## 计算机的发展史

第一代计算机，电子管

第二代，晶体管

第三代，集成电路（可以浓缩了）

第四代，超大规格集成电路，电子元件更多，元件体积更小

第五代，未来的展望，人工智能计算机，生物计算机，量子计算机，非冯诺依曼结构模式的计算机

cpu的发展史

cpu可分为4 6 8 16 32 64位数理器,工艺上是

1bit = 1b

1024B = 1KB

1024KB = 1M

1024M = 1G

cpu工艺，纳米。

## 计算机硬件组成

### 一、冯诺依曼计算机

现代计算机之父，提出了数据和程序的分离。

计算机的五大系统:硬盘/手机是内存、内存、cpu（运算器，控制器，寄存器）、输入设备、输入出设备

运算器（是用于做计算），控制器（控制哪些数据来做计算）

### 二、软硬件工作流程

打开一个word文档

操作系统调用cpu发送指令给硬盘读取文件的指令，

操作系统给cpu发送打开文件的指令

cpu计算出控制硬盘的指令发送给硬盘

word程序和文件载入内存

cpu调用内存中word程序的代码来计算出控制显示器的指令

显示器通过电路信号来显示文件内容到屏幕上

### 三、cpu中央处理器

#### cpu架构

厂商制的规范

X86架构：intel amd,X86针对windows系统，功耗高，性能好

ARM架构:智能手机

MIPS架构：国产龙芯

RISC-V：国产的，这是一个新的架构，开源免费的

操作系统和cpu是配合使用的

amd64就是指intel的x86

软件32位和64位的区别：

#### cpu针脚

以前的cpu有针脚，现在的cpu现在取消针脚，带针脚的叫pga封装，不带针脚的叫lga封装

#### cpu的性能参数介绍

CPU系列 英特尔 酷睿i7 8代系列

**CPU型号 Intel 酷睿i7 8750H**

8750代表标准版的cpu多用于台式机

8750K代表可以超频，多用于台式机，通过控制主板的参数可以超频

8750F不带核显

8750H主频低一些，多用于笔记本

8750U主频更低，主打低功耗

8750G,代表以前没有，最近几年出现的多用于笔记本，表示带有intel的锐炬核显比普通好核显好。

**CPU主频 2.2GHz**

(GHz千兆赫)一秒可以进行22亿次运算，交流ac或电磁em波频率的单位

**最高睿频** 4.1GHz运行流畅

最高睿频 代表官方自动帮我们超频，最高是4.2GHz

**核心/线程数 六核心/十二线程多任务**

（核心是指一个cpu上有6个小核，每个小核就是一个单独的cpu）

核心数越多同一时间处理的任务越多

线程，就是每个进程里会有多个小的线程

超线程提高的是一个核的处理线程的能力，在cpu核心里的单独拿出某些单元处理额外增加处理线程数。官方说提高40%，相当于假双核心的意思

**三级缓存 9MB**

比如说1+2+5 1+2等3 这个3就是缓存后再去加5

1级离cpu最近，放不下再到2级，最后放不下再放3级。速度上1>2>3，容量上是1<2<3,

cpu会优先去L1去找数据找不到再去2，再去3再找不到去内存，相当于cpu内置的内存。

**功耗 45W耗电高**

**cpu热设计功耗TDP**

45w就是打满用到45W

**内存参数**

支持的最大内存

工艺：7nm 14 22 32 65纳米

扩展性：最大支持内存

### 四、内存

cpu与硬盘之间的缓存区，断电数据消失，易失性存储设备，硬盘为永久性存储设备

内存的大小受限于cup，操作系统

ddr 133MHZ.266MHZ,内存的频率

内存的双通道技术，一个数据同进由两个内存的存取，速度更快了

### 五、硬盘

### 机械硬盘

台式是3.5寸，笔记本是2.5寸

传输速度100M-300M

### 固态硬盘ssd

比机械更快，更耐用，更薄，更贵

### 六、显卡

系列

RTX2000系列，RTX3000系列

性能：显存 电影基本就是一秒30帧。显存越大能力越强，

显存位宽512位，一次发送的数据

cuda，显卡内部的处理器

### 七、主板

支持内存的槽越多越好，显卡支持的数量

Z,B,H，系统属于高中低端

### 八、其他设置

电源，主要是功耗

### 九、计算机的分类

巨型机，中国的银河2号，神威的太湖之光

大型机，小型，微型，服务器

服务器和个人电脑的差，cpu的运行速度，数量，硬盘数量，ecc内存有纠错功能

## windows进阶

### 用户密码

WIN10专业版 修改密码 电脑》电脑管理》使用者》修改密码

u盘pe修改密码，pe进入可以修改源系统本地的所有数据

防止pe修改密码：1，可以用微软账号登陆（需联网）

2，设置bios密码(破解办法：bios密码是保存在纽扣电池里的，拆下放电后，数据就会丢失)

### 开机自启动

1. win10，设置》应用》启动
2. 添加开机自启动win+r>>shell:startup>>把快捷方式放入文件夹就可以了
3. 注册表的方式：

電腦\HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

### 文件共享

压缩分卷压缩，解压其中一个就可以把分卷压缩的文件解压成一个文件

物理工具：u盘，移动硬盘

windows共享：1、文件右键》》属性》》输入用户（通过哪个用户访问这个文件夹）

2、网络》》属性》》高级选项

第三方软件：飞秋、奶牛快传

### 远程控制、远程桌面

向日葵、qq远程协助、todesk、teamviewer等等，

工作原理A电脑B电脑不是直接连接，是连接到第三方平台，由平台中转连接。

远程桌面：A电脑》B电脑直接连接，

开启条件：1，被控方需要开启远程桌面。2，需要知道用户名和密码。3，ip地址

开启远程桌面：我的电脑》属性》远程设置》允许远程

### 注册表regedit

远程桌面注册表控制方法

電腦\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server

fDenyTSConnections 这个值不允许是1，允许是0

开机自启动

電腦\HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

将里面的数值改成0或在直接删掉数值即可

電腦\HKEY\_LOCAL\_MACHINE\SOFTWARE\Windows\CurrentVersion\Run

### Windows安全组策略gpedit.msc

禁止某应用的启动

用记配置》系统》不运行指定windows应用程序》启用，把进程的名称添加进显示里。

禁止插u盘

### windows服务

我的电脑》管理》服务

常见的服务DHCP clinet

服务的添加与删除，管理员打开cmd》sc create 自定义服务名称 binpath=主程序所在路径\某.exe 注：添加的程序得具备本身有服务的内容的软件

sc 是windows用于创建服务的指令

create是创建的意思

删除服务 sc delete 自定义服务名称

### windows创建计划任务

我的电脑》管理》计划任务

### 用户和用户组

我的电脑》管理》使用者/群组

### 指令操作

CD ..回到上一级目录 CD ../..向上回两级目录

cls 清空面板

exit 退出面板

C: 切换盘符

dir显示当前目录

md 文件夹名(创建文件夹)

type nu1 > 文件名 –创建一个空件，比如type nu1 > text.txt

del /s 文件名 –删除文件

rd /s 文件夹名 –删除文件夹

用户操作相关指令

net user #查看用户列表

net user 用户名 密码 #改密码

net user 用户名 密码 /add 创建一个新 用户

net user 用户名 /del 删除一个用户

net user 用户名 /active:yes/no #激活或禁用账户

用户组操作相关指令

net localgroup #查看组列表

net localgroup 组名 #查看该组的成员

net localgroup 组名 /add #创建一个新的组

net localgroup 组名 用户名 /add #添加用户到组

net localgroup 组名 用户名 /del #从组中踢出用户

net localgroup 组名 /del #删除组

### 激活administrator用户

cmd》管理员身份打开

net user administrator /active:yes

## linux

### linux延伸出的体系

#### redhat红帽系列

主要做企业级的操作系统，是一个非常成熟的体系，他有几个分支，fedora,redhat,centos,其中centos是纯免费的,redhat里面有一些封装的软件是需要收费的。

