

# Table of Contents

|                |       |
|----------------|-------|
| Introduction   | 1.1   |
| 目录             | 1.2   |
| 基础知识篇          | 1.3   |
| 学习Linux的经验与技巧  | 1.3.1 |
| Linux系统基本结构    | 1.3.2 |
| Linux常用命令及使用技巧 | 1.3.3 |
| 服务器搭建篇         | 1.4   |
| 系统管理篇          | 1.5   |
| 性能调试篇          | 1.6   |
| 虚拟化与集群应用篇      | 1.7   |

# Linux-Learn

学习Linux文档

## 基础知识篇

# 学习Linux的经验与技巧

## 1.1 Linux在各领域发展的现状与趋势

### 1.1.1 Linux与开源软件

Linux是一种自由和开放源代码的类UNIX操作系统，该操作系统的内核由林纳斯·托瓦兹在1991年首次发布，之后，再加上用户空间的应用程序，就成为Linux操作系统。严格来讲，Linux只是操作系统内核本身，但通常采用“Linux内核”来表达该意思。而Linux则常用来指基于Linux内核的完整操作系统，它包括GUI组件和许多其他实用工具。GNU通用公共许可协议（GNU General Public License, GNU GPL或GPU，是一个广泛使用的自由软件许可协议条款。最初由理查德·斯托曼为GNU计划而撰写，GPL给出了计算机程序自由软件的定义，任何基于GPL软件开发衍生的产品在发布时必须采用GPL许可证方式，且必须公开源代码。Linux是自由软件和开放源代码软件发展中最著名的例子。只要遵循GNUGPL，任何个人和机构都可以自由地使用Linux的所有底层源代码，也可以自由地修改和再发布。随着Linux操作系统飞速发展，在Linux上各种集成的开源软件和实用工具也得到了应用和普及，因此，Linux也成为开源软件的代名词。

### 1.1.2 Linux在服务器领域的发展

据权威部门统计，目前Linux在服务器领域已经占据75%的市场份额。同时，Linux在服务器市场的迅速崛起，已经引起全球IT产业的高度关注，并以强劲的势头成为服务器操作系统领域中的中坚力量。

### 1.1.3 Linux在桌面领域的发展

近年来，特别在国内市场，Linux桌面操作系统的发展趋势非常迅猛。国内如中标麒麟Linux、红旗Linux、深度Linux等系统软件厂商都推出的Linux桌面操作系统，目前已经在政府、企业、OEM等领域得到了广泛应用。另外，SUSE、Ubuntu也相继推出了基于Linux的桌面系统，特别是Ubuntu Linux，已经积累了大量社区用户。但是，从系统的整体功能、性能来看，Linux桌面系统与Windows系列相比还有一定的差距，主要表现在系统易用性、系统管理、软硬件兼容性、软件的丰富程度等方面。

### 1.1.4 Linux在移动嵌入式领域的发展

Linux的低成本、强大的定制功能以及良好的可移植性，使得Linux在嵌入式系统方面也得到广泛应用，目前Linux已广泛应用于手机、平板电脑、路由器、电视和电子游戏机等领域。在移动设备上广泛使用的Android操作系统就是建立在Linux内核之上的。目前，Android已经成为全球最流行的智能手机操作系统，据2015年权威部门最新统计，Android操作系统的全球市场份额已达84.6%。此外，思科公司在网络防火墙和路由器中也使用了定制的Linux，阿里云也开发了套基于Linux的操作系统“YunOS”，可用于智能手机、平板电脑和网络电视；常见的数字视频录像机、舞台灯光控制系统等都在逐渐采用定制版本的Linux来实现，而这一切均归功于Linux与开源的力量。

### 1.1.5 Linux在云计算 / 大数据领域的发展

互联网产业的迅猛发展，促使云计算、大数据产业的形成并快速发展，云计算、大数据作为一个基于开源软件的平台，Linux占据了核心优势。据Linux基金会的统计，86%的企业已经使用Linux操作系统进行云计算、大数据平台的构建。目前，Linux已开始取代UNIX成为最受青睐的云计算、大数据平台操作系统。

## 1.2 选择适合自己的Linux行版

初学者在学习Linux之前，需要有一个明确的方向，选择一个适合自己的Linux系统至关重要。下面就分类介绍Linux发行版。

## 1.2.1 常见的Linux发行版

### 1 Red Hat Linux

目前Red Hat 分为两个系列：由Red Hat公司提供收费技术支持和更新的Red Hat Enterprise Linux，以及由社区开发的免费的Fedora Core。

### 2 Fedora Core

对于用户而言，Fedora是一套功能完备、更新迅速的免费操作系统，因此，个人领域的应用，例如开发、体验新功能等，可选择此发行版本。

### 3 Red Hat Enterprise Linux

Red Hat Enterprise Linux的版本都基于Fedora。大约发布六个新版本的Fedora后就会有新版本的Red Hat Enterprise Linux发布，因此，Red Hat Enterprise Linux大约3年发布一个新版本。

### 4 CentOS

CentOS全名为“社区企业操作系统”（Community Enterprise Operating System）。它来自于Red Hat依照开放源代码规定发布的源代码编译而成，由于RHEL是商业产品，因此必须将Red Hat的所有Logo改成自己的CentOS标识，这就产生了CentOS操作系统。两者的不同在于，CentOS并不包含封闭源代码软件。因此，CentOS不但可以自由使用，而且能享受CentOS提供的长期免费升级和更新服务。这是CentOS的一个很大优势。CentOS从Red Hat的源代码包来构建，它的版本号有两部分：主要版本和次要版本，主要版本和次要版本号分别对应于Red Hat的主要版本与更新包。2014年CentOS宣布与Red Hat合作，但CentOS将会在新的委员会下继续运作，并不受Red Hat的影响。这个策略表明CentOS后续发展将由Red Hat作为强有力的支持。

### 5 SuSE Linux

SUSE Linux现在在欧洲Linux市场中占有将近80%的份额，大部分关键应用都是建立在SUSE Linux下的。而由于SUSE多次易主，再加上SUSE在中国的营销模式问题，现在SUSE在中国的Linux市场份额并不大，但是这些并不影响SUSE Linux高可靠性与稳定性的事实。随着SUSE的发展，相信SUSE Linux在中国的应用会越来越多。

### 6 Ubuntu Linux

Ubuntu（中文谐音为“友帮拓”、“优般图”、“乌班图”）是一个以桌面应用为主的Linux操作系统，基于Debian GNU/Linux, Ubuntu旨在为一般用户提供一个主要由自由软件构建而成的最新的同时又相当稳定的操作系统。Ubuntu具有庞大的社区力量，用户可以方便地从社区获得帮助。

