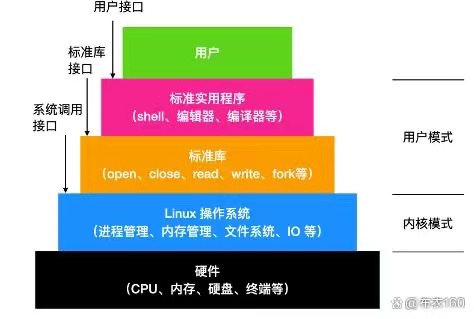
任务2：命令基础和使用帮助

4 命令的使用基础



**交互式接口**

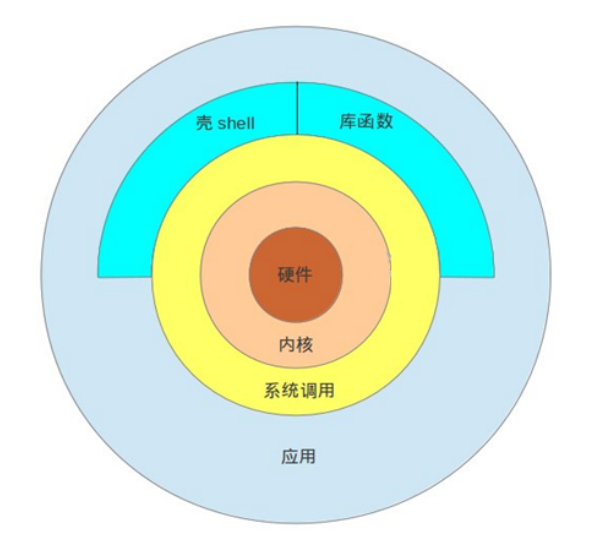
交互式接口：启动终端后，在终端设备附加一个交互式应用程序

**交互式接口类型**

GUI：Graphic User Interface X protocol, window manager, desktop Desktop: GNOME (C, 图形库gtk)， KDE (C++,图形库qt) XFCE (轻量级桌面)

CLI：Command Line Interface shell程序

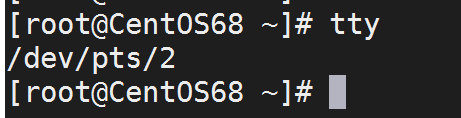
什么是shell



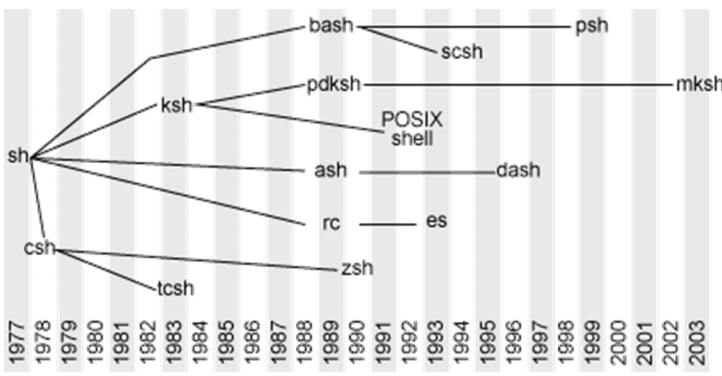
Shell 是Linux系统的用户界面，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行 shell也被称为LINUX的命令解释器（command interpreter） shell是一种高级程序设计语言

查看当前的终端设备：tty

范例：



各种Shell



sh：Steve Bourne

bash：Bourne-Again Shell，GPL，CentOS 和 Ubuntu 默认使用

csh：c shell , C 语言风格

tcsh

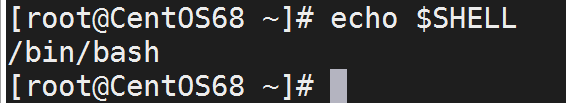
ksh ：Korn Shell, AIX 默认 shell

zsh： MacOS默认shell

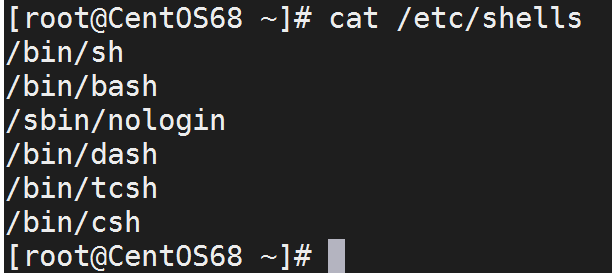
4 bash shell

GNU Bourne-Again Shell(bash)是GNU计划中重要的工具软件之一，目前也是 Linux标准的shell，与sh兼容

显示当前使用的shell



显示当前系统使用的所有shell



命令提示符

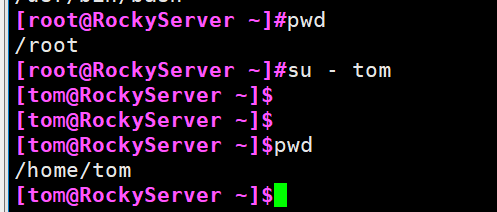
命令提示符：prompt

范例：

[root@CentOS68 ~]#

# 管理员

$ 普通用户



显示提示符格式

[root@centos8 ~]#echo $PS1

\[\e[1;35m\][\u@\h \W]\$\[\e[0m\]

修改提示符格式范例

PS1="\[\e[1;5;41;33m\][\u@\h \W]\\$\[\e[0m\]"

PS1=PS1="\[\e[1;32m\][\t \[\e[1;33m\]\u\[\e[35m\]@\h\[\e[1;31m\] \W\

[\e[1;32m\]]\[\e[0m\]\\$"

Echo 输出显示命令

提示符格式说明：

\e 控制符\033

\u 当前用户

\h 主机名简称

\H 主机名

\w 当前工作目录

\W 当前工作目录基名

\t 24小时时间格式

\T 12小时时间格式

! 命令历史数

# 开机后命令历史数

命令提示符的修改

临时修改 PS1='\[\e[1;33m\][\u@\h \W]\$\[\e[0m\]'

格式：以\[\e[F;Bm\]，以\[\e[0m\]结尾，中间是需要设置颜色的部分，其中“F“为字体颜色，编号为30-37，“B”为背景色，编号为40~47 。

F和B对应的八种颜色的顺序为：黑色，红色，绿色，黄色，蓝色，紫红色，青蓝色，白色

export PS1='\[\e[1;33m\][\u@\h \W]\$\[\e[0m\]'

**如果只是针对当前登录用户起作用，则直接在.bashrc文件最后添加上面一行**

**改成永久的**

echo 'PS1="\[\e[34m\][\u@\h \W]\\$\[\e[0m\]"' > /etc/profile.d/env.sh

## 用户类型

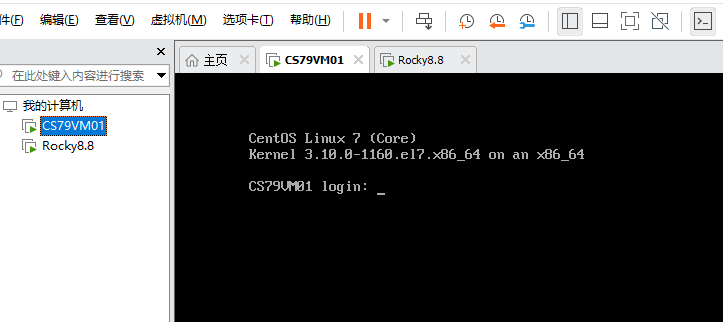
root 用户 一个特殊的管理帐户 也被称为超级用户 root已接近完整的系统控制 对系统损害几乎有

