1. tar命令用于给Linux进行打包。Tar参数分为2中，一种是命令参数，一种是选项参数。指定tar命令时要先指定命令参数，再指定选项参数。

命令参数：-c 创建一个新的打包文件。-x对打包文件进行解压操作。-t查看打包文件的内容。-v 显示操作过程。

常用选项参数：-f filename指定生成的tar包文件名（. tar.gz便是这里指定的，而不是使用-z自动生成的后缀）. -z使tar打包的同时使用gzip进行压缩。

例如：

tar cvf FileName.tar DirName # 为DirName打包成FileName.tar文件，显示打包过程并且指定生成tar包名

tar zcvf FileName.tar.gz DirName # 为DirName打包并且压缩成FileName.tar.gz文件，显示打包过程并且指定生成tar.gz包名

tar xvf FileName.tar # 解包FileName.tar

tar zxvf FileName.tar.gz # 解压FileName.tar.gz

2. 要将程序放在Linux的后台执行，并且忽略挂起信号，一直运行。nohup command >/dev/null 2>&1 &

3. 有些时候我们在Linux中通过service xxx start/stop去启动或者停止服务。其实执行service的时候，Linux 回到/etc/init.d/下面寻找服务的bash脚本。只要在/etc/init.d/下面创建一个bash脚本，那么便在 linux中创建了该服务。在Bash脚本中，我们可以去具体执行当收到start参数的时候执行带有参数的哪个目录下的具体命令，当收到stop时执行带有参数的哪个目录下的具体命令。

4. 承接3点，当我们在/etc/init.d/下面创建了服务以后。还可以使用chkconfig命令去使命令自动开机启动。

5. Linux命令cat /etc/fstab可以查看Linux下面挂载情况。

6. Crontab 中的命令在系统重启以后还会生效。

7. 在Linux图形化界面中，在某个窗口中，右键启动控制台后，控制台当前所在的目录(pwd)就是该窗口的所在目录。

8. 当我们从Linux服务器上拉下一个文件，然后修改，然后再次上传的时候，执行权限会丢失。要重新赋予。

9. 查看Linux服务器中所装的Java版本java -version

10.使用cat /etc/issue查看Ubuntu的系统版本

11. Debian, Ubuntu的包安装命令是apt-get。

RedHat Enterprise(收费产品), Fedora以及CentOS（免费，是Redhat Enterprise的社区版）的包安装命令是yum.

MacOs则是使用brew,不使用apt-get也不使用yum.

12. update 是更新 /etc/apt/sources.list 和 /etc/apt/sources.list.d 中列出的源的地址,这样才能获取到最新的软件包

upgrade 是升级已安装的所有软件包，升级之后的版本就是本地地址里的，因此，在执行 upgrade 之前一定要执行 update, 这样才能更新到最新的

apt-get的源设置在/etc/apt/sources.list中

(sources.list的配置有一个点注意，有时候里面的地址会过期。一个比较经典的例子就是，此处Ubuntu的release版本是14.0.此时安装上Ubuntu以后，source\_list中的地址是有效的，可以使用apt-get.但是2年以后，由于此release不再被supported了，那么当时指定的mirror地址还是存在的，只是该发行版不再支持了，所以当apt-get以这次发行版再去拿时便失效。失效以后有两种方法，第一种是更新系统，然后再apt-get,或者是找支持旧版本的镜像。Ubuntu对于不再支持的release版本，会搬到/old-releases.ubuntu.com/中。)

apt-get的配置在/etc/apt/apt.conf中

P.s 需要注意的是如果一个进程执行apt-get的时候，会产生一个apt-get的锁。此时其他apt-get将不会允许执行，在terminal上手动执行apt-get的时候会报错，无法获取所文件。解决的方法为：

