大致的设计思路

1. 常规输出（以进程为单位统计）

不需要携带任何命令，会以进程为单位输出所有进程调用 alloc\_pages 的统计信息：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 进程id | 命令 | 申请到的页数 | 申请失败的页数 | 总申请页数 |
| PID | COMM | SUCCESS | FAIL | ALL |

1. detail 模式

添加 -d 参数:

trace\_alloc -d

直接输出每一次的调用信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 进程id | 命令 | 线程id | 申请的页数 | GFP分配标志 | 返回的地址（struct page\*） |
| PID | COMM | TID | PAGES | MASK | ADDR |

1. 指定进程（以线程为单位统计）

指定 -p 参数：

trace\_alloc -p [pid]

可以单独监视某一个进程的调用信息，以线程为单位进行统计，目前还没有发现有申请失败的情况：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 线程id | 命令 | 申请到的页数 | 申请失败的页数 | 总申请页数 |
| TID | COMM | SUCCESS | FAIL | ALL |

1. 指定进程 detail 模式

-p 可以和 -d 一起用：

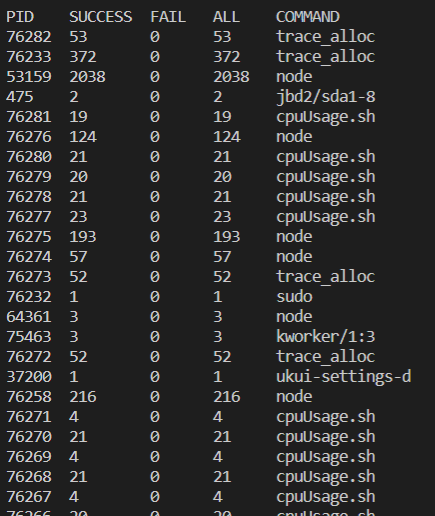
trace\_alloc -d -p [pid]

这是对该进程的每一次调用信息都进行输出:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 进程id | 命令 | 线程id | 申请的页数 | GFP分配标志 | 返回的地址 |
| PID | COMM | TID | PAGES | MASK | ADDR |

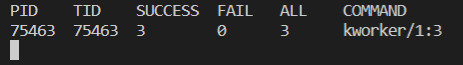
什么参数都不加，则按照pid进行显示



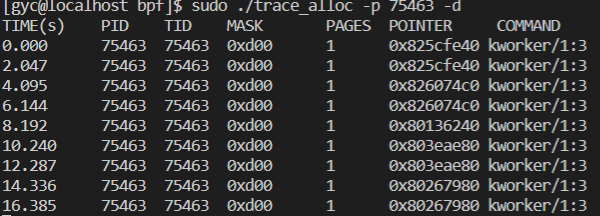


指定pid，则按照tid进行显示





指定pid并且设置-d，则输出每一次调用信息



目前就只实现了这些参数，其中刷新时间间隔还没有实现