无限的能力 除非必要,不要登录为 root

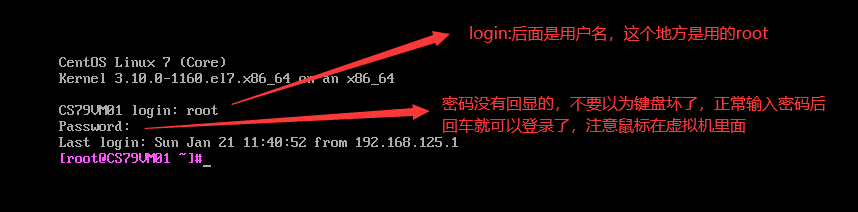
普通（非特权 ）用户 权限有限 造成损害的能力比较有限

登录系统

字符界面（也叫文本界面）

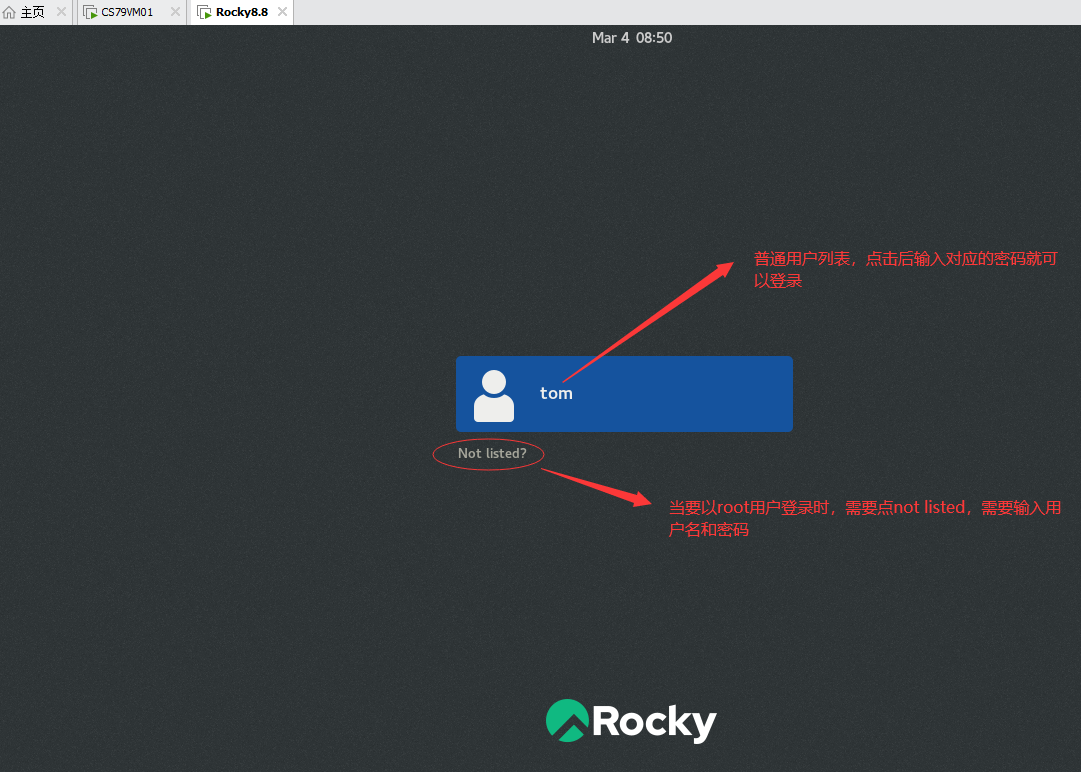


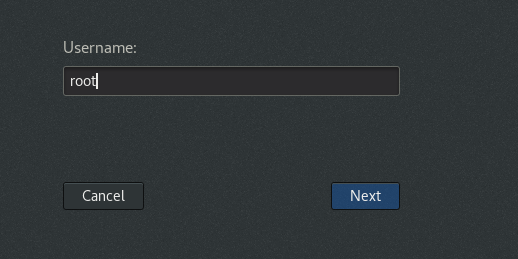
登录如下图所示

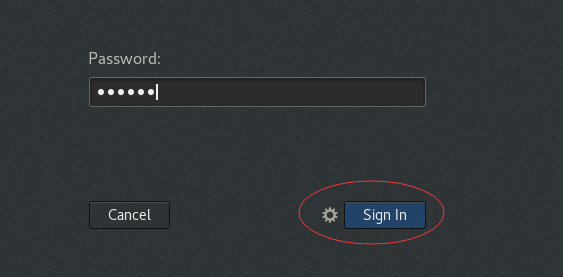


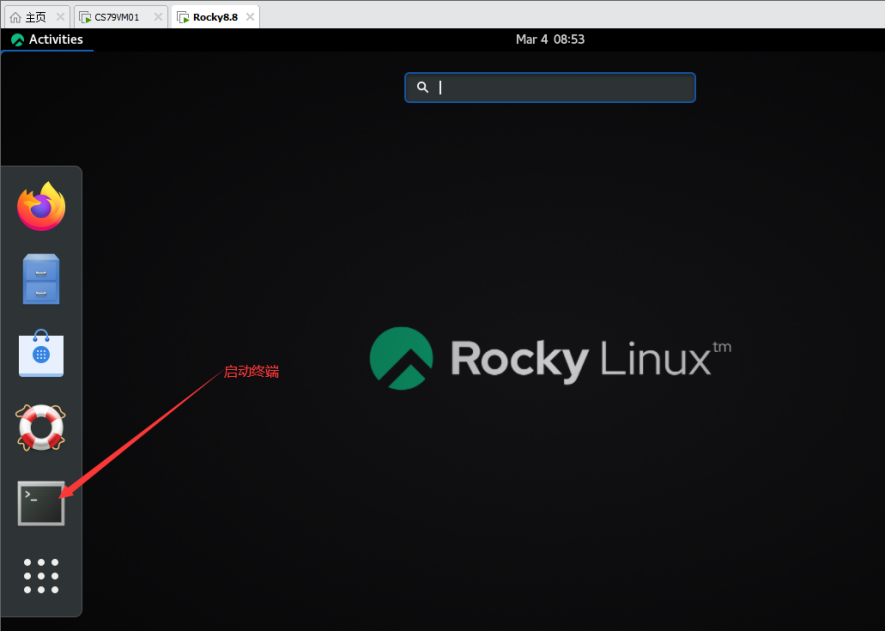
如果要进图形界面，登录成功后，输入startx

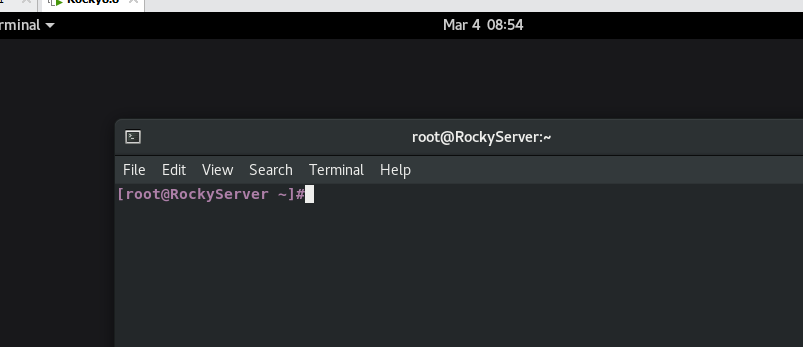
如果默认启动是图形界面，登录界面如下











**系统启动时，如何设置进入文本模式**

下面这个设置启动自动进入文本

#systemctl set-default multi-user.target

下面这个设置启动自动进入图形，这个有前提就是安装系统时要安装过图形桌面才可以

#systemctl set-default graphical.target

生产环境中一般是不会安装图形这个软件

**文字/图形界面的切换（了解）**

**Linux预设的情况下会提供六个Terminal来让用户登入，**

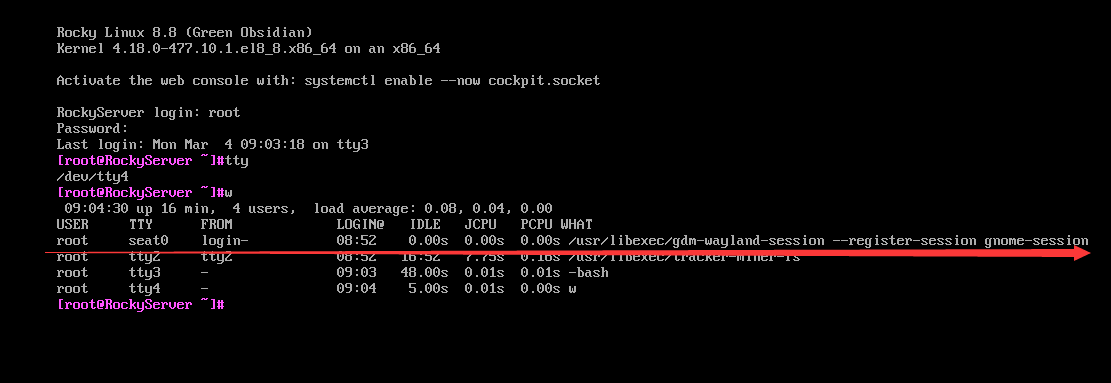
如果是你当前在**图形模式**下切换的方式为使用：[Ctrl] + [Alt] + [F1]~[F6]的组合按钮。