## 7 发行版总结

纵观Linux的各个发行版，Linux发行版本无非是朝着这两个方面发展，一是服务器市场，二是桌面市场。以Ubuntu Linux为代表的Linux发行版走的是桌面市场路线，虽然它们给用户带来很多惊喜，更新也很快，但是由于桌面市场有Windows这样强劲的对手，因此Linux桌面市场的发展不容乐观。目前Ubuntu Linux也开始向企业级服务器市场发力。以Red Hat系列版本为代表的Linux发行版现在主要面向企业级Linux的服务器市场，重点开发Linux的企业版本，其他的发行版（例如，国产红旗Linux、中标麒麟Linux等）都重点投入在了Linux服务器市场。Linux两大发

布厂商现在都走了Linux服务器市场的路线，可见Linux作为企业级服务器有着巨大的发展前途。据权威部门统计，Linux在服务器市场的占有率每年都在持续上升。其实Linux的很多应用都是针对Linux服务器的，本书的讲述也是主要针对Linux在服务器下的各种应用展开的。

## 1.2.2 初学者入门首选——CentOS系列

在了解了Linux几个主要发行版本后，我们就找到了为何选择CentOS作为初学者入门学习的理由了。

- CentOS现在拥有庞大的网络用户群体，网络Linux资源基本80%都是基于CentOS发行版的。如果在学习过程中遇到任何问题，在网络中可以较容易地搜索到解决方案。
- CentOS系列版本可以轻松获得。可以从CentOS官网或者163开源、SOHU开源、阿里云开源站下载CentOS各个版本的安装介质。如果是第一次接触Linux，那么建议先安装FedoraCore。FedoraCore的安装简单，对硬件支持很好，界面也很华丽，同时也可以体验Linux的最新功能。如果对Linux有一定的了解，需要深入学习，建议使用CentOS发行版系统。
- CentOS应用范围广，具有典型性和代表性。现在基本所有的互联网公司后台服务器都采用CentOS作为操作系统，可以说学会了CentOS，不仅能迅速融入企业的工作环境，还能触类旁通，其他类似的Linux发行版也能很快掌握。同时，现在周围学习Linux的用户一般也都是以CentOS为主的，这样交流方便，学习中出现问题，更容易得到解决。最主要的是，CentOS的安装和使用也是最简单的，因此基本上不会在“装系统”上浪费过多时间。

## 1.2.3 桌面平台首选——Ubuntu Linux

说到Linux桌面市场，Ubuntu Linux几乎占据了桌面Linux的半壁江山，Ubuntu Linux是主打Linux桌面之最，界面美观，简洁而不失华丽。如果想在Linux下进行娱乐休闲，Ubuntu Linux绝对是首选。Ubuntu的安装非常人性化，只须按照提示一步一步进行。Ubuntu被誉为对硬件支持最好、最全面的Linux发行版之一，在其他发行版上无法使用的或者在默认配置时无法使用的许多硬件，在Ubuntu上都能轻松安装、使用。因此用户可以像安装Windows一样轻松地安装Ubuntu，尽情体验UbuntuLinux带来的乐趣。

## 1.2.4 企业级应用首选——RHEL/CentOS系列

企业级应用追求的是可靠性和稳定性，这就要求构建企业级应用的系统平台具有高可靠性和高稳定性。企业级Linux的发行版本就可以解决这个问题。

RHEL与Centos两个Linux发行版本，并没有太大差别，所不同的是，RHEL属于商业Linux发行版本，如果要使用RHEL版本，则需要购买商业授权和咨询服务，RedHat公司提供系统的技术支持并提供系统的免费升级。目前Red Hat官网已经不再提供可免费下载的光盘介质，如果需要试用，可通过官网下载有试用时间限制的评估版Linux。而CentOS属于非商业发行版，可以从网上免费下载CentOS各个版本的安装介质，但CentOS并不提供商业支持，当然，使用者也不用负任何商业责任。那么，到底是选择CentOS还是RHEL呢？这取决于你所在公司是否拥有相应的技术力量，如果是单纯的业务型企业，那么建议选购RHEL发行版并购买相应服务，这样可以节省企业的IT管理费用，并可得到专业的技术支持服务。相反，如果企业技术力量比较强大，并且有多年Linux使用经验，那么CentOS发行版将是最佳选择。

# Linux系统基本结构

## 2.1 Linux控制台的使用

## 2.2 系统和硬件

Linux系统查看系统的硬件信息没有Windows那么直观，但是通过命令可以把硬件信息显示的更加清楚。我以我的腾讯云服务器来做示例

- 1 查看系统PCI设备 `lspci`命令列出所有的PCI设备，你如主板，声卡，显卡和网卡等，也会把USB接口设备列出来。

```
# lspci
```

输出结果如下

```
[root@VM_65_115_centos ~]# lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.2 USB controller: Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Red Hat, Inc Virtio network device
00:04.0 SCSI storage controller: Red Hat, Inc Virtio block device
00:05.0 Unclassified device [00ff]: Red Hat, Inc Virtio memory balloon
[root@VM_65_115_centos ~]#
```

下面我们分析一下这个机器有什么类型的参数 Host bridge: 显示的是继承主板设备为Intel Corporation 440FX - 82441FX PMC VGE: 显示的是VGE显卡设备为Cirrus Logic GD 5446 有一个USB接口设备

- 2 查看CPU信息

```
# more /proc/cpuinfo
```

输出结果

```
[root@VM_65_115_centos ~]# more /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 79
model name     : Intel(R) Xeon(R) CPU E5-26xx v4
stepping       : 1
microcode     : 0x1
cpu mhz        : 2394.446
cache size     : 4096 KB
physical id    : 0
siblings       : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx lm constant tsc rep_good nopl eagerfpu pni pclmulqdq ssse3
               : fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch bmi1 avx2 bmi2 rdseed adx xsaveopt
bogomips       : 4788.89
clflush size   : 64
cache alignment : 64
address sizes   : 40 bits physical, 48 bits virtual
power management:

[root@VM_65_115_centos ~]#
```

其中，processor是逻辑处理器的唯一标识符，vendor-id表示处理器类型，如果为英特尔处理器，则字符串是GenuineIntel,physical id表示每一个物理封装的唯一标识符，也就是一个物理CPU,siblings表示位于相同物理封装中的逻辑处理器的数量，core id表示每个内核的唯一标识符，cpu cores表示位于相同物理封装中的内核数量。

在siblings和cpu cores值之间有个对应关系，如果siblings是cpu cores的两倍，则说明系统支持超线程，并且超线程已打开，如果siblings和cpu cores一致，则说明系统不支持超线程，或者超线程未打开。

要查看系统物理CPU的个数，同通过如下命令查看。

```
[root@VM_65_115_centos ~]# cat /proc/cpuinfo | grep "physical id" | sort | uniq | wc -l
```

要查看每个物理CPU中内核的个数，可通过如下命令查看

```
[root@VM_65_115_centos ~]# cat /proc/cpuinfo | grep "cpu cores"
cpu cores          : 1
```

