

Linux

讲师: 夏磊









▶ 为什么要学Linux

开源、稳定、漏洞少,靠谱公司都用它。

➤ Linux是什么

Linux是一套免费使用和自由传播的类Unix操作系统。

Linux与Unix的那些事









Richard Stallman

- 在自由的时 代用户应该 免费享有对 软件源代码 阅读、修改 的权利。
 - 软件公司 可以靠提供 服务和训练 获得盈利。



- Unix不是你说用就能用
 - 5万美元起
 - 性能要求苛刻的高端服务器或小型机

让天下没有难学的技术







思考:安卓系统是封装了Linux的哪一部分?

让天下没有难学的技术





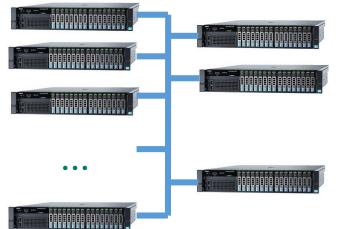
一夫当关的时代



小型机单节点架构

- ▶ 高性能
- > 极强的稳定性
- ▶ 量身定制的UNIX
- > 应用程序结构简单
- 厂商提供售后服务
- 价格...

烂机子组团的时代

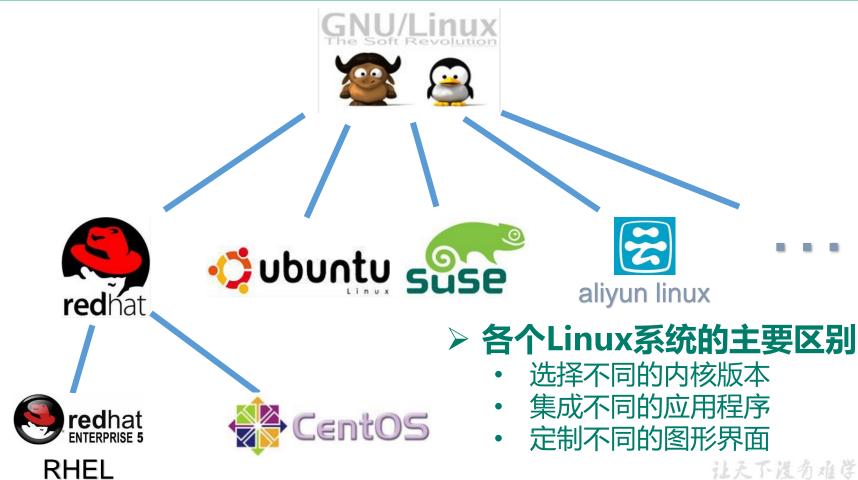


分布式架构

- 可扩展的性能
- 风险分摊
- 去IOE,免费的Linux
- 应用程序架构复杂
- 运维不依赖厂商
- 价格好商量 下没有难学的技术







让天下没有难学的技术







从哪下载

CentOS下载地址:

➤ 网易镜像: http://mirrors.163.com/centos/7/isos/

➤ 搜狐镜像: http://mirrors.sohu.com/centos/7/isos/

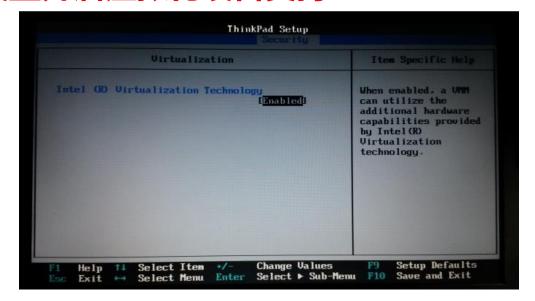




安装步骤

1. 去BIOS里修改设置开启虚拟化设备支持

- 2. 安装虚拟机
- 3. 安装镜像文件
- 4. 开始安装系统



安装步骤详见《01在VM上安装CentOS7_201802V1.4》





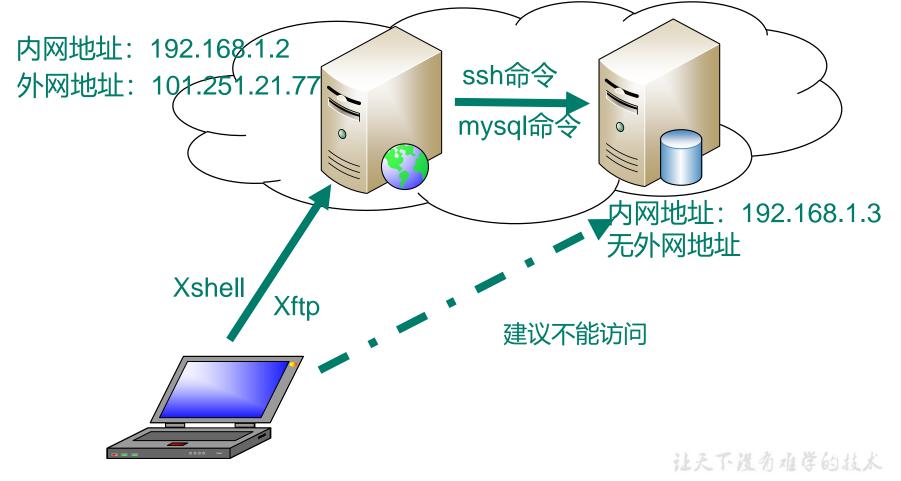
辅助软件安装

- ➤安装xshell
 - 作用:远程命令行访问
 - 详见《03通过Xshell和Xftp远程访问服务器201711_NAT模式》
- ➤安装xftp
 - 作用:远程文件访问

一个小型云服务网站架构







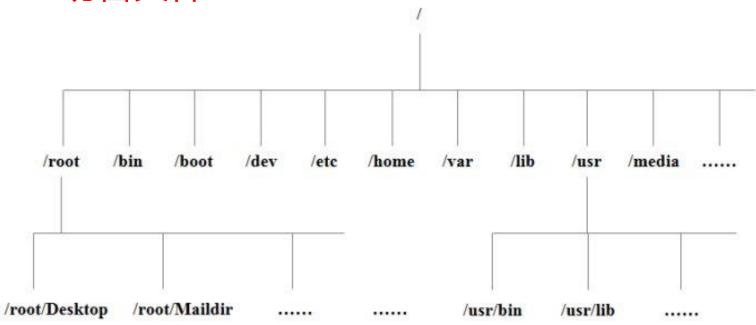






Linux 文件结构

一切皆文件





Linux 文件与目录结构



- - 是Binary的缩写,这个目录存放着最经常使用的命令
- /sbin (/usr/sbin \ /usr/local/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。
- - 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。

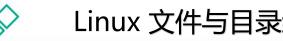
Linux 文件与目录结构





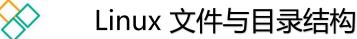
- > /lib
 - 系统开机所需要最基本的动态连接共享库,其作用类似于Windows里的 DLL文件。几乎所有的应用程序都需要用到这些共享库。
- /lost+found
 - 这个目录一般情况下是空的,当系统非法关机后,这里就存放了一些文件。
- - 所有的系统管理所需要的配置文件和子目录
- - 这是一个非常重要的目录,用户的很多应用程序和文件都放在这个目录下, 类似与windows下的program files目录。

Linux 文件与目录结构





- /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 (centos6)
 - 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。
- > /run 🛨
 - 进程产生的临时文件,虚拟机加载光盘映像在: /run/media/root/ 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ORACLE数据库则就可以放到这个目录下。默认是空的。
- /usr/local *
 - 这是另一个给主机额外安装软件所摆放的目录。一般是通过编译源码方式安装的程序。
- - 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。







