

尚硅谷嵌入式技术之 Linux (Ubuntu)

(作者：尚硅谷研究院)

版本：V1.0.0

第 1 章 Linux 入门

1.1 概述



概述



Linux内核最初只是由芬兰人**林纳斯·托瓦兹** (Linus Torvalds) 在赫尔辛基大学上学时出于个人爱好而编写的。

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。Linux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。

目前市面上较知名的发行版有：**Ubuntu**、**RedHat**、**CentOS**、Debian、Fedora、SuSE、OpenSUSE。

让天下没有难学的技术

1.2 Linux 和 Windows 区别

Linux和Windows区别		尚硅谷
比较	Window	Linux
免费与收费	收费且很贵	Linux 免费或少许费用。
软件与支持	数量和质量的劣势，不过大部分为收费软件；由微软官方提供支持和服	开源自由软件，用户可以修改定制和再发布，由于基本免费没有资金支持，部分软件质量和体验欠缺；有全球所有的Linux开发者和自由软件社区提供支持。
安全性	三天两头打补丁安装系统安全更新，还是会中病毒木马；	要说 Linux 没有安全问题，那当然是不可能的，比PC操作系统更加稳定安全，不容易产生垃圾文件，适合长期运行。
使用习惯	普通用户基本都是纯图形界面下操作使用，依靠鼠标和键盘完成一切操作，用户上手容易入门简单；	兼具图形界面操作和完全的命令行操作，可以只用键盘完成一切操作，新手入门较困难，需要一些学习和指导，一旦熟练之后效率极高。
可定制性	封闭的，系统可定制性很差；	开源，可定制化非常强。
应用场景	桌面操作系统主要使用的是window。	支撑百度，谷歌，淘宝等应用软件和服务的，是后台成千上万的Linux服务器主机。世界上大部分软件和服务都是运行在Linux之上的。 <i>让天下没有难学的技术</i>

第 2 章 VMware、Ubuntu 与 Xshell 安装



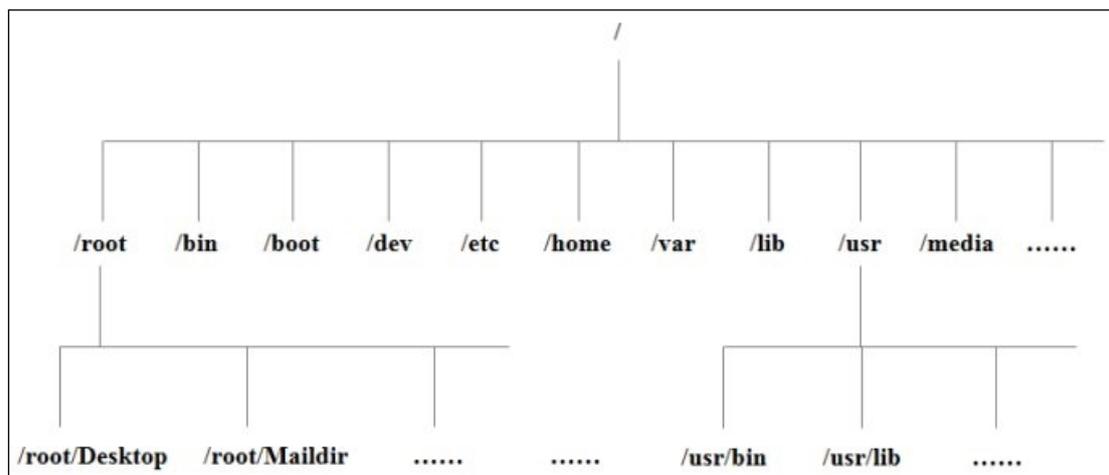
尚硅谷嵌入式技术
之VMware&Ubuntu

第 3 章 Linux 文件与目录结构

3.1 Linux 文件

Linux 系统中一切皆文件。

3.2 Linux 目录结构



➤ /bin

是 Binary 的缩写，这个目录存放着最经常使用的命令

➤ /sbin

s 就是 Super User 的意思，这里存放的是系统管理员使用的系统管理程序。

➤ /home

存放普通用户的主目录，在 Linux 中每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的。

➤ /root

该目录为系统管理员，也称作超级权限者的用户主目录。

➤ /lib

系统开机所需要最基本的动态连接共享库，其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。

➤ /lost+found

这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件。

➤ /etc

所有的系统管理所需要的配置文件和子目录。

➤ /usr

这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似于 windows 下的 program files 目录。

➤ /boot

这里存放的是启动 Linux 时使用的一些核心文件，包括一些连接文件以及镜像文件，自己的安装别放这里。

➤ /proc

这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息。

➤ /srv

service 缩写，该目录存放一些服务启动之后需要提取的数据。

➤ /sys

这是 linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 sysfs 。

➤ /tmp

这个目录是用来存放一些临时文件的。

➤ /dev

类似于 windows 的设备管理器，把所有的硬件用文件的形式存储。

➤ /media

linux 系统会自动识别一些设备，例如 U 盘、光驱等等，当识别后，linux 会把识别的设备挂载到这个目录下。

➤ /mnt

系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将外部的存储挂载在 /mnt/ 上，然后进入该目录就可以查看里的内容了。

➤ /opt

这是给主机额外安装软件所摆放的目录。比如你安装一个 mysql 数据库则就可以放到这个目录下。默认是空的。

➤ /var

这个目录中存放着在不断扩充着的东西，我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。

➤ /selinux

SELinux 是一种安全子系统,它能控制程序只能访问特定文件。

第 4 章 系统管理操作

4.1 关闭防火墙

4.1.1 systemctl

1) 基本语法

```
systemctl start | stop | restart | status 服务名
```

2) 经验技巧

查看服务的方法：查看 /usr/lib/systemd/system 目录下的文件列表，该目录下每个文件都对应一个服务。

```
atguigu@ubuntu:~$ cd /usr/lib/systemd/system
atguigu@ubuntu:/usr/lib/systemd/system$ pwd
/usr/lib/systemd/system
atguigu@ubuntu:/usr/lib/systemd/system$ ls -al
```

```
-rw-r--r--. 1 root root 275 4月 27 2018 abrt-ccpp.service
-rw-r--r--. 1 root root 380 4月 27 2018 abrt-d.service
-rw-r--r--. 1 root root 361 4月 27 2018 abrt-oops.service
-rw-r--r--. 1 root root 266 4月 27 2018 abrt-pstoreoops.service
-rw-r--r--. 1 root root 262 4月 27 2018 abrt-vmcore.service
-rw-r--r--. 1 root root 311 4月 27 2018 abrt-xorg.service
-rw-r--r--. 1 root root 751 4月 11 2018 accounts-daemon.service
-rw-r--r--. 1 root root 527 3月 25 2017 alsa-restore.service
-rw-r--r--. 1 root root 486 3月 25 2017 alsa-state.service
.....
```

3) 案例实操

(1) 查看网络服务的状态

```
atguigu@ubuntu:~/桌面$ sudo systemctl status NetworkManager
```

(2) 停止网络服务

```
atguigu@ubuntu:~/桌面$ sudo systemctl stop NetworkManager
```

(3) 启动网络服务

```
atguigu@ubuntu:~/桌面$ sudo systemctl start NetworkManager
```

(4) 重启网络服务

```
atguigu@ubuntu:~/桌面$ sudo systemctl restart NetworkManager
```

4.1.2 systemctl 设置后台服务的自启配置

1) 基本语法

`systemctl list-unit-files` (功能描述: 查看服务开机启动状态)

`systemctl disable service_name` (功能描述: 关掉指定服务的自动启动)

`systemctl enable service_name` (功能描述: 开启指定服务的自动启动)

4.1.3 关闭防火墙

1) 临时关闭防火墙

(1) 查看防火墙状态

```
atguigu@ubuntu:~/桌面$ sudo systemctl status ufw
```

(2) 临时关闭防火墙

```
atguigu@ubuntu:~/桌面$ sudo systemctl stop ufw
```

2) 开机启动时关闭防火墙

(1) 设置开机时启动防火墙

```
atguigu@ubuntu:~/桌面$ sudo systemctl enable ufw
```

(2) 设置开机时关闭防火墙

```
atguigu@ubuntu:~/桌面$ sudo systemctl disable ufw
```

(3) 查看服务是否开机自启

```
atguigu@ubuntu:~/桌面$ sudo systemctl is-enabled ufw
```

disabled 表示开机不自启

enabled 表示开机自启

4.2 关机重启命令

在 linux 领域内大多用在服务器上，很少遇到关机的操作。毕竟服务器上跑一个服务是永无止境的，除非特殊情况下，不得已才会关机。

1) sync 命令

(1) sync (功能描述：将数据由内存同步到硬盘中)

Linux 系统中为了提高磁盘的读写效率，对磁盘采取了“预读迟写”操作方式。当用户保存文件时，Linux 核心并不一定立即将保存数据写入物理磁盘中，而是将数据保存在缓冲区中，等缓冲区满时再写入磁盘，这种方式可以极大的提高磁盘写入数据的效率。但是，也带来了安全隐患，如果数据还未写入磁盘时，系统掉电或者其他严重问题出现，则将导致数据丢失。使用 sync 指令可以立即将缓冲区的数据写入磁盘。

现代 Linux 系统设计得足够智能，能够在执行关机或重启命令时自动处理必要的清理和同步操作。因此，对于绝大多数情况，直接使用关机或重启命令即可，而无需手动执行 sync。

我们的 Ubuntu 系统版本为 22.04.4，像其它现代 Linux 发行版一样，不需要在关机或重启之前手动执行 sync。

2) 关机和重启

(1) halt (功能描述：关闭系统，当前版本 Ubuntu 不会断电)

(2) poweroff (功能描述：关闭系统并断电，等同于 shutdown -h now)

(3) reboot (功能描述：重启，等同于 shutdown -r now)