要查看系统所有逻辑CPU个数(所有每个物理CPU中内核的个数加上超线程个数)，可通过如下命令查看。

```
[root@VM_65_115_centos ~]# cat /proc/cpuinfo | grep "processor" | wc -l
1
```

根据上面输出可知，此系统有1个物理cpu，每个物理CPU中有1个内核，并且没有超线程。

- 3 查看系统内存信息

```
[root@VM_65_115_centos ~]# more /proc/meminfo
MemTotal:      1883616 kB
MemFree:       80920 kB
MemAvailable:  1395720 kB
Buffers:       160548 kB
Cached:        1227276 kB
SwapCached:    0 kB
Active:        1156260 kB
Inactive:      486868 kB
Active(anon):  255564 kB
Inactive(anon): 116 kB
Active(file):  900696 kB
Inactive(file): 486752 kB
Unevictable:   0 kB
Mlocked:       0 kB
SwapTotal:     0 kB
SwapFree:      0 kB
Dirty:         520 kB
Writeback:     0 kB
AnonPages:     255344 kB
Mapped:        95456 kB
Shmem:         376 kB
Slab:          126404 kB
SReclaimable:  114840 kB
SUnreclaim:    11564 kB
KernelStack:   2096 kB
PageTables:    6000 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
```

根据这个命令，可以很清楚的知道系统的内存占用情况

- 4 查看磁盘分区信息

```
[root@VM_65_115_centos ~]# fdisk -l

Disk /dev/vda: 53.7 GB, 53687091200 bytes, 104857600 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000c5e30

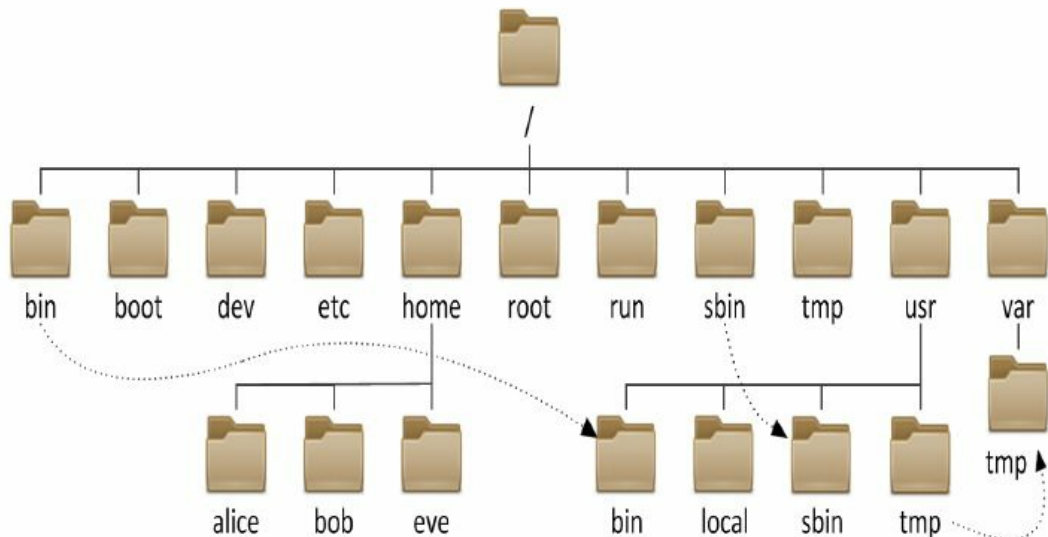
   Device Boot      Start         End      Blocks   Id  System
/dev/vda1  *        2048     104857566     52427759+   83   Linux
```

## 2.3 文件系统结构介绍

### 2.3.1 目录结构

- 1 经典树形目录





整个Linux系统以文件的形式全部存放在根目录下，同时将所有文件分类，分级分层组织在一起，形成了一个梳妆目录结构。

## ● 2 目录功能介绍

**/:** 根目录，位于Linux文件系统目录结构的顶层，一般根目录下只存放目录，不要存放文件，**/etc**、**/bin**、**/dev**、**/lib**、**/sbin**应该和根目录放置在一个分区中。

**/bin**，**/usr/bin**：该目录为命令文件目录，也称为二进制目录。包含了供系统管理员及普通用户使用的重要的linux命令和二进制（可执行）文件，包含shell解释器等。

**/boot**：该目录中存放系统的内核文件和引导装载程序文件，**/boot/vmlinuz**为linux的内核文件，以及**/boot/gurb**。建议单独分区，分区大小100M即可。

**/dev**：设备（device）文件目录，存放linux系统下的设备文件，访问该目录下某个文件，相当于访问某个设备，存放连接到计算机上的设备（终端、磁盘驱动器、光驱及网卡等）的对应文件，包括字符设备和块设备等，常用的是挂载光驱**mount /dev/cdrom/mnt**。

**/etc**：系统配置文件存放的目录，该目录存放系统的大部分配置文件和子目录，不建议在此目录下存放可执行文件，重要的配置文件有**/etc/inittab**、**/etc/fstab**、**/etc/init.d**、**/etc/X11**（X Window系统有关）、**/etc/sysconfig**（与网络有关）、**/etc/xinetd**修改配置文件之前记得备份。该目录下的文件由系统管理员来使用，普通用户对大部分文件有只读权限。

**/home**：系统默认的用户宿主目录，新增用户账号时，用户的宿主目录都存放在此目录下，**~**表示当前用户的宿主目录，**~test**表示用户**test**的宿主目录。建议单独分区，并设置较大的磁盘空间，方便用户存放数据。

**/lib**，**/usr/lib**，**/usr/local/lib**：系统使用的函数库的目录，程序在执行过程中，需要调用一些额外的参数时需要函数库的协助，该目录下存放了各种编程语言库。典型的linux系统包含了C、C++和FORTRAN语言的库文件。**/lib**目录下的库映像文件可以用来启动系统并执行一些命令，目录**/lib/modules**包含了可加载的内核模块，**/lib**目录存放了所有重要的库文件，其他的库文件则大部分存放在**/usr/lib**目录下。

**/lost+found**：在EXT2或EXT3文件系统中，当系统意外崩溃或机器意外关机，产生的一些文件碎片放在这里。在系统启动的过程中**fsck**工具会检查这里，并修复已经损坏的文件系统。有时系统发生问题，有很多的文件被移到这个目录中，可能会用手工的方法来修复，或者移动文件到运来的位置上

**/mnt**，**/media**：**mnt**目录主要用来临时挂载文件系统，为某些设备提供默认挂载点，如**floppy**，**cdrom**。这样当挂载了一个设备如光驱时，就可以通过访问目录**/mnt/cdrom**下的文件来访问相应的光驱上的文件了。