▶是什么

• Linux系统的命令行下的文本编辑器。

• 使用命令: vi xxxx文件 或者vim xxxx文件







• 武装起来的vim

```
vim
                                                                                                                               MockRequest : class
▼ FatalWarner : class
    [methods]
 - rack/
                                     49
                                               rack.run_once" => false,
                                     50
  ▶ auth/
  backports/
  handler/
                                            def initialize(app)
                                                                                                                                   flush
  ▶ multipart/
                                              @app = app
                                                                                                                                   puts
                                            end
  session/
                                                                                                                                   string
  ▶ utils/
                                                                                                                                   write
                                                                                                                                 FatalWarning : class
    body_proxy.rb
                                            def get(uri, opts={})
                                                                      request("GET", uri, opts)
                                                                                                     end
    builder rb
                                            def post(uri, opts={})
                                                                      request("POST", uri, opts)
                                                                                                     end
                                                                                                                                 get* : class
    cascade, rb
                                     58
                                            def put(uri, opts={})
                                                                      request("PUT", uri, opts)
                                                                                                     end
                                                                                                                                 ▶ post* : class
[methods]
     chunked rb
                                            def patch(uri, opts={}) request("PATCH", uri, opts)
                                                                                                    end
                                     60
                                            def delete(uri, opts={}) request("DELETE", uri, opts) end
     commonlogger.rb
                                                                                                                                   post
                                                                                                                                 [methods]
     conditionalget.rb
                                            def head(uri, opts={})
                                                                      request("HEAD", uri, opts)
     config.rb
                                            def options(uri, opts={}) request("OPTIONS", uri, opts) end
                                                                                                                                 get
                                                                                                                                 initialize
     content_length.rb
                                     64
     content_type.rb
                                            def request(method="GET", uri="", opts={})
                                              env = self.class.env_for(uri, opts.merge(:method => method))
     deflater.rb
     directory.rb
     etaa.rb
                                              if opts[:lint]
                                    68
     file.rb
                                                app = Rack::Lint.new(@app)
    handler.rb
                                    69
    head.rb
                                                app = @app
     lint.rb
     lobster.rb
     lock.rb
                                              errors = env["rack.errors"]
                                              status, headers, body = app.call(env)
     logger.rb
    methodoverride.rb
                                              MockResponse.new(status, headers, body, errors)
                                            ensure
    mime.rb
    mock.rb
                                              body.close if body.respond_to?(:close)
    multipart.rb
                                            end
    nulllogger.rb
    recursive.rb
~/.../Ruby/rack/NERD_tree_1
                                  NORMAL BR: master | lib/rack/mock.rb
                                                                                    unix | utf-8 | ruby 34% LN 68:6 [Name] mock.rb
rack.gemspec
KNOWN-ISSUES
README, rdoc
Rakefile
lib/rack.rb
Gemfile
COPYING
bin/rackup
test/cgi/test
SPEC
mru files buf
                                                                                                                          prt | path ~/Development/Ruby/rack
```





三种模式

- 一般模式 (默认模式)
- 编辑模式
- 命令模式





▶ 打开文档的默认模式,主要负责查看,和一些基础的修剪。

按键	效果
dd	删除光标当前行
dnd	删除n行
u	撤销上一步
X	删除一个字母Delete
X	删除一个字母Backspace
уу	复制光标当前行
р	米占贝占
dw	删除一个词
yw	复制一个词





▶ 打开文档的默认模式,主要负责查看,和一些基础的修剪。

按键	效果
shift+g	移动到页尾
数字1+shift+g	移动到页头
数字N+shift+g	移动到目标行
shift+6	移动到行头
shift+4	移动到行尾





三种模式

- 一般模式 (默认模式)
- 编辑模式
- 命令模式





- > 可以编写文字
- ➤ 要按下 i, a, o,等字母后才能进入编辑模式
- ➤ 进入后左下角会有[insert]或[replace]的字样
- ➤ 按下[ESC]这个按键即可退出编辑模式