(4) shutdown [选项] 时间

① shutdown 参数说明

选项	功能
-h	-h=halt 关机，不完全等同于 halt 命令
-r	-r=reboot 重启

② now 参数说明

参数	功能
now	立刻关机
时间	等待多久后关机（时间单位是分钟）

3) 案例实操

(1) 将数据由内存同步到硬盘中 (这一步可以不做)

```
atguigu@ubuntu:~/桌面$ sync
```

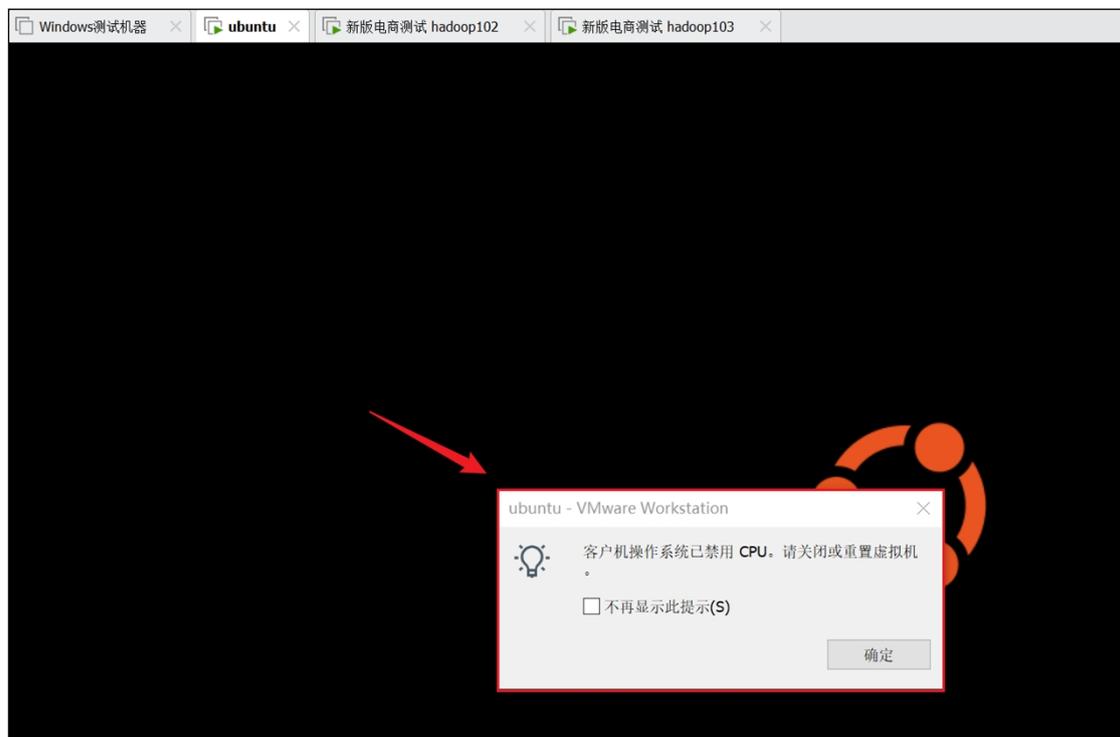
(2) 重启

```
atguigu@ubuntu:~/桌面$ sudo reboot
```

(3) 终止 CPU 的所有活动

```
atguigu@ubuntu:~/桌面$ sudo halt
```

当前系统环境下，此命令会终止 CPU 活动，但不会断电，执行上述命令后，VMWare 出现以下弹窗。



要关机或重启，需要借助 VMWare 完成。



(4) 计算机将在 1 分钟后关机，并且会显示在登录用户的当前屏幕中

```
atguigu@ubuntu:~/桌面$ sudo shutdown -h 1
```

```
Broadcast message from root@ubuntu on pts/1 (Tue 2024-03-05 14:55:53 CST):
```

```
The system is going down for poweroff at Tue 2024-03-05 14:56:53 CST!
```

```
Shutdown scheduled for Tue 2024-03-05 14:56:53 CST, use 'shutdown
```

```
-c' to cancel.
```

(5) 取消关机

```
atguigu@ubuntu:~/桌面$ sudo shutdown -c
```

```
Broadcast message from root@ubuntu on pts/1 (Tue 2024-03-05
14:56:05 CST):
```

```
The system shutdown has been cancelled
```

(6) 立马关机 (不完全等同于 `halt`, 会使系统断电)

```
atguigu@ubuntu:~/桌面$ sudo shutdown -h now
```

(7) 立马断电

```
atguigu@ubuntu:~/桌面$ sudo poweroff
```

(8) 系统立马重启 (等同于 `reboot`)

```
atguigu@ubuntu:~/桌面$ sudo shutdown -r now
```

4.3 修改主机名

```
atguigu@ubuntu-1:~/桌面$ sudo hostnamectl --static set-hostname
ubuntu
```

注: 终端命令提示符中@之后冒号之前的部分为当前的主机名: `ubuntu-1`, 上述命令更改主机名为 `ubuntu`, 执行 `hostname` 命令可以看到主机名已更改, 但命令提示符的更新需要重启机器。

4.4 修改 windows 机器的 hosts 文件

1) 修改 windows 的主机映射文件 (hosts 文件)

(1) 如果操作系统是 window7, 可以直接修改

① 进入 `C:\Windows\System32\drivers\etc` 路径

② 打开 `hosts` 文件并添加如下内容, 然后保存

```
192.168.10.150 ubuntu
```

(2) 如果操作系统是 window10, 先拷贝出来, 修改保存以后, 再覆盖即可

第 5 章 远程登录

通常在工作过程中, 公司中使用的真实服务器或者是云服务器, 都不允许除运维人员之外的员工直接接触, 因此就需要通过远程登录的方式来操作。所以, 远程登录工具就是必不可缺的, 目前, 比较主流的有 Xshell, SSH Secure Shell, SecureCRT, FinalShell 等, 同学们可以根据自己的习惯自行选择。

第 6 章 APT 软件包管理器

APT (Advanced Packaging Tools) 是 Debian 及其派生 Linux 的软件包管理器，可以自动下载，配置，安装二进制或者源代码格式的软件包，因此简化了 Unix 系统上管理软件的过程。

APT 常用命令：

用法： apt [选项] 命令

命令行软件包管理器 apt 提供软件包搜索，管理和信息查询等功能。它提供的功能与其他 APT 工具相同（像 apt-get 和 apt-cache），但是默认情况下被设置得更适合交互。

常用命令：

```
list - 根据名称列出软件包
search - 搜索软件包描述
show - 显示软件包细节
install - 安装软件包
reinstall - 重新安装软件包
remove - 移除软件包
autoremove - 卸载所有自动安装且不再使用的软件包
update - 更新可用软件包列表
upgrade - 通过 安装/升级 软件来更新系统
full-upgrade - 通过 卸载/安装/升级 来更新系统
edit-sources - 编辑软件源信息文件
satisfy - 使系统满足依赖关系字符串
```

1) 更新可用软件包列表

```
atguigu@ubuntu:~/桌面$ sudo apt update
```

2) 使用 APT 安装软件包

```
atguigu@ubuntu:~/桌面$ sudo apt install net-tools
```

常用参数：-y 不确认直接安装

3) 使用 APT 卸载软件包

```
atguigu@ubuntu:~/桌面$ sudo apt remove net-tools
```

常用参数：-y 不确认直接卸载

4) 使用 APT 搜索软件包

```
atguigu@ubuntu:~/桌面$ sudo apt search net-tools
```

第 7 章 常用基本命令

7.1 帮助命令

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

7.1.1 Manual Packages

1) 查看手册页说明文档的方式

```
atguigu@ubuntu:~/桌面$ man man
```

2) 手册页简介

(1) 名称

```
man - 系统参考手册的接口
```

(2) 概述

```
man [man 选项] [[章节] 页 ...] ...
man -k [apropos 选项] 正则表达式 ...
man -K [man 选项] [章节] 关键词 ...
man -f [whatis 选项] 页 ...
man -l [man 选项] 文件 ...
man -w|-W [man 选项] page ...
```

(3) 描述

手册页 (Manual Packages), 简称 “man pages”, 是 Unix 和类 Unix 系统 (包括 Linux 和 macOS) 上提供程序、函数、命令及文件格式文档的一种方式。手册页是用户和管理员获取命令用法、程序功能、配置文件规范和某些 API 函数描述的重要资源。

手册的章节号 (页) 及其包含的手册类型对应关系如下。

- 1-可执行程序或 shell 命令: 包含了绝大多数用户级别的外部命令或程序的文档, 这些命令通常位于用户的 PATH 环境变量指定的目录下, 如/bin、/usr/bin 等。
- 2-系统调用 (内核提供的函数): 提供了内核提供的系统调用的文档, 系统调用是应用程序与操作系统内核之间进行交互的接口。
- 3-库调用 (程序库中的函数): 包括标准 C 库函数和其他库函数的文档, 这些库函数提供了执行特定任务 (如字符串处理、文件操作) 的编程接口。
- 4-特殊文件 (通常位于/dev): 涉及到系统上的特殊文件, 如设备文件的说明。
- 5-文件格式和规范, 如/etc/passwd: 描述了各种文件格式和配置文件的结构, 比如/etc/passwd 或/etc/shadow 文件的格式。
- 6-游戏和屏保: 有些系统会在这一节中提供游戏和屏保程序的文档。
- 7-杂项 (包括宏包和规范): 包含了一些杂项文档, 如宏包、约定等。如 man(7), groff(7), man-pages(7)。
- 8-系统管理命令 (通常只针对 root 用户): 提供了系统管理员级别的命令或程序的文档, 这些命令通常位于/sbin、/usr/sbin 等目录。

➤ 9-内核例程（并非所有的发行版都有）：某些系统会提供内核级别函数和例程的文档。

我们常用的文档位于第 1、2、3、7 页。

(4) 手册页引用格式

MAN(1)
手册分页显示工具

名称
man - 系统参考手册的接口

概述
 man [man 选项] [[章节] 页 ...] ...
 man -k [apropos 选项] 正则表达式 ...
 man -K [man 选项] [章节] 关键词 ...
 man -f [whatis 选项] 页 ...
 man -l [man 选项] 文件 ...
 man -w|-W [man 选项] page ...

描述
 man 是系统的手册分页程序。指定给 man 的 页 选项通常是程序、工具或函数名。程序将显示手册页。如果指定了 章节, man 将只在手册的指定 章节 搜索。默认将按预定的顺序查找所默认值 一节), 并只显示找到的第一个 页, 即使多个 章节 中都有这个 页面。
 下表显示了手册的 章节 号及其包含的手册页类型。

执行 `man man` 命令, 进入手册页浏览模式, 左上角会显示 MAN (1), 这就是手册页引用格式或 man 引用格式, 括号前面是命令 (或系统调用等) 的名称, 括号内是命令所在的手册页编号。

当存在多个同名但功能不同的命令或调用时, 可以通过页编号区分。如: 用户命令 `write` 位于第一页, 用 `write (1)` 表示, 系统调用 `write` 位于第 2 页, 用 `write (2)` 表示。

7.1.2 man 获得帮助信息

1) 基本语法

`man [页编号] [命令或配置文件]` (功能描述: 获得帮助信息)

2) 显示说明

信息	功能
NAME	命令的名称和单行描述
SYNOPSIS	怎样使用命令
DESCRIPTION	命令功能的深入讨论
EXAMPLES	怎样使用命令的例子
SEE ALSO	相关主题 (通常是手册页)

3) 案例实操

(1) 查看 `ls` 命令的帮助信息

```
atguigu@ubuntu:~/桌面$ man ls
```

(2) 查看用户命令 `write` 的帮助信息

```
atguigu@ubuntu:~/桌面$ man write
```

WRITE(1)	User Commands	WRITE(1)
NAME		
write - send a message to another user		
write user [ttyname]		
DESCRIPTION		
write allows you to communicate with other users, by copying lines from your terminal to theirs.		
When you run the write command, the user you are writing to gets a message of the form:		
Message from yourname@yourhost on yourtty at hh:mm ...		