**/opt:** 给主机额外安装软件所摆放的目录。如: FC4使用的Fedora 社群开发软件, 如果想要自行安装新的KDE 桌面软件, 可以将该软件安装在该目录下。以前的 Linux 系统中, 习惯放置在 /usr/local 目录下。

**/proc:** 此目录的数据都在内存中, 如系统核心, 外部设备, 网络状态, 由于数据都存放于内存中, 所以不占用磁盘空间, 比较重要的目录有/proc/cpuinfo、/proc/interrupts、/proc/dma、/proc/ioports、/proc/net/\*等。

**/root:** 系统管理员root的宿主目录, 系统第一个启动的分区为/, 所以最好将/root和/放置在一个分区下。

**/sbin, /usr/sbin, /usr/local/sbin:** 放置系统管理员使用的可执行命令, 如fdisk、shutdown、mount等。与/bin不同的是, 这几个目录是给系统管理员root使用的命令, 一般用户只能"查看"而不能设置和使用。

**/tmp:** 一般用户或正在执行的程序临时存放文件的目录,任何人都可以访问,重要数据不可放置在此目录下。

**/srv:** 服务启动之后需要访问的数据目录, 如www服务需要访问的网页数据存放在/srv/www内。

**/usr:** 应用程序存放目录, /usr/bin 存放应用程序, /usr/share 存放共享数据, /usr/lib 存放不能直接运行的, 却是许多程序运行所必需的一些函数库文件, /usr/local 存放软件升级包, /usr/share/doc 系统说明文件存放目录。

**/usr/share/man:** 程序说明文件存放目录, 使用 man ls时会查询/usr/share/man/man1/ls.1.gz的内容建议单独分区, 设置较大的磁盘空间。

**/var:** 放置系统执行过程中经常变化的文件, 如随时更改的日志文件 /var/log。/var/log/message: 所有的登录文件存放目录。/var/spool/mail: 邮件存放的目录。/var/run: 程序或服务启动后。建议单独分区, 设置较大的磁盘空间。

**/tmp:** 存放临时文件目录, 一些命令和应用程序会用到这个目录。该目录下的所有文件会被定时删除, 以避免临时文件占满整个磁盘。

## 2.3.2 系统核心组成

### 1.内存管理

内存管理能够有效的管理系统的内存, 响应程序对内存的请求, Linux内存管理还支持虚拟内存, 就是Linux在运行程序和某些数据可以超过实际的物理内存, 超过物理内存的这一部分内存是通过对磁盘申请得到的, 一般情况下系统把当前运行的程序保留在内存里边, 到了内存紧缺, 迫不得已的时候就会去跟磁盘交换程序块。

### 2.进程管理

Linux使用基于优先级的进程调度算法来控制进程对CPU的访问, 当某个进程需要被执行的时候, 那么进程调度器就会基于调度算法启动新的程序, Linux支持多任务运行, 比如你在Linux上可以"同时"运行多个程序。这样看起来好像Linux可以同时并发的执行多个任务, 其实不然, 而是在系统运行的时候, 每个进程会得到被cpu执行的时间片, 当某一个进程的时间片执行完后, 调度进程就会去执行另一个进程, 这样对每个进程都进行的快速切换执行, 这样的快速快到让我们感觉好像cpu在同一时间执行多个程序一样。

### 3.进程间通信

在不同的进程之间控制数据的交换和共享, 由于不同的进程之间的进程空间不同, 所以进程间通信会借助内核的中转来实现进程。

### 4.虚拟文件系统

Linux虚拟文件系统隐藏了各种硬件的具体细节, 为所有的设备提供了统一的接口, 使得Linux可以支持多种文件类型, 虚拟文件分为逻辑文件和设备驱动文件, 逻辑文件指的是Linux支持的文件系统, 设备文件指的是为每一种硬件控制器所编写的设备驱动程序模块。

### 5.网络接口

网络接口提供了对各种网络标准的存取实现和各种网络硬件的支持。网络接口可分为网络协议和网络驱动程序。网络协议部分负责实现每一种可能的网络传输协议。网络设备驱动程序则负责与硬件设备通信，每一种可能的硬件设备都有相应的设备驱动程序。

## 2.2 系统和硬件

- 系统关机重启命令如下 `# shutdown -r now`
- 系统关机命令如下 `# shutdown -h now`
- 通过systemctl启动服务 `systemctl start [服务名]`
- 通过systemctl停止服务 `systemctl stop [服务名]`
- 通过systemctl重启服务 `systemctl stop [服务名]`

# Linux常用命令及使用技巧

## 3.1 Linux控制台的使用

### 3.1.1 什么是Shell

shell 本身是一个以C语言编写的程序，是用户和操作系统内核之间通信的桥梁。shell既是一种命令解释程序，又是一种功能强大的解释型程序设计语言。作为命令解释程序，shell解释用户输入的命令，然后提交到内核处理，最后把结果返回给用户。为了加快命令的运行，同时更有效地定制shell程序，shell中定义了一些内置命令，一般把shell自身解释执行的命令称为内置命令。例如，下面将要讲到的cd、pwd、exit和echo等命令，都是属于bash的内置命令。当用户登录系统后，shell以及内置命令就被系统载入到内存，并且一直运行，直到用户退出系统为止。除了内置命令之外，Linux系统上还有很多可执行文件。可执行文件类似于Windows下的.exe文件，这些可执行文件也可以作为shell命令来执行。其实Linux上很多命令都不是shell的内置命令。例如，ls就是一个可执行文件，存放在/bin/ls中。这些命令与shell内置命令不同，只有当它们被调用时，才由系统装入内存执行。最后，shell还是强大的解释型程序设计语言，它定义了各种选项和变量，几乎支持高级程序语言的所有程序结构，如变量、函数、表达式和循环等。利用shell可以编写shell脚本程序，类似于Windows/DOS下的批处理文件，但是shell功能更加完善，更加强大。

### 3.1.2 shell命令的语法分析

shell语法分析是指shell对命令的扫描处理过程，也就是，把命令或者用户输入的内容分解成要处理的各个部分的操作。在Linux系统下，shell语法分析包含很多内容，如重定向、文件名扩展和管道等。

1. shell的命令格式 用户登录系统后，shell命令行启动。shell遵循一定的语法格式将用户输入的命令进行分析解释并传递给系统内核。shell命令的一般格式为：`</br> command [options] {arguments} </br>` 其中，`</br>`
2. command:表示命令的名称
3. options:表示命令的选项
4. arguments:表示命令的参数