#### debian系

自己的分支，debian,ubuntu,kali

#### suse系

自己的分支opensuse,suse企业版

#### android系列

小米：miui系统

oppo:coloros系统

华为：emui,harmonyos(鸿蒙操作系统)

荣耀:magicui系统，后面带ui的其实就是提供了自己设计的图形化操作界面

#### 快照

先关机>>shutdown –h now #现在关闭

右键系统》》快照》》拍摄快照》》命名

后期系统玩坏了，到菜单虚拟机》》快照》》快照管理器》》点快照》》转到就恢复到快照节点

### centos

#### 安装与下载

清化大学镜像站<https://mirrors-i.tuna.tsinghua.edu.cn/>

硬件分配内存2G,CUP1核，无图形化界面的

#### 远程连接工具连接控制系统

查看ip ip addr

##### ssh远程连接

语法：ssh 用户名@ip地址

比如：ssh [root@192.168.61.131](mailto:root@192.168.61.131)》》回车》》输入用户名后回车》》输入密码后回车

##### 其他远程工具

xShell下载：<https://www.xshell.com/zh/xshell/>

secureCTR、tabby、openSSH等

macOS电脑上用的iterm2

##### 简单的指令总结：

断开连接：exit

重启系统：reboot 或shutdown –r now

关机指令：shutdow –h now #立即关机 shutdow –h 10#10分后关机

#### 文件操作

##### 创建文件

touch 1.txt#创建一个1.txt的文件

touch genel{1..10}.txt #创建genel.1到10个txt文件

touch .xx.txt #创建隐藏文件

##### 查看文件

ls

ls -1 #显示方式一行显示一个文件

ls –L #显示当前文件的详细信息

ls \*.txt #只查看txt的文件

##### 重命名

mv gentel.txt gt.txt #将gentel.txt 重命名为gt.txt

##### 删除文件

rm gt.txt #删除时会提示确认，>>y确认删除

rm –f 1.txt 2.txt #-f强制删除，不提示

##### 拷贝文件cp(cope)

cp 3.txt 33.txt

#### 目录操作

mkdir 目录 (make directory创建文件夹)

mkdir –p xx/kk/dev 一次性创建多级目录

cd .. 返回上一次目录 cd ../..返回上上级目录

cd / 返回根目录

mkdir test{0..3} 批量创建 test0,test1,test2,test3

rm –r test0 删除文件夹》》确认》》y

rm –r -f test1 无提示强制删除

mv old new 重命名

pwd 当前所在路径

history 历史指令

ls /root/gentel 查看root下的gentel目录的文件

#### vi文件编辑

： i\o\a

esc键 esc键

#### 查看文件内容

cat 文件 #从上往下显示文件内容 cat –n 文件 #前面加入行号

head 文件 #显示前几行数据默认显示前10行 head -5 显示前5行

tac 文件 #倒着从下往上显示文件内容

tail 文件 #默认后10行 tail -5显示后5行

#### 管道 |

ip addr | tail -5 #查看ip地址，在显示结果中显示后5行

ip addr | tail -5 | head -1 ip地址显示，在显示结果后5行, 5行中显示第一行

#### 文件内容过滤

wc –l 文件 #统计文件有多少行 -c #统计有多少字节 ls /bin | wc –l #统计bin中有多少个文件

grep “pwp” 11.txt #11.txt里显示关于pwp的内容行 –w 精准匹配

ip addr|grep “inet” 显示ip地址中有关于inet的行

awk用列过滤数据

awk ‘{print $1}’1.txt #显示1.txt中第一列的数据

awk –F ‘,’ ‘{print $1}’#显示有，为分隔符为第一列的数据

sort排序

sort 1.txt #以排序的方式显示1.txt的内容 –n 字母在前数字在后

uniq去除

uniq 1.txt #去掉重复的数显示

cat 1.txt#uniq #将1.txt中的内容去重显示（连续的有去除，不连续的不去重）

cat 1.txt|sort|uniq #这样就可以完全去重

#### 生成数字序列

seq #sequence序列的意思

seq 3 8 生成3-8的数字 seq –w 3 100 #生成等完的序列如001。。010.。100

### 目录结构介绍

/ 根目录

/bin#存放二进制的可执行文件，也就是命令，其实每个命令基本都是一个可执行代码文件，特别重要，不能删除!#window的命令文件都是exe结尾的，1inux的命令文件是没有后缀名的，如果删除了某个命令文件，那么这个命令就不能用了

/boot#开机启动需要的文件，特别重要,不要动里面的文件。

/dev#dev全称：Devices，硬件设备摆制文件，特别重要，千万不要动!

/etc#存放系统的各种配置文件，相当于windows的注册表，也就是超大的配置文件，特别重要，不能删除!

#比如改密码什么的，其实都是修改的etc下面的某个配置文件(shadow文件)中的配置，还有安装的各种软件配置文件，一般也是放到这个目录，也可以放到其他目录，但是一般都是放到这里。

/home#所有普通用户的家目录就在这个home目录下，每个用户目录中都有自己的桌面等目录，windows都是放到了Users目录下

/root#root用户的专属家目录，特别重要,不能删除!

/1ib

#1ibrary 32位库，一般是s0结尾的库文件，特别重要，不能删除!

#so结尾的库文件，类似于windows下的系统的d11动态链接库文件。千万不要尝试删除，试试就逝世。如果有快照的话你可以试试。

/1ib64#library 64位库，一般是s0结尾的库文件，特别重要，不能删除!

/media#多媒体文件目录(音乐、视频、文档等)，是一个不重要的目录，只是1inux的作者希望用户能够按照对应目录来存放内容，这个目录

普通用户是没有权限删除的，root用户可以删除它

/mnt#全称:mount，挂载的意思，一般是用来挂载光盘，U盘，也就是插入U盘、关盘等，打开之后一般都是在这个mnt目录下，这个目录

也是可以删除的

/opt#部分软件安装存储目录，安装的某些软件的时候，如果默认安装，那么它可能会将自己的程序安装到这个目录下，如果你不用这个目录，那么也是可以删除的目录

/proc#全称:process，是进程的意思，每个进程编号一个目录。通过ps-ef指令可以查看到进程编号,特别重要,不能删除!

/sbin#全称:superbin，是超级用户才能使用的命令，特别重要，不能删除!

