



Linux的启动与常用命令

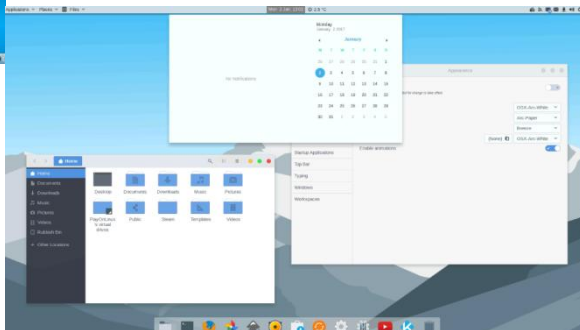
信息工程学院

一、Linux图形界面概述

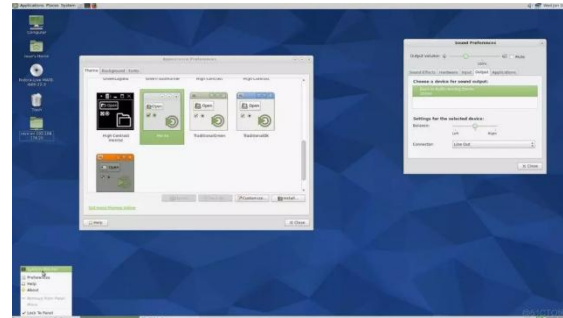
- Windows——图形界面操作系统
 - 图形环境与内核紧密结合，图形环境直接访问Windows内核
- Linux——本身无图形界面的操作系统
 - X Window System——类Unix窗口系统，是跨网络与跨操作系统的窗口系统
 - X Window System是Linux的一个可选组件



