

一、编写这个脚本的理由：

1. 抓取systrace之后，需要使用chrome在网站chrome://tracing/解析出来，但是很多时候，我们发现问题之后，需要vim systrace文件定位到具体位置。这时候问题来了，不容易查找这个task到底在文件的哪个位置。
2. 这个脚本的目的就是很容易的解决了此问题。

二、使用方式

在systrace上面作为零点判断的依据是irq、ipi或者task。目前抓取的systrace没有看到例外的。所以Python脚本里面使用上面三个之一作为零点的依据，即谁出现的最早谁就是零点。

```
python SystraceAnalysis.py -i systrace.html -o trace.txt -d delta_value
```

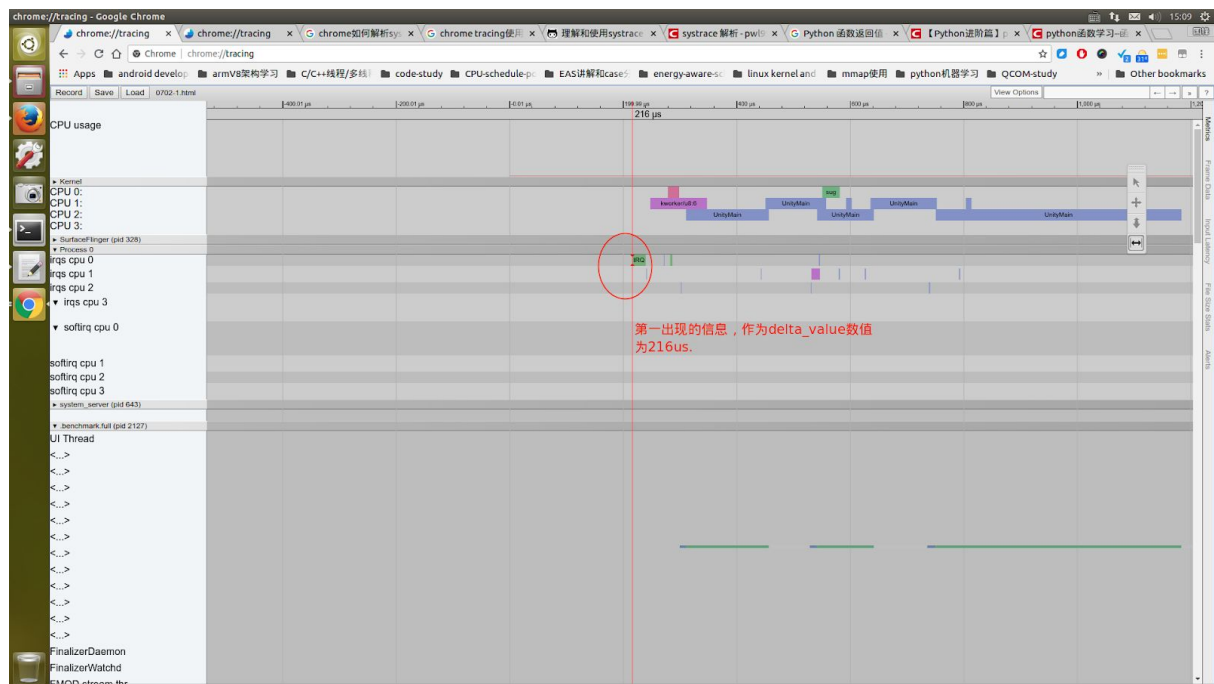
说明：

systrace.html是实际抓取的systrace

trace.txt:是时间戳修正的trace文件

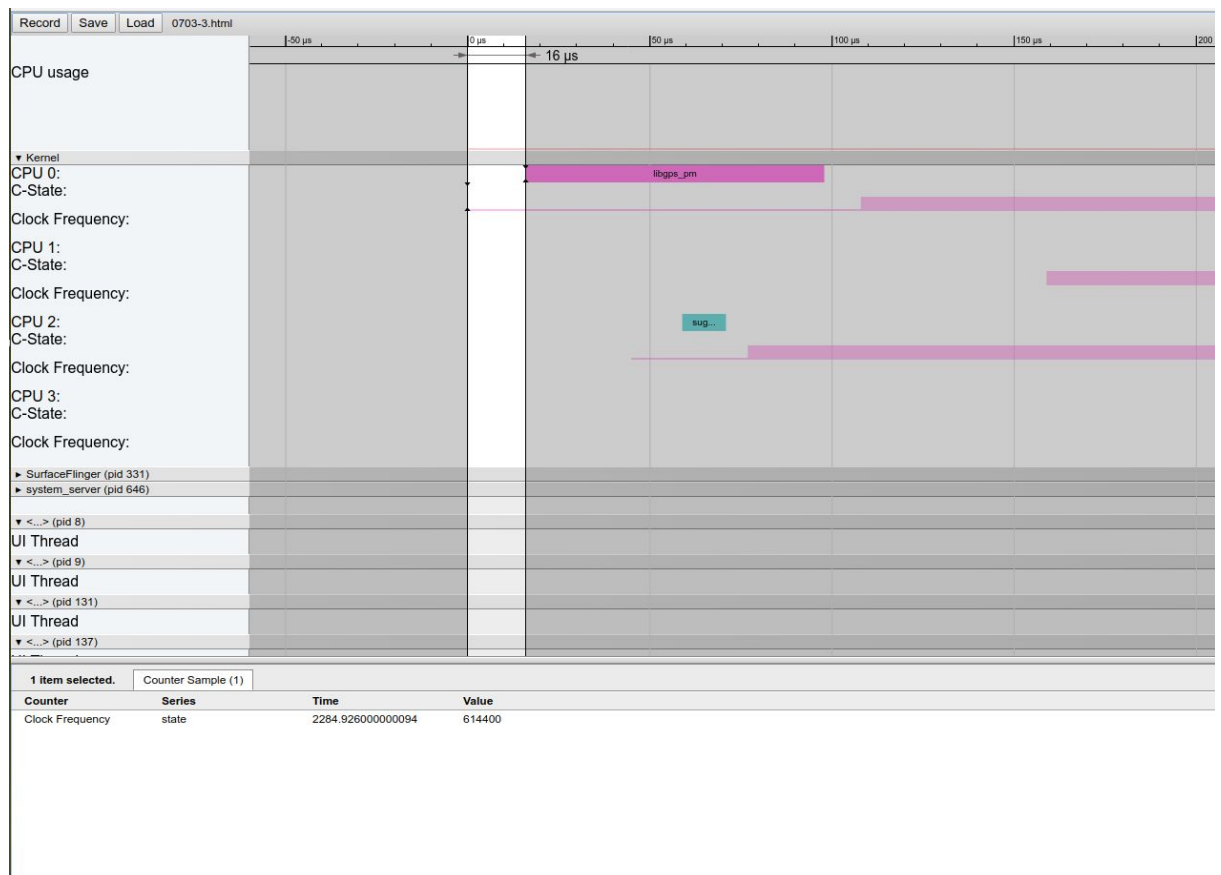
delta_value：是chrome打开systrace.html之后手动查找的在浏览器上面看到的第一个ipi/irq/task的位置，delta_value数值单位为us，如果碰到其他单位，请自动转换。如1.3ms,则delta_value=1300

如下所示：



分三种情况来解析

1. trace中第一次出现irq_handler_entry，则作为零点，delta value就是此irq在chrome显示的相对零点的数值，如上图所示为216
2. trace中第一次出现ipi_raise，上面类似
3. 在user版本中，不能使用irq event抓取systrace，所以systrace中不存在irq和ipi信息，那么这种就以sched_wakeup && sched_switch并行出现作为相对零点。如下图所以：



那么执行如下命令解析如下：

```
python SystraceAnalysis.py -i 0703-3.html -o 0703-3-1.html -d 16
```

示例在最后

三、解析之后如何使用

1. chrome打开原始systrace.html文件
2. 找到delta_value数值
3. 执行Python脚本：**python SystraceAnalysis.py -i systrace.html -o trace.txt -d delta_value**
4. **vim** 打开trace.txt，就可以对应到systrace.html上的各类信息了。