# lab8\_oom实验说明

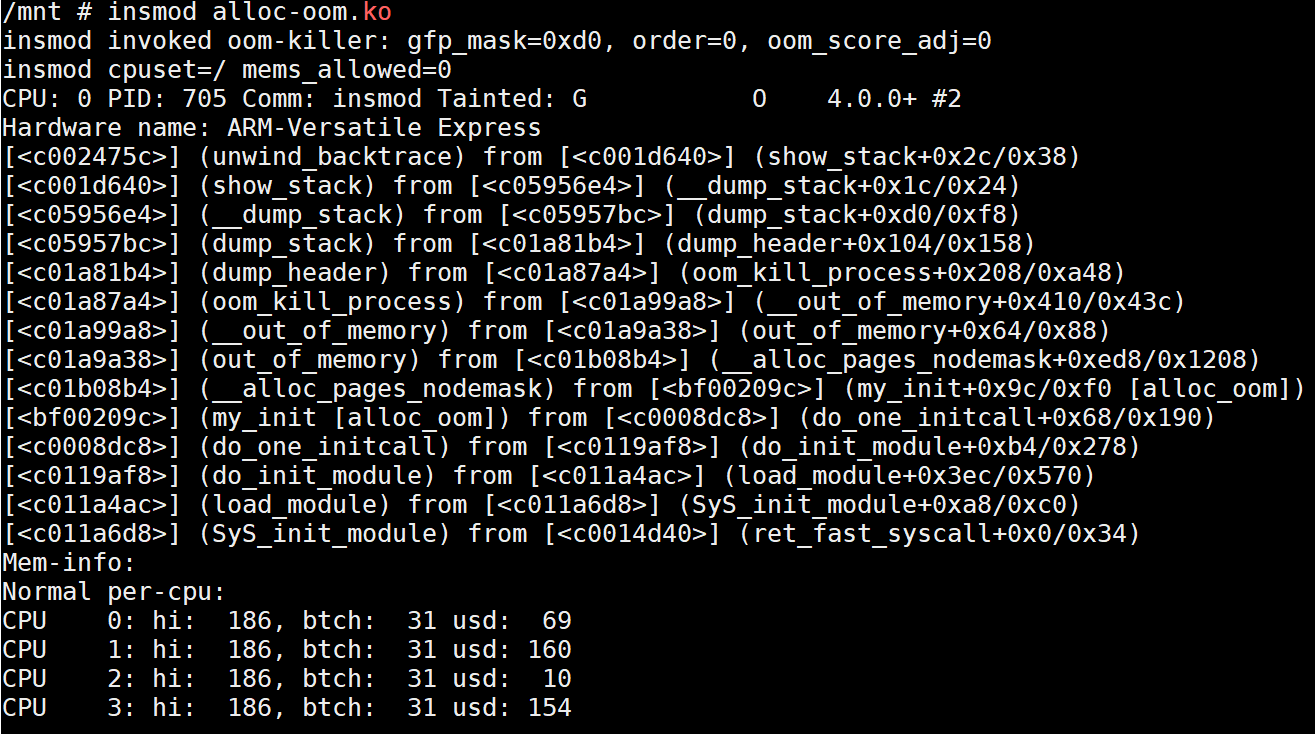
本实验是为了学习oom killer机制。

书上说，写一个用户空间的app来触发oom机制，这个可能不太容易触发OOM机制。为此，我们写一个简单的内核驱动来触发，更加直观一些。

## 基本实验步骤

1. 编译lab8的内核模块，并且拷贝到Qemu虚拟机里面。

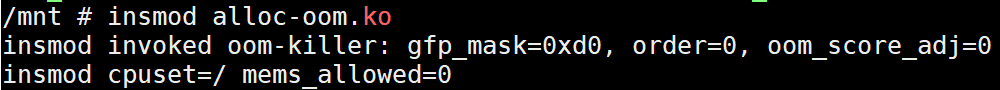
运行内核模块，过一会就会出现OOM Kill的现象了。