Kill the process

If pervious installation process stuck in middle or not responding, use following commands to kill the process

sudo ps aux | grep apt

sudo kill processnumber

sudo kill -9 processnumber

Remove the lock file

If nothing works, remove the lock files manually with following commands

sudo rm /var/lib/apt/lists/lock

sudo rm /var/cache/apt/archives/lock

sudo rm /var/lib/dpkg/lock

13. Telnet和SSH相比，Telnet是不安全的协议，因为Telnet在传输数据的时候，使用的是明文传输，包括数据密码等等。

14.sudo指令是root在/etc/sudooer指定以后，允许某些用户使用sudo指令去执行一些需要root权限的操作。需要注意的是在 sudo 的时候，环境变量在默认的设置下会被reset, 而不是跟随当前用户的环境变量。这个我们可以通过修改/etc/sudoer来改变，里面有个域env\_reset指定

15. top命令是用于查看linux系统的硬件使用情况，例如CPU,内存等。但是一般如果不加入选项，那么top会一直运行。如果想达到只输出此刻的信息则-n次数，例如-n1为只输出一次，也就是只循环一次。然后我们还可以拼接命令 | grep来达到获取某一进行某一时刻所使用的CPU率的目的。

16. 一台linux服务器主要有两个地方进行DNS的设置。

第一个是/etc/hosts文件，里面直接指定了域名以及IP地址的映射

第二个是/etc/resolve.conf, 里面指定了域名服务器的地址。

【基本介绍】  
resolv.conf配置DNS客户，它包含了主机的域名搜索顺序和DNS服务器的地址，每一行应包含一个关键字和一个或多个的由空格隔开的参数。  
  
【基本配置】

Linux代码  [收藏代码](javascript:void())

1. [root@sparkVM salt]# cat /etc/resolv.conf
2. # Generated by NetworkManager
3. domain localdomain
4. search localdomain
5. nameserver xxx.xxx.xxx.xxx

nameserver　表明DNS服务器的IP地址。可以有很多行的nameserver，每一个带一个IP地址。在查询时就按nameserver在本文件中的顺序进行，且只有当第一个nameserver没有反应时才查询下面的nameserver。  
  
domain　声明主机的域名。很多程序用到它，如邮件系统；当为没有域名的主机进行DNS查询时，也要用到。如果没有域名，主机名将被使用，  
  
search　它的多个参数指明域名查询顺序。当要查询没有域名的主机，主机将在由search声明的域中分别查找。domain和search不能共存；如果同时存在search和domain, 则后面出现的将会被使用，忽略前面的指定过。例如先指定search再指定domain则search将失效，domain生效。如果先指定domain再指定search，则search生效，domain失效。  
  
sortlist 允许将得到域名结果进行特定的排序。它的参数为网络/掩码对，允许任意的排列顺序。

17. 开启服务器的SMTP服务为/etc/init.d/postfix start

SMTP服务器配置在/etc/postfix/main.cf，其中的inet\_interfaces字段用于设置是否允许远程连接

18. 有时候使用telnet连接服务器以后一些退格(backspace)等会乱码，这样可以使用其他的连接方式，例如ssh等方式进入。

19. yum -y upgrade和yum -y update 区别

yum -y upgrade    升级后系统和软件配置不做修改

yum -y update    升级后系统和软件配置文件更新

yum info yum查看yum的版本

yum clean all 清除yum的缓存

20. service --status-all 用于展示出所有服务器中正在运行的服务。

21.在linux中定时任有两种配置方式。一种是通过crontab -e来进入黑窗口编辑，一种是直接修改文件。定时任务有两种分级，一种是特定用户模式，一种是记录所有用户模式。用户模式可以直接编辑文件：/var/spool/cron(注意这个文件的权限一般设为600,如果设为777，则有可能无法工作)或者通过crontab -e来进行编辑，crontab -e 的结果最后会反应到文件中。如果是对所有用户编辑，则需要直接编辑/etc/crontab文件。

22.