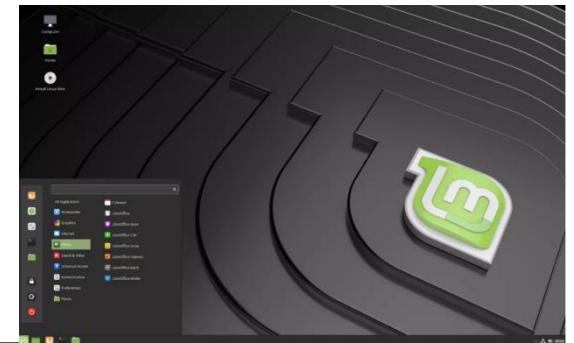
KDE



GNOME



MATE



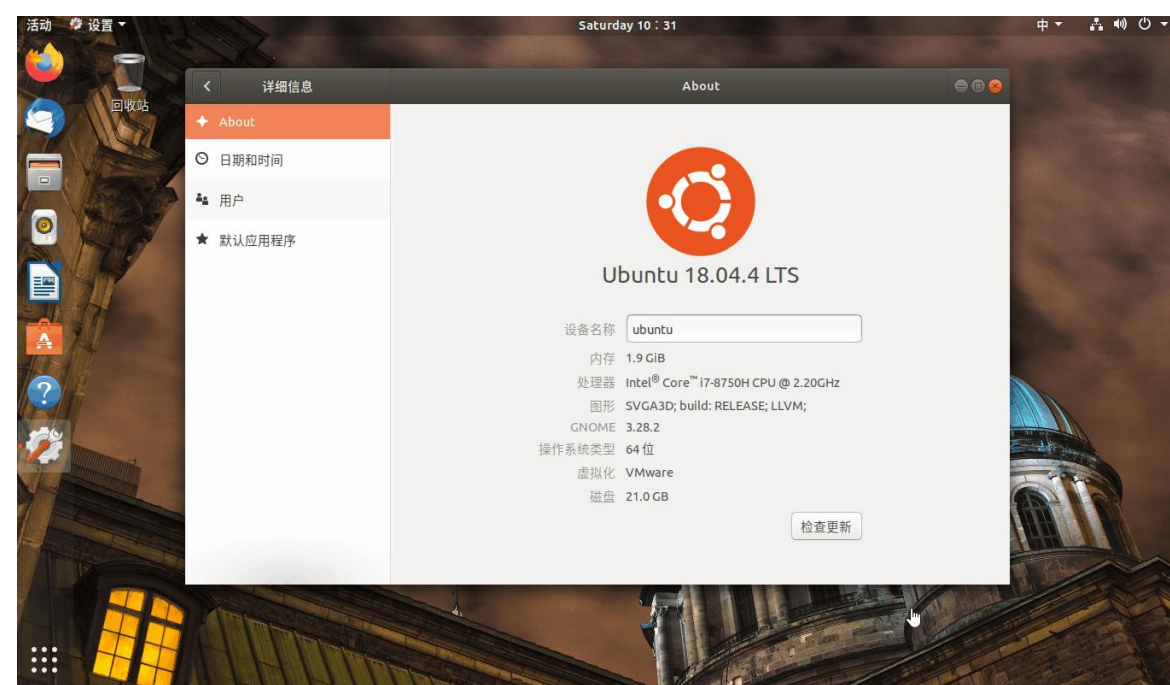
Cinnamon



Unity

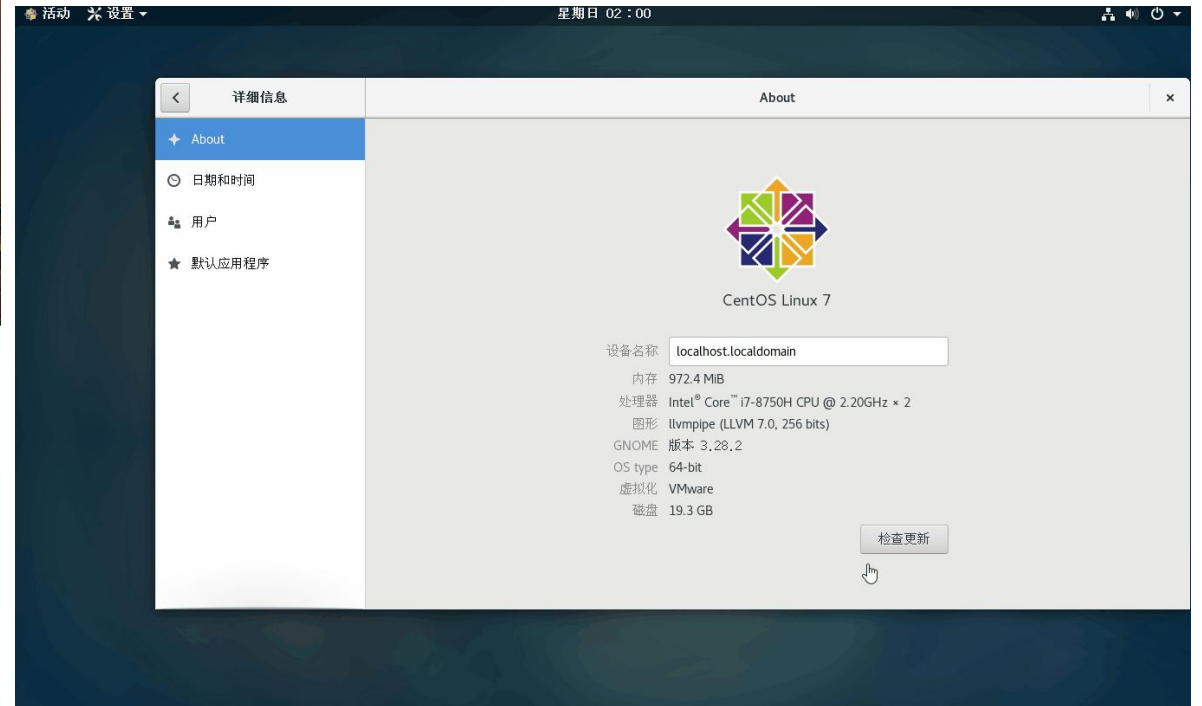
- **KDE**: 早期未开源, 推动了GNOME的发展, 更类似windows桌面
 - CentOS默认
- **GNOME**: GNU的一部分, 完全开源。
 - Red Hat、CentOS默认
- **Unity**: 是基于GNOME的非完整桌面环境, 主要实现了面板部分, 设计成可更高效地使用屏幕空间, 消耗系统资源更少。
 - 早期Ubuntu默认

一、Linux图形界面概述



Ubuntu Gnome

CentOS Gnome

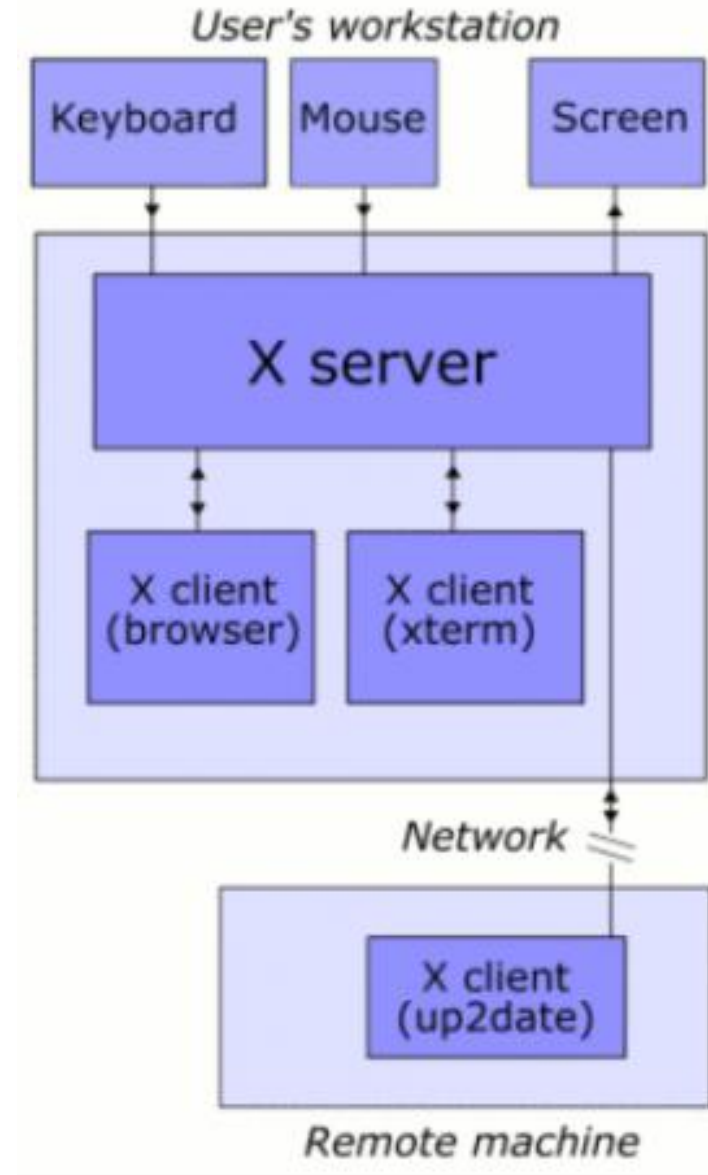


一、Linux图形界面概述

- Windowing system（窗口系统）窗口系统，是GUI的一种（也是当前计算机设备、智能设备广泛使用的一种），以WIMP（windows、icons、menus、pointer）的形式，提供人机交互接口。
- Linux系统中有很多窗口系统的实现，如X Window System、Wayland、Android SurfaceFlinger等，虽然形态各异，但思路大致相同，包含如下要点：
 - 使用Client-Server架构，Server（称作Display Server，或者Windows server、Compositor等等）管理所有输入设备，以及用于输出的显示设备。
 - 应用程序作为Display Server的一个Client，在自己窗口（Window）中运行，并绘制自己的GUI。
 - Client的绘图请求，都会提交给Display Server，Display Server响应并处理这些请求，以一定的规则混合、叠加，最终在有限的输出资源上（屏幕），显示多个应用程序的GUI。
 - Display Server和自己的Client之间，通过某种类型的通信协议交互，该通信协议通常称作Display Server Protocol。
 - Display Server Protocol可以是基于网络的，甚至是网络透明的（network transparent），如X Window System所使用的。也可以是其它类型的，如Android SurfaceFlinger所使用的Binder。

