# Shell面试题

## 一、命令实操题

### 1、用sed修改test.txt的23行test为tset；

|  |
| --- |
| sed -i ‘23s/test/tset/g’ test.txt |

### 2、查看/web.log第25行第三列的内容。

|  |
| --- |
| sed –n ‘25p’ /web.log | cut –d “ ” –f3  head –n25 /web.log | tail –n1 | cut –d “ ” –f3  awk –F “ ” ‘NR==23{print $3}’ /web.log |

### 3、删除每个临时文件的最初三行。

|  |
| --- |
| sed -i ‘1,3d’ /tmp/\*.tmp |

### 4、脚本编程：求100内的质数。

|  |
| --- |
| #!/bin/bash  i=1  while [ $i -le 100 ];do  ret=1  for (( j=2;j<$i;j++ ));do  if [ $(($i%$j)) -eq 0 ];then  ret=0  break  fi  done  if [ $ret -eq 1 ];then  echo -n "$i "  fi  i=$(( i+1 ))  done |

### 5、晚上11点到早上8点之间每两个小时查看一次系统日期与时间，写出具体配置命令

|  |
| --- |
| echo 1 23,1-8/2 \* \* \* root /tmp/walldate.sh >> /etc/crontab |

### 6、编写个shell脚本将当前目录下大于10K的文件转移到/tmp目录下

|  |
| --- |
| #!/bin/bash  fileinfo=($(du ./\*))  length=${#fileinfo[@]}  for((i=0;i<$length;i=$(( i+2 ))));do  if [ ${fileinfo[$i]} -le 10 ];then  mv ${fileinfo[$(( i+1 ))]} /tmp  fi  done |

### 7、如何将本地80端口的请求转发到8080端口，当前主机IP为192.168.2.1

|  |
| --- |
| /sbin/iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to 192.168.2.1:8080  /sbin/iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to 8080 |

### 8、在11月份内，每天的早上6点到12点中，每隔2小时执行一次/usr/bin/httpd.sh 怎么实现

|  |
| --- |
| echo "1 6-12/2 \* \* \* root /usr/bin/httpd.sh >> /etc/crontab" |

### 9、在shell环境如何杀死一个进程？

|  |
| --- |
| ps aux | grep | cut -f? 得到pid  cat /proc/pid  kill pid |

### 10、在shell环境如何查找一个文件？

|  |
| --- |
| find / -name abc.txt |

### 11、在shell里如何新建一个文件？

|  |
| --- |
| touch ~/newfile.txt |

### 12、linux下面的sed和awk的编写

|  |
| --- |
| 1） 如何显示文本file.txt中第二大列大于56789的行？  awk -F "," '{if($2>56789){print $0}}' file.txt  2） 显示file.txt的1,3,5,7,10,15行？  sed -n "1p;3p;5p;7p;10p;15p" file.txt  awk 'NR==1||NR==3||NR==5||…||NR=15{print $0}' file.txt  3） 将file.txt的制表符，即tab，全部替换成"|"  sed -i "s#\t#\|#g" file.txt |

### 13、把当前目录（包含子目录）下所有后缀为“.sh”的文件后缀变更为“.shell”

|  |
| --- |
| #!/bin/bash  str=`find ./ -name \\*.sh`  for i in $str  do  mv $i ${i%sh}shell  done |

### 14、编写shell实现自动删除50个账号功能，账号名为stud1至stud50

|  |
| --- |
| #!/bin/bash  for((i=1;i<=50;i++));do  userdel stud$i  done |

### 15、请用Iptables写出只允许10.1.8.179 访问本服务器的22端口。

|  |
| --- |
| /sbin/iptables -A input -p tcp -dport 22 -s 10.1.8.179 -j ACCEPT  /sbin/iptables -A input -p udp -dport 22 -s 10.1.8.179 -j ACCEPT  /sbin/iptables -P input -j DROP |

