

1.Linux相关

1.常用指令

1.0常用指令

cd //进入某个文件夹

ls //目录清单

ll -all //查看所有文件夹

ls -all //查看所有文件夹

“ls” 是显示当前目录下文件，“ls -l” 是显示当前目录下文件详细信息。

1.1 查看磁盘使用情况

df -h

```
root@izbp1hx8v6l7adot5p1xh2z:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G   0    3.9G   0% /dev
tmpfs           799M  3.3M   796M   1% /run
/dev/vda1       40G   11G   27G   29% /
tmpfs           3.9G  4.0K   3.9G   1% /dev/shm
tmpfs           5.0M   0    5.0M   0% /run/lock
tmpfs           3.9G   0    3.9G   0% /sys/fs/cgroup
/dev/vdb1       197G  357M  187G   1% /mnt/data1
tmpfs           799M   0    799M   0% /run/user/0
tmpfs           799M   0    799M   0% /run/user/1001
root@izbp1hx8v6l7adot5p1xh2z:/#
```

1.2对文件夹进行操作

cp directory1/directory2(需要粘贴的地址) //复制文件/文件夹

mkdir //创建文件夹

mkdir -p //创建多级文件夹

rm xxx //删除文件夹

rm -r //删除多级文件夹，递归是以树的形式

mv directory1/directory2 //移动文件夹，或者重命名文件夹

du -sh /path // 查看具体文件夹大小，例如下图：

```
[root@VM_0_6_centos ~]# du -sh /root
120M    /root
[root@VM_0_6_centos ~]#
```

1.3 查看内存使用情况(相当于任务管理器)

top

1.4 查看进程

ps -a //查看所有进程

ps -ef //用标准的格式显示进程的(显示cmd)

ps -ef|grep xxxx //查看指定进程,一般查看pid,配合kill杀掉进程
ps -ef|grep xxxx | wc -l //查看并且统计
kill -9 //强杀某个进程, 强行关闭

1.5 查看当前路径

pwd

1.6 网络相关指令

ifconfig //查看网络情况, 比如IP地址
ifconfig -a

netstat -a //查看当前所有网络进程
netstat -nap|grep xxx 查看特定端口号的进程

1.7 查看文件夹的文件内容/日志

\$ cat filename //通常是用于观看某个文件的内容的; 一次显示整个文件
\$ more //一次显示一页

tail -f xxx //动态查看日志末尾
tail -100 xxx //查看文档最后100行
head -n 100 xxx//查看文档前100行

1.8 awk命令

awk '{print \$1,\$4}' xxx.log #默认使用空格和Tab分隔符
awk -F, '{print \$1,\$4}' xxx.log #-F指定分隔符

1.9 其他常用指令

chmod 744 xxxx //修改文件或者目录访问权限

3.source sh ./ bash 区别

1. source

在当前shell内去读取、执行a.sh, 而a.sh不需要有“执行权限”

2. sh/bash

都是打开一个subshell去读取、执行a.sh, 而a.sh不需要有“执行权限”

3 ./

打开一个subshell去读取、执行a.sh, 但a.sh需要有“执行权限”

二.操作系统相关

1.死锁

死锁的起因是多个线程之间相互等待对方而被永远暂停（处于非Runnable）。

死锁的产生必须满足如下四个必要条件：

- **资源互斥：**一个资源每次只能被一个线程使用
- **请求与保持条件：**一个线程因请求资源而阻塞时，对已获得的资源保持不放
- **不剥夺条件：**线程已经获得的资源，在未使用完之前，不能强行剥夺
- **循环等待条件：**若干线程之间形成一种头尾相接的循环等待资源关系

如何避免死锁的发生

- **粗锁法：**使用一个粒度粗的锁来消除“请求与保持条件”，缺点是会明显降低程序的并发性能并且会导致资源的浪费。
- **锁排序法：（必须回答出来的点）**
 1. 如果必须在不同的操作中控制多个锁，试图在所有方法中以相同的顺序锁定它们。
 2. 然后，按照相反的顺序释放锁，并将锁及其解锁封装在一个类中。这样就不会在整个代码中分布与同步相关的代码。

<https://bbs.csdn.net/topics/250018320>

<https://blog.csdn.net/nicolastsuei/article/details/84747125>

指定获取锁的顺序，比如某个线程只有获得A锁和B锁，才能对某资源进行操作，在多线程条件下，如何避免死锁？

A：通过指定锁的获取顺序，比如规定，只有获得A锁的线程才有资格获取B锁，按顺序获取锁就可以避免死锁。这通常被认为是解决死锁很好的一种方法。

2.线程和进程

2.1线程和进程的区别和关系

1. **线程是程序执行的最小单位（资源调度的最小单位）**

进程是资源分配的最小单位；

2. 而线程是共享进程中的数据，使用相同的地址空间，因此CPU切换一个线程的花费远比进程要小很多，同时创建一个线程的开销也比进程要小很多。

进程有自己的独立地址空间，每启动一个进程，系统就会为它分配地址空间，建立数据表来维护代码段、堆栈段和数据段，这种操作非常昂贵。

3. 线程之间的通信更方便，同一进程下的线程共享全局变量、静态变量等数据，而进程之间的通信需要以通信的方式（IPC）进行。不过如何处理好同步与互斥是编写多线程程序的难点。

4. 但是多进程程序更健壮，多线程程序只要有一个线程死掉，整个进程也死掉了，而一个进程死掉并不会对另外一个进程造成影响，因为进程有自己独立的地址空间。