按键	效果
i	当前光标前
a	当前光标后
0	当前光标行的下一行
s,S	s:删除当前字符并进入编辑 S:删除整行并进入编辑
R	进入替换模式





三种模式

- 一般模式 (默认模式)
- 编辑模式
- 命令模式





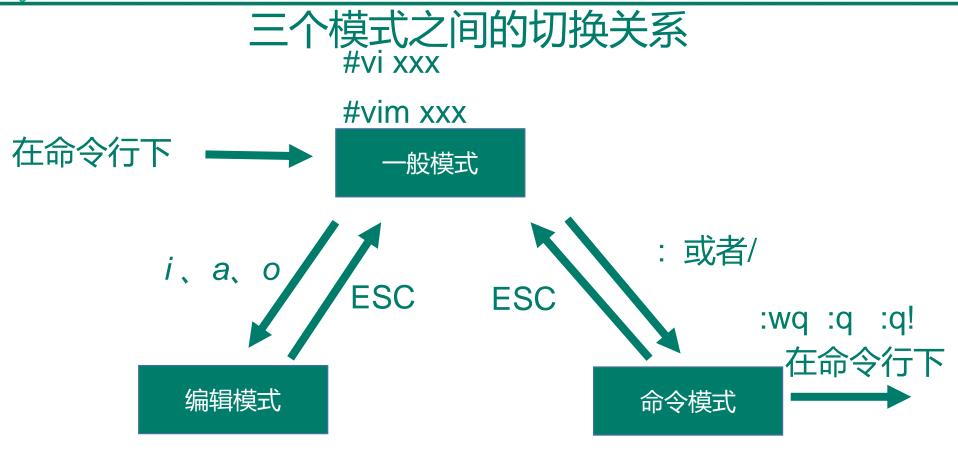


- 可以进行存盘、退出、显示行号、搜索、批量替换等操作。
- ▶ 要在一般模式下输入"/"或":" 可进入命令模式
- ▶ 进入后光标移至最下端

命令	含义
: w	保存
: q	退出
:1	强制执行
: %s/old字符/new字符/g	批量替换
/ 要查找的词	n 查找下一个,N 往上查找
	N 查找下一个, n 往上查找
:set nu / :set nonu	显示行号/关闭行号









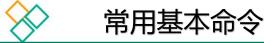
大写(B): quux(foo, bar, baz);













- ▶ 帮助手册
 - > man
 - > --help
- ▶ 日期类
 - date ,cal
- ➤ 显示当前目录 pwd
- ▶ cd切换目录
 - ➤ cd .. 返回上级目录
 - > cd / 返回至/
 - ➤ cd , cd ~ 返回家目录
 - ➤ cd /etc/sysconfig 通过绝对路径访问(利用tab键)
 - ➤ cd ./sysconfig 通过相对路径访问



常用基本命令 - 文件目录管理



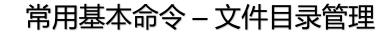
- > Is
 - -a 全部文件, 连同隐藏的文件
 - -I 列出详细列表 别名Ⅱ
 - 列表信息:



grep

- 配合显示内容的命令,根据跟随的内容显示,该行内容
- 例如: Is -I |grep xxx

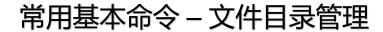
让天下没有难学的技术







- > mkdir
 - -p 可以一下建好多级目录
 - 例: mkdir 目录名
- > touch
 - 新建一个文件
 - 例: touch 文件名
- > rmdir
 - 删除一个空目录
 - 例: rmdir dirname







- > rm
 - 移除文件或目录
 - -rvf递归删除所有目录内容,提示
 - -rf 递归删除所有目录内容,不提示
- > cp
 - cp 【要复制的文件】 【到哪里】
 - -r 递归复制整个文件夹
 - -v 显示复制过程中文件的列表
 - 强制覆盖不提示的方法
 - 临时方法: 用\cp



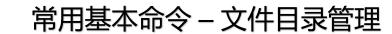
常用基本命令 - 文件目录管理



- > mv
 - mv oldFileName newFileName 重命名
 - mv /temp/movefile /targetFolder 移动文件

> cat

- cat 文件名 查看轻量级的文本文件
- cat 文件1 文件2 连接显示多个文件
- cat 文件1 文件2 > 文件3 合并为新文件





more

- ・ 查看较长的文件
- 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 more,不再显示该文件内容。
- Ctrl+F 向下滚动一屏
- Ctrl+B 返回上一屏

less

- · 同more类似,比more功能更多。
- [pagedown]: 向下翻动一页;
- [pageup]: 向上翻动一页;
- /字串 : 向下搜寻『字串』的功能;
- ?字串 : 向上搜寻『字串』的功能;
- n : 重复前一个搜寻 (与 / 或 ? 有关!)
- N : 反向的重复前一个搜寻(与/或?有关!) 让天下没有难等的技术