系统会将[F1] ~ [F6]命名为tty1 ~ tty6的操作界面环境。 也就是说，当你按下[ctrl] + [Alt] + [F1]这三个组合按钮时 (**按着[Ctrl] + [Alt]不放**，再按下[F1]功能键)， 就会进入到tty1的terminal界面中了。同样的[F2]就是tty2！ 按下 [Alt] +[Ctrl] + [F2]就可以切到tty2。整理一下登入的环境如下：

* [Ctrl] + [Alt] + [F2] ~ [F6] ：文字界面登入 tty2 ~ tty6 终端机；
* [Ctrl] + [Alt] + [F1] ：图形界面桌面。

**文本模式下切换**

[Alt] + [F1]~[F6]的组合按钮



2.1命令基础

**什么是命令**

Linux系统中凡是能在字符串操作界面中输入的能够完成特定操作和任务的字符串都可以称为“命令”。通常“命令”只代表了实现某一类功能的指令或程序的名称。其本质就是一个程序

**执行命令过程**

输入命令后回车，**提请shell程序找到键入命令所对应的可执行程序或代码**，并由其分析后提交给内核分配资源将其运行起来

**命令的分类**

命令分类：可以外部命令，也可以是内部命令

**内部命令：**

1）指的是集成于shell解释器程序内部的一些特殊指令，也成为内建（Built-in）指令；

2）属于shell的一部分，所以没有单独对应的系统文件，只要shell解释器被运行，内部指令就自动载入内存，用户可以直接使用；

3）内部命令无需从硬盘中重新读取文件，因此执行效率较高。

**外部命令：**

1）指的是Linux系统中能够完成特定功能的脚本文件或二进制程序；

2）属于shell解释器程序之外的命令，每个外部命令对应了系统中的一个文件；

3）Linux系统中必须知道外部命令对应的文件位置，才能够由shell加载并运行。

很多，几大千，由程序独立安装，但是发行版中把经常使用的大部份外部命令都安装在发行版里面去了

**区别指定的命令是内部或外部命令**

#type 命令

**内部命令相关**

help 内部命令列表

enable 管理内部命令

enable cmd 启用内部命令

enable –n cmd 禁用内部命令

enable –n 查看所有禁用的内部命令

**执行外部命令**

查看外部命令路径：

which -a |--skip-alias

Whereis

**Hash 命令缓存表**

系统初始hash表为空，当外部命令执行时，默认会从PATH路径下寻找该命令，找到后会 将这条命令的路径记录到hash表中，当再次使用该命令时，shell解释器首先会查看hash表，存在将执行之，如果不存在，将会去PATH路径下寻找，利用hash缓存表可大大提高命令的调用速率

**hash 命令常见用法**

hash 显示hash缓存

hash -l 显示hash缓存，可作为输入使用

hash -p path name 将命令全路径path起别名为name

hash -t name 打印缓存中name的路径

hash -d name 清除name缓存

hash -r 清除缓存

2.1.1 命令格式

**Linux命令格式**

**登录Linux后，我们就可以在#或$符后面去输入命令，有的时候命令后面还会跟着“选项”（英文options）或“参数”（英文arguments）。**

**即Linux中命令格式为：**

**Command [options] [arguments]**

**中括号代表是可选的，即有些命令不需要选项也不需要参数，但有的命令在运行时需要多个选项或参数。**

命令：命令即命令名称，其实就是一个独立程序或者shell中的函数调用，在Linux的字符操作界面中，使用命令字唯一确定一条命令。并且在Linux的命令环境中，无论文件名或者命令名，**英文字符需要区分大小写**；

通用的命令行格式： **命令 [选项] [参数]**

**命令 [选项] [参数]**

**注意：**

**多个选项以及多参数和命令之间使用空白字符分隔**

**取消和结束命令执行：Ctrl+c，Ctrl+d**

**多个命令可以用 ";" 符号分开**

**一个命令可以用\分成多行**

**选项options：**

选项：用于启用或关闭命令的某个或某些功能

**参数arguments：**

命令参数是命令的**处理对象**

**下面是重点**

**选项分为长选项和短选项。**

**短选项：**

短选项（short option）：由一个连字符和一个字母构成，比如-h，-l，-s等。(- 后面接单个字母)

1. 短选项都是使用‘-’引导，当有多个短选项时，各选项之间使用空格隔开。
2. **有些命令的短选项可以组合，比如-l -h 可以组合为-lh**
3. 有些命令的短选项可以不带-，这通常叫作BSD风格的选项，比如ps aux
4. 有些短选项需要带选项本身的参数（值），比如-L 512M tree -L 1 目录

**长选项：**

比如--help等。(-- 后面接单词)

1. 长选项后面都是完整的单词
2. 长选项通常不能组合
3. 如果需要参数，长选项的参数通常需要‘=’，比如--size=1G

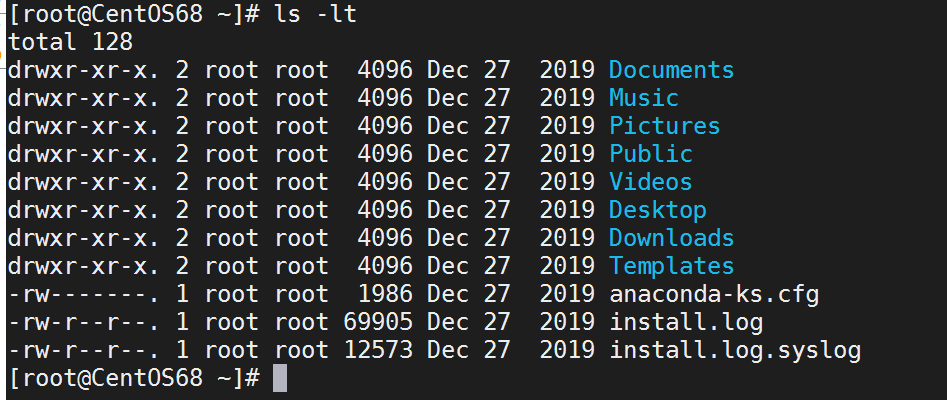
**参数arguments：**

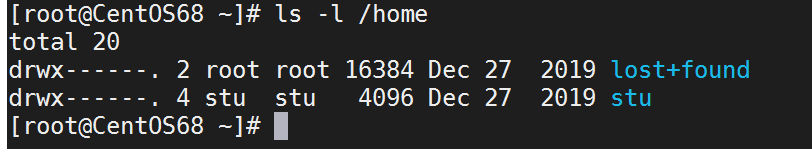
命令参数是命令的处理对象，通常情况下命令参数可以是文件名、目录（路径）名或者用户名等内容。根据使用的命令字的不同，命令参数的个数可以是零到多个。。这个会因为命令功能不同而不同

