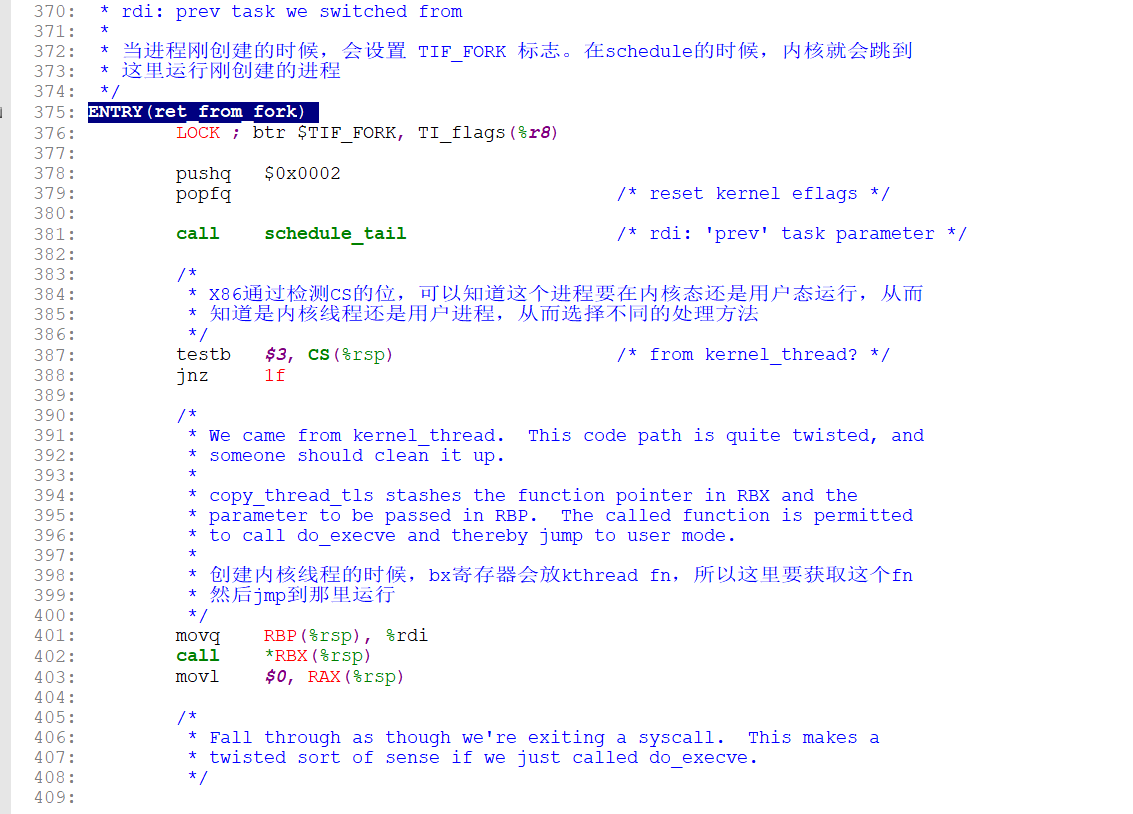




### x86\_64









# 中断

工具: <https://github.com/bytedance/trace-irqoff/>

用于检查中断/软中断关闭过长的问题

# debug

kprobe template: <https://github.com/smcdef/kprobe-template>

page cache scan: <https://github.com/smcdef/pgcache_scan>

kernel sample

kprobe, ftrace, trace\_events

<https://elixir.bootlin.com/linux/v5.6.8/source/samples>