常用基本命令 - 文件目录管理



- > tail
 - 从尾部开始查看,比较适合查看日志
 - -f 跟随查看
 - -n200 显示的行数, n可以省略。

- history
 - 查看历史命令

- > echo
 - 输出环境变量
 - 在shell脚本中当system.out.print用



常用基本命令 - 文件目录管理



- > find
 - 查找文件,提供了丰富的模糊搜索及条件搜索
 - find+搜索路径+参数+搜索关键字
 - 按文件名: find /home/esop -name "*.txt" 根据名称查找/ 目录下的filename.txt文件。

locate

- 查找文件,查询速度更快,使用更方便。
- 运行前需要 执行updatedb来更新文件索引。





> 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。
- ・ 命令 In -s 原文件或目录名 软链接名
- · 查询:通过 Is -I 就可以查看,列表属性第1位是I,尾部会有位置指向。



常用基本命令 - 压缩和解压

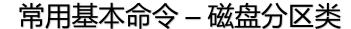


- ➤ 压缩文件(tar)
 - tar -zcvf XXX.tar.gz n1.txt
 - -c 创建一个新归档
 - -v 显示详细信息
 - -f 指定压缩后的文件名
 - -z 通过 gzip 过滤归档

- ➤ 压缩文件(zip)
 - zip mypackage.zip 1.txt 2.txt
 - zip -r pack.zip pack/*

- ➤ 解压缩文件(tar)
 - tar -zxvf XXX.tar.gz
 - -x从归档中解出文件
 - -v 显示详细信息
 - -f 指定解压前的文件名
 - -z 通过 gzip 过滤归档

- ➤ 解压缩文件(zip)
 - unzip mypackage.zip







> 分区

• windows下的磁盘分区









▶查看所有设备挂载情况

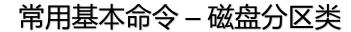
• 命令: Isblk 或者 Isblk -f

```
[root@localhost 桌面]#lsblk-f
NAME FSTYPE LABEL UUID MOUNTPOINT
sda
—sda1 swap 97c3b380-6b62-4310-834d-3a7375a49db6 [SWAP]
—sda2 xfs 3c5827fe-955b-4365-89a0-8e93c9de25f0 /
sdb
sr0 iso9660 CentOS 7_x86_64 2015-12-09-23-14-10-00 /run/media
```





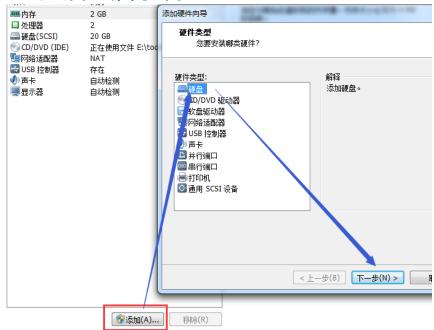
- > 如何增加一块硬盘
 - 1. 虚拟机插硬盘
 - 2. 分区
 - 3. 格式化
 - 4. 挂载

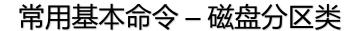






- ▶ 虚拟机增加硬盘
 - 在【虚拟机】菜单中,选择【设置】,然后设备列表里添加硬盘, 然后一路【下一步】,中间只有选择磁盘大小的地方需要修改,至 到完成。然后重启系统!





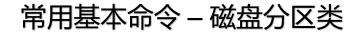




- ➢ 分区命令 fdisk /dev/sdb
 - 开始对/sda分区
 - m 显示命令列表 p 显示磁盘 n 新增分区
 - 删除分区 w 写入并退出分区 同 fdisk -I

开始分区后输入n,新增分区,然后选择p,分区类型为主分区。两次回车默 认剩余全部空间。最后输入w写入分区并退出,若不保存退出输入q。

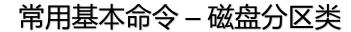
```
Command (m for help): n
Command action
   e extended
   p primary partition (1-4)
Selected partition 4
First cylinder (2233-2610, default 2233): 回车, 默认全部剩余空间
Using default value 2233
Last cylinder, +cylinders or +size{K,M,G} (2233-2610, default 2610):
Using default value 2610
Command (m for help): w
The partition table has been altered!
```