#比如普通用户是没有关机(shutdown)、重启(reboot)等危害比较大的指令，这些命令文件在sbin目录中

/srv#之前用来存放软件源代码文件的，这个目录也没啥用。源代码-->编译-->打包-->软件包，软件包在windows下叫做可执行文件,

1inux下叫做命令文件。其实叫啥都行，明白它是啥即可。

/sys#全称:system，是系统功能目录，特别重要，不能删除！

tmp#全称:temporary，临时的意思，用来存放临时文件的目录，这里面的文件如果长时间没用的话，会被系统自动清除。

#windows下也有很多这样的临时目录，多数都是隐藏目录，比如c:\users\ 户名\AppData\Loca1\Temp，好多清理系统垃圾的软件,其实都是删除了一些临时文件

/run#运行，程序运行的时候产生的文件，多数也是临时文件，但是这里的文件不会被系统自动清除

/usr #用户级的目录，usr全称是UNIX software resource，主要存放的是一些软件程序以及这些程序所需要使用的库，当然也会保存/usr一些程序需要的资源文件，特别重要，不能删除!usr目录下面的文件夹种类和/根目录下很像，比如都有bin目录，1inux作者这样的设计是想告诉用户，系统的命令文件放到根目录的bin下，用户自己安装的某些软件的命令文件，放到/usr/bin下面。#全称是variable，用来存放一些经常变动的文件，比如日志文件、网页文件、缓存等，特别重要，不能删除！

/var #这个目录下我们安全人员一般只关注1og日志目录。比如用户登录系统、什么时候登录的、登陆了几次等等，都会在1og目录下产生日志记录。

### 相对路径和绝对路径

绝对路径就是从根目录开始写路径

### 用户管理

创建用户：useradd 用户名 删除用户：userdel 用户名

彻底删除用户的所有信息userdel -r gentel

用户改密码：passwd 用户名》》》密码

查看用户是否存在： id 用户名

查看有哪些用户： cat/etc/passwd

用记的修改：usermod #modify -L(lock)锁定/禁用用户，-U(unlock)解除锁定，

查看用户的当前状态：lchage –L 用户名

密码的存放文件：cat/etc/shadow

强制替换密码：在root权限下替换shadow文件，或者把shadow中的密码替换一下

windwos只支持单用户登陆，而linux可以多用户登陆

查看当前有多少终端连接：w

### 用户组管理

添加组：groupadd 组名

查看当前有哪些组：cat /etc/group

创建用户时指定组：useradd –g 用户名 组名

查看用户属于哪些组：id 用户名

删除组：groupdel

改组名：groupmod –n old new

将用户添加到哪个组：usermod –G 用户名 组名

### 权限管理

-rw-r--r-- 1 user user 10240 Apr 1 17:25 1.py

dr-xr-xr-x. 0 root root 0 Jan 1 1970 documents

第一段的第一个字符，表示文件类型-文件，d目录，1软链接，

第一段2到4字符，表示该文件所属用户的权限(user)u

第一段5到7字符，表示该文件所属用户组的权限(group)g

第一段8到10字符，表示其他用户对该文件的权限(other)o

ugo体系

.代表开启了selinux

查看是否开启：sestatus》》》enabled表示开户状态，disabled表示禁用状态

如何关闭：vi /etc/selinux/config>>set number>>第7行selinux=enforcing,将enforcing改成disabled》》esc>>qw！保存退出》》reboot重启系统

1表示文件的硬链接数量

user user 用户和用户组 10240文件大小默认为b字节

ls –lh #h代表人类的意思，显示的文件大小以方便读的方式显示，如1.3k或1.3m

Apr 1 17:25 文件的修改时间

如何查看文件的三个时间：stat 1.py

user@:~$ stat 1.py

File: 1.py

Size: 10240 Blocks: 0 IO Block: 4096 regular file

Device: 3h/3d Inode: 4650 Links: 1

Access: (0777/-rwxrwxrwx) Uid: ( 1000/ user) Gid: ( 1000/ user)

(访问时间)Access: 1970-01-01 08:00:00.000000000 +0800

（修改时间）Modify: 2025-04-01 17:25:23.000000000 +0800

（改变时间）Change: 1970-01-01 08:00:00.000000000 +0800

#### chmod修改文件权限

如: chmod -r 1.py #将所有的读权限都减掉了

chmod +r 1.py #将所有的读权限都加上了

chmod o-r 1.py #这个操作就是将其他用户的读权限减掉了

chmod g-x,o-x 1.py #将组权限，其他权限的可执行减掉了

#### 权限值

r 4代表read读权限

w 2代表write写权限

x 1代表executable可执行权限

- 0代表空权限位

rwx r-x r-x 权限值就是755 写法：chmod 755 1.py

#### 修改目录的所属

chmod 755 documents/

#### 修改文件的所属

chown 用户名：用户组 文件名

如chown gentel:gentel 1.py #这样就所user的用户名和用户组全改成gentel了

#### 通过uid和gid的值来控制所属

chown 1001:1001 1.py

#### xftp工具上传文件

#### 可执行程序特殊目录说明

windows可执行文件的环境变量，windows中右键》》我的电脑》》属性》》高级系统设置》》高级》》环境变量》》path,添加的快捷路径后，通过电脑的cmd就可以直接开启

linux 通过 echo $PATH可以看到，类似于windows的环境变量中的PATH，反式放到这个目录中的命令程序，我们可以在任意目录下通过这个命令程序名称来直接调用命令来执行。

user@:~$ echo $PATH

/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games

将可执行文件放到这些目录里，就可以直接执行了，相当于windwos改变了程序的环境变量

### linux shell

如ls cat都是shell的命令，什么自己检测命令正确不正确

shell提示符[root@localhost ~$] #[用户名@主机名 所在目录]

默认是localhost，修改主机名：hostname gentel>>logout#退出重新登陆

~波浪符表示，当前用户的家目录

#代表当前用户是管理员,$代表当前是普通用户

提示符可自定义

#### 查看主机的详细信息hostnamectl

#### 关机shutdown

#### 重启reboot

#### shell基础语法

ls –alh

rm –rf 删除根目录

cd，mkdir，useradd 用户名

cp –a –v #-v显示拷贝过程，-a保留原文件属性

#### tab键补全

一次tab键是补全，两次tab键，提示预选命令或者路径

#### 快捷键

CTRL+a光标跳到命令最前面

CTRL+e光标跳到命令尾部

CTRL+c终止前台运行的的程序

CTRL+d退出shell

CTRL+z将任务暂停，挂至后台，执行fg命令断续执行

CTRL+l清屏和clear命令等效

CTRL+k删除从光标到行未的所有字符

CTRL+u删除从光标到行首的所有字符

CTRL+r搜索历史命令，利用关键字搜索

CTRL+w光标往前删除一个参数，以空格为分割

#### history历史命令

history –c 清除历史指令

历史命令会保存在家目录下的.bash\_history文件里，需要

#### 命令别名

alias 查看当前系统的别名

添加别名，如ls加颜色，alias ls=’ls –color=auto’

取消别名：unalias ls

这些都是临时方案

永久方案，修改配置文件，当前家目录隐藏文件.bashrc

### 优化ssh登录速度

如果你发现你的xshell或者ss指令连接linux系统的时候很慢，等好久才连接成功，那么就可以按照下面的方式优化一下，目前我们没有

遇到，所以这个暂时就不提了。

#修改配置文件，先备份

[root@localhost ~]#cp/etc/5sh/sshd\_config/tmp/

[root@localhost ~]#vi /etc/ssh/sshd\_config

#直按输入:79回车

79 GSSAPIAuthentication no

115 UseDNS no

输入:wq保存退出

#重启sshd服务

[root@localhost~]# systemctl restart sshd

#如果修改失败，还原配置文件

[root@localhost ~]# cp/tmp/sshd\_config /etc/ssh/sshd\_config

[root@localhost ~]# systemctl restart sshd

### vi常用操作

#### 移动光标

左，下，上，右，如果键盘上没有上下左右键，可以 h，j，k，l 。

进入编辑模式有三个按钮：i、a、o ，i 在光标位置编辑、a 是在光标后一位编辑、o 是换行编辑，新起一行。

ctrl + f 下翻一页

ctrl + b 上翻一页

ctrl + u 上翻半页

ctrl + d 下翻半页

0 跳至行首，不管有无缩进，就是跳到第 0 个字符

^ 跳至行首的第一个字符

$ 跳至行尾 (shift + 4)

gg 跳至文首

G 跳至文尾 (shift + g)