输入"uname -a ",可显示电脑以及操作系统的相关信息

输入"cat /proc/version",说明正在运行的内核版本

输入"cat /etc/os-release", 显示的是发行版本信息

23. source 命令， 使[Shell](http://www.51testing.com/?uid-225738-action-viewspace-itemid-206878)读入指定的Shell程序文件并依次执行文件中的所有语句

24.

du的功能：`du` reports the amount of disk space used by the specified files and for each subdirectory (of directory arguments). with no arguments,`du` reports the disk space for the current directory。

du，disk usage, 是通过搜索文件来计算每个文件的大小然后累加，du能看到的文件只是一些当前存在   
的，没有被删除的。他计算的大小就是当前他认为存在的所有文件大小的累加和

du [-abcDhHklmsSx][-L <符号连接>][-X <文件>][--block-size][--exclude=<目录或文件>][--max-depth=<目录层数>][--help][--version][目录或文件]

du常用的选项：

　　-h：以人类可读的方式显示, 以kb, mb, GB形式显示

　　-a：显示目录占用的磁盘空间大小，还要显示其下目录和文件占用磁盘空间的大小

-s：仅显示总大小，不显示各个子目录以及文件大小

-S: 显示目录大小时，不包含子目录大小

　　-c：显示几个目录或文件占用的磁盘空间大小，还要统计它们最终的总和

　　--apparent-size：显示目录或文件自身的大小

　　-l ：统计硬链接占用磁盘空间的大小

-L：统计符号链接所指向的文件占用的磁盘空间大小

--max-depth=<目录层数> 超过指定层数的目录后，予以忽略。

展示当前目录前10个占用磁盘空间最大的10个文件夹：

du -a | sort -n -r | head -n 10

展示当前目录所有目录的大小:

du -sh \*

展示某一目录下文件以及文件夹大小：

du -ah --max-depth=1 [文件夹]

df，disk free，通过文件系统来快速获取空间大小的信息

-i 只显示inodes信息，服务器在磁盘空间占满或者inodes占满的情况下都会当机

25.

scp是secure copy的简写,建立在SSH基础上使用SSH密钥，用于在Linux下进行远程拷贝文件的命令，和它类似的命令有cp，不过cp只是在本机进行拷贝不能跨服务器，而且scp传输是加密的。可能会稍微影响一下速度。当你服务器硬盘变为只读 read only system时，用scp可以帮你把文件移出来。

scp [可选参数] file\_source file\_target

-r 递归复制文件夹

例如：

从本地服务器复制文件到远程服务器：

scp local\_file remote\_username@remote\_ip:remote\_folder

从远程服务器复制文件到本地服务器：

scp [root@192.168.120.204:/opt/soft/nginx-0.5.38.tar.gz /opt/soft/](mailto:root@192.168.120.204:/opt/soft/nginx-0.5.38.tar.gz%20/opt/soft/)

🡪

使用SSH,或者Scp的时候，默认情况下我们需要输入远端机器的密码。例如本地A主机SSH/SCP远端主机B,需要输入远端主机B的密码。我们可以通过将本地主机A的公钥放入远端主机的authorized key中，便可以实现本地主机A无需密码SSH登录远端主机B, 无需密码SCP远端主机B。需要注意，这是单向的无密码认证。主机B认识主机A所以A登录B无需密码，如果要B主机无密码登录A，则需要做相同事情。

26.

find /your\_dir/ -maxdepth 1 -type d -mtime +3 -exec rm -rf {} \;

本命令可以删除 [3] 天前的所有目录。

其中：

your\_dir - 目标目录

-maxdepth - 最大目录层次，我要从第一级目录开始删除，1就可以了

-type d - 操作对象为目录， f 代表文件

-mtime +3 - 3天前的目录

-exec rm -rf {} \; - 执行命令的标准格式

配合脚本，就可以自动在服务器上清理旧的临时文件了。

27.

Git的可视化软件SourceTree. 为开源软件，可以在mac以及window下运行，提高关于git操作的可视化界面。从而我们不需要总使用命令行对远端的库进行clone, push, pull等操作， 我们可以直接在界面上操作。

28.

Linux中合法命名的字符为-, \_, 数字 字母，避免使用空格以及”:”

29.