### 16、在shell中变量的赋值有四种方法，其中，采用name=12的方法称（ A ） 。

|  |
| --- |
| A直接赋值 B使用read命令  C使用命令行参数 D使用命令的输出 |

### 17、有文件file1

|  |
| --- |
| 1) 查询file1里面空行的所在行号  grep -n ^$ file1  2) 查询file1以abc结尾的行  grep abc$ file1  3) 打印出file1文件第1到第三行  head -n3 file1  sed "3q" file1  sed -n "1,3p" file1 |

### 18、假设有一个脚本scan.sh，里面有1000行代码，并在vim模式下面，请按照如下要求写入对应的指令

|  |
| --- |
| 1） 将shutdown字符串全部替换成reboot  :%s/shutdown/reboot/g  2） 清空所有字符  :%d  3） 不保存退出  q! |

### 19、1到10数字相加，写出shell脚本

|  |
| --- |
| #!/bin/bash  j=0  for((i=1;i<=10;i++));do  j=$[j+i ]  done  echo $j |

### 20、常见shell有哪些？缺省的是哪个？

|  |
| --- |
| /bin/sh /bin/bash /bin/ash /bin/bsh /bin/csh /bin/tcsh /sbin/nologin |

### 21、Shell循环语句有哪些？

|  |
| --- |
| for、while、until |

### 22、用SHELL模拟LVS，脚本怎么写

|  |
| --- |
| /sbin/iptable -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to 192.168.1.11-192.168.1.12 |

### 23、找出系统内大于50k，小于100k的文件，并删除它们。

|  |
| --- |
| #!/bin/bash  file=`find / -size +50k -size -100k`  for i in $file;do  rm -rf $i  done |

### 24、脚本（如：目录dir1、dir2、dir3下分别有file1、file2、file2，请使用脚本将文件改为dir1\_file1、dir2\_file2、dir3\_file3）

|  |
| --- |
| #!/bin/bash  file=`ls dir[123]/file[123]`  for i in $file;do  mv $i ${i%/\*}/${i%%/\*}\_${i##\*/}  done |

### 25、将A、B、C目录下的文件A1、A2、A3文件，改名为AA1、AA2、AA3.使用shell脚本实现。

|  |
| --- |
| #!/bin/bash  file=`ls [ABC]/A[123]`  for i in $file;do  mv $i ${i%/\*}/A${i#\*/}  done |

### 26、如何获取文件的最后一行？

|  |
| --- |
| tail-1 |

### 27、如何获取文件的第一行？

|  |
| --- |
| head-1 |

### 28、如何获取变量长度？

|  |
| --- |
| ${#variable} |

### 29、统计出一个文件夹下大小大于7MB的文件夹

|  |
| --- |
| find . -size +7M -type d #在当前目录下查找 |

然后，计数要用什么呢，可以用wc，这里用到的只有计算行数，因为find把符合条件的文件夹筛选出来后，要统计文件的个数，其实就是统计行数，

|  |
| --- |
| find . -size +7M -type d | wc -l |

这样，题目基本上就解决了。当然，还可以这样写

|  |
| --- |
| find . -size +7M -type d | awk 'END{print NR}' |

## 二、每个脚本开始的 #!/bin/sh 或 #!/bin/bash 表示什么意思 ?

这一行说明要使用的shell。#!/bin/bash表示脚本使用 /bin/bash。对于python脚本，就是 #!/usr/bin/python

## 三、& 和 && 有什么区别

& - 希望脚本在后台运行的时候使用它

&& - 当前一个脚本成功完成才执行后面的命令/脚本的时候使用它

## 四、' 和 " 引号有什么区别？

单引号字符串的特点：

* 单引号里的任何字符都会原样输出，单引号字符串中的变量是无效的；
* 单引号字串中不能出现单引号（对单引号使用转义符后也不行）。

双引号的特点：

* 双引号里可以有变量
* 双引号里可以出现转义字符

## 五、[ a == b ] 和 [ a -eq b ] 有什么区别

[ a == b ]：用于字符串比较

[ a -eq b ]：用于数字比较

## 六、= 和 == 有什么区别

=： 用于为变量赋值

==：用于字符串比较

## 七、如何看当前Linux系统有几颗物理CPU和每颗CPU的核数？

|  |
| --- |
| cat /proc/cpuinfo|grep -c 'physical id'  cat /proc/cpuinfo|grep -c 'processor' |

