

一、linux常识

1、什么是linux

linux是一个服务器的操作系统，和windows，mac，包括安卓，IOS，都是一个操作系统

2、为什么要用linux

- 稳定
- 性能好
- linux免费
- 支持多用户操作

3、发行版本

- 商业：收费，红帽（主要通过提供服务支持，软件支持等收取费用）
- 社区：ubuntu,centOS,debian（免费的版本）

二、学习目的

- 通过查看服务器的日志来定位bug
- 检查系统的状态，开启或关闭一些服务/进程
- 性能测试的时候，监控系统资源使用情况
- 搭建测试环境

学习内容：

- linux的基本操作命令：创建文件，删除文件，文本编辑，用户和权限，打包安装命令
- linux系统命令：查看进程，查看任务，查看端口，cpu，内存，硬盘等情况

三、搭建学习环境

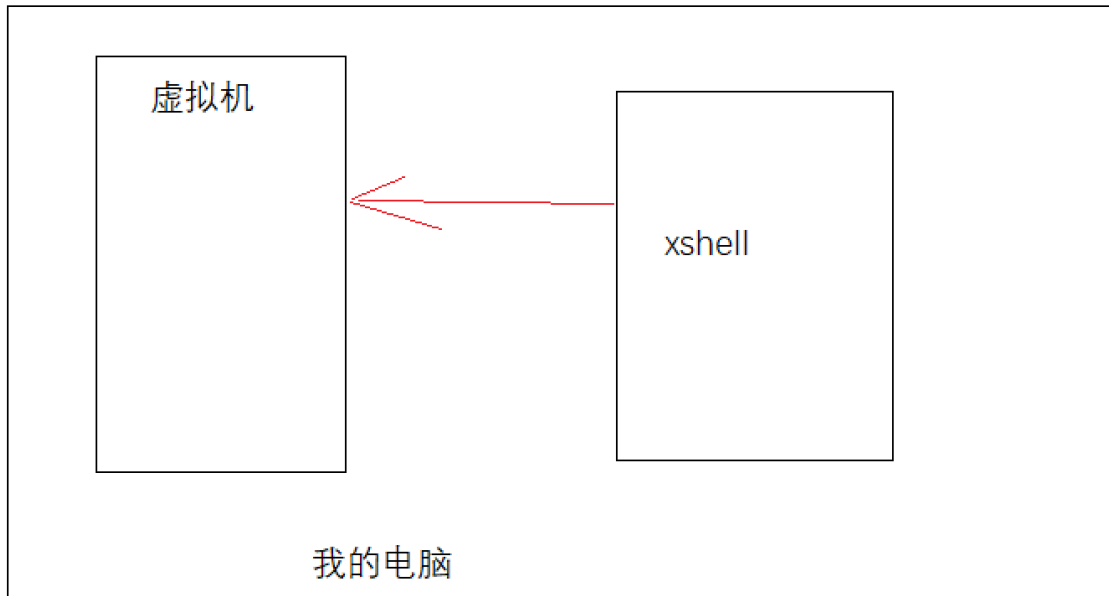
1、学习过程中

在自己电脑上创建一个虚拟机，通过linux的连接工具来进行连接，和操作系统命令

步骤：

- 打开虚拟机，登录桌面（用户名：linux 密码：cxy123456）
- 桌面右键选择‘打开终端’
- 输入ifconfig 获取到虚拟机的ip地址
- 使用xshell工具连接虚拟机
 - 填入虚拟机的ip地址，分别填入用户名和密码

