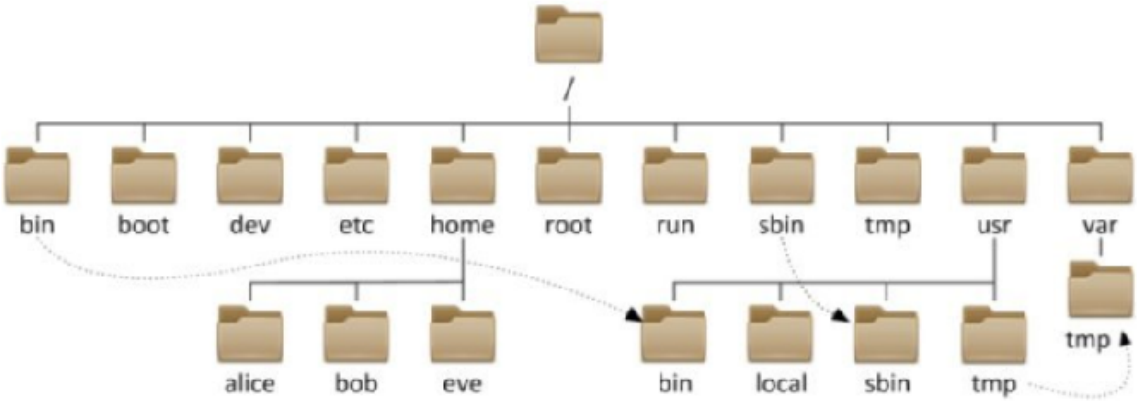
Linux基础命令

Linux的目录结构



- /，根目录是最顶级的目录了
- Linux只有一个顶级目录： /
- 路径描述的层次关系同样适用 / 来表示
- /home/itheima/a.txt，表示根目录下的home文件夹内有itheima文件夹，内有a.txt

ls命令

功能：列出文件夹信息

语法：`ls [-l -h -a] [参数]`

- 参数：被查看的文件夹，不提供参数，表示查看当前工作目录
- -l，以列表形式查看
- -h，配合-l，以更加人性化的方式显示文件大小
- -a，显示隐藏文件

隐藏文件、文件夹

在Linux中以.开头的，均是隐藏的。

默认不显示出来，需要-a选项才可查看到。

pwd命令

功能：展示当前工作目录

语法：pwd

cd命令

功能：切换工作目录

语法：cd [目标目录]

参数：目标目录，要切换去的地方，不提供默认切换到当前登录用户HOME目录

HOME目录

每一个用户在Linux系统中都有自己的专属工作目录，称之为HOME目录。

- 普通用户的HOME目录，默认在：/home/用户名
- root用户的HOME目录，在：/root

FinalShell登陆终端后，默认的工作目录就是用户的HOME目录

相对路径、绝对路径

- 相对路径，非/开头的称之为相对路径

  相对路径表示以当前目录作为起点，去描述路径，如test/a.txt，表示当前工作目录内的test文件夹内的a.txt文件

- 绝对路径，以 / 开头的称之为绝对路径

  绝对路径从根开始描述路径

特殊路径符

- `.`，表示当前，比如./a.txt，表示当前文件夹内的a.txt文件
- `..`，表示上级目录，比如 ../表示上级目录， ../../表示上级的上级目录
- `~`，表示用户的HOME目录，比如cd ~，即可切回用户HOME目录

mkdir命令

功能：创建文件夹

语法：`mkdir [-p] 参数`

- 参数：被创建文件夹的路径
- 选项：-p，可选，表示创建前置路径

touch命令

功能：创建文件

语法：`touch 参数`

- 参数：被创建的文件路径

cat命令

功能：查看文件内容

语法：`cat 参数`

- 参数：被查看的文件路径

more命令

功能：查看文件，可以支持翻页查看

语法：`more 参数`

- 参数：被查看的文件路径
- 在查看过程中：
  - 空格键翻页
  - q退出查看

cp命令

功能：复制文件、文件夹

语法：`cp [-r] 参数1 参数2`

- 参数1，被复制的
- 参数2，要复制去的地方
- 选项：-r，可选，复制文件夹使用

示例：

- `cp a.txt b.txt`，复制当前目录下a.txt为b.txt
- `cp a.txt test/`，复制当前目录a.txt到test文件夹内
- `cp -r test test2`，复制文件夹test到当前文件夹内为test2存在