5gg / 5G 调至第 5 行，或者命令行模式 :5 回车，也是跳到第 5 行，所以其实操作命令都不是唯一的

#### 二、删除复制

x 删除单个字符

10x 删除 10 个字符

dd 删除光标所在行 (其实 dd 是剪切的操作) ，# 使用 u 撤销之前的操作，使用 ctrl + r 恢复

6dd 从光标开始往下删除 6 行

dw 删除一个单词（word）

yy 复制行 ，5yy 复制 5 行，复制的内容可以通过 p\P 来黏贴

小 p 粘贴粘贴板的内容到当前行的下面，比如将 dd 剪切的行黏贴到下面

大 P 粘贴粘贴板的内容到当前行的上面

#### 三、搜索和替换

搜索：/>>回车

/pattern 向后搜索字符串 pattern #辅助小 n 向下和大 N 向上，一般都是用 / 来搜索

-?pattern 向前搜索字符串 pattern #辅助小 n 向上和大 N 向下，? 搜索用的少

替换：

:1369s/shell/jaden/g 将第 1369 行的 shell 替换为 jaden，还可以用 #或者 @符号来代替：:1369#shell#jaden#g

:1369,1379s/shell/jaden/g 将 1369 至 1379 这 10 行中的 shell 替换为 jaden

:1369,$s/shell/jaden/g 将 1369 至文末中的 shell 替换为 jaden

:% s/old/new/g #搜索整个文件，将所有的 old 替换为 new

:% s/old/new/gc #搜索整个文件，将所有的 old 替换为 new，每次都要你确认是否替换 (y/n/a/..)，y 表示确认替换一个、n 表示不替换、a 表示全部替换

#### 四、退出编辑器

:w 将缓冲区写入文件，即保存修改到硬盘上，但不退出 vi，如果我们改到一半的时候可以提前保存一下，以防断电，因为新编辑的数据是在内存中的，而且 vi 不会自动保存。

:wq 保存修改并退出

:x 保存修改并退出，和 wq 一样的效果。

:q 退出，如果对缓冲区进行过修改，则会提示

:q! 强制退出，放弃修改

:wq! 强制保存修改并退出

#### vi原理和注意问题

vi 在编写的时候会创建临时文件，如编写a.txt会生成.a.txt.swp，他的逻辑就是在临文件里编写好后，用临时文件替换原文件。但是如果在编写的时候不小心中断，临时文件未能替换原文件，这时候还可以可用vi –r a.txt还原，然后再保存后再删除临时文件

### linux输入输出重定向>

例如：head -5 services > 1.txt #将services这个文件的前五行数据保存到了1.txt里去了

**输出重定向**

echo 123 > 1.txt #清空1.txt里的内容，将123保存进1.txt中。

echo 123 >> 1.txt #将123追加到1.txt文件尾部，不会清空原有内容。

ifconfig > 1.txt 2.txt #如明指令正确，或者有内容就保存到1.txt否则就保存到2.txt

**输入重定向**

cat < 1.txt #将1.txt的内容给cat并显示

cat < 1.txt > 2.txt #将1.txt的内容给cat，cat将1.txt的内容给2.txt

cat << 0 > 1.txt #命令行输入任意内容，输入0结束输入，结束后将内容保存到1.txt

### 压缩与解压缩tar.gz

#### tar压缩

压缩tar –zcf 压缩包+路径 目标1 目标2 目标3

如tar –zcf 2.tar.gz 1.txt 2.txt

**解压tar –zxf** 解压路径

-C选项 可以指定解压到哪个文件夹，目录文件名必须存在才可以

tar –zxf tar/2py.tar.gz 将tar/2py.tar.gz的压缩文件解压到当前目录

tar –zxf tar/2py.tar.gz –C tar 将tar/2py.tar.gz解压到tar的目录

**查看压缩文件tar –tf**

#### gzip（压缩） gzip –d (解压)

区别gzip压缩好后源文件就删除了，解压后压缩包的源文件就删除了，只能单独压缩每个文件

#### zip压缩

压缩zip –r 文件名 目标1目标2目标3 如:zip –r 1.zip 1.txt 2.txt 3.txt

解压unzip 如:unzip 1.zip

#### rar在lunix需要安装工具

### 与linux文件传输

windows与linux的文件传输 windows10>>scp,xftp,rz linux>>windows wftp sz

**linux与mac scp**

把本地的文件推送到服务器：scp 1.txt [root@192.168.1.1:/tmp](mailto:root@192.168.1.1:/tmp) #用户名@ip:/路径

将远程服务文件拉取到本地：scp root@[root@192.168.1.1:/tmp](mailto:root@192.168.1.1:/tmp)/1.txt