## 进阶思考

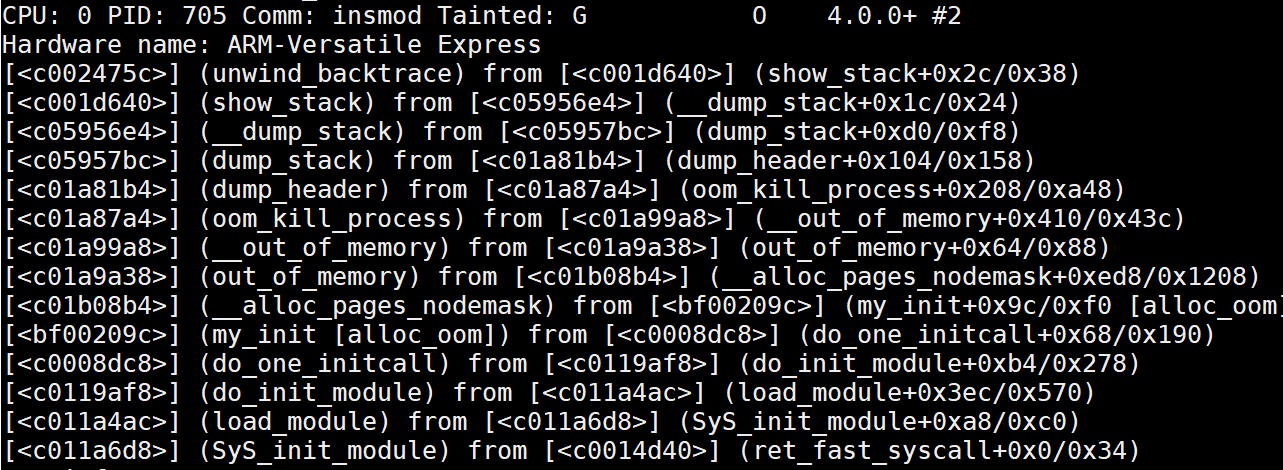
本实验重点是希望读者可以对照代码来分析oom killer的log。

比如：



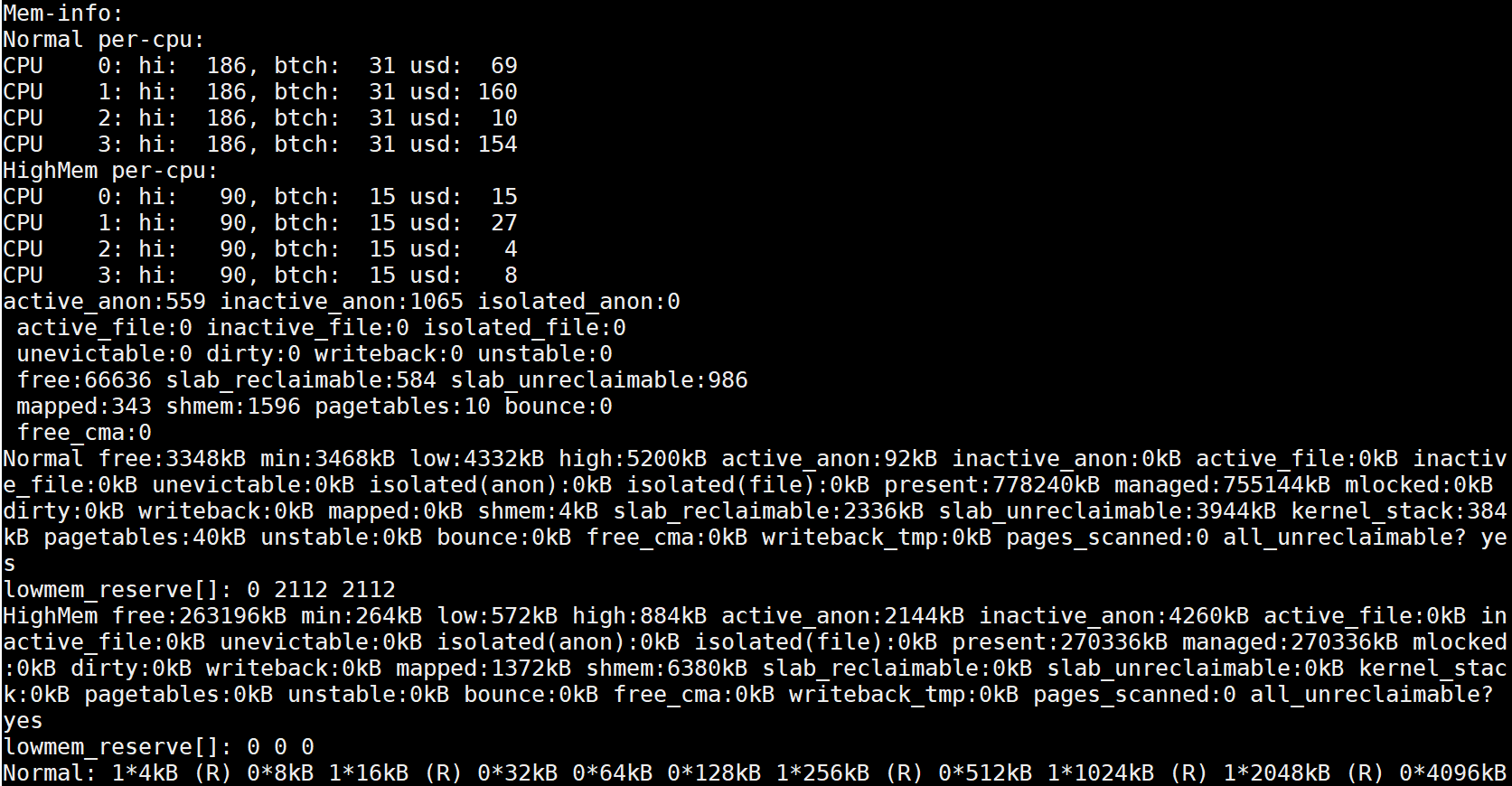
这句话是在oom\_kill\_process()->dump\_header()函数里打印出来的，告诉你那个进程触发的这次oom killer，这里显示是insmod进程，它在分配多少的物理页面的时候触发的，分配的掩码是啥，当时的oom\_score\_adj的值是啥。