mv命令

功能：移动文件、文件夹

语法：`mv 参数1 参数2`

- 参数1：被移动的
- 参数2：要移动去的地方，参数2如果不存在，则会进行改名

rm命令

功能：删除文件、文件夹

语法：`rm [-r -f] 参数...参数`

- 参数：支持多个，每一个表示被删除的，空格进行分隔
- 选项：-r，删除文件夹使用
- 选项：-f，强制删除，不会给出确认提示，一般root用户会用到

> rm命令很危险，一定要注意，特别是切换到root用户的时候。

which命令

功能：查看命令的程序本体文件路径

语法：`which 参数`

- 参数：被查看的命令

find命令

功能：搜索文件

语法1按文件名搜索：`find 路径 -name 参数`

- 路径，搜索的起始路径
- 参数，搜索的关键字，支持通配符*， 比如：*test表示搜索任意以test结尾的文件

grep命令

功能：过滤关键字

语法：`grep [-n] 关键字 文件路径`

- 选项-n，可选，表示在结果中显示匹配的行的行号。
- 参数，关键字，必填，表示过滤的关键字，带有空格或其它特殊符号，建议使用""将关键字包围起来
- 参数，文件路径，必填，表示要过滤内容的文件路径，可作为内容输入端口

> 参数文件路径，可以作为管道符的输入

wc命令

功能：统计

语法：`wc [-c -m -l -w] 文件路径`

- 选项，-c，统计bytes数量
- 选项，-m，统计字符数量
- 选项，-l，统计行数
- 选项，-w，统计单词数量
- 参数，文件路径，被统计的文件，可作为内容输入端口

> 参数文件路径，可作为管道符的输入

管道符|

写法：`|`

功能：将符号左边的结果，作为符号右边的输入

示例：

`cat a.txt | grep itheima`，将cat a.txt的结果，作为grep命令的输入，用来过滤`itheima`关键字

可以支持嵌套：

```
cat a.txt | grep itheima | grep itcast
```

echo命令

功能：输出内容

语法：`echo 参数`

- 参数：被输出的内容

`反引号

功能：被两个反引号包围的内容，会作为命令执行

示例：

- `echo `pwd``，会输出当前工作目录

tail命令

功能：查看文件尾部内容

语法：`tail [-f] 参数`

- 参数：被查看的文件
- 选项：`-f`，持续跟踪文件修改

head命令

功能：查看文件头部内容

语法：`head [-n] 参数`

- 参数：被查看的文件
- 选项：`-n`，查看的行数

重定向符

功能：将符号左边的结果，输出到右边指定的文件中去

- `>`，表示覆盖输出
- `>>`，表示追加输出

vi编辑器

命令模式快捷键

模式	命令	描述
命令模式	`i`	在当前光标位置进入`输入模式`
命令模式	`a`	在当前光标位置 之后 进入`输入模式`
命令模式	`I`	在当前行的开头，进入`输入模式`
命令模式	`A`	在当前行的结尾，进入`输入模式`
命令模式	`o`	在当前光标下一行进入`输入模式`
命令模式	`O`	在当前光标上一行进入`输入模式`
输入模式	`esc`	任何情况下输入`esc`都能回到命令模式

命令模式	键盘上、键盘k	向上移动光标
命令模式	键盘下、键盘j	向下移动光标
命令模式	键盘左、键盘h	向左移动光标
命令模式	键盘右、键盘l	向后移动光标
命令模式	0	移动光标到当前行的开头
命令模式	$	移动光标到当前行的结尾
命令模式	pageup(PgUp)	向上翻页
命令模式	pangdown(PgDn)	向下翻页
命令模式	/	进入搜索模式
命令模式	n	向下继续搜索
命令模式	N	向上继续搜索
命令模式	dd	删除光标所在行的内容
命令模式	ndd	n是数字，表示删除当前光标向下n行
命令模式	yy	复制当前行
命令模式	nyy	n是数字，复制当前行和下面的n行
命令模式	p	粘贴复制的内容
命令模式	u	撤销修改
命令模式	ctrl + r	反向撤销修改
命令模式	gg	跳到首行
命令模式	G	跳到行尾
命令模式	dG	从当前行开始，向下全部删除
命令模式	dgg	从当前行开始，向上全部删除
命令模式	d$	从当前光标开始，删除到本行的结尾
命令模式	d0	从当前光标开始，删除到本行的开头