## 八、如何查看当前系统都有哪些进程？

|  |
| --- |
| ps -aux 或者ps -elf |

## 九、如何查看系统都开启了哪些端口？

|  |
| --- |
| netstat -lnp |

## 十、如何查看网络连接状况？

|  |
| --- |
| netstat -an |

## 十一、什么是Linux

Linux是一套免费使用和自由传播的类Unix操作系统，是一个基于POSIX和Unix的多用户、多任务、支持多线程和多CPU的操作系统。它能运行主要的Unix工具软件、应用程序和网络协议。它支持32位和64位硬件。Linux继承了Unix以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

## 十二、Unix和Linux有什么区别？

Linux和Unix都是功能强大的操作系统，都是应用广泛的服务器操作系统，有很多相似之处，甚至有一部分人错误地认为Unix和Linux操作系统是一样的，然而，事实并非如此，以下是两者的区别。

**1. 开源性**

Linux是一款开源操作系统，不需要付费，即可使用；Unix是一款对源码实行知识产权保护的传统商业软件，使用需要付费授权使用。

**2. 跨平台性**

Linux操作系统具有良好的跨平台性能，可运行在多种硬件平台上；Unix操作系统跨平台性能较弱，大多需与硬件配套使用。

**3. 可视化界面**

Linux除了进行命令行操作，还有窗体管理系统；Unix只是命令行下的系统。

4. 硬件环境

Linux操作系统对硬件的要求较低，安装方法更易掌握；Unix对硬件要求比较苛刻，安装难度较大。

**5. 用户群体**

Linux的用户群体很广泛，个人和企业均可使用；Unix的用户群体比较窄，多是安全性要求高的大型企业使用，如银行、电信部门等，或者Unix硬件厂商使用，如Sun等。

相比于Unix操作系统，Linux操作系统更受广大计算机爱好者的喜爱，主要原因是Linux操作系统具有Unix操作系统的全部功能，并且能够在普通PC计算机上实现全部的Unix特性，开源免费的特性，更容易普及使用！

## 十三、什么是 Linux 内核？

Linux 系统的核心是内核。内核控制着计算机系统上的所有硬件和软件，在必要时分配硬件，并根据需要执行软件。

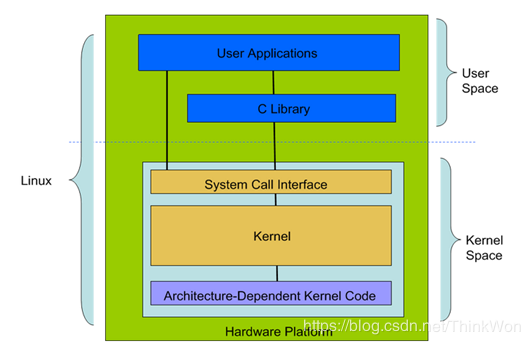
* 系统内存管理
* 应用程序管理
* 硬件设备管理
* 文件系统管理

## 十四、Linux的基本组件是什么？

就像任何其他典型的操作系统一样，Linux拥有所有这些组件：内核，shell和GUI，系统实用程序和应用程序。Linux比其他操作系统更具优势的是每个方面都附带其他功能，所有代码都可以免费下载。

## 十五、Linux 的体系结构

从大的方面讲，Linux 体系结构可以分为两块：



用户空间(User Space) ：用户空间又包括用户的应用程序(User Applications)、C 库(C Library) 。

内核空间(Kernel Space) ：内核空间又包括系统调用接口(System Call Interface)、内核(Kernel)、平台架构相关的代码(Architecture-Dependent Kernel Code) 。