(3) 查看系统调用 write 的帮助信息

```
atguigu@ubuntu:~/桌面$ man 2 write
```

WRITE(2)	Linux Programmer's Manual	WRITE(2)
NAME		
write - write to a file descriptor		
SYNOPSIS		
#include <unistd.h>		
ssize_t write(int fd, const void *buf, size_t count);		
DESCRIPTION		
write() writes up to count bytes from the buffer starting at buf to the file referred to by the file descriptor fd.		

当手册页内存在多个同名命令（或系统调用等）时，不指定页编号，会从第一页开始扫描，展示第一个名称匹配的说明文档。同一页内不会存在名称相同的命令（或系统调用等）。如果想要搜索的不是第一个名称匹配的文档，则需要指定页编号。

7.1.3 help 获取 shell 内建命令的帮助信息

1) shell 内建命令

shell 内建命令是 shell 的一部分，他们没有单独的可执行文件或手册页，这类命令的文档通过 help 命令访问。

2) 基本语法

help 命令 （功能描述：获得 shell 内建命令的帮助信息）

3) 案例实操

(1) 查看 cd 命令的帮助信息

```
atguigu@ubuntu:~/桌面$ help cd
```

<pre>atguigu@ubuntu:~/桌面\$ help cd cd: cd [-L [-P [-e]] [-@]] [目录] 改变 shell 工作目录。 改变当前目录至 <目录>。默认的 <目录> 是 shell 变量 HOME 的值。 变量 CDPATH 定义了搜索含有 <目录> 的目录的搜索路径。CDPATH 中额外的目录名称以冒号 (:) 隔开。空的 CDPATH 表示当前目录。如果 <目录> 以斜杠 (/) 开头，则不会使用 CDPATH。 如果找不到目录，并且 shell 选项 "cdable_vars" 被设定，则假定参数是一个变量名。如果该变量有值，则它的值被当作 <目录>。</pre>
--

7.1.4 常用快捷键

常用快捷键	功能
ctrl + c	停止进程
ctrl+l	清屏；彻底清屏是：reset
ctrl + q	退出
善于用 tab 键	提示(更重要的是可以防止敲错)
上下键	查找执行过的命令
ctrl + u	清除当前敲的命令

7.2 文件目录类

7.2.1 pwd 显示当前工作目录的绝对路径

pwd:print working directory 打印工作目录

1) 基本语法

pwd (功能描述：显示当前工作目录的绝对路径)

2) 案例实操

(1) 显示当前工作目录的绝对路径

```
atguigu@ubuntu:~/桌面$ pwd
```

7.2.2 ls 列出目录的内容

ls:list 列出目录内容

1) 基本语法

ls [选项] [目录或是文件]

2) 选项说明

选项	功能
-a	全部的文件，连同隐藏档(开头为 . 的文件) 一起列出来(常用)
-l	长数据串列出，包含文件的属性与权限等等数据：(常用)

3) 显示说明

每行列出的信息依次是：文件类型与权限 链接数 文件属主 文件属组 文件大小用 byte 来表示 建立或最近修改的时间 名字

4) 案例实操

(1) 查看当前目录的所有内容信息

```
atguigu@ubuntu:~/桌面$ ls -al
总用量 44
drwx-----. 5 atguigu atguigu 4096 5月 27 15:15 .
drwxr-xr-x. 3 root root 4096 5月 27 14:03 ..
```

```
drwxrwxrwx. 2 root root 4096 5月 27 14:14 hello
-rwxrw-r--. 1 atguigu atguigu 34 5月 27 14:20 test.txt
```

(2) ubuntu 中 ll 是 ls -alF 的别名，我们可以使用 ll 查看目录下的所有文件

```
atguigu@ubuntu:~/桌面$ ll
总用量 44
drwx-----. 5 atguigu atguigu 4096 5月 27 15:15 .
drwxr-xr-x. 3 root root 4096 5月 27 14:03 ..
drwxrwxrwx. 2 root root 4096 5月 27 14:14 hello
-rwxrw-r--. 1 atguigu atguigu 34 5月 27 14:20 test.txt
```

7.2.3 cd 切换目录

cd:Change Directory 切换路径

1) 基本语法

cd [参数]

2) 参数说明

参数	功能
cd 绝对路径	切换路径
cd 相对路径	切换路径
cd ~或者 cd	回到自己的家目录
cd -	回到上一次所在目录
cd ..	回到当前目录的上一级目录
cd -P	跳转到实际物理路径，而非快捷方式路径

3) 案例实操

(1) 使用绝对路径切换到 root 目录

```
atguigu@ubuntu:~/桌面$ cd /root/
```

(2) 使用相对路径切换到“公共的”目录

```
atguigu@ubuntu:~/桌面$ cd 公共的/
```

(3) 表示回到自己的家目录，亦即是/root 这个目录

```
atguigu@ubuntu:~/桌面$ cd ~
```

(4) cd- 回到上一次所在目录

```
atguigu@ubuntu:~/桌面$ cd -
```

(5) 表示回到当前目录的上一级目录，亦即是 “/root/公共的” 的上一级目录的意思；

```
atguigu@ubuntu:~/桌面$ cd ..
```

7.2.4 mkdir 创建一个新的目录

mkdir:Make directory 建立目录

1) 基本语法

mkdir [选项] 要创建的目录

2) 选项说明

选项	功能
-p	创建多层目录

3) 案例实操

(1) 创建一个目录

```
atguigu@ubuntu:~/桌面$ mkdir xiyou
atguigu@ubuntu:~/桌面$ mkdir xiyou/qujing
```

(2) 创建一个多级目录

```
atguigu@ubuntu:~/桌面$ mkdir -p xiyou/dssz/meihouwang
```

7.2.5 touch 创建空文件

1) 基本语法

touch 文件名称

2) 案例实操

```
atguigu@ubuntu:~/桌面$ touch xiyou/dssz/sunwukong.txt
```

7.2.6 cp 复制文件或目录

1) 基本语法

cp [选项] source dest (功能描述: 复制 source 文件到 dest)

2) 选项说明

选项	功能
-r	递归复制整个文件夹

3) 参数说明

参数	功能
source	源文件
dest	目标文件

4) 经验技巧

强制覆盖不提示的方法: \cp

5) 案例实操

(1) 复制文件

```
atguigu@ubuntu:~/桌面$ cp xiyou/dssz/sunwukong.txt xiyou/qujing/
```

(2) 递归复制整个文件夹

```
atguigu@ubuntu:~/桌面$ cp -r xiyou/dssz/ ./
```

7.2.7 rm 删除文件或目录

1) 基本语法

`rm [选项] deleteFile` (功能描述: 递归删除目录中所有内容)

2) 选项说明

选项	功能
<code>-r</code>	递归删除目录中所有内容
<code>-f</code>	强制执行删除操作, 而不提示用于进行确认。
<code>-v</code>	显示指令的详细执行过程

3) 案例实操

(1) 删除目录中的内容

```
atguigu@ubuntu:~/桌面$ rm xiyou/qujing/sunwukong.txt
```

(2) 递归删除目录中所有内容

```
atguigu@ubuntu:~/桌面$ rm -rf dssz/
```

7.2.8 mv 移动文件与目录或重命名

1) 基本语法

(1) `mv oldNameFile newNameFile` (功能描述: 重命名)

(2) `mv /temp/movefile /targetFolder` (功能描述: 移动文件)

2) 案例实操

(1) 重命名

```
atguigu@ubuntu:~/桌面$ mv xiyou/dssz/sunwukong.txt
xiyou/dssz/houge.txt
```

(2) 移动文件

```
atguigu@ubuntu:~/桌面$ mv xiyou/dssz/houge.txt ./
```

7.2.9 cat 查看文件内容

查看文件内容, 从第一行开始显示。

1) 基本语法

`cat [选项] 要查看的文件`

2) 选项说明

选项	功能描述
<code>-n</code>	显示所有行的行号, 包括空行。

3) 经验技巧

一般查看比较小的文件, 一屏幕能显示全的。

4) 案例实操

(1) 查看文件内容并显示行号

```
atguigu@ubuntu:~/桌面$ cat -n hougex.txt
```

7.2.10 more 文件内容分屏查看器

`more` 指令是一个基于 VI 编辑器的文本过滤器，它以全屏幕的方式按页显示文本文件的内容。`more` 指令中内置了若干快捷键，详见操作说明。

1) 基本语法

`more` 要查看的文件

2) 操作说明

操作	功能说明
空白键 (space)	代表向下翻一页；
Enter	代表向下翻『一行』；
q	代表立刻离开 <code>more</code> ，不再显示该文件内容。
Ctrl+F	向下滚动一屏
Ctrl+B	返回上一屏
=	输出当前行的行号
:f	输出文件名和当前行的行号

3) 案例实操

(1) 复制 `.bashrc` 文件到当前目录

```
atguigu@ubuntu:~/桌面$ cp ../.bashrc ./
```

(2) 采用 `more` 查看文件

```
atguigu@ubuntu:~/桌面$ more .bashrc
```

7.2.11 less 分屏显示文件内容

`less` 指令用来分屏查看文件内容，它的功能与 `more` 指令类似，但是比 `more` 指令更加强大，支持各种显示终端。`less` 指令在显示文件内容时，并不是一次将整个文件加载之后才显示，而是根据显示需要加载内容，对于显示大型文件具有较高的效率。

1) 基本语法

`less` 要查看的文件

2) 操作说明

操作	功能说明
空白键	向下翻动一页；
[pagedown]	向下翻动一行
[pageup]	向上翻动一行；
/字符串	向下搜寻『字符串』的功能；n: 向下查找；N: 向上查找；
?字符串	向上搜寻『字符串』的功能；n: 向上查找；N: 向下查找；

q	离开 less 这个程序;
---	---------------

3) 经验技巧

用 SecureCRT 时[pagedown]和[pageup]可能会出现无法识别的问题。

4) 案例实操

(1) 采用 less 查看文件

```
atguigu@ubuntu:~/桌面$ less .bashrc
```

7.2.12 tail 输出文件尾部内容

tail 用于输出文件中尾部的内容，默认情况下 tail 指令显示文件的后 10 行内容。

1) 基本语法

- (1) tail 文件 (功能描述: 查看文件尾部 10 行内容)
- (2) tail -n 5 文件 (功能描述: 查看文件尾部 5 行内容, 5 可以是任意行数)
- (3) tail -f 文件 (功能描述: 实时追踪该文档的所有更新)