网站与linux传输curl(系统自带)， wget（需安装yum install –y wget）

下载文件

curl –o 本地存放路径 文件网址。如：curl –o 123.zip <https://github.com/.../tt.zip>

wget <https://github.com/.../tt.zip>

**查看文件类型： file 1.zip**

### 软件的安装

#### 编译安装

#### rpm安装

redhat打包出来的包叫做rpm包

debian打包出来的包叫做deb包

安装rpm包 –i是安装,vb显示安装进度

如：rpm –ivb tree-1.6.0-10.e17.x86\_64.rpm

卸载：rpm –e tree

升级：rpm –Uvh xxx.rpm

查看已经安装的软件：rpm –qa|grep httpd

#### yum安装

比较便捷自动安装依赖包

安装yum仓库：yum install epel-release –y

安装nginx:yum install nginx –y

移除nginx:yum remove nginx –y

查看仓库rpm的数量：yum repolist

查看哪一个软件包中涵盖pstree这个指令：yum provides pstree

#### 总结：

编译安装优点：自由定制，痛点：难度高，步骤繁琐

rpm安装优点：安装简单，痛点：需要自己解决依赖

yum安装优点：自己解决依赖，默认安装最新版，痛点：不支持定制

#### yum仓库

### find文件查找

find /etc –maxdepth 1 –type f –name”pa” –size +100m （-2k）

命令 目录 查找深度 类型 文件名包含 大于100m的文件 小于2K的文件

-size -100 –and –size +50M 小于100大于50的文件

-imame 名字不区分大小写

根据时间查找： -mtime -1 #-1代表一天之内，+1代表一天以前

-ntime 分钟

根据用户组查找：-group gentel

对找到的文件进行处理-exec

find /root –type f –name “\*.txt” –exec rm –rf {} \; #删除找到的txt文件

find /root –type f –name “\*.txt” –exec cp –a {} /tmp \; #复制到tmp目录

### 进程管理

#### 查看进程ps

参数1：ps –ef

pid 全称process id 进程编号

uid 全称user id 里程所属用户

cmd 表示这些是系统级别的进程，或执行的系统任务

ppid 全称parent process id. 父进程

stime 全称start time。进程的启动时间

tty: 用来显示哪些是本地启动的，哪些是远程终端连接的，通过w指令可以看到哪些终端主机，远程终端会显示pts,tty代表是本地登陆连接

#### 关闭进程kill pid号

例如：kill 7851 #使用进程号，来终止进程

批量关闭：pkill 进程名 例如：pkill top

kill -9 进程名 强制关闭，慎用

有些进程需要关闭父进程（ppid）才能关闭，

### 查看计算机硬件信息

查看cpu: lscpu

查看内存: free –h

查看硬盘命令： df –h

查看计算机cpu，内存，进程等信息：top load average:负载率 Swap:虚拟内存

### linux定时任务

显示目前电脑上的有的定时任务：crontab –l

制定定时任务crontab –e

定时任务格式：

\* \* \* \* \* cmd

分 时 日 月 周 命令

分：0-59 时：0-23 日0-31 月1-12 周1-7

每5分种执行一次：\*/5 \* \* \* \* /5代表每5分钟，如果只写5就是每小时的05

每1小时的01分钟执行一次：01 \*/1 \* \* \*

第半个小时执行一次：00，30 \*/1 \* \* \*

每天晚上8点执行：00 20 \* \* \*

每个月1号晚上8点执行：00 20 1 \* \*

每年1月1日晚上8点执行：00 20 1 1 \*

时间命令

date显示时间和日期 date +%F显示日期 date +%T显示时间

user@:~$ date +%F\ %T

2025-05-24 13:11:35

修改日期和时间 date –s ‘20240723 14:40:00’

和网络时间同步：sytemctl restart chronyd

监测文件的最后一行：tail –f time.txt

哪个文件在执行定时任务：ps –ef|grep cron

自动重启这个进程：systemctl restart crond

### 系统优化

1优化ssh,以防止连接过慢

vi /etc/ssh/sshd\_config #改配置文件最好先备份

79行:GSSAPIAUthentication no

115行：UseDNS no #别忘记删除前面的#

systemctl restart sshd

2优化selinux

vi /etc/selinux/conifg

7行SELINUX=disabled

临时使用setenforce 0这个命令属于临于优化，上面改的文件是永久的

3关闭firewalld(防火墙)

systemctl stop firewalld #关闭防墙

systemctl disable firewalld #关闭防墙自启动

4安装常用软件

yum install lrzsz vim tree wget net-tools screen tcpdump bash-completion –y

net-tools:网络相关工具,查看ip和网卡ifconfig，安装的有哪些软件：rpm –qa

具体这些软件有哪些命令：rpm –ql net –tools来查看

统计软件包的个数rpm –qa|wc -l

screen：屏幕工具（网络不稳定时需要用到这个工具，操作之前先打开screen）

查看之前断开连接的任务screen –ls

继续执行之前的命令screen –r 编号（如1910）

bash-completion: 超级自动补全，TAB键的加强版，参数都会提示，安装包都会提示

tcpdump: 抓包工具

### 服务管理

服务的操作：开机自启，重启服务、启动服务、停止服务等等

安装服务：yum install httpd

查看服务列表：systemctl list-unit-files

启动服务：systemctl start httpd

停止服务：systemctl stop httpd

重启服务：systemctl restart httpd

查看服务状态：systemctl status httpd

把服务设置为开机启动：systemctl enable httpd.service

取消服务的开机启动：systemctl disable httpd.service

httpd是一个网站服务软件，我们通过浏览器访问

### linux特殊符号

# 注释符号，忽略执行

#### 1、shell指令的其他符号：

；命令分隔符，通过他可以多条指令一起执行

..代表上一级目录

.代表当前目录

#### 2、当前系统的环境变量：env

XDG\_SESSION\_ID=1

HOSTNAME=localhost.localdomain # 变量作用：比如HOSTNAME这个变量，它的值比较长，系统内部程序会经常用到这个值，那么用一个变量存放，以后想用这个值，就用这个变量即可，简单很多，而且只要修改了这个值，其他使用这个变量的地方，值都会跟着变化，方便修改。

TERM=xterm

SHELL=/bin/bash

HISTSIZE=1000 # 这就是为什么历史命令只记录1000的原因。

SSH\_CLIENT=192.168.61.1 50670 22

SSH\_TTY=/dev/pts/0

USER=root # 当前登录用户，其他用户登录的时候，这个变量对应的值就变为了其他用户名

...

LANG=zh\_CN.UTF-8 # language的简写，装系统的时候，你安装的英文，这里就是en\_US，中文就是zh\_CN

...

#### 3、查看环境变量$

**查看某个变量的值 $符号+变量名称：**

[root@localhost ~]# echo $LANG

zh\_CN.UTF-8

#### 4、修改环境变量export

#我们改一下语言变量，来看看效果，比如之前命令的参数介绍都是中文的,export用来声明环境变量、修改环境变量等，如下：

[root@localhost ~]# usermod --help

用法：usermod [选项] 登录

选项：

-c, --comment 注释 GECOS 字段的新值

-d, --home HOME\_DIR 用户的新主目录

...

[root@localhost ~]# export LANG=en\_US.UTF-8

[root@localhost ~]# export LANG=zh\_CN.UTF-8 #将系统文字改为中文

[root@localhost ~]# usermod --help # 全部变英文了

Usage: usermod [options] LOGIN

#### 5、双引号，换行，解析变量

比如：echo，本来只能输出单行文本内容加上双引号支持换行输入和输出

echo hello 输入出hello

[root@localhost ~]# echo "hello

> jaden

> "

hello

jaden

[root@localhost ~]# echo "hello

jaden

" > jaden.txt # 还可以输出到某个文件中

[root@localhost ~]# cat jaden.txt

hello

jaden

有时候比vi用起来方便。

# ""能够解析变量，如下

[root@localhost ~]# echo "$LANG"

en\_US.UTF-8

#### 6、''单引号，换行，不解析变量

# 单引号不能解析变量，其他功能和双引号类似，如下

[root@localhost ~]# echo '$LANG'

$LANG

#### 7、\和/ #这些我们都说过

\ 转义符，反斜杠

/ 路径分隔符

#### 8、! 历史命令调用，在find命令中是取反的意思。

[root@localhost ~]# history

[root@localhost ~]#!47 # 历史指令序号

#### 9、\* 通配符，我们将find的之后用到过，可以匹配任意字符

[root@localhost ~]# ls \*.txt # 查看所有.txt结尾的文件

jaden.txt

#### 10、$ 调用变量

例子1：

[root@localhost ~]# export LANG='en\_US.UTF-8'

[root@localhost ~]# echo $LANG

en\_US.UTF-8

[root@localhost ~]# stat 111.txt

File: '111.txt'

Size: 0 Blocks: 0 IO Block: 4096 regular empty file

Device: fd00h/64768d Inode: 33575641 Links: 1

Access: (0777/-rwxrwxrwx) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2021-09-12 04:40:28.399177386 +0800

Modify: 2021-09-12 04:40:28.399177386 +0800

Change: 2021-09-12 04:40:28.400260737 +0800

Birth: -

[root@localhost ~]# export LANG='zh\_CN.UTF-8'

[root@localhost ~]# stat 111.txt

文件: "111.txt"

大小: 0 块: 0 IO 块: 4096 普通空文件

设备: fd00h/64768d Inode: 33575641 硬链接: 1

权限: (0777/-rwxrwxrwx) Uid: ( 0/ root) Gid: ( 0/ root)