## 十六、为什么 Linux 体系结构要分为用户空间和内核空间的原因？

1. 现代 CPU 实现了不同的工作模式，不同模式下 CPU 可以执行的指令和访问的寄存器不同。

2. Linux 从 CPU 的角度出发，为了保护内核的安全，把系统分成了两部分。

用户空间和内核空间是程序执行的两种不同的状态，我们可以通过两种方式完成用户空间到内核空间的转移：

1）系统调用；

2）硬件中断。

## 十七、BASH和DOS之间的基本区别是什么？

BASH和DOS控制台之间的主要区别在于3个方面：

1. BASH命令区分大小写，而DOS命令则不区分;

2. 在BASH下，/ character是目录分隔符，\作为转义字符。在DOS下，/用作命令参数分隔符，\是目录分隔符

3. OS遵循命名文件中的约定，即8个字符的文件名后跟一个点，扩展名为3个字符。BASH没有遵循这样的惯例。

## 十八、Linux 开机启动过程？

1、主机加电自检，加载 BIOS 硬件信息。

2、读取 MBR 的引导文件(GRUB、LILO)。

3、引导 Linux 内核。

4、运行第一个进程 init (进程号永远为 1 )。

5、进入相应的运行级别。

6、运行终端，输入用户名和密码。

## 十九、Linux 使用的进程间通信方式？

管道(pipe)、流管道(s\_pipe)、有名管道(FIFO)。

信号(signal) 。

消息队列。

共享内存。

信号量。

套接字(socket) 。

## 二十、Linux 有哪些系统日志文件？

比较重要的是 /var/log/messages 日志文件。

该日志文件是许多进程日志文件的汇总，从该文件可以看出任何入侵企图或成功的入侵。另外，如果胖友的系统里有 ELK 日志集中收集，它也会被收集进去。

## 二十一、Linux系统安装多个桌面环境有帮助吗？

通常，一个桌面环境，如KDE或Gnome，足以在没有问题的情况下运行。尽管系统允许从一个环境切换到另一个环境，但这对用户来说都是优先考虑的问题。有些程序在一个环境中工作而在另一个环境中无法工作，因此它也可以被视为选择使用哪个环境的一个因素。

## 二十二、什么是交换空间（swap）？

交换空间是Linux使用的一定空间，用于临时保存一些并发运行的程序。当RAM没有足够的内存来容纳正在执行的所有程序时，就会发生这种情况。

## 二十三、什么是root帐户？

root帐户就像一个系统管理员帐户，允许你完全控制系统。你可以在此处创建和维护用户帐户，为每个帐户分配不同的权限。每次安装Linux时都是默认帐户。

## 二十四、什么是LILO？

LILO是Linux的引导加载程序。它主要用于将Linux操作系统加载到主内存中，以便它可以开始运行。

## 二十五、什么是BASH？

BASH是Bourne Again SHell的缩写。它由Steve Bourne编写，作为原始Bourne Shell（由/ bin / sh表示）的替代品。它结合了原始版本的Bourne Shell的所有功能，以及其他功能，使其更容易使用。从那以后，它已被改编为运行Linux的大多数系统的默认shell。

## 二十六、什么是CLI？

命令行界面（英语\*\*：command-line interface\*\*，缩写]：CLI）是在图形用户界面得到普及之前使用最为广泛的用户界面，它通常不支持鼠标，用户通过键盘输入指令，计算机接收到指令后，予以执行。也有人称之为字符用户界面（CUI）。

通常认为，命令行界面（CLI）没有图形用户界面（GUI）那么方便用户操作。因为，命令行界面的软件通常需要用户记忆操作的命令，但是，由于其本身的特点，命令行界面要较图形用户界面节约计算机系统的资源。在熟记命令的前提下，使用命令行界面往往要较使用图形用户界面的操作速度要快。所以，图形用户界面的操作系统中，都保留着可选的命令行界面。