底线命令快捷键

底线命令模式	`:wq`	保存并退出
底线命令模式	`:q`	仅退出
底线命令模式	`:q!`	强制退出
底线命令模式	`:w`	仅保存
底线命令模式	`:set nu`	显示行号
底线命令模式	`:set paste`	设置粘贴模式

命令的选项

我们学习的一系列Linux命令，它们所拥有的选项都是非常多的。

比如，简单的ls命令就有：-a -A -b -c -C -d -D -f -F -g -G -h -H -i -I -k -l -L -m -n -N -o -p -q -Q -r-R -s -S -t -T -u -U -v -w -x -X -1等选项，可以发现选项是极其多的。

课程中， 并不会将全部的选项都进行讲解，否则，一个ls命令就可能讲解2小时之久。

课程中，会对常见的选项进行讲解， 足够满足绝大多数的学习、工作场景。

查看命令的帮助

可以通过：命令 --help查看命令的帮助手册

```
[root@centos ~]# ls --help
用法：ls [选项]... [文件]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  不隐藏任何以．开始的项目
  -A, --almost-all           列出除．及．．以外的任何项目
      --author               与-l 同时使用时列出每个文件的作者
  -b, --escape               以八进制溢出序列表示不可打印的字符
      --block-size=SIZE      scale sizes by SIZE before printing them; e.g.,
                               '--block-size=M' prints sizes in units of
```

查看命令的详细手册

可以通过：man 命令查看某命令的详细手册

```
LS(1)                                User Commands                                LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information about the FILEs (the current directory by default).  Sort entries alphabetically
       if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .
```

Linux常用操作

软件安装

- CentOS系统使用：
  - yum [install remove search] [-y] 软件名称
    - install 安装
    - remove 卸载
    - search 搜索
    - -y，自动确认
- Ubuntu系统使用
  - apt [install remove search] [-y] 软件名称
    - install 安装

- remove 卸载
- search 搜索
- -y，自动确认

> yum 和 apt 均需要root权限

systemctl

功能：控制系统服务的启动关闭等

语法：`systemctl start | stop | restart | disable | enable | status 服务名`

- start，启动
- stop，停止
- status，查看状态
- disable，关闭开机自启
- enable，开启开机自启
- restart，重启

软链接

功能：创建文件、文件夹软链接（快捷方式）

语法：`ln -s 参数1 参数2`

- 参数1：被链接的
- 参数2：要链接去的地方（快捷方式的名称和存放位置）

日期

语法：`date [-d] [+格式化字符串]`

- -d 按照给定的字符串显示日期，一般用于日期计算
- 格式化字符串：通过特定的字符串标记，来控制显示的日期格式

- %Y　年%y　年份后两位数字（00..99）
- %m　月份（01..12）
- %d　日（01..31）
- %H　小时（00..23）
- %M　分钟（00..59）
- %S　秒（00..60）
- %s　自 1970-01-01 00:00:00 UTC 到现在的秒数

示例：

- 按照2022-01-01的格式显示日期

```
[itheima@bogon ~]$ date +%Y-%m-%d
2022-10-08
```

- 按照2022-01-01 10:00:00的格式显示日期

```
[itheima@bogon ~]$ date +%Y-%m-%d %H:%M:%S
date: 额外的操作数 "%H:%M:%S"
Try 'date --help' for more information.
[itheima@bogon ~]$ date "+%Y-%m-%d %H:%M:%S"
2022-10-08 00:48:12
```

- -d选项日期计算

```
date -d "+1 day" +%Y%m%d          # 显示后一天的日期
date -d "-1 day" +%Y%m%d          # 显示前一天的日期
date -d "-1 month" +%Y%m%d        # 显示上一月的日期
date -d "+1 month" +%Y%m%d        # 显示下一月的日期
date -d "-1 year" +%Y%m%d         # 显示前一年的日期
date -d "+1 year" +%Y%m%d         # 显示下一年的日期
```