最近访问: 2021-09-12 04:40:28.399177386 +0800

最近更改: 2021-09-12 04:40:28.399177386 +0800

最近改动: 2021-09-12 04:40:28.400260737 +0800

创建时间: -

# 有些系统指令的帮助信息即使设置了中文，它的帮助信息都是英文，比如curl

[root@localhost ~]# curl --help

# 因为汉化这个事情，是好多人做的，而且要做很久，但是系统还会更新，还要汉化完，就又更新系统了，还得重新汉化，所以这样不划算，汉化组就撤了，所以有些指令就不汉化了。而且有时候汉化用来的意思和英语本来的意思不同，因为参与汉化的人有些不是计算机专业的，这就很尴尬。

# 使用变量的时候要小心，比如有些人操作变量的时候，删除了根目录，如下

[root@localhost ~]# jaden=/tmp # 定义变量

[root@localhost ~]# echo $jaden # 查看变量值

/tmp

[root@localhost ~]# rm -rf $jaden/\* # 利用变量做删除，这是删除/tmp目录下的所有内容

# ！！！但是，如果变量名写错了，如下

[root@localhost ~]# rm -rf $jadin/\* # 少写了个e字母，系统帮删除的时候，找不到这个变量，那么会变成如下效果

[root@localhost ~]# rm -rf / # 就删除根目录了！！！！！系统可就瘫了。所以看到删除命令，就一定要特别的小心。

#### 11. 重定向

> 输出重定向、>>追加输出重定向、< 输入重定向、<<追加入重定向

#### 12.管道|

#### 13.||双管道

# 第一个命令失败，才执行第二个命令，第一个指令成功了，不会执行第二个指令。

# 指令连接符号，之前说过；除了它，还有 || 和 &&，分号是两个指令都会执行，哪个出错都不影响其他的指令。

[root@localhost ~]# echo 123 || ls

123

[root@localhost ~]# ech 123 || ls

-bash: ech: command not found

anaconda-ks.cfg jaden.txt

#### 14、&&

# 和，两个一起执行，如果第一个失败了，两个都不执行，如果第一个成功了，第二个失败了，就执行第一个，如果两个都成功了，就都执行。

[root@localhost ~]# echo 123 && ls

123

anaconda-ks.cfg jaden.txt

[root@localhost ~]# ech 123 && ls

-bash: ech: command not found

[root@localhost ~]# echo 123 && lx

123

-bash: lx: command not found

#### 15、&

# 后台运行

# 比如top指令，之前运行top，会占住终端，除非结束top，不然不能执行其他指令

[root@localhost ~]# top &

[1] 16567

[root@localhost ~]# ps -ef |grep top

root 16567 1467 0 16:56 pts/0 00:00:00 top

root 16569 1467 0 16:57 pts/0 00:00:00 grep --color=auto top

16、~ 代表家目录，不同的用户家目录不同。

[root@localhost ~]# cd /tmp/

[root@localhost tmp]# cd ~

[root@localhost ~]# cd /tmp

[root@localhost tmp]# cd # 其实cd什么参数都不加，也是回到家目录

[root@localhost ~]#

#### 17、` 反引号（Esc下边的按钮）

# 用来嵌套命令，反引号中的命令先执行

例子1：

[root@localhost ~]# mkdir `echo jaden`

[root@localhost ~]# ls

anaconda-ks.cfg jaden jaden.txt

例子2：

[root@localhost opt]# touch test\_`date +%T`.txt

[root@localhost opt]# ls test\_04\;56\;22.txt

test\_04:56:22.txt

例子3：

[root@localhost ~]# tar zcf /tmp/etc.tar.gz `find /etc -type f -name "\*.conf"`

tar: Removing leading `/' from member names

[root@localhost ~]# ls /tmp/

etc.tar.gz

[root@localhost ~]# tar tf /tmp/etc.tar.gz # 查看压缩包里面的文件内容

### 三剑客和正则表达式

三剑客就是三个强大的命令。后期讲应急溯源的时候会用到。grep用的最多，所以我们重点讲解这个指令。三剑客配合正则表达式可以发挥很牛的作用。

grep

#擅长过滤

例子：/var/log/secure是记录用户登录系统的行为的，登录成功还是失败，系统都会自动记录这次登录动作。

[root@localhost ~]# grep 'Failed password' /var/log/secure

Mar 25 18:20:48 localhost sshd[16905]: Failed password for root from 192.168.61.1 port 55577 ssh2

# 如果我们想将所有登录失败的ip地址找出来，可以再次过滤

# 比如查看某个ip地址登录了多少次

[root@localhost ~]# grep 'Failed password' /var/log/secure|grep '192.168.61.1'|wc -l

3

# 可以借助正则表达式来进行过滤，可以将内容过滤的很精确，有些在线的网站也可以帮我们生成一些常用的正则表达式，比如https://www.hake.cc/tools/regexcode/

[root@localhost ~]# grep 'Failed password' /var/log/secure|grep --color -P "(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)"

Mar 25 18:20:48 localhost sshd[16905]: Failed password for root from 192.168.61.1 port 55577 ssh2

# ip地址都标红色了，表示匹配出来了。

# 如果只要ip地址-Po

[root@localhost ~]# grep 'Failed password' /var/log/secure|grep --color -Po "(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)"

192.168.61.1

# 有些ip地址是重复的，因为尝试登录了很多次，那么可以先排序，再去重，来查看ip地址

[root@localhost ~]# grep 'Failed password' /var/log/secure|grep --color -Po "(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|[0-1]\d{2}|[1-9]?\d)"|sort -n |uniq -c

2 192.168.2.113

3 192.168.2.116

1 192.168.61.1

# 比如，如果看到了这么个ip地址164.90.232.121、164.90.232.241等尝试登录了你上千上万次，就是在爆破攻击我们，百度一查，德国的

# 我们可以直接拉黑这个ip地址，也叫做加黑，加入黑名单的意思，这个等后面我再教大家。

#### grep擅长过滤行

#grep参数

-n 显示行号

[root@localhost ~]# grep -n 'tcp' test.txt

# vi test.txt +48，表示进入vi的时候，光标直接定位的48行起始位置。

-c 对结果行计数

[root@localhost ~]# grep -c 'tcp' test.txt

14

-i 不区分大小写

[root@localhost ~]# grep -n 'tcp' test.txt -i

-v 反向搜索，取反

[root@localhost ~]# grep -n 'udp' test.txt -v # 将不含有udp的行全部过滤出来

-w 精准匹配

[root@localhost ~]# grep -w 'tcp' test.txt

-o 只显示匹配的结果，前面的第一个示例中我们用过-o参数

[root@localhost ~]# grep -o -n 'tcp' test.txt

-A1 同时打印搜索结果行的后一行 ，A是after的简写

[root@localhost ~]# grep -A2 'ftp' test.txt

[root@localhost ~]# grep -A 2 'ftp' test.txt

-B3 同时打印搜索结果行的前三行，B是before的简写

[root@localhost ~]# grep -B2 'ftp' test.txt

-C2 同时打印搜索结果行的上下各两行

[root@localhost ~]# grep -C2 'ftp' test.txt

-E 扩展正则表达式

# 正则我们下面会细讲，先简单了解一下。

[root@localhost ~]# grep -E '.tp' test.txt #.就是正则表达式，表示任意的一个字符

[root@localhost ~]# grep -E 'ftp|ssh' test.txt # 查找ftp或者ssh，|是或者的意思，可以用多个ftp|ssh|telnet...