2) 选项说明

选项	功能
-n<行数>	输出文件尾部 n 行内容
-f	显示文件最新追加的内容, 监视文件变化

3) 案例实操

(1) 查看文件尾 1 行内容

```
atguigu@ubuntu:~/桌面$ tail -n 1 .bashrc
```

(2) 实时追踪该档的所有更新

```
atguigu@ubuntu:~/桌面$ tail -f hougou.txt
```

7.2.13 echo

echo 输出内容到控制台

1) 基本语法

echo [选项] [输出内容]

➤ 选项:

➤ -e: 支持反斜线控制的字符转换

控制字符	作用
\\	输出\本身
\n	换行符
\t	制表符, 也就是 Tab 键

2) 案例实操

```
atguigu@ubuntu:~/桌面$ echo "hello\tworld"
hello\tworld
atguigu@ubuntu:~/桌面$ echo -e "hello\tworld"
hello    world
```

7.2.14 >输出重定向和>>追加

1) 基本语法

- (1) `ls -l > 文件` (功能描述: 列表的内容写入文件 `a.txt` 中 (覆盖写))
- (2) `ls -al >> 文件` (功能描述: 列表的内容追加到文件 `aa.txt` 的末尾)
- (3) `cat 文件 1 > 文件 2` (功能描述: 将文件 1 的内容覆盖到文件 2)
- (4) `echo "内容" >> 文件`

2) 案例实操

- (1) 将 `ls` 查看信息写入到文件中

```
atguigu@ubuntu:~/桌面$ ls -l>houge.txt
```

- (2) 将 `ls` 查看信息追加到文件中

```
atguigu@ubuntu:~/桌面$ ls -l>>houge.txt
```

- (3) 采用 `echo` 将 `hello` 单词追加到文件中

```
atguigu@ubuntu:~/桌面$ echo hello>>houge.txt
```

7.2.15 ln 软链接

软链接也成为符号链接, 类似于 windows 里的快捷方式, 有自己的数据块, 主要存放了链接其他文件的路径。

1) 基本语法

- `ln -s [原文件或目录] [软链接名]` (功能描述: 给原文件创建一个软链接)

2) 经验技巧

删除软链接: `rm -rf 软链接名`, 而不是 `rm -rf 软链接名/`

查询: 通过 `ll` 就可以查看, 列表属性第 1 位是 `l`, 尾部会有位置指向。

3) 案例实操

- (1) 创建软连接

```
atguigu@ubuntu:~/桌面$ mv houge.txt xiyou/dssz/
atguigu@ubuntu:~/桌面$ ln -s xiyou/dssz/houge.txt ./houzi
atguigu@ubuntu:~/桌面$ ll
lrwxrwxrwx. 1 root  root    20 6月 17 12:56 houzi ->
xiyou/dssz/houge.txt
```

- (2) 删除软连接

```
atguigu@ubuntu:~/桌面$ rm -rf houzi
```

注意：`rm -rf houzi/` 这样删是删不掉的 不能再软连接后面加/

(3) 进入软连接实际物理路径

```
atguigu@ubuntu:~/桌面$ ln -s xiyou/dssz/ ./dssz
atguigu@ubuntu:~/桌面$ cd -P dssz/
```

7.2.16 history 查看已经执行过历史命令

1) 基本语法

`history` (功能描述：查看已经执行过历史命令)

2) 案例实操

(1) 查看已经执行过的历史命令

```
atguigu@ubuntu:~/桌面$ history
```

7.3 VI/VIM 编辑器

7.3.1 vi/vim 是什么

VI 是 Unix 操作系统和类 Unix 操作系统中最通用的文本编辑器。

VIM 编辑器是从 VI 发展出来的一个性能更强大的文本编辑器。可以主动的以字体颜色辨别语法的正确性，方便程序设计。VIM 与 VI 编辑器完全兼容。

在终端中执行以下命令安装 `vim`

```
atguigu@ubuntu:~/桌面$ sudo apt install vim
```

7.3.2 测试数据准备

(1) 拷贝 `/etc/profile` 数据到当前目录下

```
atguigu@ubuntu:~/桌面$ cp /etc/profile ./
```

7.3.3 一般模式

以 `vim` 打开一个档案就直接进入一般模式了（这是默认的模式）。在这个模式中，你可以使用『上下左右』按键来移动光标，你可以使用『删除字符』或『删除整行』来处理档案内容，也可以使用『复制、贴上』来处理你的文件数据。

1) 常用语法

语法	功能描述
<code>yy</code>	复制光标当前一行
<code>y 数字 y</code>	复制一段（从光标当前行到后 <code>n</code> 行）
<code>p</code>	箭头移动到目的行粘贴
<code>u</code>	撤销上一步

o	当前光标行的下一行
I	光标所在行最前
A	光标所在行最后
O	当前光标行的上一行

2) 退出编辑模式

按『Esc』键

7.3.5 指令模式

在一般模式当中，输入『:/?』3个中的任何一个按钮，就可以将光标移动到最底下那一行。

在这个模式当中，可以提供你『搜寻资料』的动作，而读取、存盘、大量取代字符、离开 vi、显示行号等动作是在此模式中达成的！

1) 基本语法

命令	功能
:w	保存
:q	退出
:! 	强制执行
/要查找的词	n 查找下一个，N 往上查找
:noh	取消高亮显示
:set nu	显示行号
:set nonu	关闭行号
:%s/old/new/g	替换内容 /g global 替换匹配到的所有内容

2) 案例实操

(1) 保存退出

对于有写权限的文件，修改后，保存并退出。

```
:wq
```

(2) 直接退出

没有修改文件内容，直接退出。

```
:q
```

(3) 强制退出

修改了文件内容，但是不想保存，此时需要强制退出。

```
:q!
```

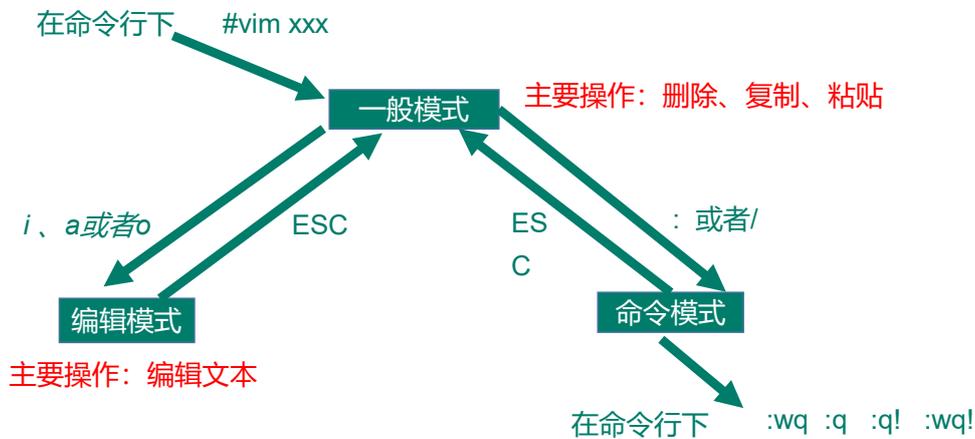
(4) 强制保存退出

对于没有写权限的文件，修改后，必须强制保存退出方可保留更改。

```
:wq!
```

7.3.6 模式间转换

模式间转换



让天下没有难学的技术

7.4 时间日期类

1) 基本语法

date [OPTION]... [+FORMAT]

2) 选项说明

选项	功能
-d<时间字符串>	显示指定的“时间字符串”表示的时间，而非当前时间
-s<日期时间>	设置系统日期时间

3) 参数说明

参数	功能
<+日期时间格式>	指定显示时使用的日期时间格式

7.4.2 date 显示当前时间

1) 基本语法

- (1) date (功能描述：显示当前时间)
- (2) date +%Y (功能描述：显示当前年份)
- (3) date +%m (功能描述：显示当前月份)
- (4) date +%d (功能描述：显示当前是哪一天)
- (5) date "+%Y-%m-%d %H:%M:%S" (功能描述：显示年月日时分秒)

2) 案例实操

- (1) 显示当前时间信息

```
atguigu@ubuntu:~/桌面$ date
2017年 06月 19日 星期一 20:53:30 CST
```

(2) 显示当前时间年月日

```
atguigu@ubuntu:~/桌面$ date +%Y%m%d
20170619
```

(3) 显示当前时间年月日时分秒

```
atguigu@ubuntu:~/桌面$ date "+%Y-%m-%d %H:%M:%S"
2017-06-19 20:54:58
```

7.4.3 date 显示非当前时间

1) 基本语法

(1) `date -d '1 days ago'` (功能描述: 显示前一天时间)

(2) `date -d '-1 days ago'` (功能描述: 显示明天时间)

2) 案例实操

(1) 显示前一天

```
atguigu@ubuntu:~/桌面$ date -d '1 days ago'
2017年 06月 18日 星期日 21:07:22 CST
```

(2) 显示明天时间

```
atguigu@ubuntu:~/桌面$ date -d '-1 days ago'
2017年 06月 20日 星期日 21:07:22 CST
```

7.4.4 date 设置系统时间

1) 基本语法

`date -s 字符串时间`

2) 案例实操

(1) 设置系统当前时间

```
atguigu@ubuntu:~/桌面$ date -s "2017-06-19 20:52:18"
```

7.5 用户管理命令

7.5.1 adduser 添加新用户

1) 基本语法

`adduser 用户名` (功能描述: 添加新用户)

应用场景 1: 企业开发, 多人协同 (也会有多人使用相同的一个低权限用户)。

应用场景 2: 框架协同 `gitlab` `mysql` `redis`

2) 案例实操

(1) 添加一个用户

```
atguigu@ubuntu:/home$ sudo adduser tangseng
正在添加用户 "tangseng"...
正在添加新组 "tangseng" (1001)...
正在添加新用户 "tangseng" (1001) 到组 "tangseng"...
创建主目录 "/home/tangseng"...
正在从 "/etc/skel" 复制文件...
新的密码:
无效的密码: 密码是一个回文
重新输入新的密码:
passwd: 已成功更新密码
正在改变 tangseng 的用户信息
请输入新值, 或直接敲回车键以使用默认值
  全名 []: 唐僧
  房间号码 []: 1
  工作电话 []:
  家庭电话 []:
  其它 []:
chfn: 名称中有非 ASCII 字符: "唐僧"
这些信息是否正确? [Y/n] y
```

按提示输入密码、用户信息即可。

```
atguigu@ubuntu:~/桌面$ ll /home/
```

可以看到以下内容。

```
总计 16
drwxr-xr-x  4 root    root    4096  3月  5 21:22 ./
drwxr-xr-x 20 root    root    4096  3月  4 15:50 ../
drwxr-x--- 17 atguigu atguigu 4096  3月  5 19:06 atguigu/
drwxr-x---  2 tangseng tangseng 4096  3月  5 21:22 tangseng/
```