在命令行中，选项是包含一个或多个字母的代码，主要用于改变命令的执行方式。一般在选项前面有一个“-”符号，用于区别参数。

1. shell的通配符 通配符主要是为了方便用户对文件或者目录的描述，例如，当用户仅仅需要以“.conf”结尾的文件时，使用通配符就能很方便地实现。各个版本的shell都有通配符，这些通配符是一些特殊字符，用户可以在命令行的参数中使用这些字符，进行文件名或者路径名的匹配。shell将把与命令行中指定的匹配规则符合的所有文件名或者路径名作为命令的参数，然后执行这个命令。bash中常用的通配符有“\*”、“?”、“[]” (1) “\*” 匹配任意一个或多个字符 (2) “?” 匹配任意单一字符 (3) “[]” 匹配任何包含在方括号内的单字符 (4) 通配符的组合使用 `#ls [0~9]?.conf` 这条命令列出当前目录下以数字开头，随后一个是任意字符，接着以“.conf”结尾的所有文件。
2. shell的重定向 Linux下系统打开3个文件，即标准输入、标准输出和标准错误输出。用户的shell将键盘设为默认的标准输入，默认的标准输出和标准错误输出为屏幕。也就是用户从键盘输入命令，然后将结果和错误消息输出到屏幕。所谓的重定向，就是不使用系统默认的标准输入/输出，而是重新指定，因此重定向分为输入重定向、输出重定向和错误输出重定向。要实现重定向就需要了解重定向操作符，shell就是根据重定向操作符来决定重定向操作的。(1) 输入重定向 输入重定向用于改变命令的输入源，利用输入重定向，就可以将一个文件的内容作为命令的输入，而不从键盘输入。用于输入重定向的操作符有“<”和“<<”。

```
[root@VM_65_115_centos bin]# wc<su
75   932 32096
```

这里用WC命令统计输入给它的文件 / etc/inittab的行

数、单词数和字符数。</br> 连有一种输入重定向操作符“<<”这种重定向告诉shell，当前命令的标准输入

```
[root@VM_65_115_centos bin]# wc<< aa
> #defds
> #fdsa
> aa
2 2 13
```

为来自 命令行中一对分隔号之间的内容。 上面的

命令将一对分隔号aa之间的内容作为WC命令的输入。分隔号可以是任意字符。shell将在第一个分隔号后开始读取内容，直到出现另一个分隔号读取结束，然后将内容送给WC命令处理。(2) 输出重定向 输出重定向不是将命令的输出结果在屏幕输出，而是输出到一个指定文件中。在Linux下输出重定向用得很多。例如，某个命令的输出很长，一个屏幕无法显示完毕，这时可以将命令的输出指定到一个文件，然后用 more命令查看这个文件，从而得到命令输出的完整信息。用于输出重定向的操作符有“>”和“>>”。 #ps -ef >ps.txt 这条命令将ps -ef输出的系统运行进程信息全部输入到了ps.txt文件，而不输出到屏幕，可以用more命令查看ps.txt文件中系统运行的进程信息。

```
[root@VM_65_115_centos bin]# ps -ef >ps.txt
[root@VM_65_115_centos bin]# more ps.txt
UID      PID  PPID  C  STIME TTY          TIME CMD
root      1      0  0  Mar23 ?        00:02:15 /usr/lib/systemd/systemd --switched-root --system --deserialize 21
root      2      0  0  Mar23 ?        00:00:00 [kthreadd]
root      3      2  0  Mar23 ?        00:00:24 [ksoftirqd/0]
root      5      2  0  Mar23 ?        00:00:00 [kworker/0:0H]
root      7      2  0  Mar23 ?        00:00:00 [migration/0]
root      8      2  0  Mar23 ?        00:00:00 [rcu_bh]
root      9      2  0  Mar23 ?        00:07:35 [rcu_sched]
root     10      2  0  Mar23 ?        00:00:11 [watchdog/0]
root     12      2  0  Mar23 ?        00:00:00 [kdevtmpfs]
root     13      2  0  Mar23 ?        00:00:00 [netns]
root     14      2  0  Mar23 ?        00:00:00 [khungtaskd]
root     15      2  0  Mar23 ?        00:00:00 [writeback]
root     16      2  0  Mar23 ?        00:00:00 [kintegrityd]
root     17      2  0  Mar23 ?        00:00:00 [bioset]
root     18      2  0  Mar23 ?        00:00:00 [kblockd]
root     19      2  0  Mar23 ?        00:00:00 [md]
root     25      2  0  Mar23 ?        00:00:02 [kswapd0]
root     26      2  0  Mar23 ?        00:00:00 [ksmd]
root     27      2  0  Mar23 ?        00:00:04 [khugepaged]
root     28      2  0  Mar23 ?        00:00:00 [crypto]
root     36      2  0  Mar23 ?        00:00:00 [kthrotld]
root     38      2  0  Mar23 ?        00:00:00 [kmpath_rdacd]
root     39      2  0  Mar23 ?        00:00:00 [kpsmoused]
root     40      2  0  Mar23 ?        00:00:00 [ipv6_addrconf]
root     59      2  0  Mar23 ?        00:00:00 [deferwq]
root     97      2  0  Mar23 ?        00:00:16 [kauditd]
root    229      2  0  Mar23 ?        00:00:00 [ata_sff]
```

#more file1 file2 file3 >file 其中，more命令用于查看文件的内容，上面的命令是将自file2、file2和自file3的内容全部输出到file文件中，类似于文件内容的合并。如果在“>”后面指定的文件不存在，shell就会自动重建一个;如果文件存在，那么这个文件原有的内容将被覆盖；如果不想覆盖存在的文件，可以使用“>>”操作符。

例如 [root@VM\_65\_115\_centos ~]# ls -al /etc/\* /root/install.log 这条命令将 / etc 目录及其子目录下的所有文件信息追加到 / root/install.log 文件的后面。/root/install.log文件原来的内容仍然存在。

(3) 错误重定向 错误重定向和标准输出重定向一样，可以使用操作符“2>”和“2>>”实现对错误输出,例如

```
[root@VM_65_115_centos ~]# tar zxvf text.tar.gz 2> error.txt
```

其中，即是打包命令，可以在屏幕上看到tar的解压过程。如果“text.tar.gz”是个损坏的压缩包，就会把错误消息输出到error.txt文件。

3. shell的管道 管道可以把很多命令连接起来，可以把第1个命令的输入当作第2个命令的输出，第2个命令的输出当作第3个命令的输入，依此类推。因此，管道的作用就是把一个命令的输出当作下一个命令的输入，而不经过程

任何中间文件。 [root@VM\_65\_115\_centos ~]# ls -al /etc/\* | more 这条命令表示将 / etc 目录以及子目录下的所有文件分屏显示。

4. shell中的引用 (1)转义字符“\” [root@VM\_65\_115\_centos ~]# mv abc\?\\* abc 上面是将 abc? \* 重命名为 abc (2) 单引号“'” 如果将字符串放到一对单引号之间，那么字符串中所有字符的特殊含义将被忽略， (3) 双引号“” 双引号的引用与单引号基本相同， 包含在双引号内的大部分特殊字符可以当作普通字符处理，但是仍有一些特殊字符即使用双引号括起来，也仍然保留自己的特殊含义， 比如 “\$”, “\”, “”