Linux中每个用户有自己的主目录，例如stack用户默认用户目录在/home/stack中。如果在/home/stack/下建立.bashrc文件，则用户登录时会自动执行/home/stack/.bashrc文件。我们可以在此文件建立一些命令别名，则stack用户登陆以后便可以使用这些命令别名，则作为其他用户登录时没有这些命令别名。

30.

Crontab的设置会根据用户的不同而不同，使用root用户进行crontab -e 的时候编辑的是root用户下的定时任务。使用其他用户进行crontrab -e的时候，是其他用户的定时任务。

31.

Linux wc命令用于计算字数。利用wc指令我们可以计算文件的Byte数、字数、或是列数，若不指定文件名称、或是所给予的文件名为"-"，则wc指令会从标准输入设备读取数据。

wc -l 只显示行数

wc -w 只显示字数

wc -c 只显示bytes数

32.

Linux下的内存管理命令

free 展示内存情况 -g以GB单位显示， -h以人比较容易读取的格式显示

cat /proc/meminfo meminfo文件是内存文件，free中显示的信息实际上由此文件显示。但是输出的格式不如free指令输出的信息友好。

内存交换空间(Swap Space): swap space 是**磁盘**上的一块区域，可以是一个分区，也可以是一个文件。所以具体的实现可以是 swap 分区也可以是 swap 文件。当系统物理内存吃紧时，Linux 会将内存中不常访问的数据保存到 swap 上，这样系统就有更多的物理内存为各个进程服务，而当系统需要访问 swap 上存储的内容时，再将 swap 上的数据加载到内存中，这就是常说的换出和换入。交换空间可以在一定程度上缓解内存不足的情况，但是它需要读写磁盘数据，所以性能不是很高。

Buff/Cache: 通过把磁盘中的数据缓存到物理内存中，把对磁盘的访问变为对物理内存的访问。当内核要读一个文件 (比如 /etc/hosts) 时，它会先检查这个文件的数据是不是已经在页高速缓存中了。如果在，就放弃访问磁盘，直接从内存中读取。这个行为称为缓存命中。如果数据不在缓存中，就是未命中缓存，此时内核就要调度块 I/O 操作从磁盘去读取数据。然后内核将读来的数据放入页高速缓存中。

Free/Available内存定义：

free 是真正尚未被使用的物理内存数量。至于 available 就比较有意思了，它是从应用程序的角度看到的可用内存数量。Linux 内核为了提升磁盘操作的性能，会消耗一部分内存去缓存磁盘数据，就是我们介绍的 buffer 和 cache。所以对于内核来说，buffer 和 cache 都属于已经被使用的内存。当应用程序需要内存时，如果没有足够的 free 内存可以用，内核就会从 buffer 和 cache 中回收内存来满足应用程序的请求。所以从应用程序的角度来说，available = free + buffer + cache

free指令输出示例：

total used free shared buffers cached

Mem: 62 62 0 0 0 15

-/+ buffers/cache: 46 16

Swap: 73 11 62

总内存的62GB， 输出时使用了62GB, 0GB的空闲内存。在62GB使用的内存中，他们都在 buffer/cache中，其中46GB是使用的16GB的cache/buffer空闲。

所以buffers/cache used + buffers/cache free = Mem Used.

Mem used + Mem free = Mem total

buffers/cache used + buffers/cache free + Mem free = Mem total

33. 显示Linux系统中，各进程占用的 CPU以及内存

ps axu | head -n 10 # a 显示现行终端机下的所有程序，包括其他用户的程序. u以用户为主的格式来显示程序状况。X 显示所有程序，不以终端机来区分。

通过上述指令可以输出进程的用户，进程号，CPU利用率，内存利用率，启动进程的命令。我们再通过命令拼接可以对指令列以大小的顺序进行排序，达到输出前10各占用CPU或者内存最多的进程。

ps aux | sort -k4nr | head -n 10 # 占用内存前10个进程

其中 k表示对某一列排序， 4表示对第四列排序，n表示依照数值大小默认进行从小到大排序，r表示进行倒序，以从大到小进行排序。由于ps aux的第4列是内存所以这将输出按照占用内存输出占用最多内存的前十个进程。

ps aux | sort -k3nr | head -n 10 # 占用CPU前10个进程

其中 k表示对某一列排序， 3表示对第三列排序，n表示依照数值大小默认进行从小到大排序，r表示进行倒序，以从大到小进行排序。由于ps aux的第3列是CPU利用率所以这将输出按照占用CPU输出占用最多CPU的前十个进程。

34.