一、Linux图形界面概述

- 用户通过鼠标键盘对X server下达操作命令
- X server利用Event传递用户操作信息给X client
- X client进行程序运算
- X client利用Request传回所要显示的结果
- X server将结果显示在屏幕上



二、Linux操作系统有6种运行级别：

- 0: 停机
- 1: 单用户模式，就像Windows 9x下的安全模式
- 2: 多用户，但是没有NFS
- 3: 完全多用户模式，标准运行级别（命令行启动）
- 4: 保留，一般不用
- 5: X Window图形模式（默认为图形启动方式）
- 6: 重新启动

- 图形模式->文本模式启动——传统命令init 3
- 文本模式->图形模式启动——传统命令init 5

(CentOS 6) 查看和修改默认级别- vi /etc/inittab

```
[root@CM00 ~]# cat /etc/inittab
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/init/rcS.conf
#
# In runlevel, the system is started by /etc/init/rc.conf
#
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/
# with configuration in /etc/sysconfig/init.
#
# For information on how
# upstart works, see
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```



读取inittab
文件



CentOS 7 设置系统默认运行级别



init级别	systemctl target
0	shutdown.target
1	emergency.target
2	rescue.target
3	multi-user.target
4	无
5	graphical.target
6	无



CentOS 7设置系统默认运行级别



```
jiangnan@localhost:/  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[root@localhost ~]# systemctl set-default multi-user.target  
[root@localhost ~]# systemctl get-default  
multi-user.target  
[root@localhost ~]# systemctl set-default graphical.target  
Removed symlink /etc/systemd/system/default.target.  
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/graphical.target.  
[root@localhost ~]# systemctl get-default  
graphical.target  
[root@localhost ~]#
```

图形模式->文本模式启动

- **rm /etc/systemd/system/default.target** //删除原来的默认启动模式
- **ln -s /lib/systemd/system/multi-user.target /etc/systemd/system/default.target**
//添加新的启动模式，将文本模式设置成默认启动模式

文本模式->图形模式启动

- **rm /etc/systemd/system/default.target** //删除原来的默认启动模式
- **ln -s /lib/systemd/system/graphical.target /etc/systemd/system/default.target**
//添加新的启动模式，将图形模式设置成默认启动模式

默认情况下，系统中有两个主要的**targets**:

- **multi-user.target** ——文本模式（黑底白字）
- **graphical.target** ——图形模式。



文本模式下启动图形模式



当启动操作系统为文本模式

- ◆ 若还想进入图形界面，则输入命令：

startx 或 **sudo startx**

//以普通用户登陆图形界面 或 以超级用户进入图形界面

- 若要切回文本模式，则找到用户名点击**log out**重启

- 默认的情况下安装完成后是从X Window启动的。

X-Window可以通过终端切换到命令行模式

- 若文本模式启动，在系统被引导后：