7.5.2 passwd 设置或更改用户密码

1) 基本语法

`passwd 用户名` (功能描述: 设置用户密码)

2) 案例实操

(1) 更改用户的密码

```
atguigu@ubuntu:~/桌面$ sudo passwd tangseng
```

7.5.3 id 查看用户是否存在

1) 基本语法

`id 用户名`

2) 案例实操

- (1) 查看用户是否存在

```
atguigu@ubuntu:~/桌面$ id tangseng
```

7.5.4 cat /etc/passwd 查看创建了哪些用户

1) 基本语法

```
atguigu@ubuntu:~/桌面$ cat /etc/passwd
```

7.5.5 su 切换用户

su: swith user 切换用户

1) 基本语法

su 用户名称 (功能描述: 切换用户, 只能获得用户的执行权限, 不能获得环境变量)

su - 用户名称 (功能描述: 切换到用户并获得该用户的环境变量及执行权限)

2) 案例实操

- (1) 切换用户

```
atguigu@ubuntu:~/桌面$ su tangseng

atguigu@ubuntu:~/桌面$ echo $PATH
/usr/lib64/qt-
3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/
bin:/root/bin
atguigu@ubuntu:~/桌面$ exit
atguigu@ubuntu:~/桌面$ su - tangseng
atguigu@ubuntu:~/桌面$ echo $PATH
/usr/lib64/qt-
3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/s
bin:/home/tangseng/bin
```

7.5.6 userdel 删除用户

1) 基本语法

(1) userdel 用户名 (功能描述: 删除用户但保存用户主目录)

(2) userdel -r 用户名 (功能描述: 用户和用户主目录, 都删除)

2) 选项说明

选项	功能
-r	删除用户的同时, 删除与用户相关的所有文件。

3) 案例实操

- (1) 删除用户但保存用户主目录

```
atguigu@ubuntu:~/桌面$ sudo userdel tangseng
```

```
atguigu@ubuntu:~/桌面$ ll /home/
```

(2) 删除用户和用户主目录，都删除

```
atguigu@ubuntu:~/桌面$ sudo adduser zhubajie
```

```
atguigu@ubuntu:~/桌面$ ll /home/
```

```
atguigu@ubuntu:~/桌面$ sudo userdel -r zhubajie
```

```
atguigu@ubuntu:~/桌面$ ll /home/
```

7.5.7 usermod 修改用户

1) 基本语法

```
usermod -l 新用户名 老用户名
```

```
usermod -d /home/新用户名 -m 新用户名
```

2) 选项说明

选项	功能
-l	改变用户名
-d	修改家目录

3) 案例实操

(1) 改变用户名

```
atguigu@ubuntu:~$ sudo usermod -l meihouwang sunwukong
```

(2) 更改家目录

```
atguigu@ubuntu:~$ sudo usermod -d /home/meihouwang -m meihouwang
```

7.6 用户组管理命令

每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同 Linux 系统对用户组的规定有所不同。

如 Linux 下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建。

用户组的管理涉及用户组的添加、删除和修改。组的增加、删除和修改实际上就是对 `/etc/group` 文件的更新。

7.6.1 groupadd 新增组

1) 基本语法

```
groupadd 组名
```

2) 案例实操

(1) 添加一个 xitianqujing 组

```
atguigu@ubuntu:~$ sudo groupadd xitianqujing
```

7.6.2 groupdel 删除组

1) 基本语法

groupdel 组名

2) 案例实操

(1) 删除 xitianqujing 组

```
atguigu@ubuntu:~$ sudo groupdel xitianqujing
```

7.6.3 groupmod 修改组

1) 基本语法

groupmod -n 新组名 老组名

2) 选项说明

选项	功能描述
-n<新组名>	指定工作组的新组名

3) 案例实操

(1) 修改 xitianqujing 组名称为 xitian

```
atguigu@ubuntu:~$ sudo groupadd xitianqujing
atguigu@ubuntu:~$ sudo groupmod -n xitian xitianqujing
```

7.6.4 usermod 修改用户主组

在 Linux 和 Unix 系统中，每个用户都有一个主组（primary group）和可能的多个附加组（secondary groups 或 additional groups）。

用户的主组在用户创建时被指定，默认与用户名称相同，当用户创建一个新文件或目录时，默认情况下，这些文件或目录会被分配给用户的主组。

1) 基本语法

usermod -g 组名 用户名

2) 选项说明

选项	功能描述
-g	指定用户的新主组

3) 案例实操

(1) 查看用户主组

默认情况下用户的家目录会被分配给主组。

```
atguigu@ubuntu:~$ sudo adduser zhubajie
atguigu@ubuntu:~$ ll /home
```

```
atguigu@ubuntu:~$ ll /home
总计 24
drwxr-xr-x 6 root      root      4096  3月  6 14:20 ./
drwxr-xr-x 20 root     root      4096  3月  4 15:50 ../
drwxr-x--- 17 atguigu  atguigu   4096  3月  6 14:44 atguigu/
drwxr-x---  2 meihouwang sunwukong 4096  3月  6 13:51 meihouwang/
drwxr-x---  2 meihouwang sunwukong 4096  3月  5 21:25 tangseng/
drwxr-x---  2 zhubajie  zhubajie  4096  3月  6 14:20 zhubajie/
```

用户组名

可以看到，zhubajie 所属的主组为 zhubajie。

(2) 切换用户主组

```
atguigu@ubuntu:~$ sudo usermod -g xitian zhubajie
atguigu@ubuntu:~$ ll /home
总计 24
drwxr-xr-x 6 root      root      4096  3月  6 14:20 ./
drwxr-xr-x 20 root     root      4096  3月  4 15:50 ../
drwxr-x--- 17 atguigu  atguigu   4096  3月  6 14:44 atguigu/
drwxr-x---  2 meihouwang sunwukong 4096  3月  6 13:51 meihouwang/
drwxr-x---  2 meihouwang sunwukong 4096  3月  5 21:25 tangseng/
drwxr-x---  2 zhubajie  xitian    4096  3月  6 14:20 zhubajie/
```

7.6.5 查看附加组和用户的映射关系

/etc/group 文件存储了用户和附加组的映射关系，每一行对应一个用户组，第三个冒号后面是以该组作为附加组的用户列表，列表为空表示没有用户将其作为附加组。

1) 基本操作

```
atguigu@ubuntu:~/桌面$ cat /etc/group
```

7.6.6 将用户添加到附加组

1) 基本语法

usermod -aG 组名 用户名

2) 选项说明

选项	功能描述
-aG	指定用户需要加入的附加组

3) 案例实操

(1) 查看 atguigu 组的用户列表

```
atguigu@ubuntu:~$ sudo cat /etc/group
lxd:x:135:atguigu
atguigu:x:1000:
sambashare:x:136:atguigu
sunwukong:x:1001:
xitian:x:1002:
```

atguigu 组的用户列表为空。

(2) 将 atguigu 作为 zhubajie 的附加组

```
atguigu@ubuntu:~$ sudo usermod -aG atguigu zhubajie
atguigu@ubuntu:~$ sudo cat /etc/group
```

```
pulse-access:x:133:
gdm:x:134:
lxd:x:135:atguigu
atguigu:x:1000:zhubajie
sambashare:x:136:atguigu
sunwukong:x:1001:
```

7.6.7 将用户从组中移除

1) 基本语法

deluser 用户名 组名

2) 案例实操

```
atguigu@ubuntu:~$ sudo deluser zhubajie atguigu
正在将用户"zhubajie"从组"atguigu"中删除...
完成。
```

7.6.8 sudo 设置普通用户具有 root 权限

sudo 是将对应的命令给到 root 用户去执行。

1) 将 meihouwang 更名为 sunwukong

```
atguigu@ubuntu:~$ sudo usermod -l sunwukong meihouwang
atguigu@ubuntu:~$ sudo usermod -d /home/sunwukong -m sunwukong
```

2) 修改配置文件

```
atguigu@ubuntu:~/桌面$ sudo vim /etc/sudoers
```

找到下面一行(50行)，如下所示：

```
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL
```

这行的作用是允许 sudo 组的所有成员执行任何命令，换言之，该组成员都拥有了 root 权限。但是通过 sudo 命令操作时需要输入密码。在最后一个 ALL 前添加 NOPASSWD:，则该组的成员通过 sudo 命令操作时不必输入密码。

```
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) NOPASSWD:ALL
```

保存退出。注意：sudoers 文件没有写权限，保存退出要用 wq!。

3) 查看 sudo 组的成员

```
atguigu@ubuntu:~$ sudo cat /etc/group
```

执行上述命令可以查看 `sudo` 组的成员，如下。

```
cdrom:x:24:atguigu
floppy:x:25:
tape:x:26:
sudo:x:27:atguigu
audio:x:29:pulse
dip:x:30:atguigu
```

可以看到，`atguigu` 用户已经在 `sudo` 组中，因此之前我们并没有做 `sudo` 相关的配置，但是 `atguigu` 用户却可以获得 `root` 权限。

4) 将 `sunwukong` 添加到 `sudo` 组中

```
atguigu@ubuntu:~/home$ sudo usermod -aG sudo sunwukong
```

5) 重新查看 `sudo` 组的成员

```
atguigu@ubuntu:~$ sudo cat /etc/group
```

```
cdrom:x:24:atguigu
floppy:x:25:
tape:x:26:
sudo:x:27:atguigu,sunwukong
audio:x:29:pulse
dip:x:30:atguigu
```

6) 案例实操

(1) 切换到 `sunwukong`

```
atguigu@ubuntu:~$ su - sunwukong
```

(2) 用普通用户 `sunwukong` 查看 `/etc` 下的 `hostname` 文件

```
sunwukong@ubuntu:~$ cat /etc/sudoers
```

```
cat: /etc/sudoers: 权限不够
```

使用 `root` 权限查看。

```
sunwukong@ubuntu:~$ sudo cat /etc/sudoers
```

普通控制台会打印 `sudoers` 文件的内容

```
sunwukong@ubuntu:~$ sudo cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
```

(3) 切换回 `atguigu` 用户

```
sunwukong@ubuntu:~$ exit
```

注销

7.7 文件权限类

7.7.1 文件属性

能力越大，责任越大。权限越小，责任越小。

Linux 系统是一种典型的多用户系统，不同的用户处于不同的地位，拥有不同的权限。为了保护系统的安全性，Linux 系统对不同的用户访问同一文件（包括目录文件）的权限做了不同的规定。在 Linux 中我们可以使用 ll 或者 ls -l 命令来显示一个文件的属性以及文件所属的用户和组。

1) 文件属性：从左到右的 10 个字符表示

文件类型	属主权限			属组权限			其他用户权限		
0	1	2	3	4	5	6	7	8	9
d	r	w	x	r	-	x	r	-	x
目录文件	读	写	执行	读	写	执行	读	写	执行