  - 支持的时间标记为：

其中支持的时间标记为：

- year年
- month月
- day天
- hour小 时
- minute分 钟
- second秒

-d选项可以和 格式化字符串配合一起使用哦

时区

修改时区为中国时区

```
rm -f /etc/localtime
sudo ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

ntp

功能：同步时间

安装：`yum install -y ntp`

启动管理：`systemctl start | stop | restart | status | disable | enable ntpd`

手动校准时间：`ntpdate -u ntp.aliyun.com`

ip地址

格式：a.b.c.d

- abcd为0~255的数字

特殊IP：

- 127.0.0.1，表示本机
- 0.0.0.0
  - 可以表示本机
  - 也可以表示任意IP（看使用场景）

查看ip：`ifconfig`

主机名

功能：Linux系统的名称

查看：`hostname`

设置：`hostnamectl set-hostname 主机名`

配置VMware固定IP

1. 修改VMware网络，参阅PPT，图太多
2. 设置Linux内部固定IP

   修改文件：`/etc/sysconfig/network-scripts/ifcfg-ens33`

   示例文件内容：

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"          # 改为static，固定IP
DEFROUTE="yes"
```

```
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="1b0011cb-0d2e-4eaa-8a11-af7d50ebc876"
DEVICE="ens33"
ONBOOT="yes"
IPADDR="192.168.88.131"      # IP地址，自己设置，要匹配网络范围
NETMASK="255.255.255.0"      # 子网掩码，固定写法255.255.255.0
GATEWAY="192.168.88.2"       # 网关，要和VMware中配置的一致
DNS1="192.168.88.2"          # DNS1服务器，和网关一致即可
```

ps命令

功能：查看进程信息

语法：`ps -ef`，查看全部进程信息，可以搭配grep做过滤：`ps -ef | grep xxx`

kill命令

语法：`kill [-9] 进程ID`

选项：-9，表示强制关闭进程。不使用此选项会向进程发送信号要求其关闭，但是否关闭看进程自身的处理机制。

```
[itheima@centos ~]$ ps -ef|grep tail
itheima   46712   8021  0 17:14 pts/0    00:00:00 tail
itheima   47217  46758  0 17:14 pts/2    00:00:00 grep --color=auto tail
```

```
[itheima@centos ~]$ tail
已终止
```

```
[itheima@centos ~]$ ps -ef|grep tail
itheima    2339   8021  0 19:00 pts/0    00:00:00 tail
itheima    2659  46758  0 19:00 pts/2    00:00:00 grep --color=auto tail
[itheima@centos ~]$ kill -9 2339
```

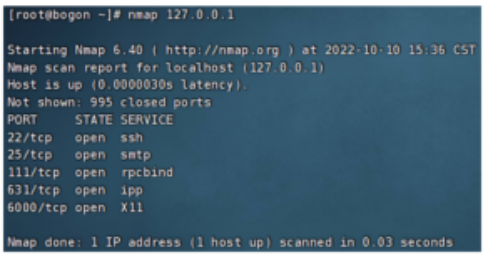
```
[itheima@centos ~]$ tail
已杀死
```

nmap命令

可以通过Linux命令去查看端口的占用情况

- 使用nmap命令，安装nmap：yum -y install nmap

语法：nmap 被查看的IP地址



可以看到，本机（127.0.0.1）上有5个端口现在被程序占用了。

其中：

- 22端口，一般是SSH服务使用，即FinalShell远程连接Linux所使用的端口

netstat命令

功能：查看端口占用

用法：`netstat -anp | grep xxx`

ping命令

测试网络是否联通

语法：`ping [-c num] 参数`

- 选项：-c，检查的次数，不使用-c选项，将无限次数持续检查
- 参数：ip或主机名，被检查的服务器的ip地址或主机名地址

示例：

- 检查到baidu.com是否联通

```
[itheima@bogon ~]$ ping baidu.com
PING baidu.com (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=128 time=8.64 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=128 time=7.65 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=128 time=7.70 ms
```