```
CentOS release 6.5 (Final)
Kernel 2.6.32-431.el6.i686 on an i686

www login: root
Password: _
```

```
[root@www ~]# _
```

```
Ubuntu 20.04.3 LTS ubuntu tty1

ubuntu login: wlw
Password:
```

```
root@ubuntu:~#
```

说明:

- Fedora和CentOS在安装时会设置root用户的密码。
- Ubuntu在安装时root初始密码随机的，要登录第一个用户给 root 设置密码。

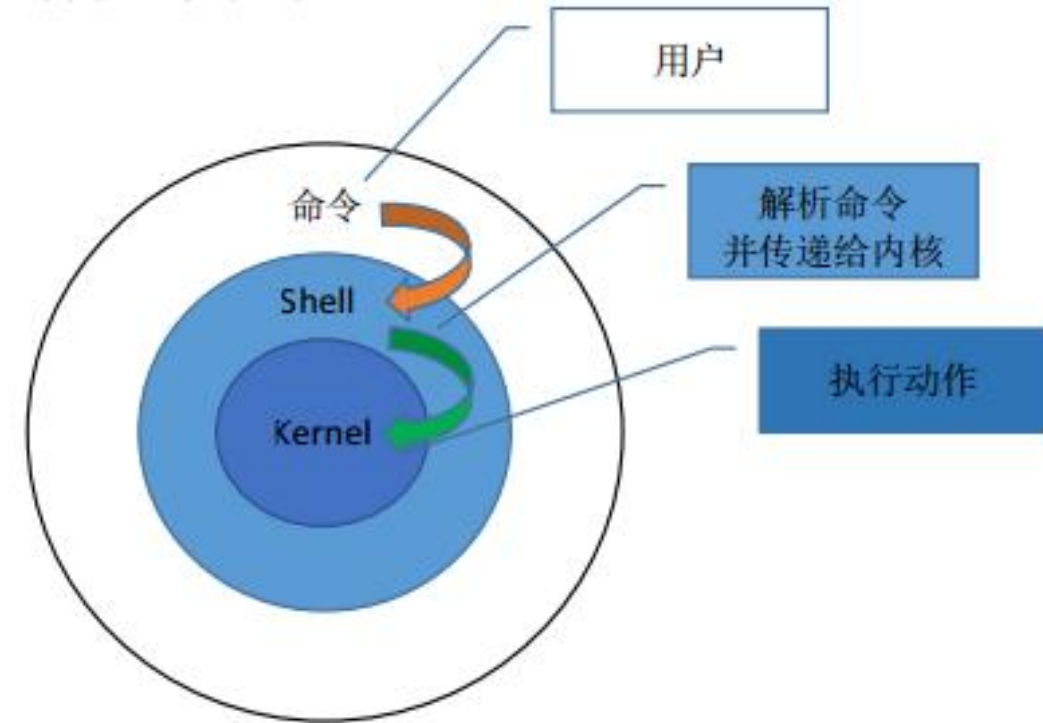
```
dreamhai@ubntuserver:/$ su root
Password:
su: Authentication failure
dreamhai@ubntuserver:/$ sudo passwd
[sudo] password for dreamhai:
New password:
Retype new password:
passwd: password updated successfully
dreamhai@ubntuserver:/$ su root
Password:
root@ubntuserver:/# su dreamhai
dreamhai@ubntuserver:/$
dreamhai@ubntuserver:/$
dreamhai@ubntuserver:/$
dreamhai@ubntuserver:/$
```

由于初始化时,root的密码是随机的,所以认证失败

输入sudo passwd, 修改密码
本次改为:123
修改成功后,
> sudo root
输入密码:123
这里可以看到用户已经切换为root

三、Linux Shell

- Shell 是一个 C 语言编写的脚本语言
- 用户与 Linux 的桥梁
- 用户输入命令交给 Shell 处理
- Shell 将相应的操作传递给内核内核把处理的结果输出给用户



JavaScript (JS, 前端), VBScript, ASP, JSP, PHP (后端), SQL (数据库操作语言), Perl, Shell, python, Ruby, JavaFX, Lua等。

常见的脚本



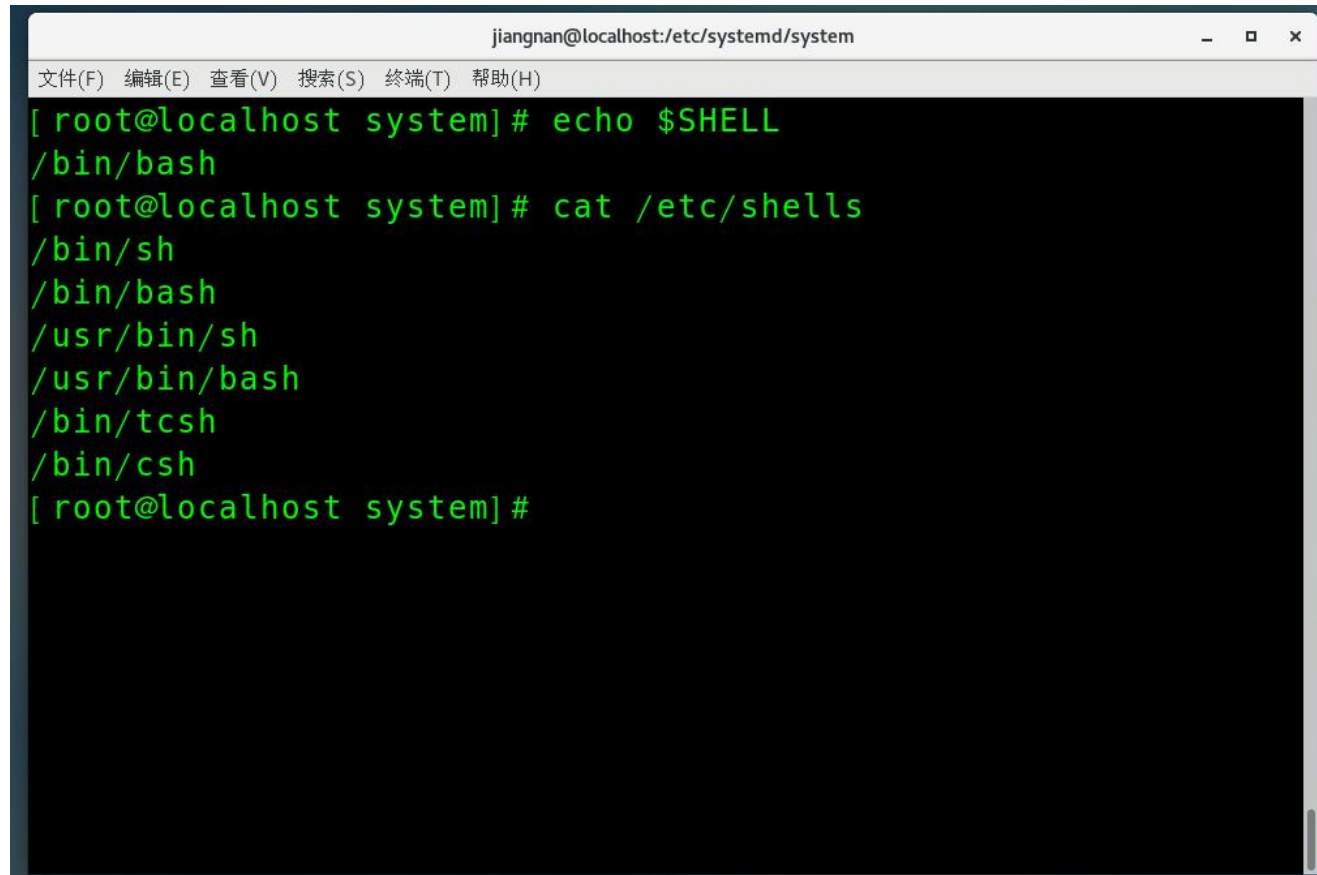
三、Linux Shell



- 图形界面Shell（GUI Shell）
 - GUI 为 Unix 或者类 Unix 操作系统构造一个功能完善、操作简单以及界面友好的桌面环境。主流桌面环境有 KDE，Gnome 等。
- 命令行界面Shell（CTL Shell）
 - CLI 是在用户提示符下键入可执行指令的界面，用户通过键盘输入指令，完成一系列操作。在 Linux 系统上主流的 CLI 实现是 Bash，是许多 Linux 发行版默认的 Shell。还有许多 Unix 上 Shell，例如 tcsh、csh、ash、bsh、ksh 等。

三、Linux Shell

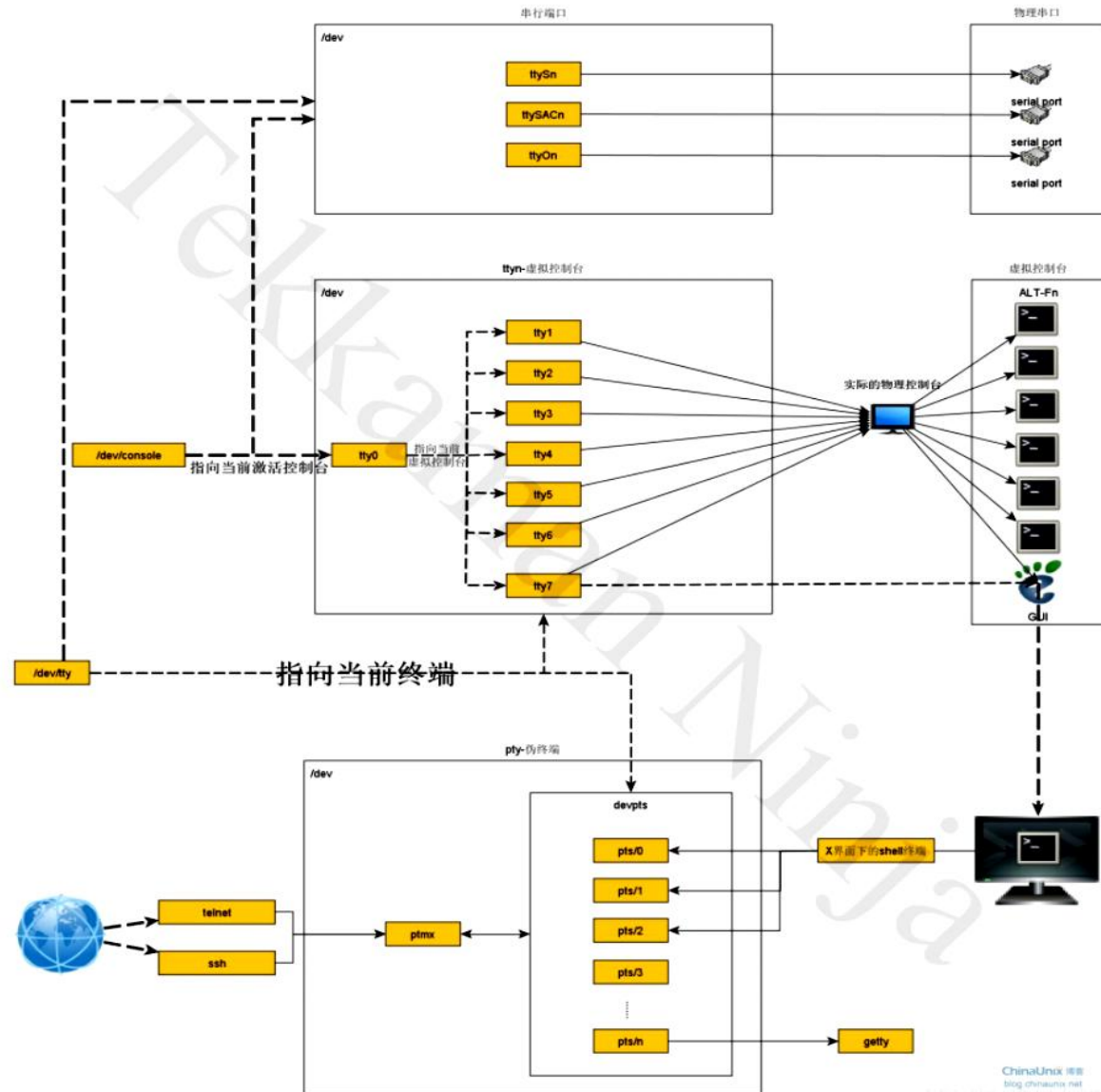
- 查看文件/etc/shells显示当前系统有效的shell程序
- 命令 echo \$SHELL 可以查看当前使用 shell程序