如果没有权限，就会出现减号[-]而已。从左至右用 0-9 这些数字来表示：

(1) 0 首位表示类型

在 Linux 中第一个字符代表这个文件是目录、文件或链接文件等等

- 代表文件

d 代表目录

l 链接文档(link file);

(2) 第 1-3 位确定属主（该文件的所有者）拥有该文件的权限。---User

(3) 第 4-6 位确定属组（所有者的同组用户）拥有该文件的权限，---Group

(4) 第 7-9 位确定其他用户拥有该文件的权限 ---Other

2) rxw 作用文件和目录的不同解释

(1) 作用到文件：

➤ [r]代表可读（read）：可以读取，查看

➤ [w]代表可写（write）：可以修改，但是不代表可以删除该文件，删除一个文件的前提条件是对该文件所在的目录有写权限，才能删除该文件。

➤ [x]代表可执行（execute）：可以被系统执行

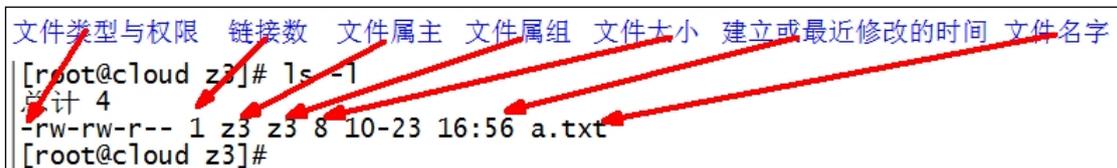
(2) 作用到目录：0

- [r]代表可读 (read) : 可以读取, ls 查看目录内容
- [w]代表可写 (write) : 可以修改, 目录内创建+删除+重命名目录
- [x]代表可执行 (execute) : 可以进入该目录

3) 案例实操

```
atguigu@ubuntu:~/桌面$ ll
总用量 104
-rw-----. 1 root root 1248 1月 8 17:36 anaconda-ks.cfg
drwxr-xr-x. 2 root root 4096 1月 12 14:02 dssz
lrwxrwxrwx. 1 root root 20 1月 12 14:32 houzi ->
xiyou/dssz/houge.tx
```

(1) 文件基本属性介绍



(2) 如果查看到是文件: 链接数指的是硬链接个数。创建硬链接方法

ln [原文件] [目标文件]

```
atguigu@ubuntu:~/桌面$ ln xiyou/dssz/houge.txt ./hg.txt
```

(3) 如果查看的是文件夹: 链接数指的是子文件夹个数。

```
atguigu@ubuntu:~/桌面$ ls -al xiyou/
总用量 16
drwxr-xr-x. 4 root root 4096 1月 12 14:00 .
dr-xr-x---. 29 root root 4096 1月 12 14:32 ..
drwxr-xr-x. 2 root root 4096 1月 12 14:30 dssz
drwxr-xr-x. 2 root root 4096 1月 12 14:04 qujing
```

7.7.2 chmod 改变权限

1) 基本语法

文件类型	属主	属组	其他用户
0	1 u 2 3	4 g 5 6	7 o 8 9
d	rwX	r-X	r-X
目录文件	读 写 执行	读 写 执行	读 写 执行

(1) 第一种方式变更权限

chmod [{ugoa}{+|=}{rwx]} 文件或目录

(2) 第二种方式变更权限

```
chmod [mode=421 ] [文件或目录]
```

2) 经验技巧

U: 所有者 g: 所有组 o: 其他人 a: 所有人 (u、g、o 的总和)

r=4 w=2 x=1 rwx=4+2+1=7

3) 案例实操

(1) 修改文件使其所属主用户具有执行权限

```
atguigu@ubuntu:~/桌面$ cp xiyou/dssz/houge.txt ./
atguigu@ubuntu:~/桌面$ chmod u+x houge.txt
```

(2) 修改文件使其所属组用户具有执行权限

```
atguigu@ubuntu:~/桌面$ chmod g+x houge.txt
```

(3) 修改文件所属主用户执行权限,并使其他用户具有执行权限

```
atguigu@ubuntu:~/桌面$ chmod u-x,o+x houge.txt
```

(4) 采用数字的方式, 设置文件所有者、所属组、其他用户都具有可读可写可执行权限。

```
atguigu@ubuntu:~/桌面$ chmod 777 houge.txt
```

(5) 修改整个文件夹里面的所有文件的所有者、所属组、其他用户都具有可读可写可执行权限。

```
atguigu@ubuntu:~/桌面$ chmod -R 777 xiyou/
```

7.7.3 chown 改变所有者

1) 基本语法

chown [选项] [最终用户] [文件或目录] (功能描述 改变文件或者目录的所有者)

2) 选项说明

选项	功能
-R	递归操作

3) 案例实操

(1) 修改文件所有者

```
atguigu@ubuntu:~/桌面$ sudo chown sunwukong houge.txt
atguigu@ubuntu:~/桌面$ ll
...
-rwxrwxrwx  1 sunwukong atguigu  367  3月  6 15:44 houge.txt
...
```

(2) 递归改变文件所有者和所有组

```
atguigu@ubuntu:~/桌面$ cd xiyou/
```

```

atguigu@ubuntu:~/桌面/xiyou$ ll
总计 16
drwxrwxrwx 4 atguigu atguigu 4096 3月 6 15:42 ./
drwxr-xr-x 3 atguigu atguigu 4096 3月 6 15:44 ../
-rwxrwxrwx 1 atguigu atguigu 0 3月 6 15:42 a*
drwxrwxrwx 3 atguigu atguigu 4096 3月 5 19:06 dssz/
drwxrwxrwx 2 atguigu atguigu 4096 3月 5 18:57 qujing/

atguigu@ubuntu:~/桌面/xiyou$ sudo chown -R
sunwukong:sunwukong ../xiyou/
atguigu@ubuntu:~/桌面/xiyou$ ll
总计 16
drwxrwxrwx 4 sunwukong sunwukong 4096 3月 6 15:42 ./
drwxr-xr-x 3 atguigu atguigu 4096 3月 6 15:44 ../
-rwxrwxrwx 1 sunwukong sunwukong 0 3月 6 15:42 a*
drwxrwxrwx 3 sunwukong sunwukong 4096 3月 5 19:06 dssz/
drwxrwxrwx 2 sunwukong sunwukong 4096 3月 5 18:57 qujing/
    
```

7.7.4 chgrp 改变所属组

1) 基本语法

`chgrp [最终用户组] [文件或目录]` (功能描述: 改变文件或者目录的所属组)

2) 案例实操

(1) 修改文件的所属组

```

atguigu@ubuntu:~/桌面/xiyou$ cd ..
atguigu@ubuntu:~/桌面$ ll
总计 28
drwxr-xr-x 3 atguigu atguigu 4096 3月 6 15:44 ./
drwxr-x--- 17 atguigu atguigu 4096 3月 6 15:27 ../
-rw-rw-r-- 1 atguigu atguigu 0 3月 6 15:42 a
-rw-r--r-- 1 atguigu atguigu 3788 3月 5 18:59 .bashrc
lrwxrwxrwx 1 atguigu atguigu 11 3月 5 19:07 dssz ->
xiyou/dssz//
-rwxrwxrwx 2 sunwukong sunwukong 367 3月 5 19:05 hg.txt*
-rwxrwxrwx 1 sunwukong atguigu 367 3月 6 15:44 hougex.txt*
-rw-r--r-- 1 atguigu atguigu 582 3月 5 15:31 profile
drwxrwxrwx 4 sunwukong sunwukong 4096 3月 6 15:42 xiyou/
atguigu@ubuntu:~/桌面$ sudo chgrp root hougex.txt
atguigu@ubuntu:~/桌面$ ll
总计 28
drwxr-xr-x 3 atguigu atguigu 4096 3月 6 15:44 ./
drwxr-x--- 17 atguigu atguigu 4096 3月 6 15:27 ../
-rw-rw-r-- 1 atguigu atguigu 0 3月 6 15:42 a
    
```

```

-rw-r--r--  1 atguigu  atguigu  3788  3月  5  18:59  .bashrc
lrwxrwxrwx  1 atguigu  atguigu   11  3月  5  19:07  dssz ->
xiyou/dssz//
-rwxrwxrwx  2 sunwukong sunwukong  367  3月  5  19:05  hg.txt*
-rwxrwxrwx  1 sunwukong root      367  3月  6  15:44  houge.txt*
-rw-r--r--  1 atguigu  atguigu   582  3月  5  15:31  profile
drwxrwxrwx  4 sunwukong sunwukong 4096  3月  6  15:42  xiyou/
    
```

7.8 搜索查找类

7.8.1 find 查找文件或者目录

find 指令将从指定目录向下递归地遍历其各个子目录，将满足条件的文件显示在终端。

1) 基本语法

find [搜索范围] [选项]

2) 选项说明

选项	功能
-name<查询方式>	按照指定的文件名查找模式查找文件
-user<查询方式>	查找属于指定用户名所有文件
-size<文件大小>	按照指定的文件大小查找文件，单位为： b —— 块（512 字节） c —— 字节 w —— 字（2 字节） k —— 千字节 M —— 兆字节 G —— 吉字节

3) 案例实操

(1) 按文件名：根据名称查找当前目录下所有以.txt 结尾的文件。

```
atguigu@ubuntu:~/桌面$ find ./ -name "*.txt"
```

(2) 按拥有者：查找当前目录下，用户名称为-user 的文件

```
atguigu@ubuntu:~/桌面$ find ./ -user atguigu
```

(3) 按文件大小：在当前目录下查找大于 200 字节的文件（+n 大于 -n 小于 n 等于）

```
atguigu@ubuntu:~/桌面$ find ./ -size +200c
```

7.8.2 grep 过滤查找及“|”管道符

管道符，“|”，表示将前一个命令的处理结果输出传递给后面的命令处理。

1) 基本语法

grep 选项 查找内容 源文件

2) 选项说明

选项	功能
-n	显示匹配行及行号。

3) 案例实操

(1) 查找某文件在第几行

```
atguigu@ubuntu:~/桌面$ ls | grep -n houg
```

7.9 压缩和解压类

7.9.1 gzip/gunzip 压缩

1) 基本语法

`gzip 文件` (功能描述: 压缩文件, 只能将文件压缩为*.gz 文件)

`gunzip 文件.gz` (功能描述: 解压缩文件命令)

2) 经验技巧

(1) 只能压缩文件不能压缩目录

(2) 不保留原来的文件

3) 案例实操

(1) gzip 压缩

```
atguigu@ubuntu:~/桌面$ ls
a dssz hg.txt houg.txt profile xiyou
atguigu@ubuntu:~/桌面$ gzip houg.txt
atguigu@ubuntu:~/桌面$ ls | grep houg
houg.txt.gz
```