**以上简要说明了选项及参数的区别，但具体Linux中哪条命令有哪些选项及参数，需要我们靠经验积累或者查看Linux的帮助了。**

**栗子**

如ls命令，不加参数的时候显示是当前目录，也可以加参数，如ls /dev, 则输出结果是/dev目录。





上面这个例子中，

ls是命令，

-lt是两个短选项，l(字母L的小写) 表示以详细信息格式显示内容，t表示按时间排序

/home代表操作的对象

2.1.2 命令的执行顺序

内部命令和外部命令一般都是可以直接敲命令后就能执行的

外部命令的执行过程

因为外部命令是第三方安装程序，那么是一定要找到命令的可执行路径才能执行，当一个命令敲入时

**原则：命令别名优先**

**别名命令查看**

**alias 命令名称**

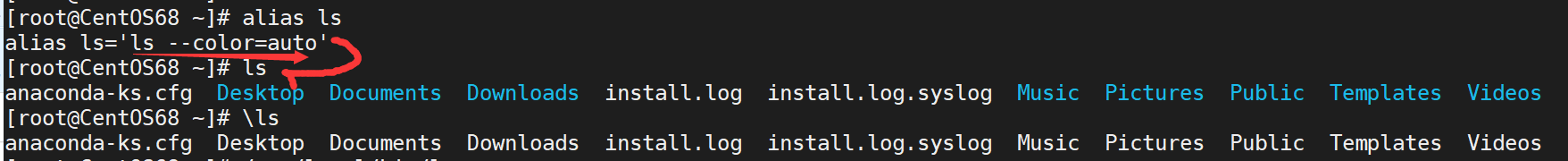
**注意：如果别名名称和原命令名称是相同的**

**如果要执行原命令怎么办？**

**命令前加 \**

**比如要执行ls这个命令为例子**

**大家注意看下图的区别**



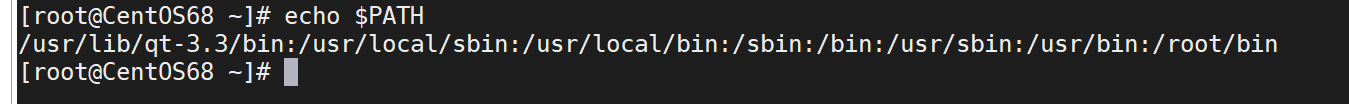
**原则：命令别名优先**

1 查看是否为内部命令，如果是直接执行

2 如果是外部命令，则查看hash缓存表中是否有这个命令存在，如果有，直接按hash缓存中的给出路径执行这个命令

3 如果是外部命令，但hash表中也没有这个命令的缓存，那么这个时候系统就会按照**指定的**几个目录里面去搜索是否有这个可执行的命令，系统将执行搜索到的第一个，如果搜索完成这指定的几个目录里面都没有，则会显示没有这个命令。

这几个**指定的目录**是在一个系统环境变量是设置好的，环境变量名称为**$PATH**



目录与目录之间是用冒号分开的哈

什么是操作系统环境变量，**环境变量是在操作系统中一个具有特定名字的对象,它包含了一个或者多个应用程序所将使用到的信息。**环境变量相对于给系统或用户应用程序设置的一些变量。 环境变量一般是指在操作系统中用来指定操作系统运行环境的一些参数,比如临时文件夹位置和系统文件夹位置等

2.1.3命令别名

对于经常执行的较长的命令，可以将其定义成较短的别名，以方便执行

显示当前shell进程所有可用的命令别名

alias

定义别名NAME，其相当于执行命令VALUE

alias NAME='VALUE'

[root@RockyServer ~]#alias vimip='vim /etc/sysconfig/network-scripts/ifcfg-ens160'

例子：

撤消别名：unalias

unalias [-a] name [name ...]

unalias -a #取消所有别名

注意：在命令行中定义的别名，仅对当前shell进程有效

如果想永久有效，要定义在配置文件中

仅对当前用户：~/.bashrc

对所有用户有效：/etc/bashrc

编辑配置给出的新配置不会立即生效，bash进程重新读取配置文件

source /path/to/config\_file

. /path/to/config\_file

**如果别名同原命令同名，如果要执行原命令，可使用**

\ALIASNAME

“ALIASNAME”

‘ALIASNAME’

command ALIASNAME

/path/commmand

2.1.4 编辑命令行的辅助操作：

反斜杠“\”：如果输入的一行命令内容太长，终端会自动换行。有时候为了美观和方便查看，可以插入“"来进行强制换行，在下一行出现的”>“提示符后可以继续输入内容，作为上一行命令的延续。

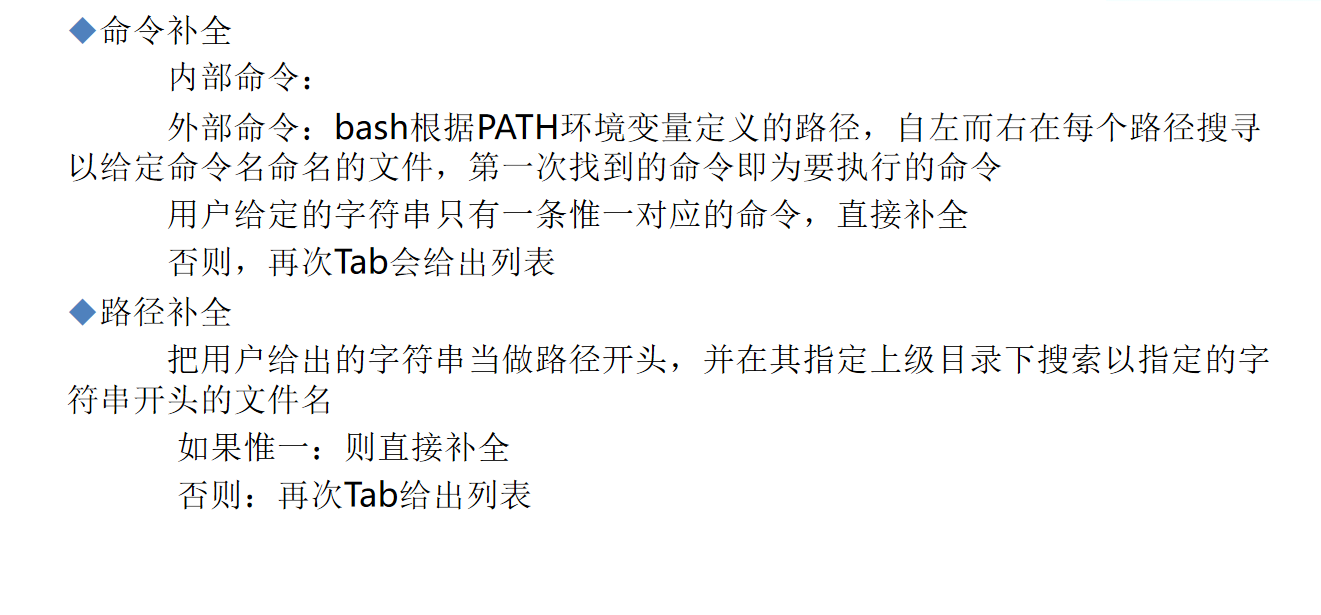
Ctrl+U快捷键：快速删除当前光标处之前的所有字符内容。

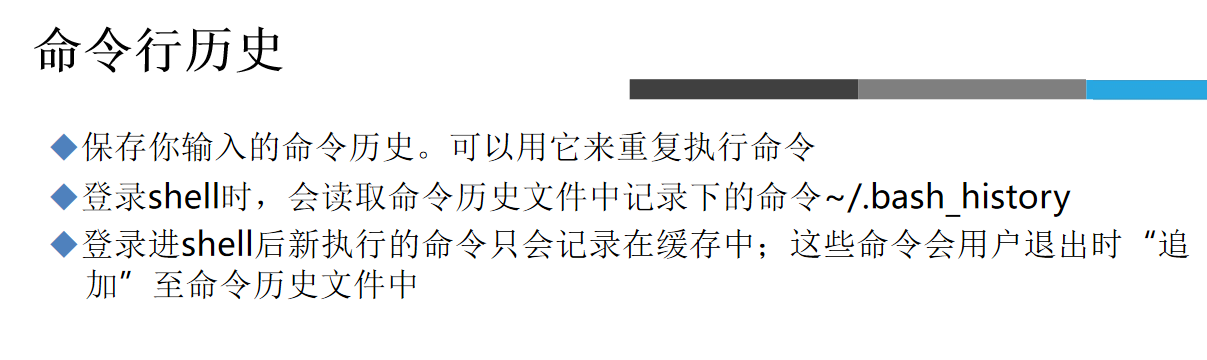
Ctrl+K快捷键：快速删除从当前光标处之后的所有字符内容。

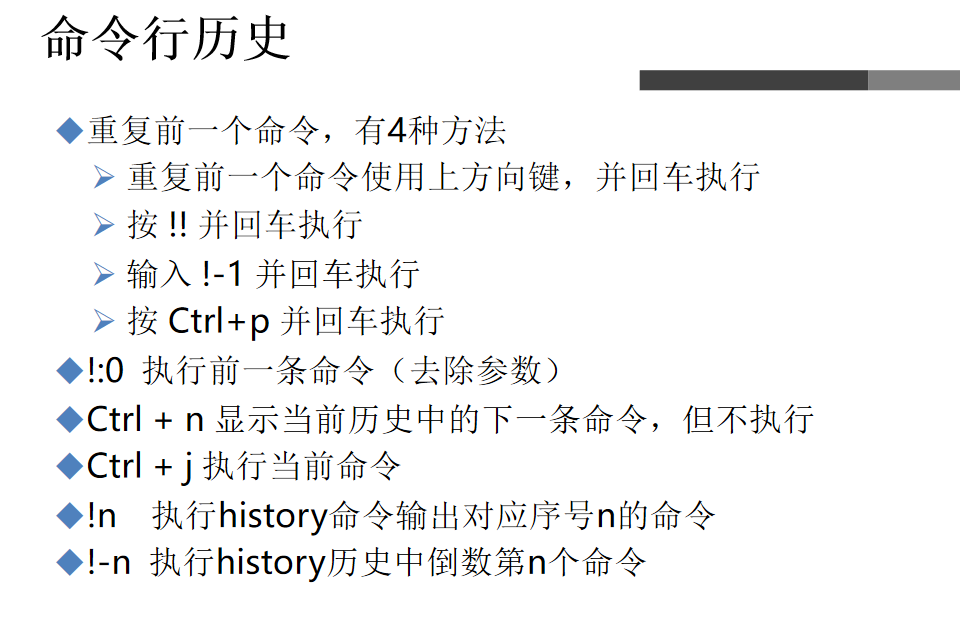
Ctrl+L快捷键：快速清空当前屏幕内容的显示内容，只在左上角显示命令提示符。相当于clear命令