- 学习电脑默认：用户名 linux 密码：123456



2、实际工作中

需要获取到公司服务器的IP地址，账号和密码，然后使用通过linux的连接工具来进行连接，和操作系统命令

四、linux操作命令

1、命令行展示

```
[root @localhost ~]#
```

- root 使用的用户的名称
- @localhost 服务器地址或名称
- ~ 宿主目录（路径）

2、linux的目录结构

/	根目录
home	用户目录
root	超级管理员的目录
usr	用于存放项目相关的程序或用户文件

3、linux的命令的语法

command **【-options】** **【parameter】**

command : 命令名称 一般是对应功能的单词或者单词的缩写

-options : 选项，可以用来对命令进行控制，实现不同的效果

parameter : 执行命令的参数，可以是一个或者多个。参数也可以是文件名，或者目录名

4、快捷键

tab键 : 补全，可以补全当前的文件或目录名，也可以补全路径

ctrl + c : 弃用当前的输入，或者终止当前的任务或进程

ctrl + insert : 鼠标选中后，按下后复制

shift + insert : 粘贴

ctrl + l : 清空屏幕

5、当前路径

pwd 展示当前处于的目录位置，从根目录出发

/root/a 第一个/代表根目录，第二个/是目录分隔符

绝对路径：从根目录一层一层切换的路径 举例：**cd /home/jc/vitest/**

相对路径：从当前路径出发，切换到对应的目录 举例：**cd ../../test**

6、查看历史命令

history

7、查看文件或目录

ls #展示当前目录下的子目录和子文件

ls -a #查看所有文件和目录，包括隐藏目录和隐藏文件（隐藏文件是以.开头的）

ls -l #查看当前目录下的所有文件的详情

ls -al #查看当前目录下的所有文件（包括隐藏文件）的详情

ls /root #查看根目录下的**root**目录中的子目录和子文件

8、切换目录

`cd` 参数 切换到当前文件的下级目录

`cd .` `#.`代表当前目录

`cd ..` `#`返回上层目录

`cd ~` `#`宿主（用户）目录，与登录的用户有关，如果是`root`，使用该命令会切换到`/root`，如果是使用普通用户执行这个命令则切换到对应的用户目录下

`cd -` `#`切换到上一次的目录

9、重启系统

`reboot`

10、创建目录

`mkdir` 目录名

`mkdir a b c` `#`在当前目录下创建三个目录

`mkdir -p bb5/bb6/bb7` `#-p`递归创建，在`bb5`下创建`bb6`，然后再`bb6`下创建`bb7`

`mkdir bb5/bb6/bb7/bb8` `#`在`bb5/bb6/bb7`目录下创建`bb8`

11、删除空目录

`rmdir` 空目录名 `#`删除空目录

`rmdir -p` `#`递归删除空目录 举例：`rmdir -p bb5/bb6/bb7/bb8/` 依次执行`rmdir bb8`，`rmdir bb7`，。。。

12、创建文件

`touch` 文件名

举例：`touch 2242.txt`

重定向输出：

`> :` `#`将内容覆盖写入到文件中

举例：`echo 今天是周四! > 2242.txt`

`>> :` `#`将内容追加写入到文件末尾

举例：`echo 今天是周五! >> 2242.txt`

`ls >> 2242.txt` `#`将`ls`的结果写入到`2242.txt`文件中

13、删除文件

```
rm 文件名
rm -r 文件或目录名 #强制删除文件或目录
rm -f 文件 #不询问删除
rm -rf 文件或目录名 #不询问，且强制删除文件或目录
```

课堂练习：

- 1、在用户目录下创建一个目录2242
- 2、在2242中创建一个三个文件，a.txt,b.txt,c.txt
- 3、往a.txt中写入：今天是周三，阴天！
- 4.删除b.txt
- 5.将ls的结果写入c.txt中

14、移动/改名

```
mv 目录/文件 新名称 #改名
举例：mv 2242_1/ 2242

mv 目录/文件 目录 #移动
举例：mv 2242 /home

mv 目录/文件 路径/新名字 #移动的同时改名
举例： mv /home/2242/ 2242_1 #将home目录下的2242移动到当前目录中并改名
      mv 2242.txt /home/22421.txt #将当前目录下的2242.txt移动到/home目录下并改名为22421.txt
```

15、复制文件

```
cp 文件名称 新文件名字
举例：cp a.txt a_copy.txt #复制文件，副本文件名称叫做a_copy.txt
举例：cp a.txt / #将文件复制到根目录下

cp -r 目录名 新目录名
举例：cp -r 2242 2242_1 #复制目录
```

16、文件编辑

进入文件：

```
vi 文件名  
vim 文件名
```

普通模式：

使用vi/vim命令进入文件后的第一种模式

```
yy  复制光标中的一行  
p   粘贴  
u   撤销改动  
x   剪贴光标位置的字符  
dd  删除光标位置的一行  
o/o  向下/向上插入一行空行，会自动进入到编辑模式  
gg  光标跳转到第一行  
G   光标跳转到最后一行  
  
num + 操作命令  
3yy  复制光标位置后3行  
2p   粘贴两次  
5dd  删除5行  
10↑  光标上移10行  
5→  光标右移5个字符
```

编辑模式：

按下i 或a 键进入编辑模式

在编辑模式下可以一个一个插入字符

退出编辑模式：

按下 esc键即可

底行模式

英文输入法中，按下shift + :

底行模式需要在普通模式下才可以按出来

底行模式用于保存退出文件

```
: q      退出文件
: q!     强制退出文件
: w      保存
: w!     强制保存
: wq     保存退出
: wq!    强制保存退出
: set nu  展示行号
: set nonu
: num    快速跳转到num行
```