## 二十七、什么是GUI？

图形用户界面（Graphical User Interface，简称 GUI，又称图形用户接口）是指采用图形方式显示的计算机操作用户界面。

图形用户界面是一种人与计算机通信的界面显示格式，允许用户使用鼠标等输入设备操纵屏幕上的图标或菜单选项，以选择命令、调用文件、启动程序或执行其它一些日常任务。与通过键盘输入文本或字符命令来完成例行任务的字符界面相比，图形用户界面有许多优点。

## 二十八、开源的优势是什么？

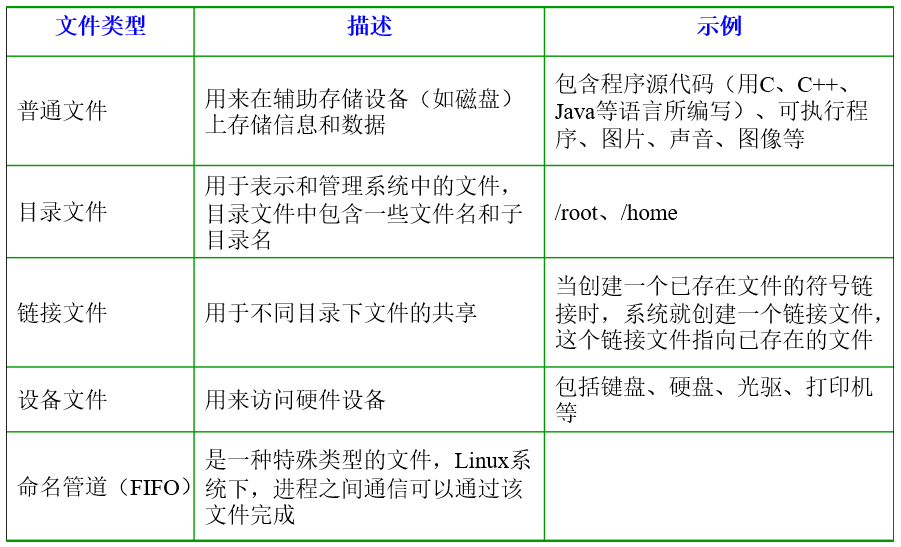
开源允许你将软件（包括源代码）免费分发给任何感兴趣的人。然后，人们可以添加功能，甚至可以调试和更正源代码中的错误。它们甚至可以让它运行得更好，然后再次自由地重新分配这些增强的源代码。这最终使社区中的每个人受益。

## 二十九、简述 Linux 文件系统？

在 Linux 操作系统中，所有被操作系统管理的资源，例如网络接口卡、磁盘驱动器、打印机、输入输出设备、普通文件或是目录都被看作是一个文件。

也就是说在 Linux 系统中有一个重要的概念\*\*：一切都是文件\*\*。其实这是 Unix 哲学的一个体现，而 Linux 是重写 Unix 而来，所以这个概念也就传承了下来。在 Unix 系统中，把一切资源都看作是文件，包括硬件设备。UNIX系统把每个硬件都看成是一个文件，通常称为设备文件，这样用户就可以用读写文件的方式实现对硬件的访问。