**Ctrl+C快捷键：取消当前命令行的编辑，并切换为新的一行命令提示符。**

2.1.5 tab键及历史记录







2.2 常用的基础命令

2.2.1日期和时间命令

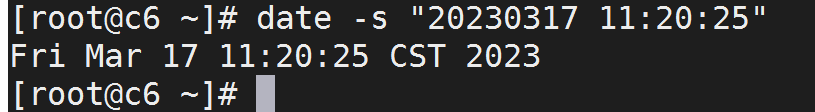
Linux的两种时钟 系统时钟：由Linux内核通过CPU的工作频率进行的 硬件时钟：主板，clock，hwclock: 显示硬件时钟 -s, --hctosys 以硬件时钟为准，校正系统时钟 -w, --systohc 以系统

**相关命令**

**date**

**1 显示和设置系统时间**

将系统的当前时间设置为2023年3月17日11点20分的date命令如下所示：



2 日期格式字符串列表

%H 小时，24小时制（00~23）

%I 小时，12小时制（01~12）

%k 小时，24小时制（0~23）

%l 小时，12小时制（1~12）

%M 分钟（00~59）

%p 显示出AM或PM

%r 显示时间，12小时制（hh:mm:ss %p）

%s 从1970年1月1日00:00:00到目前经历的秒数

%S 显示秒（00~59）

%T 显示时间，24小时制（hh:mm:ss）

%X 显示时间的格式（%H:%M:%S）

%Z 显示时区，日期域（CST）

%a 星期的简称（Sun~Sat）

%A 星期的全称（Sunday~Saturday）

%h,%b 月的简称（Jan~Dec）

%B 月的全称（January~December）

%c 日期和时间（Tue Nov 20 14:12:58 2012）

%d 一个月的第几天（01~31）

%x,%D 日期（mm/dd/yy）

%j 一年的第几天（001~366）

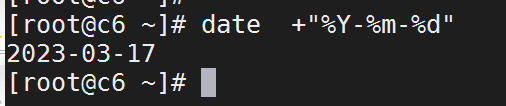
%m 月份（01~12）

%w 一个星期的第几天（0代表星期天）

%W 一年的第几个星期（00~53，星期一为第一天）

%y 年的最后两个数字（1999则是99）

格式化输出时间：



**hwclock命令**

**hwclock命令**是一个硬件时钟访问工具，它可以显示当前时间、设置硬件时钟的时间和设置硬件时钟为系统时间，也可设置系统时间为硬件时钟的时间。

在Linux中有硬件时钟与系统时钟等两种时钟。硬件时钟是指主机板上的时钟设备，也就是通常可在BIOS画面设定的时钟。系统时钟则是指kernel中的时钟。当Linux启动时，系统时钟会去读取硬件时钟的设定，之后系统时钟即独立运作。所有Linux相关指令与函数都是读取系统时钟的设定。

**选项**

--adjust：hwclock每次更改硬件时钟时，都会记录在/etc/adjtime文件中。使用--adjust参数，可使hwclock根据先前的记录来估算硬件时钟的偏差，并用来校正目前的硬件时钟；--debug：显示hwclock执行时详细的信息；--directisa：hwclock预设从/dev/rtc设备来存取硬件时钟。若无法存取时，可用此参数直接以I/O指令来存取硬件时钟；--hctosys：将系统时钟调整为与目前的硬件时钟一致；

--set --date=<日期与时间>：设定硬件时钟；

--show：显示硬件时钟的时间与日期；

--systohc：将硬件时钟调整为与目前的系统时钟一致；

--test：仅测试程序，而不会实际更改硬件时钟；

--utc：若要使用格林威治时间，请加入此参数，hwclock会执行转换的工作；

--version：显示版本信息。

**设置硬件时间要依赖于操作系统时间，具体方法如下：**

hwclock –systohc

hwclock --systohc –-utc

**不加任何参数使用hwclock，可以查看当前的硬件日期和时间。**

hwclock

范例：修改系统时间同步

RHEL8系统，必须同时修改系统时间和硬件时间，才可以保证修改有效，单纯的使用date命令修改系统时间，是立即生效，重启后系统还原。具体操作如下：

1.timedatectl show （）查看目前本地的时间和时区）

2.hwclock --show {查看硬件的时间}

3.时区设定：timedatectl set-timezone Asia/Shanghai

4.如果硬件时间和系统时间不同，那就对硬件的时间进行修改

5.hwclock --set --date '014-12-15 15:15:15' （设置硬件时间为14年12月15日15点15分15秒）

6.

系统时间同步到硬件时间： hwclock -w

硬件时间同步到系统时间： hwclock -s

7.hwclock --hctosys （设置系统时间和硬件时间同步）

8.clock -w （保存时钟）

如果内网有NTP服务器或者使用公网的NTP服务器可能就更加能够确保时间同步完全一致，能够精确到毫秒

修改/etc/chrony.conf配置文件

pool ntp1.aliyun.com iburst

chronyc sources -v 查看同步源状态

2.2.2关机和重启命令

关机：

shutdown -h now

halt

poweroff

init 0

重启：

shutdown -r now

reboot -f: 强制，不调用shutdown -p: 切断电源

关机或重启：shutdown

-r: reboot -h: halt -c：cancel TIME：无指定，默认相当于+1（CentOS7） now: 立刻,相当于+0 +#:

相对时间表示法，几分钟之后；例如 +3 hh:mm: 绝对时间表示，指明具体时间

2.2.3 用户登录信息查看命令

whoami: 显示当前登录有效用户

who: 系统当前所有的登录会话

w: 系统当前所有的登录会话及所做的操作

last

2.2.4查看系统相关信息的几个命令

查看系统基本信息

**修改主机名**

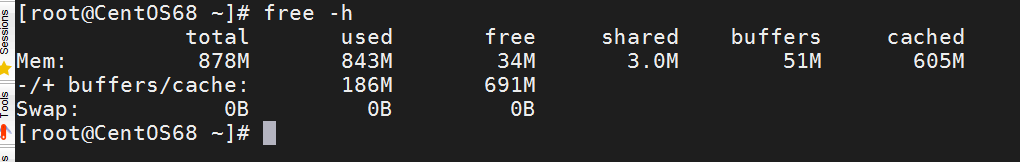
hostnamectl set-hostname centos

[root@CentOS68 ~]# lscpu

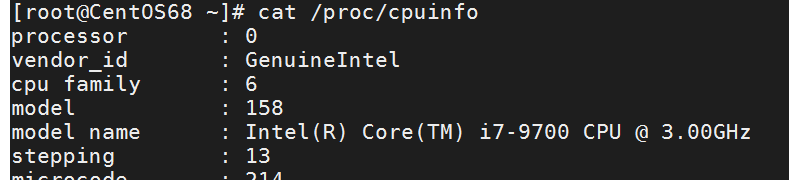
查看CPU信息

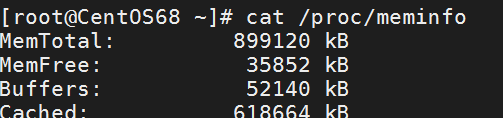
[root@CentOS68 ~]# free -h

查看内存



对应的还有两个也可以看





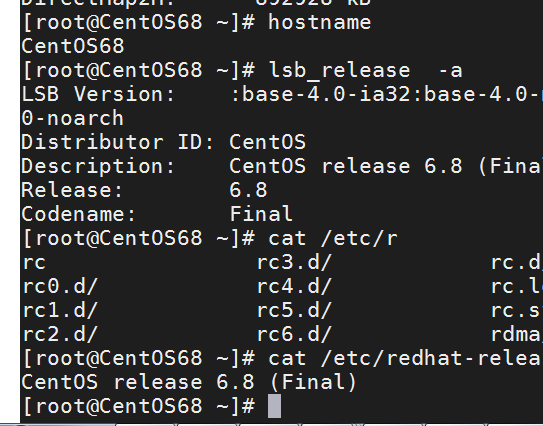
主机名称查看和系统版本

#hostname 查看主机名称

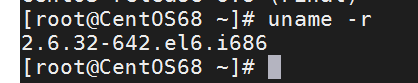
#lsb\_release -a 查看系统发行版，注意这个命令默认安装时不会安装上去，如果需要使用，需要安装一个软件上去