(2) gunzip 解压缩文件

```
atguigu@ubuntu:~/桌面$ gunzip houg.txt.gz
atguigu@ubuntu:~/桌面$ ls | grep houg
houg.txt
```

7.9.2 tar 打包

1) 基本语法

`tar [选项] XXX.tar.gz` 将要打包进去的内容 (功能描述: 打包目录, 压缩后的文件格式.tar.gz)

2) 选项说明

选项	功能
-c	产生.tar 打包文件
-v	显示详细信息

-f	指定压缩后的文件名
-z	打包同时压缩
-x	解包.tar 文件

3) 案例实操

(1) 压缩多个文件

```
atguigu@ubuntu:~/桌面$ touch bailongma.txt

atguigu@ubuntu:~/桌面$ tar -zcvf houma.tar.gz houge.txt
bailongma.txt

[root@hadoop101 opt]# ls
a bailongma.txt dssz hg.txt houge.txt houma.tar.gz profile
xiyou
```

(2) 压缩目录

```
atguigu@ubuntu:~/桌面$ tar -zcvf xiyou.tar.gz xiyou/
xiyou/
xiyou/qujing/
xiyou/dssz/
xiyou/dssz/houge.txt

atguigu@ubuntu:~/桌面$ ls
a bailongma.txt dssz hg.txt houge.txt houma.tar.gz profile
xiyou xiyou.tar.gz
```

(3) 解压到当前目录

```
atguigu@ubuntu:~/桌面$ mkdir houma_test
atguigu@ubuntu:~/桌面$ cd houma_test/
atguigu@ubuntu:~/桌面/houma_test$ mv ../houma.tar.gz ./
atguigu@ubuntu:~/桌面/houma_test$ ls
houma.tar.gz

atguigu@ubuntu:~/桌面/houma_test$ tar -zxvf houma.tar.gz
houge.txt
bailongma.txt
atguigu@ubuntu:~/桌面/houma_test$ ls
bailongma.txt houge.txt houma.tar.gz
```

(4) 解压到指定目录

```
atguigu@ubuntu:~/桌面$ mkdir xiyou2
atguigu@ubuntu:~/桌面$ tar -zxvf xiyou.tar.gz -C ./xiyou2
atguigu@ubuntu:~/桌面$ ls xiyou2/
xiyou
```

7.10 磁盘类

7.10.1 df 查看磁盘空间使用情况

df: disk free 空余硬盘

1) 基本语法

df 选项 (功能描述: 列出文件系统的整体磁盘使用量, 检查文件系统的磁盘空间占用情况)

2) 选项说明

选项	功能
-h	以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示;

3) 案例实操

(1) 查看磁盘使用情况

```
atguigu@ubuntu:~/桌面$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       15G   3.5G   11G   26% /
tmpfs           939M   224K   939M    1% /dev/shm
/dev/sda1       190M   39M   142M   22% /boot
```

7.10.2 du 文件和目录的磁盘使用空间

1) 基本语法

du 目录/文件 (功能描述: 统计文件或递归显示目录及子目录的磁盘使用空间)

2) 选项说明

选项	功能
-a	显示当前目录下所有的文件目录及子目录大小

3) 案例实操

(1) 查看文件的空间使用情况

```
atguigu@ubuntu:~/桌面$ du houg.txt
4 houg.txt
```

(2) 查看目录及子目录的空间使用情况

```
atguigu@ubuntu:~/桌面$ du xiyou
4 xiyou/dssz/meihouwang
12 xiyou/dssz
4 xiyou/qujing
20 xiyou
```

(3) 查看目录、子目录及目录下所有文件的空间使用情况

```
atguigu@ubuntu:~/桌面$ du -a xiyou
4 xiyou/dssz/meihouwang
4 xiyou/dssz/houg.txt
```

```
12 xiyou/dssz
4 xiyou/qujing
0 xiyou/a
20 xiyou
```

7.11 进程线程类

进程是正在执行的一个程序或命令，每一个进程都是一个运行的实体，都有自己的地址空间，并占用一定的系统资源。

7.11.1 ps 查看当前系统进程状态

ps:process status 进程状态

1) 基本语法

ps -aux | grep xxx (功能描述：查看系统中所有进程)

ps -ef | grep xxx (功能描述：可以查看子父进程之间的关系)

2) 选项说明

选项	功能
-a	选择所有进程
-u	显示所有用户的所有进程
-x	显示没有终端的进程

3) 功能说明

(1) ps -aux 显示信息说明

- USER: 该进程是由哪个用户产生的
- PID: 进程的 ID 号
- %CPU: 该进程占用 CPU 资源的百分比，占用越高，进程越耗费资源；
- %MEM: 该进程占用物理内存的百分比，占用越高，进程越耗费资源；
- VSZ: 该进程占用虚拟内存的大小，单位 KB；
- RSS: 该进程占用实际物理内存的大小，单位 KB；
- TTY: 该进程是在哪个终端中运行的。其中 tty1-tty7 代表本地控制台终端，tty1-tty6 是本地的字符界面终端，tty7 是图形终端。pts/0-255 代表虚拟终端。

是本地的字符界面终端，tty7 是图形终端。pts/0-255 代表虚拟终端。

➤ STAT: 进程状态。常见的状态有：R: 运行、S: 睡眠、T: 停止状态、s: 包含子进程、+: 位于后台

➤ START: 该进程的启动时间

➤ TIME: 该进程占用 CPU 的运算时间，注意不是系统时间

- COMMAND: 产生此进程的命令名
- (2) ps -ef 显示信息说明
- UID: 用户 ID
- PID: 进程 ID
- PPID: 父进程 ID
- C: CPU 用于计算执行优先级的因子。数值越大, 表明进程是 CPU 密集型运算,

执行优先级会降低; 数值越小, 表明进程是 I/O 密集型运算, 执行优先级会提高

- STIME: 进程启动的时间
- TTY: 完整的终端名称
- TIME: CPU 时间
- CMD: 启动进程所用的命令和参数

4) 经验技巧

如果想查看进程的 CPU 占用率和内存占用率, 可以使用 `aux`;

如果想查看进程的父进程 ID 可以使用 `ef`;

5) 案例实操

- (1) 查看进程的 CPU 占用率和内存占用率

```
atguigu@ubuntu:~/桌面$ ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	19348	1544	?	Ss	10:55	0:01	/sbin/init
root	2	0.0	0.0	0	0	?	S	10:55	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	10:55	0:00	[migration/0]
root	4	0.0	0.0	0	0	?	S	10:55	0:00	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S	10:55	0:00	[stopper/0]

- (2) 查看进程的父进程 ID

```
atguigu@ubuntu:~/桌面$ ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	10:26	?	00:00:01	/sbin/init
root	2	0	0	10:26	?	00:00:00	[kthreadd]
root	3	2	0	10:26	?	00:00:00	[migration/0]
root	4	2	0	10:26	?	00:00:00	[ksoftirqd/0]
root	5	2	0	10:26	?	00:00:00	[stopper/0]
root	6	2	0	10:26	?	00:00:00	[watchdog/0]
root	7	2	0	10:26	?	00:00:00	[migration/1]
root	8	2	0	10:26	?	00:00:00	[stopper/1]
root	9	2	0	10:26	?	00:00:00	[ksoftirqd/1]

7.11.2 kill 终止进程

1) 基本语法

kill [选项] 进程号 (功能描述: 通过进程号杀死进程)

killall 进程名称 (功能描述: 通过进程名称杀死进程, 也支持通配符, 这在系统因负载过大而变得很慢时很有用)

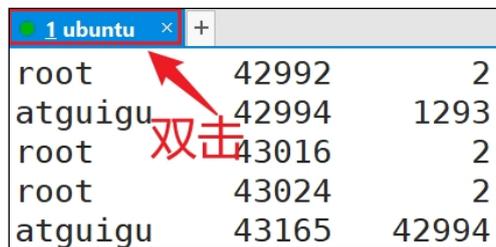
2) 选项说明

选项	功能
-9	表示强迫进程立即停止

3) 案例实操

(1) 开启多个终端

在 XShell 中双击开启的标签, 即可打开新的终端。



准备两个终端。

(2) 监控 hougex.txt

在其中一个终端中执行以下命令。

```
atguigu@ubuntu:~/桌面$ tail -F hougex.txt
```

(3) 查看 tail 进程号

在另一个终端中查看进程号。

```
atguigu@ubuntu:~/桌面$ ps -ef | grep tail
atguigu 43823 38358 0 17:14 pts/0 00:00:00 tail -F
hougex.txt
```

(4) 杀死 tail 进程

```
atguigu@ubuntu:~/桌面$ kill -9 43823
```

```
atguigu@ubuntu:~/桌面$ tail -F hougex.txt
总计 8
-rw-rw-r-- 1 atguigu atguigu 0 3月 5 19:05 hougex.txt
-rw-r--r-- 1 atguigu atguigu 582 3月 5 15:31 profile
drwxrwxr-x 4 atguigu atguigu 4096 3月 5 18:57 xiyou
总计 12
-rw-rw-r-- 1 atguigu atguigu 180 3月 5 19:05 hougex.txt
-rw-r--r-- 1 atguigu atguigu 582 3月 5 15:31 profile
drwxrwxr-x 4 atguigu atguigu 4096 3月 5 18:57 xiyou
hello
已杀死
```

此时, 另一个终端可以看到提示, 进程已被杀死, 如上图所示。

(5) 通过名称杀死进程

killall 命令可以根据名称杀死进程，此处的进程名称是精确匹配。通常进程名称为启动命令中可执行文件的名称。对于 tail -F hougex.txt 启动的进程，其进程名称为 tail。

再开启一个终端，在其中两个终端中执行以下命令。

```
atguigu@ubuntu:~/桌面$ tail -F hougex.txt
```

在最后一个终端中执行以下命令。

```
atguigu@ubuntu:~/桌面$ killall tail
```

可以看到另外两个终端的 tail 进程均被杀死。

7.11.3 查看服务器总体内存

1) 基本语法

free -m (功能描述：查看服务器总体内存)

2) 案例实操

```
atguigu@ubuntu:~/桌面$ man free
```

	total	used	free	shared	buff/cache
available					
Mem:	3934	543	2879	12	511
	3093				
Swap:	4095	0	4095		

7.11.4 top 查看系统健康状态

1) 基本命令

top [选项]

2) 选项说明

选项	功能
-d 秒数	指定 top 命令每隔几秒更新。
-i	使 top 不显示任何闲置或者僵死进程。
-p	通过指定监控进程 ID 来仅仅监控某个进程的状态。

3) 操作说明

操作	功能
P	以 CPU 使用率排序，默认就是此项
M	以内存的使用率排序
N	以 PID 排序
q	退出 top

4) 查询结果字段解释

(1) 第一行信息为任务队列信息

内容	说明
12:26:46	系统当前时间
up 1 day, 13:32	系统的运行时间，本机已经运行 1 天 13 小时 32 分钟
2 users	当前登录了两个用户
load average: 0.00, 0.00, 0.00	系统在之前 1 分钟，5 分钟，15 分钟的平均负载。一般认为小于 1 时，负载较小。如果大于 1，系统已经超出负荷。