```
jiangnan@localhost:/etc/systemd/system
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost system]# echo $SHELL
/bin/bash
[root@localhost system]# cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
/bin/tcsh
/bin/csh
[root@localhost system]#
```

四、Linux终端

- 在Linux系统的设备特殊文件目录/dev/下，终端特殊设备文件一般有以下几种：
- 串行端口终端(/dev/ttySn)
- 伪终端(/dev/pty/)
- 控制台终端(/dev/ttyn, /dev/console)---显示器



• 串行端口终端(/dev/ttySn)

- 串行端口终端是使用电脑串行端口连接的终端设备。电脑把每个串行端口都看作是个字符设，通常被称为终端设备。
- 串行端口所对应的设备名称是/dev/tts/0(或/dev/ttyS0),/dev/tts/1(或/dev/ttyS1)等，设备号分别是(4,0),(4,1)等，分别对应于DOS系统下的COM1、COM2等。若要向一个端口发送数据，能够在命令行上把标准输出重定向到这些特别文档名上即可。
- 例如，键入命令行：echo test >/dev/ttyS1，把” test” 发送到连接在ttyS1(COM2)端口的设备上，可接串口来验证。

• 控制终端(/dev/tty)

- tty(TeleType)源于Teletypes或者teletypewriters, 原指电传打字机, 通过串行线传送信息, 后来被键盘与显示器取代, 现在是终端设备的统称。
- /dev/tty代表当前tty设备, 是当前进程的控制终端的设备特别文档。能够使用命令”ps ax”来查看进程和哪个控制终端相连。使用命令”tty”能够查看具体对应哪个实际终端设备。
- 例如: 输入 echo “hello” > /dev/tty , 都会直接显示在当前的终端中。

```
[root@CM00 ~]# echo "thank ">/dev/tty
thank
[root@CM00 ~]# echo "thank ">/dev/tty0
thank
[root@CM00 ~]# tty
/dev/tty1
[root@CM00 ~]# echo "thank ">/dev/tty1
thank
[root@CM00 ~]# _
```

• 控制台终端(/dev/ttyn, /dev/console)

- 控制台终端(/dev/console)是电脑显示器，现在只有在单用户模式下，才允许用户登录控制台console。/dev/console仿真了Linux的一种终端，有一些设备特别文档ttyn等与之相关联，**控制台终端 (/dev/ttyn) 称为虚拟终端。**
- 当在控制台上登录时，使用的是tty1。可以使用Alt+[F1—F6]组合键在tty1...tty6间切换，能够让系统同时有几个不同的会话存在。Alt+F7是切换到tty7图形界面。