[root@RockyServer ~]#cat /etc/redhat-release

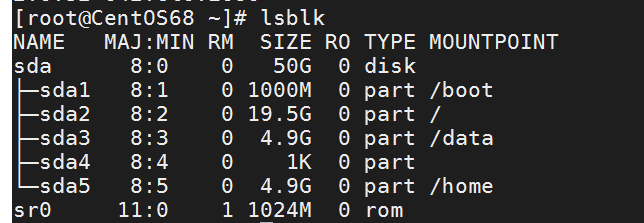
Rocky Linux release 8.8 (Green Obsidian)



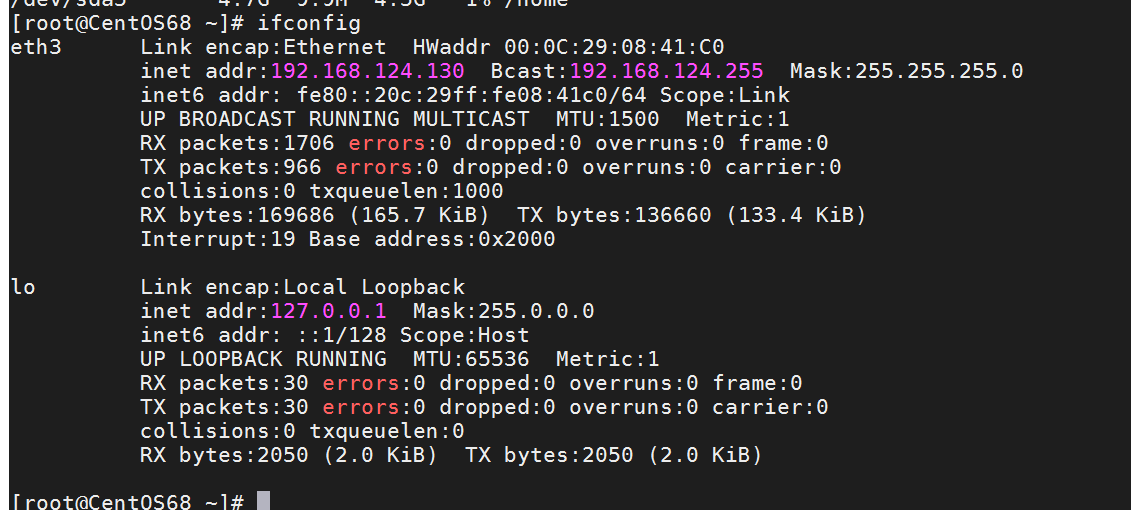
查看内核版本



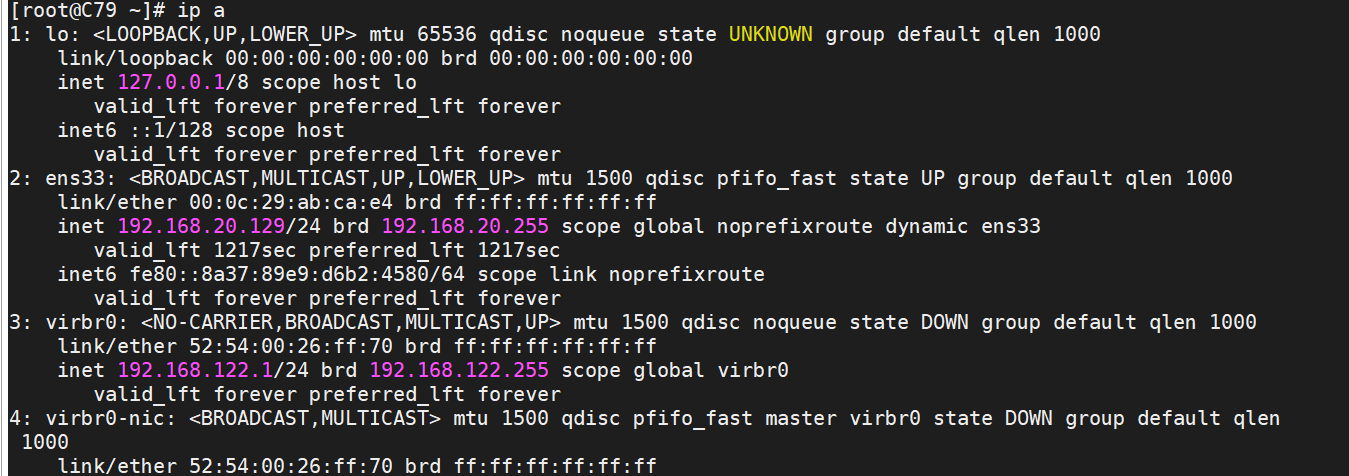
查看硬盘信息



查看IP地址 ifconfig



第2个是命令是 ip a



2.3 使用帮助

**帮助概述**

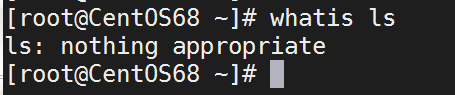
Linux系统中能够使用的命令数量繁多，具体选项也各不相同，使用格式也存在细微区别。对于Linux系统命令的更多选项和使用格式，可以通过在线帮助查询。