在系统中除了内存将被耗尽的时候可以清缓存以外，我们还可以使用下面这个文件来人工触发缓存清除的操作：

第一步: resync (执行sync同步数据，防止数据或操作丢失（重要），将未写的系统缓冲区写到磁盘中。包含已修改的 i-node、已延迟的块 I/O 和读写映射文件)

第二步：echo 3 > /proc/sys/vm/drop\_caches （释放缓存）

35./tmp目录的自动清理机制

Ubuntu默认的tmp目录下的文件每次重启后自动删除。可以修该/etc/default/rcS文件中的TMPTIME字段内容而改变为不自动删除。

Redhat默认中/etc/cron.daily/tmpwatch将清除/tmp下30天没有访问的文件。

36.

首先需要区分Linux内核与Linux发行版

* Linux内核是Linux操作系统的核心, 负责硬件管理, 比如管理内存、管理磁盘（文件系统）、管理CPU (进程)等等...
* Linux发行版是在Linux内核的基础上添加了一些工具软件, 比如图形界面、函数库、软件包管理系统等等...

CentOS与Ubuntu是不同的Linux发行版, 它们都是基于Linux内核， 只是添加的工具软件不同。比如， 他们的软件包管理系统不同， CentOS使用yum命令安装软件， 而Ubuntu使用apt-get命令安装软件。

因此CentOS与Ubuntu的内核是相同的(版本可能不同), 只是所安装的软件不同， 即文件系统不同。

37.

Linux $() 是命令行结果替换，它会在命令执行中开启一个subshell然后将运行的结果放入命令当中。

例如：docker rm $(docker ps -aq)

38.

查找本目录下含有带有hello关键字的文件

find . -name '\*' |xargs grep 'hello'

39.

Linux AutoFS（也就是AutoMount）

在Linux中使用挂载可以有多种方式。

方式1：使用mount指令临时挂载。当系统重启的时候，挂载盘将消失。

方式2：使用/etc/fstab配置挂载，当系统重启时候，挂载盘会被重新挂载。

方式3：使用AutoFS. AutoFS和/etc/fstab的区别在于，AutoFS将在用户访问挂载盘的时候进行挂在，当用户不访问挂载盘一段时间后，将自动unmount实现动态挂载。而/etc/fstab将不会根据用户活动进行动态变化。

P.s Unmount指令无法去除正在使用的设备

40.

Linux中运行一个命令的时候，Linux将从环境变量PATH中寻找。并且如果找到命令则不再寻找。

例如：PATH=/usr/local/bin:/usr/bin:/bin

当运行Python的时候，将从/usr/local/bin中寻找，如果没有再从/usr/bin中寻找如果没有再从/bin中寻找。

41.

在一个文件夹中新建文件，删除文件，建立link连接等会更新文件夹的最新修改时间。但是在该文件夹中修改其中的文件不会更新文件夹的最新修改时间。

42.

netstat 命令：

netstat命令用于显示与IP、TCP、UDP和ICMP协议相关的统计数据，一般用于检验本机各端口的网络连接情况。netstat是在内核中访问网络及相关信息的程序，它能提供TCP连接，TCP和UDP监听，进程内存管理的相关报告。

其中命令选项为：

-t 显示tcp协议端口

-u 显示udp协议端口

-l 显示监听状态套接字

-p 显示使用该端口的进程号

43.

Linux的file命令，用于辨识文件类型。

例如：

file zipcheck.pl 将返回结果：zipcheck.pl: a /usr/bin/perl script text executable

file zipdiff\_README.txt将返回结果：zipdiff\_README.txt: ASCII text

44.