- ▶ 格式化磁盘
 - 分区命令:mkfs -t ext4 /dev/sdb1
 - 其中ext4是分区类型





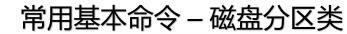


> 挂载

- 将一个分区与一个目录联系起来,
- mount 设备名称 挂载目录
 - 例如: mount /dev/sdb1 /newdisk
- · umount 设备名称 或者 挂载目录
 - 例如: umount /dev/sdb1 或者 umount /newdisk
- · 用命令行挂载重启后会失效

> 永久挂载

- 通过修改/etc/fstab实现挂载
- · 添加完成后 执行mount -a 即刻生效



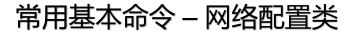




磁盘情况查询

- > df -h
 - 查询系统整体磁盘使用情况
- ➤ du -h /目录
 - 查询指定目录的磁盘占用情况,默认为当前目录
 - -s 指定目录占用大小汇总
 - -h 带计量单位
 - -a 含文件
 - --max-depth=1 子目录深度
 - -c 列出明细的同时,增加汇总值

例: du -ach --max-depth=1 /opt







- ifconfig
 - 查看网络配置
 - 如何修改IP
 - 图形化操作
 - 命令行方式 vi /etc/sysconfig/network-scripts/ifcfg-ens33

```
DEVICE=eth0
TYPE=Ethernet
UUID=4f146062-dea7-4374-9e60-ff99dafad44b
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
HWADDR=00: 0C: 29: D4: 6C: 59
IPADDR=192. 168. 154. 167
PREFIX=24
GATEWAY=192. 168. 154. 1
LAST_CONNECT=1509971498
```

- 刷新网络设置
 - service network restart



常用基本命令 – 网络配置类



DEVICE=eth0 #接口名(设备,网卡)

BOOTPROTO=none

IP的配置方法[none|static|bootp|dhcp](引导时不使用协议|静态分配

IP|BOOTP协议|DHCP协议)

BROADCAST=192.168.1.255 #广播地址

HWADDR=00:0C:2x:6x:0x:xx #MAC地址

IPADDR=192.168.1.23 #IP地址

NETMASK=255.255.255.0 # 网络掩码

NETWORK=192.168.1.0 #网络地址

ONBOOT=yes #系统启动的时候网络接口是否有效 (yes/no)

TYPE=Ethernet #网络类型 (通常是Ethemet)







ps (process)



ps 显示的信息选项:

字段	说明
PID	进程识别号
TTY	终端机号
TIME	此进程所消 CPU 时间
CMD	正在执行的命令或进程名
500 0107	

ps -aux

ps -a: 显示当前终端的所有进程信息

ps-u: 以用户的格式显示进程信息

ps-x: 显示后台进程运行的参数

让天下没有难学的技术







> ps



- ps –aux|grep xxx
- System V展示风格
- USER: 用户名称
- PID: 进程号
- ・ %CPU: 进程占用CPU的百分比
- · %MEM:进程占用物理内存的百分比
- ・ VSZ: 进程占用的虚拟内存大小 (单位: KB)
- ・ RSS: 进程占用的物理内存大小 (单位: KB)
- TT:终端名称,缩写
- STAT: 进程状态,其中S-睡眠,s-表示该进程是会话的先导进程,N-表示进程拥有比普通优先级更低的优先级,R-正在运行,D-短期等待,Z-僵死进程,T-被跟踪或者被停止等等
- STARTED: 进程的启动时间
- TIME: CPU时间,即进程使用CPU的总时间
- COMMAND: 启动进程所用的命令和参数,如果过长会被截断显示。

常用基本命令 – 进程类





> ps



- ps -ef是以全格式显示当前所有的进程
- -e 显示所有进程。-f 全格式。
- ps -ef|grep xxx
 - ・ 是BSD风格
 - 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







> ps

综上:

- > 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
- > pid
 - 每个进程都有一个pid作为唯一标识。
- > kill
 - kill pid 通过唯一标识pid杀死进程
 - -9 强行杀死进程
- killall
 - killall name 通过进程名称杀死进程







> service (centos6)