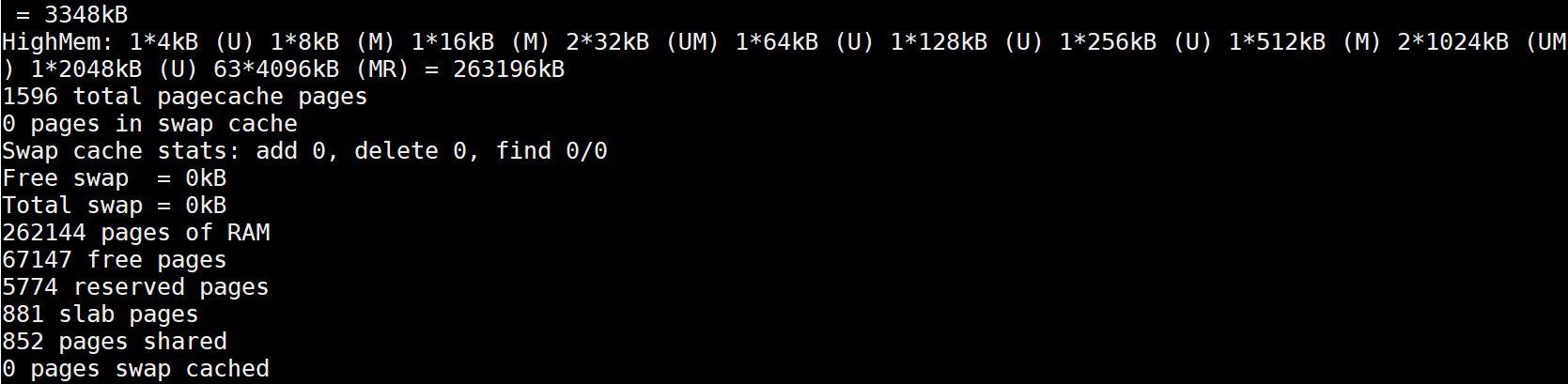
接下来。



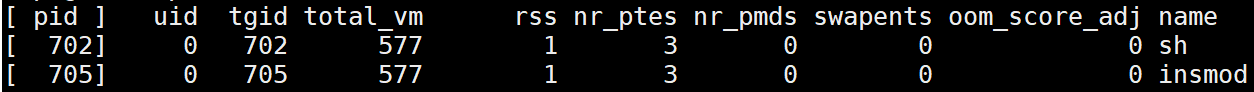
这是在oom\_kill\_process()->dump\_header()->dump\_stack()函数里打印的，这也是打印触发这次oom killer的函数调用栈的关系。

