Linux运维故障排查

Linux系统故障解决方法：重视报错提示信息->查阅日志文件(系统日志文件/var/log,和程序日志)->分析和定位问题->解决问题

Linux系统无法启动：通过单用户修复(如root密码忘记了，磁盘坏了或者、/etc/fstab配置错误等问题)。如果单用户都不行，就只能使用关盘修复。

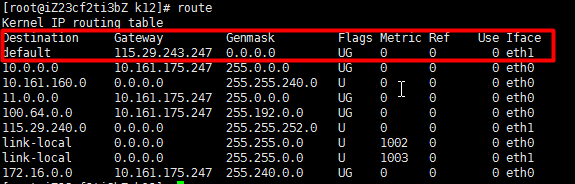
# 网络故障修复

## 检查网络硬件问题

检查网络故障，首先要排除网络硬件设备是否存在问题，比如网线，网卡集线器，路由器，交换机等是否正常工作。

## 检查网卡是否正常工作

通过lsmod,ifconfig命令检查网卡是否正常加载，如果通过ifconfig可以显示网卡接口的配置信息说明网卡驱动程序检测到网络设备，网卡加载正常。通过route检查路由器是否设置正确。



## 检查DNS解析文件是否设置正确

在Linux系统中有两个文件用来指定系统到哪里寻找相关解析域名的库：分别是文件/etc/host.conf和/etc/nsswitch.conf文件用于指定系统如何解析主机名，Linux通过域名解析库来获取主机名对应的ip地址。

/etc/host.conf 文件指定如何解析主机名，Linux 通过解析器库来获得主机名对应的IP:

|  |
| --- |
| order hosts bind  multi on |

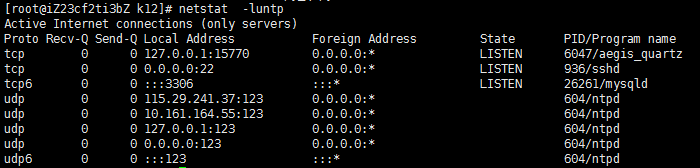
* "order" 指定主机名查询顺序，其参数为用逗号隔开的查找方法，支持的查找方法为bind、 hosts 和nis ，分别代表DNS 、/etc/hosts 和NIS， 这里规定先查询"/etc/hosts"文件然后再使用DNS 来解析域名。
* "trim" 表明当通过DNS 进行地址到主机名的转换时，域名将从主机名中被裁剪掉，trim 可以被多个域包含多次，对/etc/hosts 和NIS 查询方法不起作用，注意在/etc/hosts 和NIS 表中主机名是被适当地(有或没有全域名)列出的。
* "multi" 指定是否"/etc/hosts"文件中指定的主机可以有多个地址，值为on 表示允许，拥有多个IP 地址的主机一般称为具有多个网络界面。
* "nospoof " 指是否允许对该服务器进行IP 地址欺骗，值为on 表示不允许。IP欺骗是一种攻击系统安全的手段，通过把IP 地址伪装成别的计算机，来取得其它计算机的信任。
* "alert" 当nospoof 指令为on 时，alert 控制欺骗的企图是否用syslog 工具进行记录，值为on 表示使用，缺省值为off。
* "rccorder" 如果被设置为on ，所有的查询将被重新排序，所以在同一子网中的主机将首选被返，回缺省值为off。

/etc/nsswitch.conf 文件是由Sun 公司开发并用于管理系统中多个配置文件查找的顺序，它比/etc/host.conf 文件提供了更多的功能。/etc/nsswitch.conf 中的每一行或者是注释(以#号开头)或者是一个关键字后跟冒号和一系列要试用的有顺序的方法。每一个关键字是在/etc/目录可以被/etc/nsswitch.conf 控制的/etc 文件的名字。下面是可以被包含的关键字：

|  |
| --- |
| aliases 邮件别名;  passwd 系统用户;  group 用户组;  shadow 隐蔽口令;  hosts 主机名和I P 地址;  networks 网络名和号;  protocols 网络协议;  services 端口号和服务名称;  ethers 以太网号;  rpc 远程进程调用的名称和号;  netgroup 网内组; |

## 检查服务是否打开

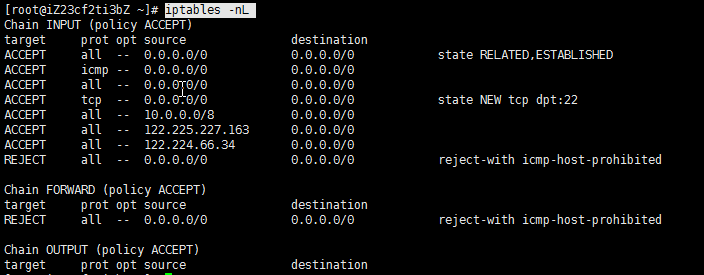
netstat -luntp



## 检查访问权限是否打开

### 检查防火墙iptables状态

iptables –nL

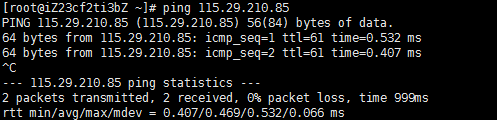


### 检查SELinux是否打开

getenforce



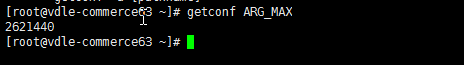
## 检查局域网主机是否联机正常



# 常见故障解决

## "Argument list too long"错误解决

问题产生的原因：当linux系统中尝试传递太多参数给一个系统命令的时候，会出现"Argument list too long"错误。查看这个限制可以使用命令getconf ARG\_MAX查看。