-P 使用perl正则

# perl语言中设计的正则表达式写法规则，很强大，很多领域都支持perl正则的语法结构。

[root@localhost ~]# grep -P "\d+" test.txt # 匹配所有的4位数

[root@localhost ~]# grep -P "\d{4,}" test.txt # 匹配4位的数字

#### sed查找替换删除

#擅长取行和修改替换[

用法：sed [-nri] [动作] 目标文件文件

选项与参数：

-n ：使用安静(silent)模式。在一般 sed 的用法中，所有来自 STDIN 的数据一般都会被列出到终端上。但如果加上 -n 参数后，则只有经sed 特殊处理的那一行(或者动作)才会被列出来。

-r ：sed 的动作支持的是延伸型正则表示法的语法。(默认是基础正则表示法语法)

-i ：直接修改读取的文件内容，而不是输出到终端。

动作说明：[n1[,n2]]function

n1, n2一般表示为行号

function:

a ：指定行后面插入一行

d ：删除

i ：指定行前面插入一行

p ：打印，#一般和前面的-n参数一起用

s ：替换 需要I忽略大小写，全局替换需要g

# sed删除匹配的行数据

[root@localhost ~]# sed '/tcp/d' test.txt # 删除所有带tcp的行，并不是删除原文件中的数据，而是将删除之后的结果打印出来了。

# 所以只需要一个重定向，就拿到删除之后的结果了

[root@localhost ~]# sed '/tcp/d' test.txt > 2.txt

[root@localhost ~]# cat -n 2.txt

# -i就可以直接删除原文件的数据

[root@localhost ~]# sed -i '/tcp/d' test.txt

[root@localhost ~]# cat -n test.txt

[root@localhost ~]# sed -i '/udp/d' test.txt

[root@localhost ~]# cat -n test.txt

[root@localhost ~]# sed '/^#/d' test.txt # 删除以#开头的行，^表示以什么开头的正则表达式，我没有加-i昂

# 指定行号来删除

[root@localhost ~]# sed '1,10d' test.txt # 删除1 - 10行的数据

[root@localhost ~]# sed -i '1,10d' test.txt

# 只要1 - 5行的数据

[root@localhost ~]# sed -n '1,5p' test.txt

# 只要第5行的数据

[root@localhost ~]# sed -n '5p' test.txt

# 插入数据

# 在第三行后面插入一行数据

[root@localhost ~]# sed '3a hello jaden' 1.txt

1

2c

3a

hello jaden

4b

5a

# 在第2行前面插入一行数据

[root@localhost ~]# sed '2i hello jaden' 1.txt

1

hello jaden

2c

3a

4b

5a

# 加 -i 参数就能直接修改原文件

[root@localhost ~]# sed -i '2i hello jaden' 1.txt

[root@localhost ~]# cat 1.txt

1

hello jaden

2c

3a

4b

# 替换数据

# 把3a替换成wulaoban

[root@localhost ~]# sed's/#3a#/wulaoban#' 1.txt

1

hello jaden

2c

wulaoban

4b

5a

[root@localhost ~]# sed -i's/#3a#/wulaoban#' 1.txt

[root@localhost ~]# cat 1.txt

1

hello jaden

2c

wulaoban

4b

5a

# 把A替换成xxx，每行只替换一次，每行的第一个及之后的A都不再进行替换

[root@localhost ~]# sed's/A#/xxx#' 1.txt

1

hello jaden

2c

wulxxxOBAN

4b

5a

# 把所有的A都替换成xxx

[root@localhost ~]# sed's#A#xxx#g' 1.txt

1

hello jaden

2c

wuLxxxOBxxxN

4b

5a

# 把所有的A和a都替换成xxx，忽略大小写，参数I

[root@localhost ~]# sed's#A#xxx#gI' 1.txt

1

hello jxxxden

2c

wuLxxxOBxxxN

4b

5xxx

#### awk

# 擅长取列

用法，取列

比如3.txt内容如下：

2 this is a test

3 Do you like awk

This's a test

10 There are orange,apple,mongo

例子1： 取列

[root@localhost ~]# vi 3.txt

[root@localhost ~]# cat 3.txt

2 this is a test

3 Do you like awk

This's a test

10 There are orange,apple,mongo

[root@localhost ~]# awk '{print $1}' 3.txt

2

3

This's

10

[root@localhost ~]# awk '{print $3}' 3.txt # 取第三列

[root@localhost ~]# awk '{print $NF}' 3.txt # 取每一行的最后一列，NF是固定写法

test

awk

test

orange,apple,mongo

[root@localhost ~]# awk '{print $1,$NF}' 3.txt # 取第一列和最后一列

2 test

3 awk

This's test

10 orange,apple,mongo

[root@localhost ~]# awk '{print $NF,$1}' 3.txt # 还可以反着写，所以通过awk，可以将列顺序重新排版

[root@localhost ~]# awk '{print $1,$4}' jaden.txt # 取出第一列和第四列

例子2：计算

vi 4.txt，写上如下内容

# 水果，每斤多少钱，总共多少斤

orange 10 20

apple 20 30

mongo 50 10

banana 5 200

# 开始计算

[root@localhost ~]# awk '{print $1,$2\*$3}' 4.txt

orange 200

apple 600

mongo 500

banana 1000

# 还可以加备注信息：

[root@localhost ~]# awk '{print $1"总价为：",$2\*$3"元"}' 4.txt

orange总价为： 200元

apple总价为： 600元

mongo总价为： 500元

banana总价为： 1000元

# 例子3：根据行号来筛选内容

# a = 1表示变量赋值，让a=1

# a == 1，表示判断一下a的值是不是等于1，等于1那么条件判断结果为真，不等1那么条件判断结果为假

# 支持符号： > < == >= <=

[root@localhost ~]# awk 'NR==1' 4.txt # 取出第一行数据，grep不会取出特定的行，只能筛选某些行

orange 10 20

[root@localhost ~]# awk 'NR>2' 4.txt # 取出行号大于2的行数据

mongo 50 10

banana 5 200

[root@localhost ~]# awk 'NR<=3' 4.txt # 取出行号小于等于3的行数据

orange 10 20

apple 20 30

mongo 50 10

[root@localhost ~]# awk 'NR<=3 && NR>1' 4.txt # 取出行号大于1并且小于等于3的行

apple 20 30

mongo 50 10

# 还可以取行的同时来取列

[root@localhost ~]# awk 'NR<=3{print $1}' 4.txt

orange

apple

mongo

# 例子4:

# 还可以过滤出指定的行，awk也能做过滤，但是

[root@localhost ~]# awk '/apple/' 4.txt # 取出含有apple数据的行数据

apple 20 30

# grep、sed、awk过滤对比

grep 'apple' 4.txt

sed -n '1,2p' 4.txt

awk 'NR>1 && NR<=3' 4.txt

# 例子5：

# 再比如我们刚才取ip地址：比直接写正则要方便很多

[root@localhost log]# grep 'Failed password' secure

Mar 25 18:20:48 localhost sshd[16905]: Failed password for root from 192.168.61.1 port 55577 ssh2

Mar 25 18:49:26 localhost sshd[1498]: Failed password for root from 192.168.2.113 port 49991 ssh2

...

[root@localhost log]# grep 'Failed password' secure|awk '{print $11}'

192.168.61.1

192.168.2.113

192.168.2.116

...