结果表示联通，延迟8ms左右

- 检查到39.156.66.10是否联通，并检查3次

```
[itheima@bogon ~]$ ping -c 3 39.156.66.10
PING 39.156.66.10 (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10: icmp_seq=1 ttl=128 time=8.20 ms
64 bytes from 39.156.66.10: icmp_seq=2 ttl=128 time=6.59 ms
64 bytes from 39.156.66.10: icmp_seq=3 ttl=128 time=8.18 ms

--- 39.156.66.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 6.598/7.663/8.208/0.756 ms
```

wget命令

wget是非交互式的文件下载器，可以在命令行内下载网络文件

语法：`wget [-b] url`

- 选项：-b，可选，后台下载，会将日志写入到当前工作目录的wget-log文件
- 参数：url，下载链接

示例：

- 下载apache-hadoop 3.3.0版本：wget http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz

- 在后台下载：wget -b http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
- 通过tail命令可以监控后台下载进度：tail -f wget-log

注意：无论下载是否完成，都会生成要下载的文件，如果下载未完成，请及时清理未完成的不可用文件。

curl命令

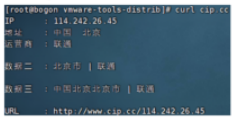
curl可以发送http网络请求，可用于：下载文件、获取信息等

语法：`curl [-O] url`

- 选项：-O，用于下载文件，当url是下载链接时，可以使用此选项保存文件
- 参数：url，要发起请求的网络地址

示例：

- 向cip.cc发起网络请求：curl cip.cc



- 向python.itheima.com发起网络请求：curl python.itheima.com

- 通过curl下载hadoop-3.3.0安装包：curl -O http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz

top命令

功能：查看主机运行状态

语法：`top`，查看基础信息

可用选项：

```
选项   功能
-p   只显示某个进程的信息
-d   设置刷新时间，默认是5s
-c   显示产生进程的完整命令，默认是进程名
-n   指定刷新次数，比如 top -n 3，刷新输出3次后退出
-b   以非交互非全屏模式运行，以批次的方式执行top，一般配合-n指定输出几次统计信息，将输出重定向到指定文件，比如 top -b -n 3 > /tmp/top.tmp
-i   不显示任何闲置 (idle) 或无用 (zombie) 的进程
-u   查找特定用户启动的进程
```

交互式模式中，可用快捷键：

```
按键    功能
h键    按下h键，会显示帮助画面
c键    按下c键，会显示产生进程的完整命令，等同于-c参数，再次按下c键，变为默认显示
f键    按下f键，可以选择需要展示的项目
M键    按下M键，根据驻留内存大小（RES）排序
P键    按下P键，根据CPU使用百分比大小进行排序
T键    按下T键，根据时间/累计时间进行排序
E键    按下E键，切换顶部内存显示单位
e键    按下e键，切换进程内存显示单位
l键    按下l键，切换显示平均负载和启动时间信息。
i键    按下i键，不显示闲置或无用的进程，等同于-i参数，再次按下，变为默认显示
t键    按下t键，切换显示CPU状态信息
m键    按下m键，切换显示内存信息
```

df命令

查看磁盘占用

语法: df [-h]

选项: -h, 以更加人性化的单位显示

```
文件系统           1K-块     已用      可用 已用% 挂载点
/dev/sda3      39517336 5720616 33796720   15% /
devtmpfs         482152       0   482152    0% /dev
tmpfs            497944       0   497944    0% /dev/shm
tmpfs            497944    8512   489432    2% /run
tmpfs            497944       0   497944    0% /sys/fs/cgroup
/dev/sda1        303780  149528   154252   50% /boot
tmpfs             99592      12    99580    1% /run/user/42
tmpfs             99592       0    99592    0% /run/user/1000
[itheima@centos ~]$ df -h
文件系统         容量  已用  可用 已用% 挂载点
/dev/sda3         38G  5.5G   33G   15% /
devtmpfs         471M     0  471M    0% /dev
tmpfs            487M     0  487M    0% /dev/shm
tmpfs            487M  8.4M  478M    2% /run
tmpfs            487M     0  487M    0% /sys/fs/cgroup
/dev/sda1        297M  147M  151M   50% /boot
tmpfs             98M   12K   98M    1% /run/user/42
tmpfs             98M     0   98M    0% /run/user/1000
```

iostat命令

查看CPU、磁盘的相关信息

语法: iostat [-x] [num1] [num2]