```
CentOS Linux 8
Kernel 4.18.0-240.el8.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

localhost login: root
Password:
Last login: Thu Dec 10 12:30:02 on tty2
[root@localhost ~]# ps
  PID TTY          TIME CMD
 13649 tty1        00:00:00 bash
 13678 tty1        00:00:00 ps
[root@localhost ~]# _
```

```
CentOS Linux 8
Kernel 4.18.0-240.el8.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

localhost login: root
Password:
Last login: Thu Dec 10 14:24:28 on tty1
[root@localhost ~]# ps
  PID TTY          TIME CMD
 14086 tty2        00:00:00 bash
 14115 tty2        00:00:00 ps
[root@localhost ~]#
```

• 控制台终端(/dev/ttyn, /dev/console)

- tty0是当前虚拟终端的别名，只有系统或root用户能向/dev/tty0进行写操作。

```
[root@CM00 ~]# echo "thank ">/dev/tty
thank
[root@CM00 ~]# echo "thank ">/dev/tty0
thank
[root@CM00 ~]# tty
/dev/tty1
[root@CM00 ~]# echo "thank ">/dev/tty1
thank
[root@CM00 ~]# echo "thank ">/dev/tty2
[root@CM00 ~]# _
```

```
[wlw@CM00 ~]$ tty
/dev/tty2
[wlw@CM00 ~]$ thank

[wlw@CM00 ~]$ echo "thank">/dev/tty0
-bash: /dev/tty0: Permission denied
[wlw@CM00 ~]$
```

在tty1执行echo "thank">/dev/tty2，不会显示；
按Alt+F2切换到tty2则可见显示了thank。

• 伪终端(/dev/pty/) —— pty(pseudo-tty)

- **逻辑上的终端设备**。例如，在X Window下打开的终端，或在Windows使用telnet 或ssh等方式登录Linux主机，此时均在使用pty设备。伪终端是成对的逻辑终端设备（即master和slave设备，对master的操作会反映到slave上）。
- 在使用设备文档系统之前，为了得到大量的伪终端设备特别文档，使用了比较复杂的文档名命名方式。为了得到更多的逻辑设备对，就使用了象q、r、s等字符来代替p。例如，ttys8和ptys8就是个伪终端设备对。但现在使用了/dev/ptmx，它的对应端则会被自动地创建成/dev/pts/x。

```
[root@CM00 ~]# ll /dev/pt*  
crw-rw-rw-. 1 root tty 5, 2 Mar 1 16:54 /dev/ptmx  
  
/dev/pts:  
total 0  
crw--w----. 1 root tty 136, 0 Mar 1 16:54 0  
c----- . 1 root root 5, 2 Mar 2 2023 ptmx
```

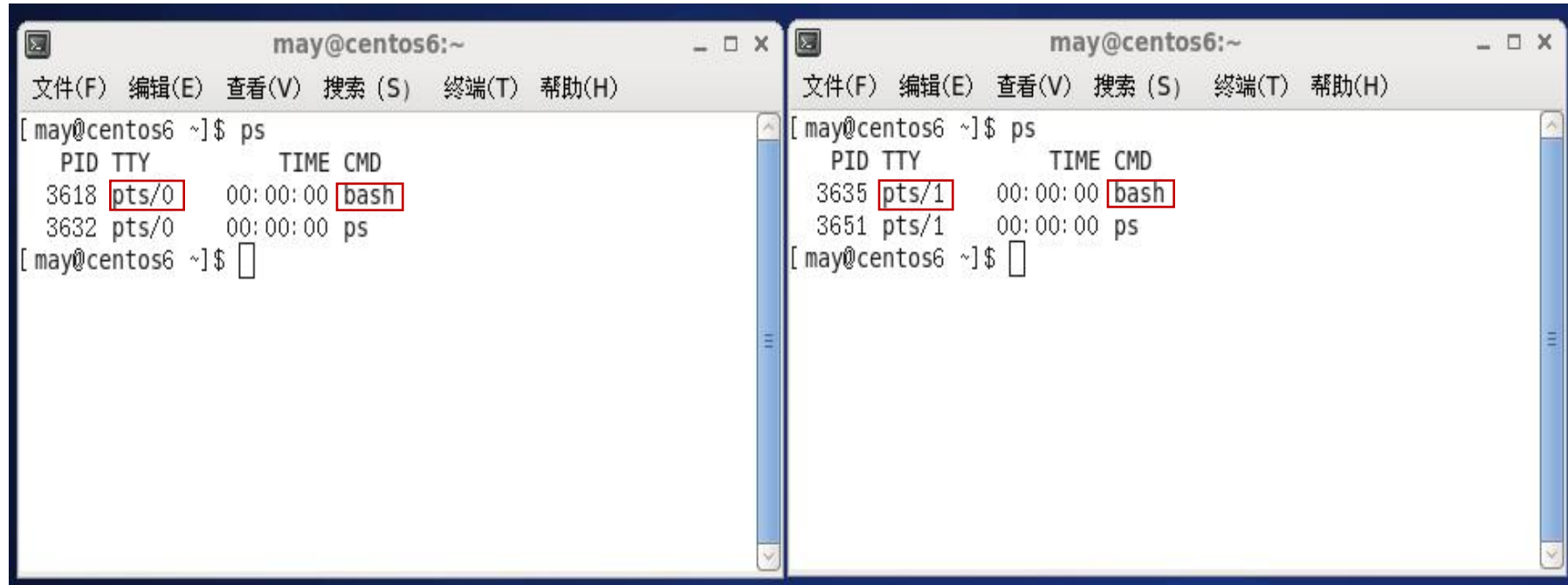
打开一个终端窗口

```
[root@CM00 ~]# ll /dev/pt*  
crw-rw-rw-. 1 root tty 5, 2 Mar 1 16:57 /dev/ptmx  
  