- 注册在系统中的标准化程序
- 有方便统一的管理方式(常用的方法)
 - service 服务名 start
 - service 服务名 stop
 - service 服务名 restart
 - service 服务名 reload
 - service 服务名 status
- · 查看服务的方法 /etc/init.d/服务名
- · 通过chkconfig 命令设置自启动
 - 查看服务 chkconfig --list|grep xxx
 - chkconfig --level 5 服务名 on





➤ 运行级别runlevel(centos6)

开机



BIOS



/boot



init进程



运行级别



查看默认级别: vi/etc/inittab

运行级对应的服务

Linux系统有7种运行级别(runlevel): 常用的是级别3和5

- **运行级别0**: 系统停机状态,系统默认运行级别不能设为0,否则不能正常启动
- **运行级别1**: 单用户工作状态,root权限,用于系统维护,禁止远程登陆
- 运行级别2: 多用户状态(没有NFS),不支持网络
- 运行级别3: 完全的多用户状态(有NFS), 登陆后进入控制台命令行模式
- 运行级别4: 系统未使用, 保留
- 运行级别5: X11控制台, 登陆后进入图形GUI模式
- 运行级别6: 系统正常关闭并重启,默认运行级别不能设为6, 否则不能正常启动



常用基本命令 - 进程类



> systemctl (centos7)

- 注册在系统中的标准化程序
- 有方便统一的管理方式(常用的方法)
 - systemctl start 服务名(xxxx.service)
 - systemctl restart 服务名(xxxx.service)
 - systemctl stop 服务名(xxxx.service)
 - systemctl reload 服务名(xxxx.service)
 - systemctl status 服务名(xxxx.service)
- 查看服务的方法 /usr/lib/systemd/system
- · 查看服务的命令
 - systemctl list-unit-files
 - systemctl --type service
- · 通过systemctl 命令设置自启动
 - 自启动systemctl enable service_name
 - 不自启动systemctl disable service name





➤ 运行级别runlevel(centos7)



查看默认级别: vim /etc/inittab Centos7运行级别简化为:

运行级对应的服务

multi-user.target 等价于原运行级别3(多用户有网,无图形界面)

multi-user.target 等价于原运行级别5(多用户有网,有图形界面)







netstat

- 查看系统的网络情况
- -an 按一定顺序排列输出
- -p 显示哪个进程在调用
- netstat -anp|grep 8080 查看占用8080端口的进程









> 用户

- Linux系统是一个多用户多任务的操作系统,任何一个要使用系统资的用户,都必须首先向系统管理员申请一个账号,然后以这个账号的身份进入系统。
- 新增用户:useradd 新用户名
- 设置密码: passwd 用户名
- 用户是否存在: id 用户名
- 切换用户: su 切换用户名
- 查看当前用户/登录用户: whoami/ who am I
- 删除用户: userdel







> 用户组

- 类似于角色,系统可以对有共性的多个用户进行统一的管理。
- 新增组:groupadd 组名
- 删除组:groupdel 组名
- 修改用户的组: usermod -g 用户组 用户名
- 增加用户时直接加上组: useradd -g 用户组 用户名







- > 系统中用户和组的相关文件
 - 用户 (user) 的配置文件: /etc/passwd
 - 每行的含义: **用户名:口令:用户标识号:组标识号:注释性描述:主 目录:登录**Shell
 - 口令的配置文件: /etc/shadow
 - 每行的含义: 登录名:加密口令:最后一次修改时间:最小时间间隔: 最大时间间隔:警告时间:不活动时间:失效时间:标志
 - 组(group)的配置文件: /etc/group
 - 每行含义: 组名:口令:组标识号:组内用户列表





文件的权限管理

➤ 再说Is -I

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	789
d	rwx	r-x	r-x
目录 文件	读写执行	读写执	读写执

> 0-9位说明

- 第0位确定文件类型(d, -, I, c, b)
- 第1-3位确定所有者(该文件的所有者)拥有该文件的权限。---User
- 第4-6位确定所属组(同用户组的)拥有该文件的权限,---Group
- 第7-9位确定其他用户拥有该文件的权限 ---Other