17、查看文件

cat

顺序查看

```
cat 文件名    #将文件的内容打印到页面中
举例: cat a.txt
```

tac

倒序查看

```
tac 文件名    #将文件内容打印到页面中
举例: tac a.txt
```

head

查看头部内容

```
head -num 文件名    #展示头部的num行数据，不填选项则默认展示头部10行数据
举例: head -2 a.txt
```

tail

查看尾部内容

```
tail -num 文件名    #展示尾部的num行数据，不填选项则默认展示尾部10行数据

tail -f 文件名    #查看动态文件内容
在实际工作中：需要找到日志文件，使用 tail -f 文件名 去查看动态日志即可。
```

练习查看服务器日志

服务器:

47.107.114.9

root

Cxy123456

日志文件位置: /var/log/httpd/access_log

浏览器访问: http://47.107.114.9/upload/

使用xshell连接服务器 使用命令去查看/var/log/httpd/access_log

然后使用浏览器去访问http://47.107.114.9/upload/, 每一次范围都可以在access_log中查看到访问的日志

more

分页查看, 将内容打印到页面中

more 文件名

enter键 #下一行

空格键 #下一页

b键 #上一页

q键 #退出

less

进入到文件中查看文件内容, 也可以搜索需要的内容

less 文件名

enter键 #下一行

空格键 #下一页

b键 #上一页

q键 #退出

按下/后, 输入需要查找关键字回车

符合条件的数据会高亮起来

n键 #跳转到下一项匹配内容

系统出现bug或异常的时候, 常见的几个报错关键字

exeption 异常

failed 失败

error 错误

在实际工作中, 发现了bug, 如果在相关日志中没有找到这三个关键字, 可以问一下开写的报错关键字是什么。

18、通配符

可以代替其他的符号、

可以通配多个任意的字符

举例：rm -rf a*

?

可以通配单个字符

举例：rm -rf 1?.txt #删除1开头，并且文件名称为2个字符的，并且结尾为.txt的文件

19、用户相关命令

切换用户

普通用户切换到其他用户是需要输入密码的
root用户切换到其他普通用户不需要输入密码

命令

su **#切换到超级管理员**
su 用户名 切换到对应的用户
举例： **su** linux

注意：输入密码是界面是没有密码位数展示

创建用户

useradd 用户名

举例：

useradd g2242

每创建一个新用户，会在home目录下自动创建一个用户目录，目录名称和用户名相同

修改密码

passwd 用户名

举例：

passwd g2242

需要输入两次相同的密码

可以忽略提示

删除用户

```
userdel 用户名  
举例：  
userdel g2242
```

20、管道符和grep

管道符： |

使用管道符可以再次执行另外一条命令，第一条命令的执行结果可以作为第二条命令执行的条件

举例：
`ls | grep yum` #查找当前目录下的所有文件和目录中，带有yum关键字的文件或目录

grep

```
grep 内容a 文件b #在文件b中查找内容a,并且返回内容a所在的一整行  
举例：  
grep 周四 1.txt #返回周四所在的一整行
```

21、文件权限

文件详细信息：

`d` `rwX` `r-X` `r-X` `3` `root` `root` `21` `7月 7 2021` `pkcs11`

`d` :

- : 表这是一个文件
- d : 表示这个是一个目录
- l : link 连接文件。类似于快捷方式
- b : 表示存储 (块) 设备文件
- c : 字符设别文件

`rwX` : (r:可读, w:可写, x:可执行, - : 没有该权限)

用户权限 (u)

在 `linux` 中, 每个文件都有所属用户。所属用户对文件一般是拥有 `rwX` 权限。
当然超级管理员是拥有改动所有文件的权限

`r-X` :

组权限 (g)

该文件属于那个用户, 然后用户属于那个组, 同一个组内的用户对这个文件拥有对应的权限。一个用户可以属于多个组, 但是只有一个主组。这个权限主要是主组

`r-X` :

其他用户权限 (o)

除了用户和组用户的其他用户的权限

`3`

1 代表是文件,

2 以及 2 以上的, 表示目录内有多少个子目录

`Root` :

这个字段表示该文件或目录属于那个用户

`Root` :

文件或目录的所属用户所在的组

`21` :

文件的大小 kb

`7月 7 2021` :

最后的修改时间, 2021 年 7 月 7 日

`pkcs11` :

文件的名称

修改文件权限

`chmod` 权限 文件/目录 #修改权限需要用root用户去操作

方法1:

`chmod g+w 2242` #给组加上可写权限
`chmod o-r 2242` #给其他用户减去可读权限

方法2:

`chmod g=rwx,o=rwx 2242`

方式3:

可读: 4

可写: 2

可执行: 1

`chmod 762 2242` #用户: `rwX` , 组: `rw-` , 其他: `-w-`

`-R` :