5. shell的自动补全命令行 当用户输入某个命令的一部分后，按Tab键，shell就会根据系统环境变量信息提示出与用户输入命令相似的所有命令和文件，

## 3.2 系统管理与维护

### 3.2.1 ls命令

ls命令显示指定工作目录下的内容，列出工作目录所含的文件及子目录。此命令与Windows下的dir类似。另外，Linux也提供了dir命令，用户也可以用dir命令代替ls命令。举例

```
[root@VM_65_115_centos ~]# ls -l /usr
total 112
dr-xr-xr-x.  2 root root 32768 Apr 30 22:53 bin
drwxr-xr-x.  2 root root 4096 Nov  5  2016 etc
drwxr-xr-x.  2 root root 4096 Nov  5  2016 games
drwxr-xr-x. 34 root root 4096 Mar 23 14:51 include
drwxr-xr-x.  7 root root 4096 Mar 26 09:26 Java
dr-xr-xr-x. 29 root root 4096 Mar 23 14:51 lib
dr-xr-xr-x. 42 root root 24576 Mar 26 09:33 lib64
drwxr-xr-x. 22 root root 4096 Mar 23 14:51 libexec
drwxr-xr-x. 14 root root 4096 Mar 26 09:27 local
dr-xr-xr-x.  2 root root 20480 Mar 26 09:33 sbin
drwxr-xr-x. 85 root root 4096 Jan  9 18:23 share
drwxr-xr-x.  4 root root 4096 Jan  9 18:19 src
lrwxrwxrwx.  1 root root    10 Jan  9 18:19 tmp -> ../var/tmp
```

以上命令列

出/root 目录下文件及其子目录

```
[root@VM_65_115_centos ~]# ls -alF
total 88
dr-xr-x---.  6 root root 4096 Apr  2 00:01 ./
dr-xr-xr-x. 19 root root 4096 Apr 30 23:40 ../
-rw-----.  1 root root 31160 Apr 30 23:40 .bash_history
-rw-r--r--.  1 root root   18 Dec 29  2013 .bash_logout
-rw-r--r--.  1 root root  176 Dec 29  2013 .bash_profile
-rw-r--r--.  1 root root  176 Dec 29  2013 .bashrc
drwxr-xr-x.  3 root root 4096 Mar  8 15:46 .cache/
drwxr-xr-x.  3 root root 4096 Mar  8 15:46 .config/
-rw-r--r--.  1 root root  100 Dec 29  2013 .cshrc
drwxr-xr-x.  2 root root 4096 Mar 23 09:59 .oracle_jre_usage/
-rw-----.  1 root root   21 Mar 23 16:07 .rediscli_history
drwx-----.  2 root root 4096 Mar  8 15:45 .ssh/
-rw-r--r--.  1 root root  129 Dec 29  2013 .tcshrc
-rw-----.  1 root root  5106 Apr  2 00:01 .viminfo
```

以上命令显示/root

下的所有文件及其子目录的详细信息，并显示文件类型标记

### 3.2.2 ls命令

显示当前的工作目录，执行pwd指令可立刻得知目前所在的工作目录的绝对路径名称，

```
[root@VM_65_115_centos ~]# pwd
/root
```

### 3.2.3 cd 命令

改变当前工作目录，其用法与Windows下的cd类似。具体的语法格式如下。

```
[root@VM_65_115_centos ~]# cd /usr
[root@VM_65_115_centos usr]#
```

以上命令进入到usr目录下

### 3.2.4 date命令

显示或者修改系统时间与日期。只有超级用户才能用 **date** 命令设置时间，一般用户只能用 **date** 命令显示时间。

```
[root@VM_65_115_centos ~]# date
Tue May 1 11:21:48 CST 2018
[root@VM_65_115_centos ~]# date '+This date now is:%x. time is now:%X. thanks'
This date now is:05/01/2018. time is now:11:21:50 AM. thanks
[root@VM_65_115_centos ~]# date -s 20170715
Sat Jul 15 00:00:00 CST 2017
```

以上命令是用指定的格式显示时间和日期。

以

```
[root@VM_65_115_centos ~]# date '+%Y-%m-%d'
2018-04-17
[root@VM_65_115_centos ~]# date -d "2 days ago" +%Y-%m-%d
2018-04-15
```

上命令为修改系统时间。

显示两天前的时间。

### 3.2.5 passwd命令

```
[root@VM_65_115_centos ~]# passwd
Changing password for user root.
New password:
```

用于设置用户密码

然后根据提示连续两次输入密码即可。

### 3.2.6 SU 命令

SU 命令主要用于改变用户身份 普通用户要成为超级用户，可执行如下命令。 **su -** 根据系统提示输入超级用户密码，即可转变为超级用户。

### 3.2.7 clear命令

clear命令用来清除屏幕信息 **clear**

### 3.2.8 man命令

man命令用来显示指定命令的帮助信息

```
[root@VM_65_115_centos ~]# man ls

LS(1)                                User Commands                                LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default). Sort
  entries alphabetically if none of -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .
  -A, --almost-all
      do not list implied . and ..
  --author
      with -l, print the author of each file
  -b, --escape
```

### 3.2.9 who命令

who命令显示目前登录到系统的用户  
什么运行级别

```
[root@VM_65_115_centos ~]# who -r
run-level 3 2018-03-23 15:40
```

查询系统处于什

```
[root@VM_65_115_centos ~]# who -but
system boot 2018-03-23 15:40
root + pts/0 2018-04-17 00:10 . 15691 (202.100.50.100)
```

显示

系统最近启动日期，以及当前每个用户的登录详情、终端状态

### 3.2.10 w命令

用于显示登录到系统的用户信息

```
[root@VM_65_115_centos ~]# w
00:18:03 up 38 days, 20:09, 1 user, load average: 2.00, 3.54, 3.28
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU WHAT
root      pts/0    202.100.50.100 00:10       3.00s  0.01s  0.00s w
```

•上面第1行

输出显示了当前的系统时间、系统从启动到现在已经运行的时间、登录到系统中的用户数和系统平均负载。平均负载是指在1分钟、5分钟、15分钟内系统负载状况。•USER: 表示登录系统的用户。•TTY: 表示用户使用的TTY名称。•FROM: 表示用户从哪里登录进来，一般显示远程登录主机的E地址或者主机名。•LOGIN@: 用户登录的日期和时间。•IDLE: 表示某个程序上次从终端开始执行到现在所持续的时间。•JCPU: 表示该终端上的所有进程及子进程使用系统的总时间。•PCPU: 当前活动进程使用的系统时间。•WHAT: 当前用户执行的进程名称和选1页。