/dev/pts:  
total 0  
crw--w----. 1 root tty 136, 0 Mar 1 16:54 0  
crw--w----. 1 root tty 136, 1 Mar 1 16:57 1  
c----- . 1 root root 5, 2 Mar 2 2023 ptmx  
[root@CM00 ~]#
```

打开两个终端窗口

- 虚拟终端(/dev/pts/n)——pts(pseudo-terminal slave)

- 在X Window下打开的终端或通过telnet或者ssh等连上的Linux系统显示出来的。
- pts是pty的实现方法，与ptmx(pseudo-terminal master)配合使用。



```
may@centos6:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[ may@centos6 ~]$ ps  
  PID TTY          TIME CMD  
  3618 pts/0        00:00:00 bash  
  3632 pts/0        00:00:00 ps  
[ may@centos6 ~]$  
  
may@centos6:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[ may@centos6 ~]$ ps  
  PID TTY          TIME CMD  
  3635 pts/1        00:00:00 bash  
  3651 pts/1        00:00:00 ps  
[ may@centos6 ~]$
```

登录之后用PS命令能看到当前SHELL进程，一般为BASH

五、Linux命令格式

- Linux命令的各组成部分间用空格分隔。



- 短选项：选项之间用空格隔开
 - 有些短选项可以组合：-l -h = -lh
 - 有些短选项需要参数：-L 512M
- 长选项：--help, --list
 - 长选项通常不能组合
 - 长选项的参数需要用=号，--size=1G

- ❖ ls
- ❖ ls -l 显示详细列表
- ❖ ls -a 显示隐藏文件
- ❖ ls -la
- ❖ ls -help
- ❖ ls /root
- ❖ ls /etc

五、Linux命令格式

```
jiangnan@localhost:/usr
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost usr]# ls-l
bash: ls-l: 未找到命令...
[root@localhost usr]# ls -l
总用量 256
dr-xr-xr-x.  2 root root 49152 2月 22 20:17 bin
drwxr-xr-x.  2 root root    6 4月 11 2018 etc
drwxr-xr-x.  2 root root    6 4月 11 2018 games
drwxr-xr-x.  9 root root 4096 2月 21 17:01 include
dr-xr-xr-x. 42 root root 4096 2月 22 20:17 lib
dr-xr-xr-x. 144 root root 77824 2月 21 17:07 lib64
drwxr-xr-x. 46 root root 12288 2月 21 17:02 libexec
drwxr-xr-x. 12 root root   131 2月 21 16:54 local
dr-xr-xr-x.  2 root root 20480 2月 22 20:17 sbin
drwxr-xr-x. 239 root root 8192 2月 21 17:02 share
drwxr-xr-x.  4 root root    34 2月 21 16:54 src
lrwxrwxrwx.  1 root root    10 2月 21 16:54 tmp -> ../var/tmp
mp
[root@localhost usr]#
```

```
jiangnan@localhost:/usr
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost usr]# ls /home
jiangnan
[root@localhost usr]#
```



六、常用命令



- 登录
- 用户切换
- 修改主机名
- 关机/注销/离开系统
- 查看当前进程

Linux系统是严格区分大小写的，无论是用户名、文件名、设备名都是如此。
即Ubuntu、ubuntu、UBUNTU是三个不同的用户名或文件名。

命令：su

功能：用来切换用户的工作环境

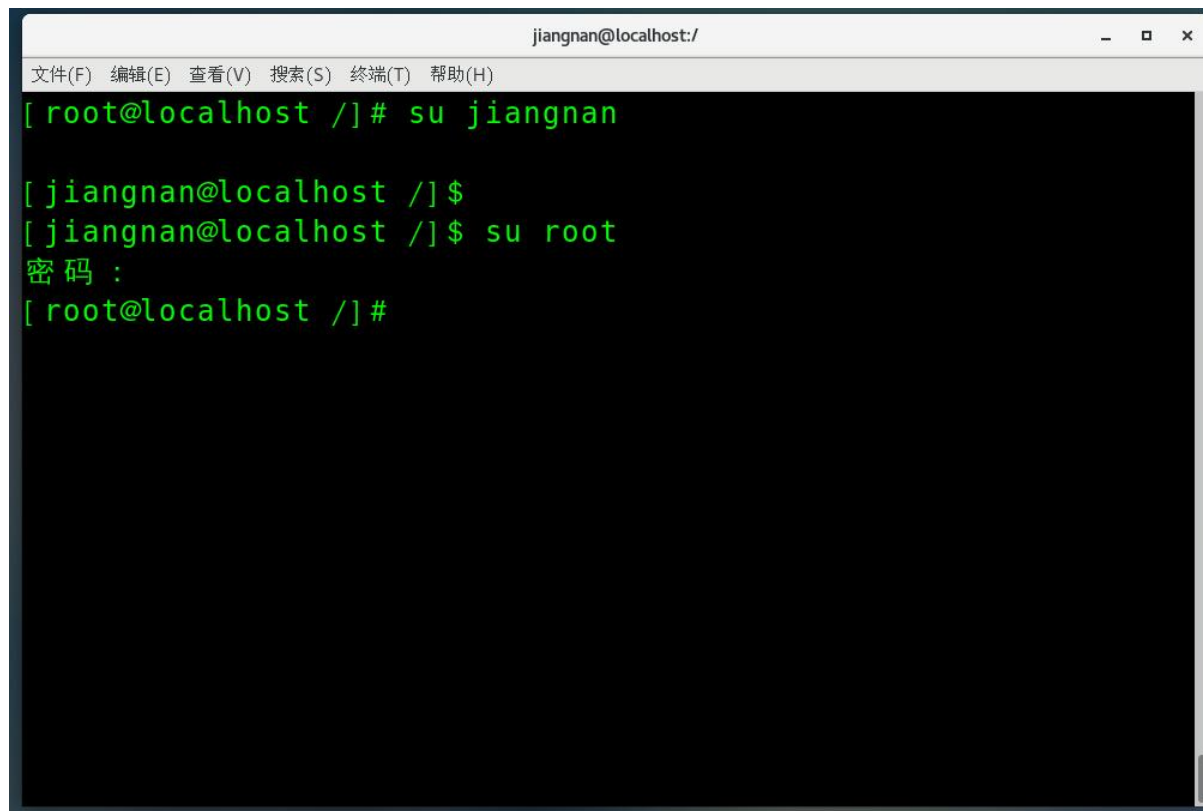
- 从普通用户进入root用户

su root

- 从root用户进入普通用户

su 用户名

- 命令提示符表示命令输入的状态
 - 管理员root用户的提示符 “**#**”
 - 普通用户提示符 “**\$**”