给该目录下的所有子目录和子文件都改成同样的权限

```
chmod -R 762 2242 #注意：大写的R
```

22、查找

find 需要查找的路径 选项 参数

注意：

如果查找使用到通配符，有时候会产生找不到的情况，此时在通配符前加上一个转义符\

举例：

```
find . -name a\*.txt
```

选项：

-name 按照名称查找：

```
find / -name a.log #在根目录下查找符合a.log名的文件
```

-size 按照大小查找：

c：字节

k：千字节

M（大写）：兆字节

G：千兆字节

+numG ：大于numG的文件

-numG ：小于numG的文件

numG ：等于numG的文件

举例：

```
find . -size -6M #在当前目录下查找小于6M的文件
```

按照多条个条件仅限查找

查找小于2k的，且结尾为.log的文件

```
find / -size -2k -name *.log
```

23、打包和解压

打包：将多个文件/目录添加到一个打包文件你我

压缩：将文件压缩，压缩后的文件会变小并且不可以编辑，传输速度也会加快

大部分情况下打包和压缩进行了整合。包的后缀名决定的解压和解包的方式。

tar

tar 选项 包名.tar (包名.gz.tar) 需要打包的文件/目录

选项:

这个两个选项只能二选一

-c : 建立打包档案

-x : 解包解压

下面的选项是必填:

-f : 使用档案名字, 这个选项必须是在最后, 后面只能接档案名

一下选项是选填:

-z : 有gzip属性, 压缩时, 先打包成tar包, 然后再使用gzip进行压缩, 解包时反过来

-v : 显示所有打包或压缩的过程

举例:

```
tar -cvf aab.tar aa #打包当前目录下的aa
```

```
tar -czvf bb.tar.gz bb #压缩和打包带有gzip属性
```

解包:

```
tar -xvf aab.tar
```

如果需要指定路径解包, 加上-C (大写)

```
tar -xvf aab.tar -C / #将包解到/中
```

```
tar -xzvf bb.tar.gz #解压带有gzip属性的文件
```

zip

zip 选项 包名.zip 需要压缩的文件/目录

举例:

```
zip bb.zip bb.log #
```

选项:

-r 压缩目录内的子目录和子文件

举例:

```
zip -r bb.zip bb/
```

解压:

```
unzip 包名
```

举例:

```
unzip 2242.zip
```

gzip

只能压缩文件，压缩/解压会自动删除原文件

```
gzip 文件      #这里不需要手动输入生成的包名，会自动生成：xx.gz
gzip 1.txt
```

解压：

```
gzip -d 包名.gz
```

举例：

```
gzip -d 1.txt.gz
```

24、监控系统资源

vmstat

```
procs -----memory----- ---swap-- -----io---- -system-- -----cpu----- r b swpd free buff cache si
so bi bo in cs us sy id wa st 3 0 520 74280 0 262196 0 0 23 6 57 148 0 0 100 0 0
```

procs

r : 等待执行的任务数

b : 等待IO的进程数

memory

swpd : 正在使用虚拟内存的大小单位K

free : 空闲内存

buff : 缓存

cache : 缓存

swap

si : 每秒从交换区写入内存的大小 (kb/s)

so : 每秒从内存写到交换区的大小 (kb/s)

io

bi : 每秒读取的块数

bo : 每秒写入的块数

cpu

us : 用户进程执行消耗cpu的时间

top

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	125724	4020	2412	S	0.0	0.4	0:05.32	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd

pid : 进程号

user: 用户名

S : 进程的状态, s=sleep 睡眠, R =run 运行, T=stop 暂停

%CPU : 占用cpu的百分比

%MEM : 占用内存

25、查看进程

ps -ef

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	09:27	?	00:00:05	/usr/lib/systemd/systemd --switched-root --system --deserialize 22
root	2	0	0	09:27	?	00:00:00	[kthreadd]
root	4	2	0	09:27	?	00:00:00	[kworker/0:0H]
root	6	2	0	09:27	?	00:00:00	[ksoftirqd/0]

UID : 用户名

pid :进程号

PPID : 父进程

C : cpu的占比

CMD : 命令的名称

26、关闭进程

kill pid

这里需要找到对应进程的pid

```
kill 21851
```

27、查看物理内存

```
free      #默认展示kb
           total      used      free      shared  buff/cache  available
Mem:      939452      603772      64016      13624      271664      138260
Swap:     2097148        520     2096628
```

```
free -m    #以兆为单位展示内存情况
```

```
free -s 3  #按照时间刷新
```

28、查看文件占用磁盘空间

```
du      #当前目录下的文件占用磁盘空间的大小
```

```
du -h    #更加易读
```

29、查看磁盘空间

```
df
```

```
df -h    #方便阅读
```