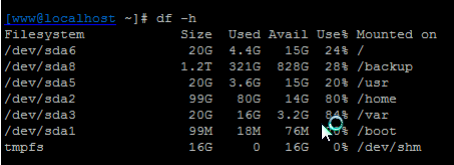


解决方法：使用find命令帅选，然后执行rm –rf

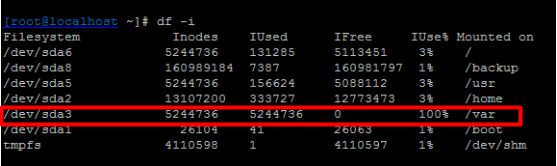
|  |
| --- |
| find /var/spool/clientmqueue -type f -print -exec rm -f {} \; |

## No space left on device

问题产生的原因：既然错误提示与磁盘空间有关，那就深入研究下关于磁盘空间的问题，在 Linux 系统中 对磁盘空间的占用分为三个部分：第一个是物理磁盘空间，第二个是 inode 节点所占用的磁 盘空间，第三个是 Linux 用来存放信号量的空间，而平时接触较多的是物理磁盘空间，对第 二个和第三个空间的问题接触较少。



通过输出可以看出不是物理磁盘空间的问题，接着检查是否是 inode 节点耗尽的问题，通过执行"df -i"查看系统可用的 inode 节点，说明inode用完。



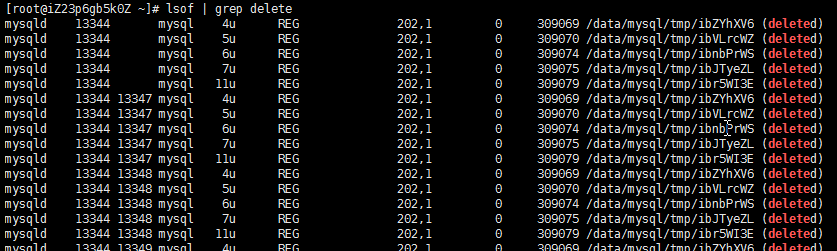
解决方法：把环境中不需要的文件清除掉。

## 文件删除但空间不释放

问题产生的原因：一个文件在文件系统中的存放分为两个部分：数据部分和指针部分，指针位于文件系统的 meta-data 中，在将数据删除后，这个指针就从 meta-data 中清除了，而数据部分存储在磁盘中。在将数据对应的指针从 meta-data 中清除后，文件数据部分占用的空间就可以被覆盖并写入新的内容，之所以在出现删除 access\_log 文件后，空间还没释放，就是因为 httpd 进

程还在一直向这个文件写入内容，导致虽然删除了 access\_log 文件，但是由于进程锁定，文件对应的指针部分并未从 meta-data 中清除，而由于指针并未被删除，系统内核就认为文件并未被删除。

解决方法：以获取一个已经被删除但仍然被应用程序占用的文件列表



然后执行echo " " >/tmp/access\_log 释放空间

## "Too many open file"

问题产生的原因：系统可用的文件描述符不够，可以通过ulimit –n查看系统打开最大文件描述符的数量。



ulimit命令的使用: ulimit [options] [limit]

参数 含义

* -a 显示当前系统所有的 limit 资源信息
* -H 设置硬资源限制，一旦设置不能增加
* -S 设置软资源限制，设置后可以增加，但是不能超过硬资源设置
* -c 最大的 core 文件的大小，以 blocks 为单位
* -f 进程可以创建文件的最大值，以 blocks 为单位
* -d 进程最大的数据段的大小，以 Kbytes 为单位
* -m 最大内存大小，以 Kbytes 为单位
* -n 可以打开的最大文件描述符的数量
* -s 线程栈大小，以 Kbytes 为单位
* -p 管道缓冲区的大小，以 Kbytes 为单位
* -u 用户最大可用的进程数
* -v 进程最大可用的虚拟内存，以 Kbytes 为单位
* -t 最大 CPU 占用时间，以秒为单位
* -l 最大可加锁内存大小，以 Kbytes 为单位

解决方法：

方法一：在环境变量中(.bashrc或者.bash\_profile)配置"ulimit -n 65535"

方法二：使用配置/etc/security/limits.conf配置

格式：<Domain> <type> <item> <value>

* Domain： 表示用户或组的名字，还可以使用 \* 作为通配符，表示任何用户或用户组。
* type ：表示限制的类型，可以有两个值，soft 和 hard，分别表示软、硬资源限制。
* Item： 表示需要限定的资源名称，常用的有 nofile、cpu、stack 等。分别表示最大打开句柄数、占用的 cpu 时间、最大的堆栈大小。
* value 表示限制各种资源的具体数值。

除了 limits.conf 文件之外，还有一个/etc/security/limits.d 目录，可以将资源限制创建一 个文件放到这个目录中，默认系统会首先去读取这个目录下的所有文件，然后才去读取 limits.conf 文件。在所有资源限制设置完成后，退出 shell 终端，再次登录 shell 终端后，ulimit 设置即可自动生效。

## "could not bind to address 0.0.0.0.80"

问题产生的原因：80端口正在被其他进程占用

解决方法：fuser –k tcp/80