```
jiangnan@localhost: /
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost ~]# su jiangnan

[jiangnan@localhost ~]$
[jiangnan@localhost ~]$ su root
密码:
[root@localhost ~]#
```

修改主机名

[root@localhost ~]\$ **hostname** 姓名全拼+小班序号

[用户名@**主机名** 主目录]\$

```
[root@localhost ~]# hostname  
localhost.localdomain  
[root@localhost ~]#
```

- 主机名是为计算机取的名字，用于区分不同的计算机。
- 主机名是内网用的名字，可重复但建议使用唯一名字。
- 域名是申请的公网名字，公网唯一。

永久修改主机名的步骤

- 第一步：更改/etc/sysconfig下的network文件，在提示符下输入 vi /etc/sysconfig/network，然后把HOSTNAME后面的值更改为想要设置的主机名。

```
[root@localhost ~]# hostname
localhost.localdomain
[root@localhost ~]# vi /etc/sys
sysconfig/          sysctl.conf          system-release          system-release-cpe
[root@localhost ~]# vi /etc/sysconfig/network

[1]+  Stopped                  vi /etc/sysconfig/network
[root@localhost ~]# vi /etc/sysconfig/network
```

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
~
~
~
~
```

更改为想要的主机名字

- 第二步:

更改/etc下的hosts文件, 在提示符下输入 vi /etc/hosts, 然后将localhost.localdomian改为想要设置的主机名

```
[3]+ Stopped vi /etc/hosts
[root@localhost ~]# vi /etc/hosts
```

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
~
~
~
~
~
~
~
~
```

更改为想要设置的主机名字

- 第三步 关闭防火墙:

意义: 开放出所有的端口号(80,443,22...), 8001端口号的时候, 防火墙会直抢端口号, 最终需要把防火墙关闭

```
[root@localhost ~]# service iptables stop
iptables: Setting chains to policy ACCEPT: filter      [ OK ]
iptables: Flushing firewall rules:                    [ OK ]
iptables: Unloading modules:                          [ OK ]
[root@localhost ~]#
```

- 第四步:

重启虚拟机: reboot

注销/关机/重启

- **logout** //注销用户，与login对应
- **exit** //退出目前的shell。
- **poweroff** //关机
- **reboot** //重启系统。
- **shutdown** //重启/关机（系统管理者）
 - r //重启
 - h //关闭计算机
 - now //立即

重启

-reboot
-shutdown -r now

关机

-shutdown -h now
-poweroff

- **sync 把内存数据同步到磁盘**

注意：当系统重启或关机时，都应该先执行**sync**命令，防止数据丢失。

联机帮助指令man（manual）

- 执行man cd 显示如下:列出几乎所有内部命令

```
BASH_BUILTINS(1)                                BASH_BUILTINS(1)
NAME
bash, !, ., [, alias, bg, bind, break, builtin, caller, cd, command, compgen, complete, com-
popt, continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, false, fc, fg,
getopts, hash, help, history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd,
pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap, true,
type, typeset, ulimit, umask, unalias, unset, wait - bash built-in commands, see bash(1)
```

- man文档一般都放在/usr/share/man/目录中分类存放，可通过whereis命令查看man文档的路径，一般都是以.gz结尾。

获取命令帮助的方式有：

- man 命令名
- help 命令名
- 命令名 --help

- **Tab键**
 - 补全功能：可以用来补全指令名或文件名
 - 列表功能：两次点击Tab键，将会把所有可选指令或文件的名称列举出来，避免用户输入错误
- **向上箭头**：上一条指令
- **查看历史命令记录用history**
- **Ctrl+c** 终止当前正在执行的程序
- **Ctrl+d** 终止当前键盘输入
 - 如要终止当前打开的终端，就可以使用该方法

- 查看当前进程 ps
- 查看当前路径 pwd
- 切换路径 cd
- 查看文件信息 ls
- 查看文件内容 cat
- 查看当前用户 who

```
jiangnan@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[root@localhost ~]# ps  
  PID TTY          TIME CMD  
 2890 pts/0        00:00:00 su  
 2899 pts/0        00:00:00 bash  
 5284 pts/0        00:00:01 su  
 5334 pts/0        00:00:00 su  
 5342 pts/0        00:00:00 bash  
 5445 pts/0        00:00:00 ps  
[root@localhost ~]# pwd  
/root  
[root@localhost ~]# pwd  
/root  
[root@localhost ~]# cd  
[root@localhost ~]# ls  
anaconda-ks.cfg      公共    视频    文档    音乐  
initial-setup-ks.cfg 模板    图片    下载    桌面  
[root@localhost ~]#
```

4 Bash 常用快捷键

快捷键	作 用
ctrl+A	把光标移动到命令行开头。如果我们输入的命令过长，想要把光标移动到命令行开头时使用。
ctrl+E	把光标移动到命令行结尾。
ctrl+C	强制终止当前的命令。
ctrl+L	清屏，相当于 clear 命令。
ctrl+U	删除或剪切光标之前的命令。我输入了一行很长的命令，不用使用退格键一个一个字符的删除，使用这个快捷键会更加方便
ctrl+K	删除或剪切光标之后的内容。
ctrl+Y	粘贴 ctrl+U 或 ctrl+K 剪切的内容。
ctrl+R	在历史命令中搜索，按下 ctrl+R 之后，就会出现搜索界面，只要输入搜索内容，就会从历史命令中搜索。