- 选项: -x, 显示更多信息
- num1: 数字, 刷新间隔, num2: 数字, 刷新几次

```
[itheima@centos ~]$ iostat
Linux 3.10.0-957.el7.x86_64 (centos)    2022年10月12日   _x86_64_        (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.42    0.00    2.05    0.02    0.00   97.52

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               5.89       159.98         9.37     349348      20455
```

CPU相关信息

tps: 该设备每秒的传输次数（Indicate the number of transfers per second that were issued to the device.）。

"一次传输"意思是"一次I/O请求"。多个逻辑请求可能会被合并为"一次I/O请求"。"一次传输"请求的大小是未知的。

- 使用iostat的-x选项, 可以显示更多信息

```
[itheima@centos ~]$ iostat -x
Linux 3.10.0-957.el7.x86_64 (centos)    2022年10月12日   _x86_64_        (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.40    0.00    1.98    0.01    0.00   97.60

Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sda               0.00     0.07    3.50    1.30   128.29     8.05    56.86     0.00    0.34    0.39    0.23   0.28   0.13
```

rrqm/s: 每秒这个设备相关的读取请求有多少被Merge了（当系统调用需要读取数据的时候, VFS将请求发到各个FS, 如果FS发现不同的读取请求读取的是相同Block的数据, FS会将这个请求合并Merge, 提高IO利用率, 避免重复调用）;
wrqm/s: 每秒这个设备相关的写入请求有多少被Merge了。
rsec/s: 每秒读取的扇区数; sectors
wsec/: 每秒写入的扇区数。
rKB/s: 每秒发送到设备的读取请求数
wKB/s: 每秒发送到设备的写入请求数
avgrq-sz 平均请求扇区的大小
avgqu-sz 平均请求队列的长度。毫无疑问, 队列长度越短越好。
await: 每一个IO请求的处理的平均时间（单位是微秒毫秒）。
svctm 表示平均每次设备I/O操作的服务时间（以毫秒为单位）
%util: 磁盘利用率

sar命令

查看网络统计

语法: sar -n DEV num1 num2

选项: -n, 查看网络, DEV表示查看网络接口

num1: 刷新间隔（不填就查看一次结束）, num2: 查看次数（不填无限次数）

```
[itheima@centos ~]$ sar -n DEV 3 2
Linux 3.10.0-957.el7.x86_64 (centos)    2022年10月12日   _x86_64_        (1 CPU)

15时16分04秒     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst/s
15时16分07秒        lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时16分07秒 virbr0-nic      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时16分07秒    virbr0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时16分07秒     ens33     13.85     29.05      1.18      6.58      0.00      0.00      0.00

15时16分07秒     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst/s
15时16分10秒        lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时16分10秒 virbr0-nic      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时16分10秒    virbr0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时16分10秒     ens33     11.11     21.89      0.96      6.22      0.00      0.00      0.00

平均时间:     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst/s
平均时间:        lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间: virbr0-nic      0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间:    virbr0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间:     ens33     12.48     25.46      1.07      6.40      0.00      0.00      0.00
```

信息解读:

- **IFACE 本地网卡接口的名称**
- **rxpck/s 每秒钟接受的数据包**
- **txpck/s 每秒钟发送的数据包**
- **rxKB/S 每秒钟接受的数据包大小，单位为KB**
- **txKB/S 每秒钟发送的数据包大小，单位为KB**
- **rxcmp/s 每秒钟接受的压缩数据包**
- **txcmp/s 每秒钟发送的压缩包**
- **rxmcst/s 每秒钟接收的多播数据包**

如图, 查看2次, 隔3秒刷新一次, 并最终汇总平均记录

环境变量

- 临时设置：export 变量名=变量值
- 永久设置：
  - 针对用户，设置用户HOME目录内：.bashrc文件
  - 针对全局，设置/etc/profile