# 例子6： 指定分隔符，默认是按照空格作为分隔符的

awk -F ":" '{print $7,$1}' /etc/passwd # 这个文件都是用:做的分隔符

[root@localhost ~]# awk -F ':' 'NR==3 || NR==5 {print $1}' /etc/passwd # 取出第三行和第五行的第一列数据，分隔符为:

daemon

lp

# &&表示and，两个条件同时成立

# ||表示or，满足一个条件即可

# 例子7： 拼凑指定文本，双引号之间原样输出

# awk -F ":" '{print $1":123:"$7}' /etc/passwd

[root@localhost ~]# awk -F ":" '{print $1":123:"$7}' /etc/passwd

root:123:/bin/bash

bin:123:/sbin/nologin

...

# 例子8： 过滤文本

# awk -F "[ /]+" '$2~/\^47/' 1.txt

# 找出第一列数据中带h的，并取出第一列和第七列的数据

[root@localhost ~]# awk -F ':' '$1~/h/{print $1,$7}' /etc/passwd

shutdown /sbin/shutdown

halt /sbin/halt

sshd /sbin/nologin

chrony /sbin/nologin

apache /sbin/nologin

#### 正则表达式

1. 什么是正则表达式？

简单的说，正则表达式就是一套处理大量的字符串而定义的规则和方法。

例如：假设 @代表12345

通过正则表达式这些特殊符号，我们可以快速的过滤、替换需要的内容。

Linux正则一般以行为单位处理的。

2. 正则表达式

1) ^ 表示搜索以什么开头。

2) $ 表示搜索以什么结尾。

3) ^$ 表示空行，不是空格。

4). 代表且只能代表任意一个字符。

5) \ 转义字符，让有着特殊身份意义的字符，脱掉马甲，还原原型。

例如：\.只表示小数点，还原始小数点的意义。

6) \* 重复0个或多个前面的一个字符。不代表所有了。

7).\* 匹配所有的字符。^.\* 任意多个字符开头。

8) [abc] 匹配字符集合内任意一个字符[a-z]

9) [^abc] ^在中括号里面表示非，不包含a或b或c。

10) {n,m} 重复n到m次，前一个字符。

11) + 重复1次到多次

12)? 重复0次到多次

示例：

1) ^ 表示搜索以什么开头。

[root@localhost ~]# grep '^#' test.txt # 找出开头为#号的的行数据

[root@localhost ~]# grep -v '^#' test.txt # 找出开头不是#号的行数据

2) $ 表示搜索以什么结尾。

[root@localhost ~]# grep 'ol$' test.txt # 找出结尾为ol字母的行数据

3) ^$ 表示空行，不是空格。

[root@localhost ~]# grep '^$' test.txt # 找出所有的空白行。

[root@localhost ~]# grep -v '^$' test.txt # 找出所有的非空白行数据，如果保存一下，那么相当于快速删除所有空行。

# sed删除空行，更容易，因为它更擅长修改文件内容

[root@localhost ~]# sed '/^$/d' test.txt # 加上-i就改了原文件

4) . 代表且只能代表任意一个字符。跟癞子斗地主中的癞子一样。

[root@localhost ~]# grep '.dp' test.txt # 找出xdp，x可以是任意一个字符，注意，是一个！

5) \ 转义字符，让有着特殊身份意义的字符，脱掉马甲，还原原型。

例如：\. 只表示小数点，还原原始小数点的意义。

[root@localhost ~]# grep '\.ia' test.txt # 这就表示，找到含有.dp的行数据

6）\* 重复 0 个或多个前面的一个字符。不代表所有了。

准备一个测试数据：

[root@localhost ~]# vi num.txt

[root@localhost ~]# cat num.txt

a 12222

b 12222222

c 1

b 3

d 12222222222222

e 12

a 4

看下面几个的效果就知道 \* 的意思了。

[root@localhost ~]# grep '2' num.txt

[root@localhost ~]# grep '22' num.txt

[root@localhost ~]# grep -o '22' num.txt # -o 就能看出来，它是两个字符两个字符的匹配

[root@localhost ~]# grep '12' num.txt

[root@localhost ~]# grep '12\*' num.txt #对于2这个字符进行0次或者多次匹配

[root@localhost ~]# grep '.' num.txt # 匹配所有字符

[root@localhost ~]# grep '12.' num.txt

7）.\* 匹配所有的字符。^.\* 任意多个字符开头。

[root@localhost ~]# grep '12.\*' num.txt

小任务：

1. 复制一份 /etc/services 的前 100 行数据到 serv 文件中，然后将去掉 serv 文件中所有含有 #的行和空行

[root@localhost ~]# head -100 /etc/services > serv

用 sed 来玩：

[root@localhost ~]# sed '/#/d' serv # 去掉含有 #号的行

[root@localhost ~]# sed '/^$/d' serv # 去掉所有空行

用 grep 来玩：

[root@localhost ~]# grep -Ev '^$|#' serv > ss.txt

[root@localhost ~]# cat ss.txt

2. 去掉 aaa.txt 文件中所有的空行和注释（注意：一行中可能只有部分是注释）

[root@localhost ~]# cat aaa.txt

hello -a #a 表示友好

hello -b #b 表示不友好

hello -c #c 表示骂人

哈哈，这是个啥

what?

hello -d #d 表示非常爱你

[root@localhost ~]# grep '#.' aaa.txt # 这么写，可以将所有 #号注释的内容全部匹配到，所以这个正则表达式可以用

[root@localhost ~]# sed -i 's/#./g' aaa.txt # 将 #.\* 匹配到的内容全部替换为空。

[root@localhost ~]# sed -i '/^$/d' aaa.txt # 再去除空行

[root@localhost ~]# cat aaa.txt # 结果就 ok 了。

hello -a

hello -b

hello -c

hello –d

8） [abc] 匹配字符集合内任意一个字符 [a-z]、[0-9]、[A-Z]，多选一，[0,9] 这个是 2 选 1.

[root@localhost ~]# grep -E '[0-9]' serv

[root@localhost ~]# grep -E '[0,9]' serv

9） [^abc] ^ 在中括号里面表示非，不包含 a 或 b 或 c。

[root@localhost ~]# grep -E '[^0-9]\*' serv # 不要数字

[root@localhost ~]# grep -E '[^a-z]' serv # 不要小写字母

[root@localhost ~]# grep -E '[^a-z0-9A-Z]' serv # 不要字母和数字

10） {n,m} 重复 n 到 m 次，前一个字符。

# 匹配一下 ip 地址：0.0.0.0 -- 255.255.255.255，这是 ip 地址的范围，个数上就是 7-15 个

[root@localhost ~]# grep -Eo '[0-9.]{7,15}' /var/log/secure # 但是如果有个大于 7 位的数字，也会被匹配上

[root@localhost ~]# grep -Eo '[0-9.]{7,15}' /var/log/secure|grep '.'

11） + 重复 1 次到多次，和不同

[root@localhost ~]# grep -E '19' serv

[root@localhost ~]# grep -E '19+' serv

12） ? 重复 0 次到多次，和 \* 号很像。

[root@localhost ~]# grep -Eo '[0-9]?' serv

13） \d 和 [0-9] 是一样的，但是它属于 perl 正则，grep 使用 perl 正则的时候需要用 - P 参数来指定

取ip的例子:

ip addr|grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'|sed -n '2p'

grep 'Failed password' secure |grep -Eo '[0-9]{1,3}\.[0-9.]+'

取 root 用户登录时间

lastlog|sed -n '/root/p'|grep -Eo '[0-9]{2}:[0-9]{2}:[0-9]{2}'

关于正则，我们简单学习这么多即可，不要深究昂，因为正则的东西太多了，后面遇到了复杂的我们在分析。

perl 正则：https://www.runoob.com/perl/perl-regular-expressions.html