### 3.2.11 uname命令

用来显示操作系统相关信息



```

[root@VM_65_115_centos ~]# uname -a
Linux VM_65_115_centos 3.10.0-693.el7.x86_64 #1 SMP Tue Aug 22 21:09:27 UTC 2017 x86_64
x86_64 x86_64 GNU/Linux
[root@VM_65_115_centos ~]# uname -m
x86_64
[root@VM_65_115_centos ~]# uname -n
VM_65_115_centos
[root@VM_65_115_centos ~]# uname -s
Linux
[root@VM_65_115_centos ~]# uname -r
3.10.0-693.el7.x86_64

```

-a:显示当前操作系统的全部信息 -m:显示的是系统的CPU类型，是32位还是64位系统 -n:显示的是操作系统的主机名 -s:显示的是操作系统的类型 -r:显示的操作系统的内核版本 图中显示的是一个64位的Linux系统，内核版本为3.10.0-693.el7.x86\_6，主机名称为VM\_65\_115\_centos

### 3.2.12 uptime命令

uptime命令用来输出系统任务队列信息

```

[root@VM_65_115_centos ~]# uptime
 00:30:35 up 38 days, 20:22, 1 user, load average: 0.00, 0.29, 1.46

```

上面的输出表示现在系统时间是 00:30:35，系统已经运行38天零20小时22分钟，目前有1个用户在钱，系统在1分钟、5分钟、15分钟前到现在的平均负载为0.0、0.29、1.46

### 3.2.13 last 命令

列出目前与过去登入系统的用户相关信息。当执行last指令时，它会默认读取位于/var/log目录下名称为wtmp的文件，并把该给文件记录的登入系统的用户名单全部显示出来。

```

[root@VM_65_115_centos ~]# last -a
root      pts/0      Tue Apr 17 00:10   still logged in   202.100.50.100
root      pts/0      Tue May  1 11:20 - 00:10 (-14+-11:-9) 202.100.50.100
root      pts/0      Mon Apr 30 23:38 - 02:41 (03:03) 202.100.50.100
root      pts/0      Mon Apr 30 23:22 - 23:30 (00:07) 202.100.50.100
root      pts/0      Mon Apr 30 23:19 - 23:20 (00:00) 202.100.50.100
root      pts/0      Mon Apr 30 23:09 - 23:19 (00:09) 202.100.50.100
root      pts/0      Mon Apr 30 22:28 - 23:09 (00:40) 202.100.50.100
root      pts/0      Mon Apr 30 16:30 - 17:54 (01:23) 202.100.50.100
root      pts/0      Mon Apr 30 16:29 - 16:30 (00:00) 202.100.50.100
root      pts/0      Mon Apr 30 15:29 - 16:29 (01:00) 202.100.50.100
root      pts/1      Mon Apr 30 15:29 - 15:29 (00:00) 202.100.50.100
root      pts/0      Mon Apr 30 15:28 - 15:29 (00:00) 211.159.185.38
root      pts/0      Fri Apr 27 16:32 - 16:34 (00:01) 222.91.125.254
root      pts/0      Fri Apr 27 16:25 - 16:29 (00:03) 222.91.125.254
root      pts/0      Thu Apr 19 14:09 - 14:09 (00:00) 222.91.125.254
root      pts/0      Thu Apr 19 10:46 - 10:56 (00:09) 222.91.125.254
root      pts/0      Thu Apr 19 10:45 - 10:46 (00:00) 222.91.125.254
root      pts/0      Fri Apr 13 09:48 - 09:48 (00:00) 222.91.125.254
root      pts/1      Thu Apr 12 13:54 - 13:54 (00:00) 222.91.125.254
root      pts/0      Thu Apr 12 13:54 - 13:55 (00:00) 222.91.125.254
root      pts/0      Thu Apr 12 12:41 - 12:41 (00:00) 222.91.125.254
root      pts/0      Thu Apr 12 12:40 - 12:41 (00:00) 222.91.125.254
root      pts/0      Thu Apr 12 10:39 - 10:39 (00:00) 222.91.125.254
root      pts/0      Thu Apr 12 10:39 - 10:39 (00:00) 222.91.125.254

```

### 3.2.14 dmesg命令

显示开机信息。内核会将开机信息存储在系统缓冲区（ring buffer）中，如果开机来不及查看相关信息，可以在开机后利用dmesg命令查看，也可以在 /var/log / 目录中查看dmesg文件。

```
[root@VM_65_115_centos ~]# dmesg -c
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Initializing cgroup subsys cpuacct
[ 0.000000] Linux version 3.10.0-693.el7.x86_64 (builder@kbuilder.dev.centos.org) (gc
c version 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC) ) #1 SMP Tue Aug 22 21:09:27 UTC 2017
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-3.10.0-693.el7.x86_64 root=UUID=2c
04c946-7fee-41c2-a99f-f53e2532e4f7 ro crashkernel=auto console=ttyS0 console=tty0 panic=
5 net.ifnames=0 biosdevname=0
[ 0.000000] e820: BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000007ffdfdf] usable
[ 0.000000] BIOS-e820: [mem 0x00000000007ffdfdf-0x00000000007ffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000007ffffff-0x0000000000ffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000ffffff-0x0000000000ffffff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 2.4 present.
[ 0.000000] DMI: Bochs Bochs, BIOS Bochs 01/01/2011
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[ 0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[ 0.000000] e820: last_pfn = 0x7ffffe max_arch_pfn = 0x400000000
[ 0.000000] MTRR default type: write-back
[ 0.000000] MTRR fixed ranges enabled:
[ 0.000000] 00000-9ffff write-back
[ 0.000000] a0000-bffff uncachable
[ 0.000000] c0000-fffff write-protect
[ 0.000000] MTRR variable ranges enabled:
[ 0.000000] 0 base 00e000000 mask ffe000000 uncachable
[ 0.000000] 1 disabled
```

### 3.2.15 free命令

free 命令用来显示系统内存状态，具体包括系统物理内存、虚拟内存、共享内存和系统缓存。

```
[root@VM_65_115_centos ~]# free -m
              total        used         free       shared    buff/cache   available
Mem:           1839         650          160           0         1028         997
Swap:            0           0            0
```

