## strace是什么？

按照strace官网的描述, strace是一个可用于诊断、调试和教学的Linux用户空间跟踪器。我们用它来监控用户空间进程和内核的交互，比如系统调用、信号传递、进程状态变更等。

strace底层使用内核的ptrace特性来实现其功能。

在运维的日常工作中，故障处理和问题诊断是个主要的内容，也是必备的技能。strace作为一种动态跟踪工具，能够帮助运维高效地定位进程和服务故障。它像是一个侦探，通过系统调用的蛛丝马迹，告诉你异常的真相。

### strace常用选项：

从一个示例命令来看：

strace -tt -T -v -f -e trace=file -o /tmp/log/strace.log -s 1024 -p 23489

-tt 在每行输出的前面，显示毫秒级别的时间  
-T 显示每次系统调用所花费的时间  
-v 对于某些相关调用，把完整的环境变量，文件stat结构等打出来。  
-f 跟踪目标进程，以及目标进程创建的所有子进程  
-e 控制要跟踪的事件和跟踪行为,比如指定要跟踪的系统调用名称  
-o 把strace的输出单独写到指定的文件  
-s 当系统调用的某个参数是字符串时，最多输出指定长度的内容，默认是32个字节  
-p 指定要跟踪的进程pid, 要同时跟踪多个pid, 重复多次-p选项即可。

* **实例：跟踪nginx, 看其启动时都访问了哪些文件**

strace -tt -T -f -e trace=file -o /tmp/log/strace.log -s 1024 ./nginx

这里的输出只显示和文件访问有关的内容，这是因为我们通过-e trace=file 选项指定了。

* **实例：统计进程调用系统调用的情况**

strace -q -f -c -p 1234

* **strace的-e trace选项**

要跟踪某个具体的系统调用，-e trace=xxx即可。但有时候我们要跟踪一类系统调用，比如所有和文件名有关的调用、所有和内存分配有关的调用。

如果人工输入每一个具体的系统调用名称，可能容易遗漏。于是strace提供了几类常用的系统调用组合名字。

-e trace=file     跟踪和文件访问相关的调用(参数中有文件名)  
-e trace=process  和进程管理相关的调用，比如fork/exec/exit\_group  
-e trace=network  和网络通信相关的调用，比如socket/sendto/connect  
-e trace=signal    信号发送和处理相关，比如kill/sigaction  
-e trace=desc  和文件描述符相关，比如write/read/select/epoll等  
-e trace=ipc 进程见同学相关，比如shmget等

绝大多数情况，我们使用上面的组合名字就够了。实在需要跟踪具体的系统调用时，可能需要注意C库实现的差异。

比如我们知道创建进程使用的是fork系统调用，但在glibc里面，fork的调用实际上映射到了更底层的clone系统调用。使用strace时，得指定-e trace=clone, 指定-e trace=fork什么也匹配不上。