> 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

> 作用到目录

- [r]代表可读(read): 可以读取, Is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







> chmod

- 第一种方式: +、-、= 变更权限
 - u:所有者 g:所有组 o:其他人 a:所有人(u、g、o的总和)
 - chmod u=rwx,g=rx,o=x 文件目录名
 - chmod o+w 文件目录名
 - chmod a-x 文件目录名
- 第二种方式:通过数字变更权限
 - r=4 w=2 x=1 rwx=4+2+1=7
 - chmod u=rwx,g=rx,o=x 文件目录名
 - 相当于 chmod 751 文件目录名

文件 类型	1 2 3	4 3 6	基プ ^庁 789
d	rwx	r-x	r-x
目录 文件	读写执	读写执	读写执





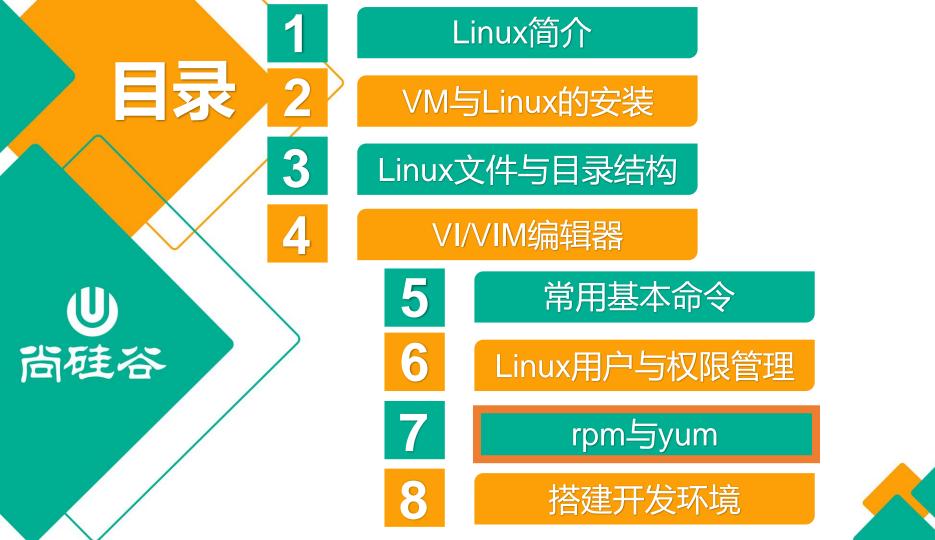


> chown

- chown newowner file 改变文件的所有者
- chown newowner:newgroup file 改变用户的所有者和所有组
- -R 如果是目录则使其下所有子文件或目录递归生效

chgrp

• chgrp newgroup file 改变文件的所有组

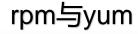






> rpm

- RPM (RedHat Package Manager), Rethat软件包管理工具, 类似windows里面的setup.exe
- 是Linux这系列操作系统里面的打包安装工具,它虽然是 RedHat的标志,但理念是通用的。
- 查询已安装的rpm列表 rpm -qa|grep xx
- rpm包的名称:
 - firefox-52.5.0-1.el7.centos.x86_64
 - 名称:firefox
 - 版本号: 52.6.0-1
 - 适用操作系统: el7.centos.x86_64
 - 表示centos7.x的64位系统。







➢ 安装rpm

- rpm -ivh rpm包名
- -i 安装 install
- -v 查看信息
- -h 查看进度条

▶ 卸载rpm

• rpm -e RPM软件包







> yum

- 类似于我们java开发中的maven工具,可以从镜像网站上下载应用程序,并直接安装
- yum list|grep xx软件列表
- yum install xxx 下载安装









> 安装jdk

- 解压缩到
- 配置环境变量的配置文件vim /etc/profile
- JAVA_HOME=/opt/jdk1.8.0_152
- PATH=/opt/jdk1.8.0_152/bin:\$PATH
- export JAVA_HOME PATH

- · 安装完成注销重新登录一下
- source /etc/profile
- 重启系统







➤ 安装Tomcat

- 解压缩到/opt
- 启动tomcat ./startup.sh

> 安装Eclipse

- 解压缩到/opt
- 启动eclipse, 配置jre和server
- 编写Hello world
- 编写jsp 页面