1. whatis命令
2. 使用help命令
3. 使用man手册

2.3.1 whatis

功能：**简单的查看命令的功能**

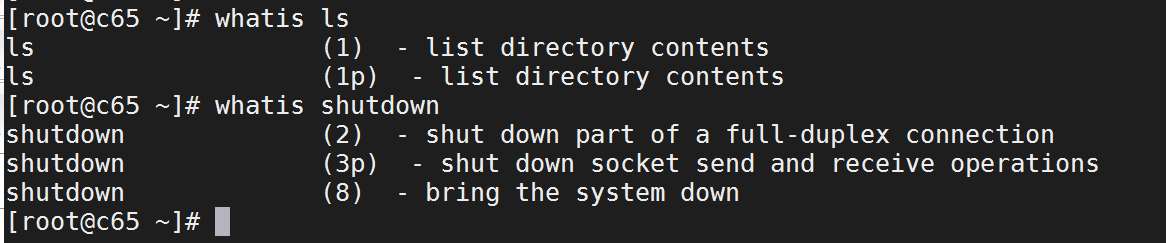
第一次执行如果出现下面情况



在安装Linux的时候，可能没有生成whatis的数据库，所以whatis的命令无法使用。如果出现无法使用whatis命令失效，那就生成whatis数据库。

RHEL7以前生成数据库命令： makewhatis

RHEL7以后以前生成数据库命令： mandb



2.3.2 help

功能：

1. 查看是外部还是内部命令
2. 提供命令帮助

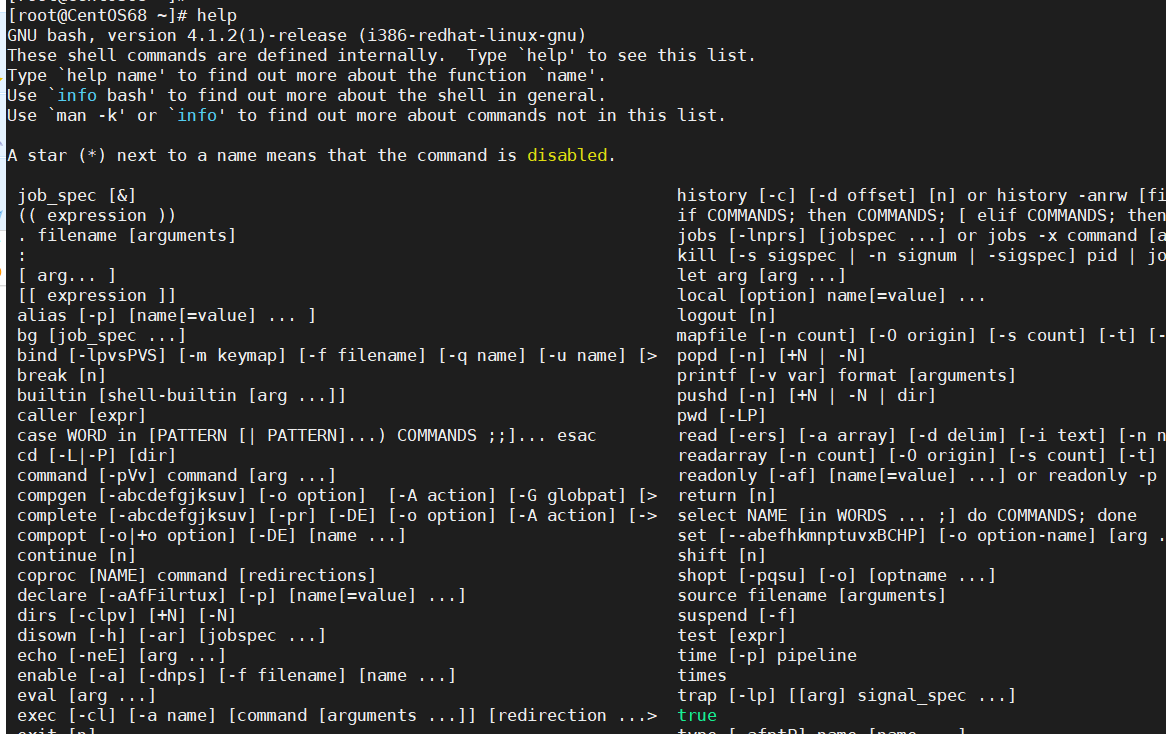
help命令本身是Linux shell中的一个内建指令，用途是查看各shell各内部命令的帮助信息。使用help命令时，只需要添加内部指令的名称作为参数即可。例如“help pwd”、“help cd”等；

对于大多数Linux外部命令，可以使用一个通用的命令长选项“--help”，用于显示对应命令字的格式及选项等帮助信息。若该命令字没有“--help”选项，一般只会提示简单的命令格式。例如执行“ls–help”