PATH变量

记录了执行程序的搜索路径

可以将自定义路径加入PATH内，实现自定义命令在任意地方均可执行的效果

$符号

可以取出指定的环境变量的值

语法：$变量名

示例：

echo $PATH，输出PATH环境变量的值

echo ${PATH}ABC，输出PATH环境变量的值以及ABC

如果变量名和其它内容混淆在一起，可以使用${}

压缩解压

压缩

tar -zcvf 压缩包 被压缩1...被压缩2...被压缩N

- -z表示使用gzip，可以不写

`zip [-r] 参数1 参数2 参数N`

- -r，被压缩的包含文件夹的时候，需要使用-r选项，和rm、cp等命令的-r效果一致

示例：

- zip test.zip a.txt b.txt c.txt

将a.txt b.txt c.txt 压缩到test.zip文件内

- zip -r test.zip test itheima a.txt

将test、itheima两个文件夹和a.txt文件，压缩到test.zip文件内

解压

`tar -zxvf 被解压的文件 -C 要解压去的地方`

- -z表示使用gzip，可以省略
- -C，可以省略，指定要解压去的地方，不写解压到当前目录

`unzip [-d] 参数`

语法：`unzip [-d] 参数`

- -d，指定要解压去的位置，同tar的-C选项
- 参数，被解压的zip压缩包文件

示例：

- unzip test.zip，将test.zip解压到当前目录
- unzip test.zip -d /home/itheima，将test.zip解压到指定文件夹内（/home/itheima）

su命令

切换用户

语法：`su [-] [用户]`

- - 表示切换后加载环境变量，建议带上
- 用户可以省略，省略默认切换到root

sudo命令

- 可以让一条普通命令带有root权限，语法：`sudo 其它命令`
- 需要以root用户执行visudo命令，增加配置方可让普通用户有sudo命令的执行权限

比如：

```
itheima ALL=(ALL)       NOPASSWD: ALL
```

在visudo内配置如上内容，可以让itheima用户，无需密码直接使用`sudo`

chmod命令

修改文件、文件夹权限

语法：`chmod [-R] 权限 参数`

- 权限，要设置的权限，比如755，表示：`rwxr-xr-x`

- 0: 无任何权限，　即 ---
- 1: 仅有x权限，　即 --x
- 2: 仅有w权限　　即 -w-
- 3: 有w和x权限　即 -wx
- 4: 仅有r权限　　即 r--
- 5: 有r和x权限　　即 r-x
- 6: 有r和w权限　即 rw-
- 7: 有全部权限　　即 rwx

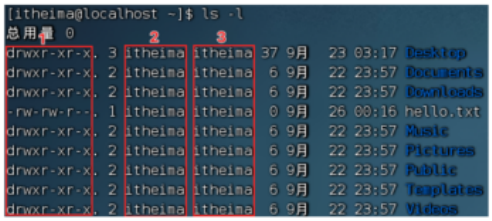
- 参数，被修改的文件、文件夹
- 选项-R，设置文件夹和其内部全部内容一样生效

chown命令

修改文件、文件夹所属用户、组

语法：chown [-R] [用户][:][用户组] 文件或文件夹

语法：chown [-R] [用户][:][用户组] 文件或文件夹



- 选项，-R，同chmod，对文件夹内全部内容应用相同规则
- 选项，用户，修改所属用户
- 选项，用户组，修改所属用户组
- :用于分隔用户和用户组

示例：

- chown root hello.txt，将hello.txt所属用户修改为root
- chown :root hello.txt，将hello.txt所属用户组修改为root
- chown root:itheima hello.txt，将hello.txt所属用户修改为root，用户组修改为itheima
- chown -R root test，将文件夹test的所属用户修改为root并对文件夹内全部内容应用同样规则

用户组管理

以下命令需root用户执行

- 创建用户组

groupadd 用户组名

- 删除用户组

groupdel 用户组名

用户管理

以下命令需root用户执行

- 创建用户

useradd [-g -d] 用户名

  - 选项：-g指定用户的组，不指定-g，会创建同名组并自动加入，指定-g需要组已经存在，如已存在同名组，必须使用-g
  - 选项：-d指定用户HOME路径，不指定，HOME目录默认在：/home/用户名

- 删除用户

userdel [-r] 用户名

  - 选项：-r，删除用户的HOME目录，不使用-r，删除用户时，HOME目录保留

- 查看用户所属组

id [用户名]

  - 参数：用户名，被查看的用户，如果不提供则查看自身

- 修改用户所属组

usermod -aG 用户组 用户名, 将指定用户加入指定用户组

genenv命令

- `getenv group`，查看系统全部的用户组

```
[itheima@localhost ~]$ getent group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
```

- `getenv passwd`，查看系统全部的用户

```
[itheima@localhost ~]$ getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```

env命令

查看系统全部的环境变量

语法：`env`