系统总的物理内存为1839MB,已经使用了650MB，可用内存有160mb。

### 3.2.16 ps命令

```
[root@VM_65_115_centos ~]# ps
  PID TTY          TIME CMD
 15691 pts/0        00:00:00 bash
 17330 pts/0        00:00:00 ps
```

ps命令显示系统进程在瞬间的运行动态  
使用这的进程。 以上命令查看系统目前



```
[root@VM_65_115_centos ~]# ps -ef
UID          PID    PPID  C  STATE TTY          TIME CMD
root         1      0  0  Mar09 ?    00:02:16 /usr/lib/systemd/systemd --switched-root --system --deserialize 21
root         2      0  0  Mar09 ?    00:00:00 [kthreadd]
root         3      2  0  Mar09 ?    00:00:24 [ksoftirqd/0]
root         5      2  0  Mar09 ?    00:00:00 [kworker/0:0H]
root         7      2  0  Mar09 ?    00:00:00 [migration/0]
root         8      2  0  Mar09 ?    00:00:00 [rcu_bh]
root         9      2  0  Mar09 ?    00:07:40 [rcu_sched]
root        10      2  0  Mar09 ?    00:00:11 [watchdog/0]
root        12      2  0  Mar09 ?    00:00:00 [kdevtmpfs]
root        13      2  0  Mar09 ?    00:00:00 [netns]
root        14      2  0  Mar09 ?    00:00:00 [khungtaskd]
root        15      2  0  Mar09 ?    00:00:00 [writeback]
root        16      2  0  Mar09 ?    00:00:00 [kintegrityd]
root        17      2  0  Mar09 ?    00:00:00 [bioaset]
root        18      2  0  Mar09 ?    00:00:00 [kblockd]
root        19      2  0  Mar09 ?    00:00:00 [md]
root        25      2  0  Mar09 ?    00:00:02 [kswapd0]
root        26      2  0  Mar09 ?    00:00:00 [ksmd]
root        27      2  0  Mar09 ?    00:00:04 [khugepaged]
root        28      2  0  Mar09 ?    00:00:00 [crypto]
root        36      2  0  Mar09 ?    00:00:00 [kthrotld]
root        38      2  0  Mar09 ?    00:00:00 [kmpath_rdad]
root        39      2  0  Mar09 ?    00:00:00 [kpsmouse]
root        40      2  0  Mar09 ?    00:00:00 [ipv6_addrconf]
root        59      2  0  Mar09 ?    00:00:00 [deferwq]
root        97      2  0  Mar09 ?    00:00:16 [auditd]
root       229      2  0  Mar09 ?    00:00:00 [ata_sff]
```

以上命令查看系统所有的进程。

### 3.2.17 top命令

top命令提供了对系统处理器状态的实时监控，它能够实时显示系统中各个进程的资源占用状况。该命令可以按照对CPU的使用、内存使用和执行时间对系统任务进程进行排序显示，同时top命令还可以通过交互式命令进行设定显示。

```
top - 00:55:20 up 38 days, 20:47, 2 users, load average: 0.00, 0.01, 0.30
Tasks: 72 total, 2 running, 70 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.0 sy, 0.0 ni, 99.3 id, 0.3 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1883616 total, 160408 free, 669112 used, 1054096 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 1017968 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
   1 root        20   0   43340   3020   1816 S   0.0   0.2    2:16.81 systemd
   2 root        20   0       0       0       0 S   0.0   0.0    0:00.09 kthreadd
   3 root        20   0       0       0       0 S   0.0   0.0    0:24.72 ksoftirqd/0
   5 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 kworker/0:0H
   7 root        rt    0       0       0       0 S   0.0   0.0    0:00.00 migration/0
   8 root        20   0       0       0       0 S   0.0   0.0    0:00.00 rcu_bh
   9 root        20   0       0       0       0 S   0.0   0.0    7:40.77 rcu_sched
  10 root        rt    0       0       0       0 S   0.0   0.0    0:11.65 watchdog/0
  12 root        20   0       0       0       0 S   0.0   0.0    0:00.00 kdevtmpfs
  13 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 netns
  14 root        20   0       0       0       0 S   0.0   0.0    0:00.71 khungtaskd
  15 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 writeback
  16 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 kintegrityd
  17 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 bioaset
  18 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 kblockd
  19 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 md
  25 root        20   0       0       0       0 S   0.0   0.0    0:02.72 kswapd0
  26 root        25   5       0       0       0 S   0.0   0.0    0:00.00 ksmd
  27 root        39  19       0       0       0 S   0.0   0.0    0:04.43 khugepaged
  28 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 crypto
  36 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 kthrotld
  38 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 kmpath_rdad
  39 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 kpsmouse
  40 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 ipv6_addrconf
  59 root         0 -20       0       0       0 S   0.0   0.0    0:00.00 deferwq
  97 root        20   0       0       0       0 S   0.0   0.0    0:16.73 auditd
```

，top的输出可以分为统计信息区和进程信息区两个部分，即前5行为统计信息区，下面几行为进程信息区，下面分别介绍两种信息区。(1)统计信息区 第1行为任务队列信息，含义如下。

- 00:55:20: 表示当前系统时间。
- up 38 days, 20: 47: 表示系统已经启动38天零20小时47分钟了。
- 2 users: 当前登录系统的用户数。
- load average: 0.00, 0.01, 0.30: 表示系统平均负载，3个数值分别表示1分钟、5分钟、15分钟前到现在的系统平均负载值。

第2行和第3行分别为进程和CPU信息，具体含义如下。

- Tasks: 72 total: 进程的总数。
- 2 running: 正在运行的进程数。
- 70 sleeping: 处于休眠状态的进程数。
- 0 stopped: 停止的进程数。
- 0

zombie: 僵死的进程数。• Cpu(s): 0.3% us: 表示用户进程占用CPU的百分比。• 0.0% sy: 系统进程占用CPU的百分比。• 0.0% ni: 用户进程空间内改变过优先级的进程占用CPU的百分比。• 99.3% id: 空闲CPU占用的百分比。• 0.3% wa: 等待输入、输出的进程占用CPU的百分比。最后两行输出的是内存信息，具体含义如下。• Mem: 1883616 total: 系统的物理内存大小。• 669112 used: 已经使用的物理内存大小。• 160488 free: 目前空余内存大小。• 1054096 buffers: 用作内核缓冲区的内存大小。• Swap: 0total: 交换分区内存大小。• 0k used: 已经使用的交换分区大小。• 1017968 free: 空闲的交换分区大小。• 2320396k cached: 缓存大小。(2)进程信息区 进程信息区显示了每个进程的运行状态，我们先来看每一列输出的含义。• PID: 进程的id。• USER: 进程所有者的用户名。• PR: 进程优先级。• NI: nice值。负值表示高优先级，正值表示低优先级。• VIRI: 进程使用的虚拟内存总量，单位KB，VIRT-SWAP+RES。• RES:进程使用的，未被换出的物理内存大小，单位KB。RES=CODE+DATA,其中，CODE为执行代码占用的物理内存大小，DATA为数据占用的内存大小。• %SHR: 共享内存大小，单位KB。• %S: 进程状态，D代表不可终端的睡眠状态，R代表运行状态，S表示睡眠状态，T代表跟踪/停止，Z表示僵尸进程。• %CPU: 上次更新到现在的CPU时间占用百分比。• %MEM: 进程占用的物理内存百分比。• TIME+: 进程总计使用的CPU时间，单位为1/100秒。• COMMAND: 正在运行进程的命令名或者命令路径。

### 3.3 文件管理与编辑

## 服务器搭建篇

## 系统管理篇

## 性能调试篇

## 虚拟化与集群应用篇