Linux 支持 5 种文件类型，如下图所示：文件类型

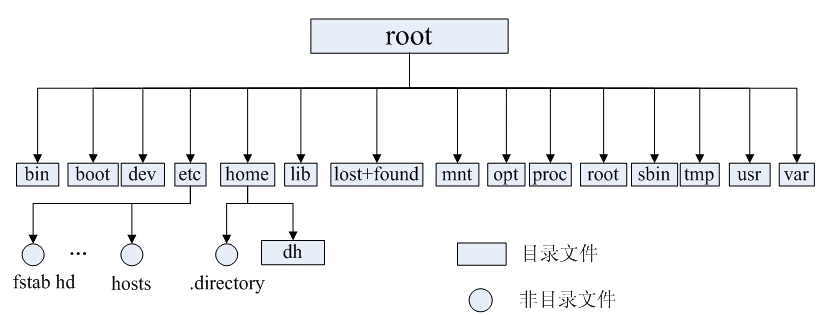


## 三十、Linux 的目录结构是怎样的？

这个问题，一般不会问。更多是实际使用时，需要知道。

Linux 文件系统的结构层次鲜明，就像一棵倒立的树，最顶层是其根目录：

Linux的目录结构



常见目录说明：

* /bin：存放二进制可执行文件(ls,cat,mkdir等)，常用命令一般都在这里；
* /etc：存放系统管理和配置文件；
* /home：存放所有用户文件的根目录，是用户主目录的基点，比如用户user的主目录就是/home/user，可以用~user表示；
* \*\*/usr \*\*：用于存放系统应用程序/opt：额外安装的可选应用程序包所放置的位置。一般情况下，我们可以把tomcat等都安装到这里；
* /proc：虚拟文件系统目录，是系统内存的映射。可直接访问这个目录来获取系统信息；
* /root：超级用户（系统管理员）的主目录（特权阶级o）；
* /sbin: 存放二进制可执行文件，只有root才能访问。这里存放的是系统管理员使用的系统级别的管理命令和程序。如ifconfig等；
* /dev：用于存放设备文件；
* /mnt：系统管理员安装临时文件系统的安装点，系统提供这个目录是让用户临时挂载其他的文件系统；
* /boot：存放用于系统引导时使用的各种文件；
* \*\*/lib \*\*：存放着和系统运行相关的库文件 ；
* /tmp：用于存放各种临时文件，是公用的临时文件存储点；
* /var：用于存放运行时需要改变数据的文件，也是某些大文件的溢出区，比方说各种服务的日志文件（系统启动日志等。）等；
* /lost+found：这个目录平时是空的，系统非正常关机而留下“无家可归”的文件（windows下叫什么.chk）就在这里。

## 三十一、什么是硬链接和软链接？

**1）硬链接**

由于 Linux 下的文件是通过索引节点(inode)来识别文件，硬链接可以认为是一个指针，指向文件索引节点的指针，系统并不为它重新分配 inode 。每添加一个一个硬链接，文件的链接数就加 1 。

不足：1）不可以在不同文件系统的文件间建立链接；2）只有超级用户才可以为目录创建硬链接。

**2）软链接**

软链接克服了硬链接的不足，没有任何文件系统的限制，任何用户可以创建指向目录的符号链接。因而现在更为广泛使用，它具有更大的灵活性，甚至可以跨越不同机器、不同网络对文件进行链接。

不足：因为链接文件包含有原文件的路径信息，所以当原文件从一个目录下移到其他目录中，再访问链接文件，系统就找不到了，而硬链接就没有这个缺陷，你想怎么移就怎么移；还有它要系统分配额外的空间用于建立新的索引节点和保存原文件的路径。

实际场景下，基本是使用软链接。总结区别如下：

* 硬链接不可以跨分区，软件链可以跨分区。
* 硬链接指向一个 inode 节点，而软链接则是创建一个新的 inode 节点。
* 删除硬链接文件，不会删除原文件，删除软链接文件，会把原文件删除。

## 三十二、Linux 性能调优都有哪几种方法？

1、Disabling daemons (关闭 daemons)。

2、Shutting down the GUI (关闭 GUI)。

3、Changing kernel parameters (改变内核参数)。

4、Kernel parameters (内核参数)。

5、Tuning the processor subsystem (处理器子系统调优)。

6、Tuning the memory subsystem (内存子系统调优)。

7、Tuning the file system (文件系统子系统调优)。

8、Tuning the network subsystem（网络子系统调优)。

## 三十三、linux的权限标识

**权限范围：**

u ：目录或者文件的当前的用户

g ：目录或者文件的当前的群组

o ：除了目录或者文件的当前用户或群组之外的用户或者群组

a ：所有的用户及群组

**权限代号：**

r ：读权限，用数字4表示

w ：写权限，用数字2表示

x ：执行权限，用数字1表示

- ：删除权限，用数字0表示

s ：特殊权限