接下来：





这么一大段都是在oom\_kill\_process()->dump\_header()->show\_mem ()函数里打印的，告诉你现在触发oom killer的那个时刻，系统所有的内存系统，包括有多少匿名页面，多少page cache等等。

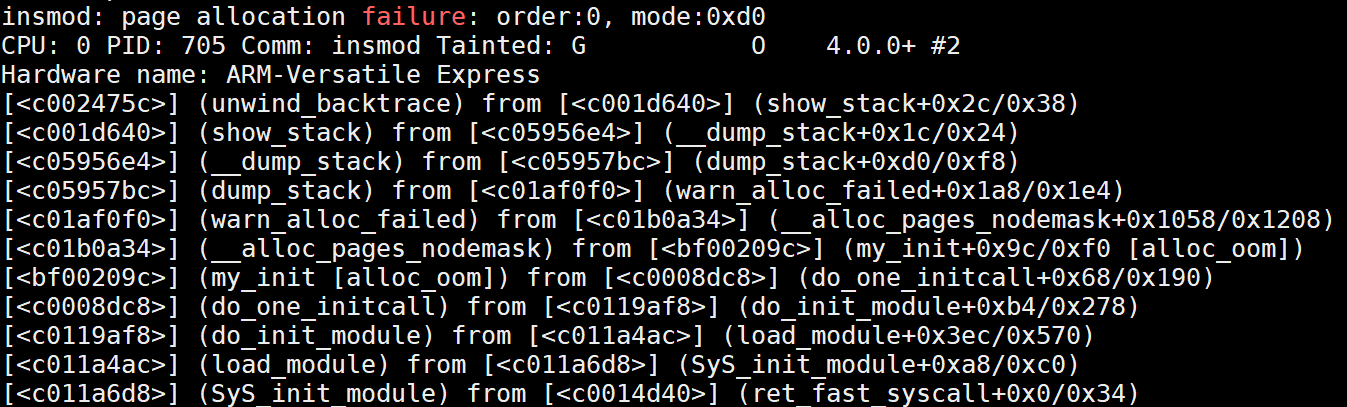


这里是在oom\_kill\_process()->dump\_header()->dump\_tasks ()函数里打印的，列举出有可能被杀的进程。



这里是在oom\_kill\_process()函数打印的，告诉你，我现在选择705进程来杀了，痛快！

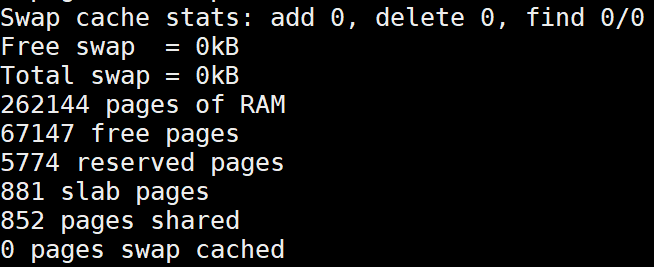
接下来：



杀完之后，insmod进程有发现分配内存失败了，又继续重复刚才的故事。。。

**进阶思考：**

我们从log里看到：



系统明明空闲页面还有67147个，为啥我分配一个页面都会让失败呢？

笨叔在这里没有太详细解析每个参数，如果大家对这个问题感兴趣，可以关注笨叔的第一季旗舰篇视频，笨叔会在视频中和大家详细解答。

请关注奔跑吧Linux社区微信公众号，里面有更多更有料更好玩的Linux干货：



奔跑吧Linux社区的旗舰篇视频节目，帮您打通Linux和职场任督二脉！





笨叔最新力作，全程约5小时高清，140多页ppt，8大实验，基于x86\_64的Centos 7.6和arm64，提供全套实验素材和环境。全面介绍kdump+crash在死机黑屏方面的实战应用，全部案例源自线上云服务器和嵌入式产品开发实际案例！

不能为客户和老板解决死机黑屏问题的都不是合格的linux运维或者研发攻城狮！





订阅视频请到淘宝店：<https://shop115683645.taobao.com>