1. 查看是外部还是内部命令

#help，不需要带任何选项和参数

功能：显示所有内部命令

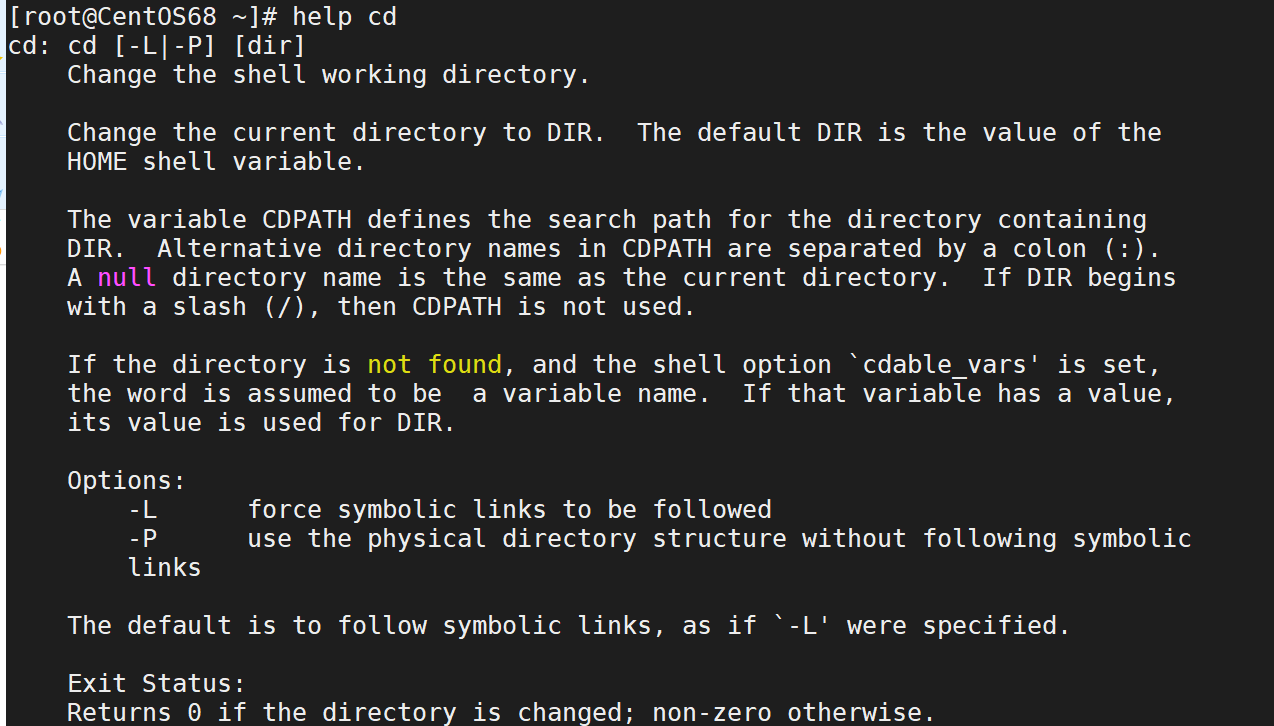


常用的两个帮助 help man

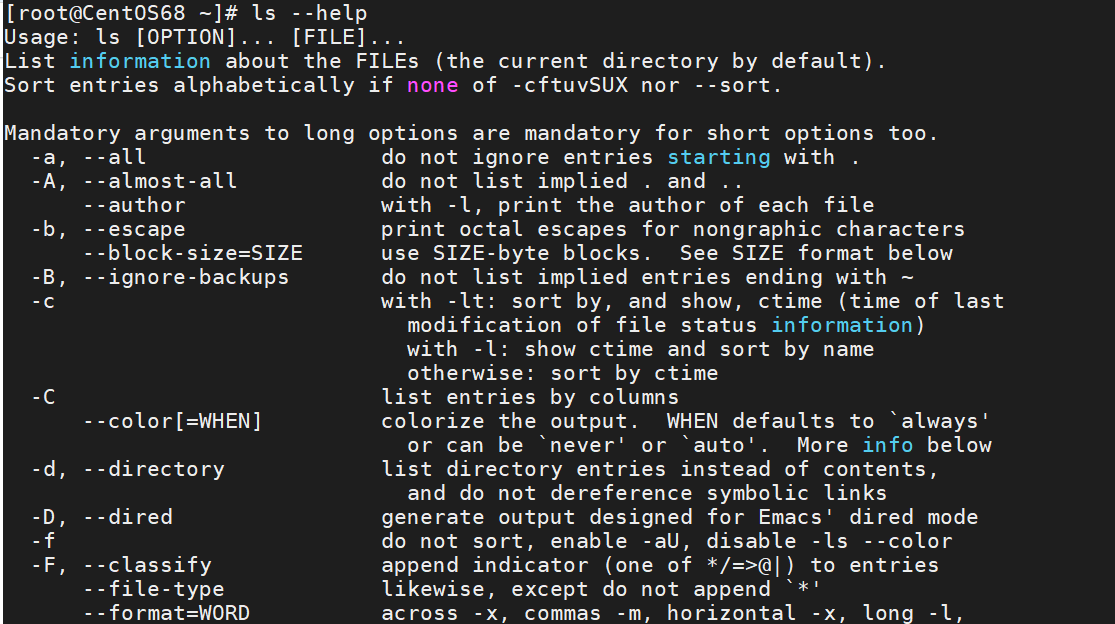
2） help帮助显示内部命令帮助

#help 命令

#help cd



#help帮助显示外部命令帮助时是 命令 --help



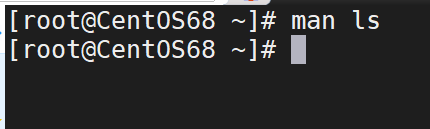
2.3.3 man

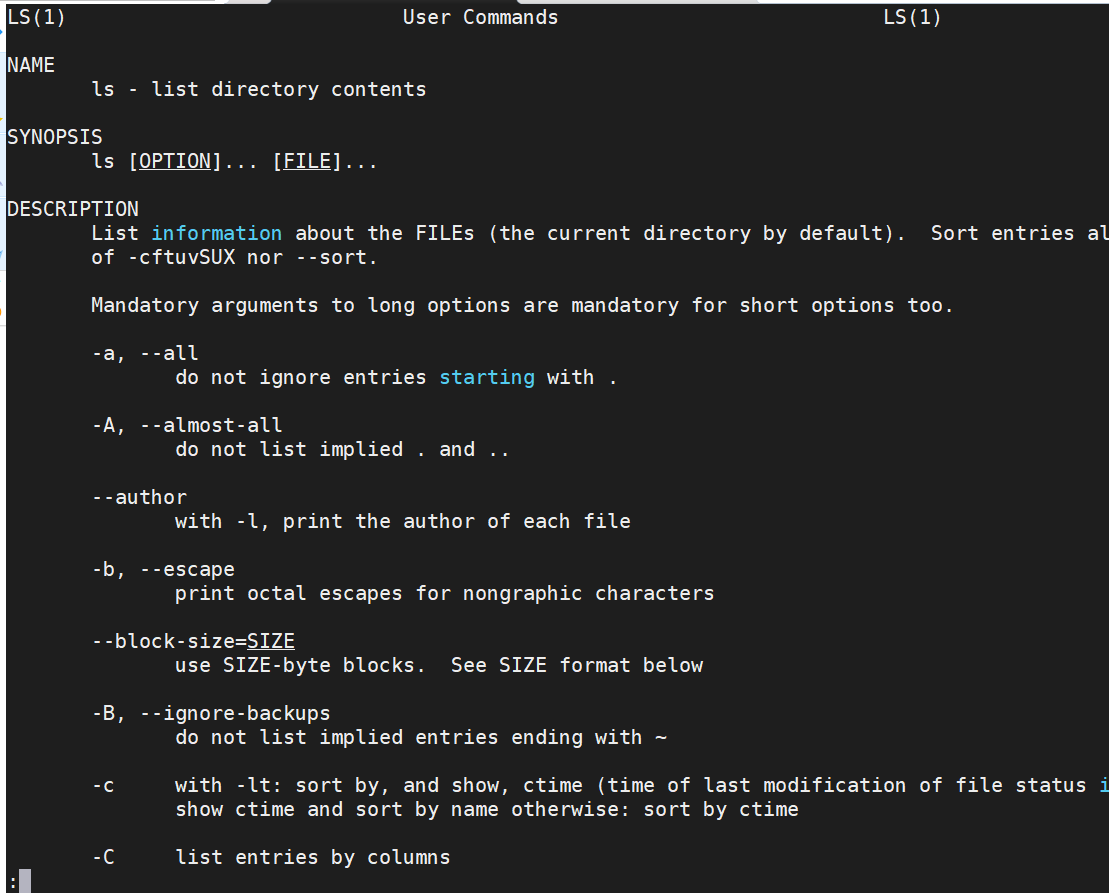
man手册页（manual page）是Linux系统中最常用的一种系统级别帮助形式。绝大部分的外部软件在安装时为执行程序、配置文件提供了详细的帮助手册页。例如执行“man cd”。

阅读man手册页时，按page up键和page down可以向上向下翻页显示，按“Q"或”q“可以随时退出手册页的阅读环境。按”/"键后可以对手册内容进行查找，若找到的结果有多个，可以按“n"或者”N“分别向下、向上进行定位选择。

#man 命令

#man可以显示所有命令帮助





再来，man page的内容也分成好几个部分来加以介绍该指令呢！就是上头man date那个表格内， 以NAME作为开始介绍，最后还有个SEE ALSO来作为结束。基本上，man page大致分成底下这几个部分：

|  |  |
| --- | --- |
| 代号 | 内容说明 |
| NAME | 简短的指令、资料名称说明 |
| SYNOPSIS | 简短的指令下达语法(syntax)简介 |
| DESCRIPTION | 较为完整的说明，这部分最好仔细看看！ |
| OPTIONS | 针对 SYNOPSIS 部分中，有列举的所有可用的选项说明 |
| COMMANDS | 当这个程序(软件)在执行的时候，可以在此程序(软件)中下达的指令 |
| FILES | 这个程序或资料所使用或参考或连结到的某些档案 |
| SEE ALSO | 可以参考的，跟这个指令或资料有相关的其他说明！ |
| EXAMPLE | 一些可以参考的范例 |

有时候除了这些外，还可能会看到Authors与Copyright等，不过也有很多时候仅有NAME与DESCRIPTION等部分。

按q退出man

5 man page的分类

|  |  |
| --- | --- |
| 代号 | 代表内容 |
| 1 | 使用者在shell环境中可以操作的指令或可执行档 |
| 2 | 系统核心可呼叫的函数与工具等 |
| 3 | 一些常用的函数(function)与函式库(library)，大部分为C的函式库(libc) |
| 4 | 装置档案的说明，通常在/dev下的档案 |
| 5 | **设定档或者是某些档案的格式** |
| 6 | 游戏(games) |
| 7 | 惯例与协定等，例如Linux档案系统、网络协定、ASCII code等等的说明 |
| 8 | 系统管理员可用的管理指令 |
| 9 | 跟kernel有关的文件 |

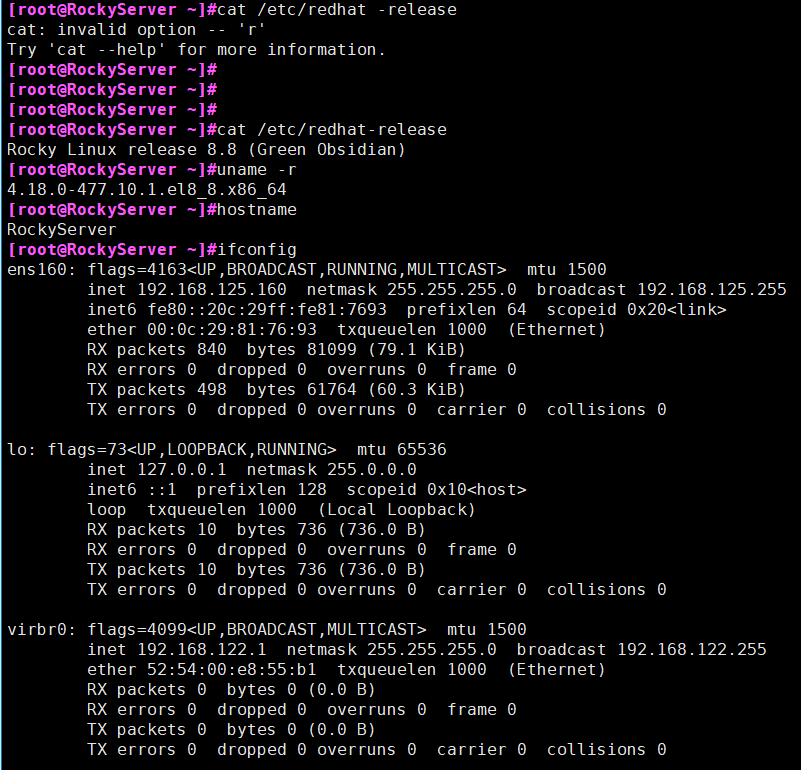
上述的表格内容可以使用『man man』来更详细的取得说明。透过这张表格的说明， 未来你如果使用man page在察看某些资料时，就会知道该指令/档案所代表的基本意义是什么了。 举例来说，如果你下达了『man null』时，会出现的第一行是：『NULL(4)』，对照一下上面的数字意义， 嘿嘿！原来null这个玩意儿竟然是一个『装置档案』呢！很容易了解了吧！

上表中的1, 5, 8这三个号码特别重要，一定要将这三个数字所代表的意义背下来喔！

实验2

1 使用mobaxterm xshell 终端连接服务器

2 查看服务器系统基本信息



3 用命令关机和重启服务器



4 修改系统时间为当前时间半个小时以后

5 使用帮助，按照下面命令输入看帮助结果