(2) 第二行为进程信息

Tasks: 95 total	系统中的进程总数
1 running	正在运行的进程数
94 sleeping	睡眠的进程
0 stopped	正在停止的进程
0 zombie	僵尸进程。如果不是 0，需要手工检查僵尸进程

(3) 第三行为 CPU 信息

Cpu(s): 0.1%us	用户模式占用的 CPU 百分比
0.1%sy	系统模式占用的 CPU 百分比
0.0%ni	改变过优先级的用户进程占用的 CPU 百分比
99.7%id	空闲 CPU 的 CPU 百分比
0.1%wa	等待输入/输出的进程的占用 CPU 百分比
0.0%hi	硬中断请求服务占用的 CPU 百分比
0.1%si	软中断请求服务占用的 CPU 百分比
0.0%st	st (Steal time) 虚拟时间百分比。就是当有虚拟机时，虚拟 CPU 等待实际 CPU 的时间百分比。

(4) 第四行为物理内存信息

Mem: 625344k total	物理内存的总量，单位 KB
571504k used	已经使用的物理内存数量
53840k free	空闲的物理内存数量，我们使用的是虚拟机，总共只分配了 628MB 内存，所以只有 53MB 的空闲内存了
65800k buffers	作为缓冲的内存数量

(5) 第五行为交换分区 (swap) 信息

Swap: 524280k total	交换分区 (虚拟内存) 的总大小
0k used	已经使用的交互分区的大小
524280k free	空闲交换分区的大小
409280k cached	作为缓存的交互分区的大小

5) 案例实操

```
atguigu@ubuntu:~/桌面$ top -d 1
atguigu@ubuntu:~/桌面$ top -i
atguigu@ubuntu:~/桌面$ top -p 2575
```

执行上述命令后，可以按 P、M、N 对查询出的进程结果进行排序。

7.11.5 netstat 显示网络统计信息和端口占用情况

1) 基本语法

`netstat -anp |grep 进程号` (功能描述: 查看该进程网络信息)

`netstat -nlp |grep 端口号` (功能描述: 查看网络端口号占用情况)

2) 选项说明

选项	功能
<code>-n</code>	拒绝显示别名, 能显示数字的全部转化成数字
<code>-l</code>	仅列出有在 listen (监听) 的服务状态
<code>-p</code>	表示显示哪个进程在调用

3) 案例实操

(1) 通过 `nc` 命令监听某个端口

```
atguigu@ubuntu:~/桌面$ nc -lk 12345
```

上述命令表示监听本机 12345 端口。

(2) 在另一个终端查看进程号

```
atguigu@ubuntu:~/桌面$ ps -ef | grep "nc -lk 12345"
atguigu  44118  38358  0 18:08 pts/0    00:00:00 nc -lk 12345
atguigu  44152  43767  0 18:11 pts/1    00:00:00 grep --
color=auto nc -lk 12345
```

(3) 通过进程号查看该进程的网络信息

```
atguigu@ubuntu:~/桌面$ netstat -anp | grep 44118
```

(并非所有进程都能被检测到, 所有非本用户的进程信息将不会显示, 如果想看到所有信息, 则必须切换到 `root` 用户)

```
tcp      0      0 0.0.0.0:12345          0.0.0.0:*
LISTEN  44118/nc
```

(4) 查看某端口号是否被占用

```
atguigu@ubuntu:~/桌面$ netstat -nlp | grep 12345
```

(并非所有进程都能被检测到, 所有非本用户的进程信息将不会显示, 如果想看到所有信息, 则必须切换到 `root` 用户)

```
tcp      0      0 0.0.0.0:12345          0.0.0.0:*
LISTEN  44118/nc
```

7.12 crontab 系统定时任务

7.12.1 crontab 服务管理

1) 重新启动 `crond` 服务

```
atguigu@ubuntu:~/桌面$ sudo systemctl restart cron
```

7.12.2 crontab 定时任务设置

1) 基本语法

```
crontab [选项]
```

2) 选项说明

选项	功能
-e	编辑 crontab 定时任务
-l	查询 crontab 任务
-r	删除当前用户所有的 crontab 任务

3) 选择编辑器

```
atguigu@ubuntu:~/桌面$ crontab -e
```

执行上述命令，系统会提示我们选择编辑器，此处没有 vim。

```
no crontab for atguigu - using an empty one
```

```
Select an editor. To change later, run 'select-editor'.
```

1. /bin/nano <---- easiest
2. /usr/bin/vim.basic
3. /usr/bin/vim.tiny
4. /bin/ed

```
Choose 1-4 [1]:
```

我们可以通过 EDITOR 环境变量在执行 crontab 时选择编辑器，命令如下。

```
atguigu@ubuntu:~/桌面$ EDITOR=vim crontab -e
```

4) 参数说明

(1) 执行上述命令会进入 crontab 编辑界面，并打开 vim 编辑定时任务。

***** 执行的任务

项目	含义	范围
第一个 “*”	一小时当中的第几分钟	0-59
第二个 “*”	一天当中的第几小时	0-23
第三个 “*”	一个月当中的第几天	1-31
第四个 “*”	一年当中的第几个月	1-12
第五个 “*”	一周当中的星期几	0-7 (0 和 7 都代表星期日)

(2) 特殊符号

特殊符号	含义
*	代表任何时间。比如第一个 “*” 就代表一小时中每分钟都执行一次的意思。
,	代表不连续的时间。比如 “0 8,12,16 * * * 命令”，就代表在每天的 8 点 0 分，12 点 0 分，16 点 0 分都执行一次命令
-	代表连续的时间范围。比如 “0 5 * * 1-6 命令”，代表在周一到周六的凌晨 5 点 0 分执行命令

<code>*/n</code>	代表每隔多久执行一次。比如“*/10 * * * * 命令”，代表每隔 10 分钟就执行一遍命令
------------------	--

(3) 特定时间执行命令

时间	含义
<code>45 22 * * * 命令</code>	在 22 点 45 分执行命令
<code>0 17 * * 1 命令</code>	每周 1 的 17 点 0 分执行命令
<code>0 5 1,15 * * 命令</code>	每月 1 号和 15 号的凌晨 5 点 0 分执行命令
<code>40 4 * * 1-5 命令</code>	每周一到周五的凌晨 4 点 40 分执行命令
<code>*/10 4 * * * 命令</code>	每天的凌晨 4 点，每隔 10 分钟执行一次命令
<code>0 0 1,15 * 1 命令</code>	每月 1 号和 15 号，每周 1 的 0 点 0 分都会执行命令。注意：星期几和几号最好不要同时出现，因为他们定义的都是天。非常容易让管理员混乱。

5) 案例实操

(1) 监听 bailongma.txt

```
atguigu@ubuntu:~/桌面$ tail -F bailongma.txt
```

(2) 每隔 1 分钟，向/home/atguigu/桌面/bailongma.txt 文件中添加一个 11 的数字

```
* /1 * * * * /bin/echo "11" >> /home/atguigu/桌面/bailongma.txt
```

(3) 查看效果

```
atguigu@ubuntu:~/桌面$ tail -F bailongma.txt
11
```

第 8 章 常见错误及解决方案

1) 虚拟化支持异常情况如下几种情况

(1) 异常情况一

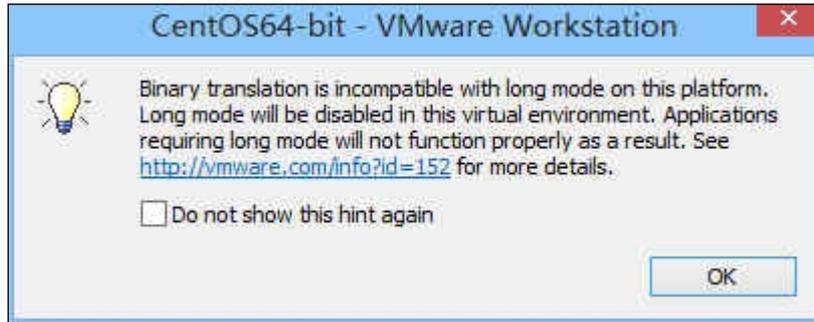
```

Loading vmlinuz.....
Loading initrd.img.....ready.
This kernel requires an x86-64 CPU, but only detected an i686 CPU.
Unable to boot - please use a kernel appropriate for your CPU.
    
```

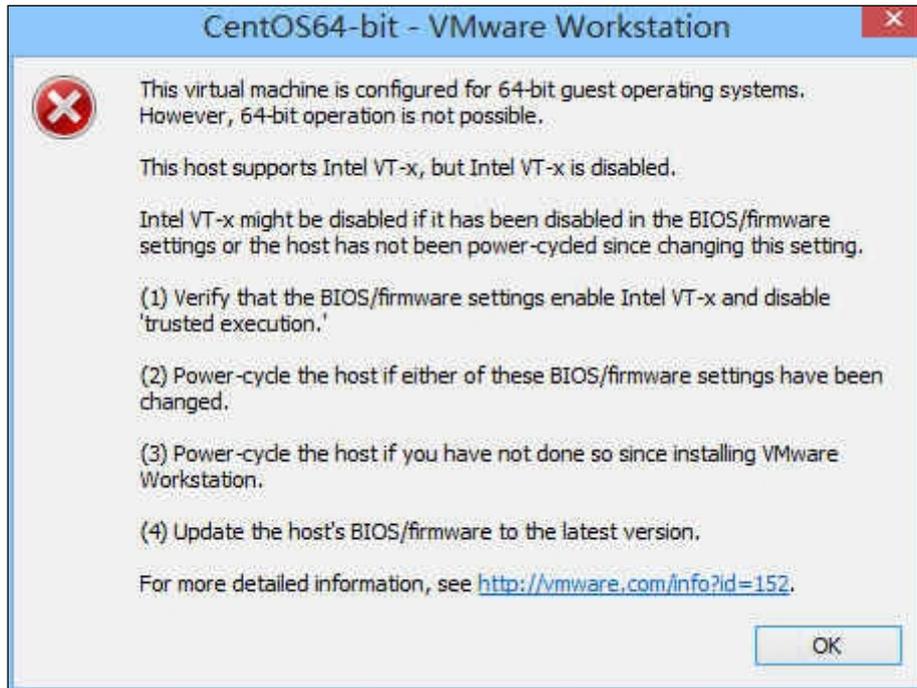
(2) 异常情况二



(3) 异常情况三



(4) 异常情况四



2) 问题原因：宿主机 BIOS 设置中的硬件虚拟化被禁用了

3) 解决办法：需要打开笔记本 BIOS 中的 IVT 对虚拟化的支持

