

# 计算机杂记

李佳

xmujiali@163.com

2021 年 4 月 15 日

# 目录

<b>第一章 Linux 配置</b>	<b>3</b>
1.1 修改启动级别 . . . . .	3
1.2 添加用户和用户组 . . . . .	3
<b>第二章 Linux 网络配置</b>	<b>5</b>
2.1 SSH 自动掉线的解决办法 . . . . .	5
2.2 通过 sshfs 将远程服务器的文件挂载到本地 . . . . .	5
<b>第三章 Linux 环境下编译、开发环境配置</b>	<b>7</b>
<b>第四章 CentOS 相关</b>	<b>8</b>
4.1 CentOS 上网 . . . . .	8
4.2 CentOS 拨号上网 . . . . .	13
4.3 yum . . . . .	13
4.4 安装 32 位的 lib . . . . .	13
4.5 NSF 服务 . . . . .	13
4.6 远程安装系统 . . . . .	13
<b>第五章 Ubuntu 相关</b>	<b>14</b>
<b>第六章 on L<sup>A</sup>T<sub>E</sub>X</b>	<b>15</b>
6.1 L <sup>A</sup> T <sub>E</sub> X 基本命令 . . . . .	15
6.1.1 使用 L <sup>A</sup> T <sub>E</sub> X 抄录代码 . . . . .	15
6.2 字体 . . . . .	16
6.3 版面设计 . . . . .	16
6.4 标题 . . . . .	16

目录	2
6.5 表格	16
6.6 列表	16
6.7 数学式	16
6.8 插图	16
6.9 正文工具	16
6.10 浮动体	16
6.11 幻灯片: beamer	16
6.11.1 tableofcontents	16
第七章 on Python	17
第八章 on C	18
第九章 on Haskell	19

# 第一章 Linux 配置

## 1.1 修改启动级别

该方案是过时的!!!

安装 Linux 时如果选择安装了图形界面，系统会启动后会默认启动 Xwindows，如不希望如此可做如下操作：

以 root 用户修改/etc/inittab，找到 id:x:initdefault: 一行，x=3 为文本方式 x=5 为 Xwindow 方式，重启机器即可生效

## 1.2 添加用户和用户组

- 建用户  
adduser phpq //新建 phpq 用户  
passwd phpq //给 phpq 用户设置密码
- 建工作组  
groupadd test //新建 test 工作组
- 新建用户同时增加工作组  
useradd -g test phpq //新建 phpq 用户并增加到 test 工作组  
参数说明：-g 所属组 -d 家目录 -s 所用的 SHELL 类型
- 给已有的用户增加工作组  
usermod -G groupname username  
或  
gpasswd -a user groupadd

- 临时关闭

在/etc/shadow 文件中属于该用户的行的第二个字段（密码）前面加上 \* 就可以了。想恢复该用户，去掉 \* 即可。或者使用如下命令关闭用户账号：

```
passwd peter -l
```

重新释放：

```
passwd peter -u
```

- 永久性删除用户账号

```
userdel peter
```

```
groupdel peter
```

usermod -G peter peter // 强制删除该用户的主目录和主目录下的所有文件和子目录

- 从组中删除用户

编辑/etc/group 找到 GROUP1 那一行，删除 A。或者用命令：

```
gpasswd -d A GROUP
```

- 显示用户信息

```
id user
```

```
cat /etc/passwd
```

## 第二章 Linux 网络配置

### 2.1 SSH 自动掉线的解决办法

SSH 连接总是隔一段时间没有输入时就断开，解决办法如下：

- 服务端配置

```
sudo vi /etc/ssh/sshd_config
ClientAliveInterval 60 # 服务端主动向客户端请求响应的间隔
ClientAliveCountMax 10 # 服务器发出请求后客户端没有响应的次数达到一定
sudo restart ssh
```

- 客户端配置

```
sudo vi /etc/ssh/ssh_config # 或 vi ~/.ssh/config
TCPKeepAlive=yes
ServerAliveInterval 60 # 客户端主动向服务端请求响应的间隔
```

- ssh 时添加选项：

```
ssh -i <key-file> -o StrictHostKeyChecking=no -o TCPKeepAlive=yes -o
```

上面方式任选一种，推荐客户端配置方式。

### 2.2 通过 sshfs 将远程服务器的文件挂载到本地

假定用户 jli 可以通过 SSH 连接到远程服务器。为避免反复使用 scp 交换文件，可以直接把远程服务器上的目录挂载到本地。具体方案如下：

- 安装 sshfs

```
sudo apt install sshfs
# 要检查 fuse 组是否存在
cat /etc/group | grep 'fuse'
# 如果该组不存在，须创建 fuse 组
sudo groupadd fuse
# 如果用户 jli 不在 fuse 组中，将其加入
sudo usermod -a -G fuse jli
# 搞完后最好重新登录下
```

- 安装 sshfs

```
# sshfs 语法如下：
sshfs [user@]host:[directory] mountpoint [options]
# 我要把服务器 chem174 上的 /home/jli 目录
# 挂载到本地的 /home/jli/remote/174 下：
sshfs jli@chem174:/home/ /home/jli/remote/174
```

搞定！就像操作本地文件操作远程文件吧！

## 第三章 Linux 环境下编译、开发 环境配置



## 第四章 CentOS 相关

### 4.1 CentOS 上网

在 RHEL 或者 CentOS 等 Redhat 系的 Linux 系统里，跟网络有关的主要设置文件如下：

/etc/host.conf	# 配置域名服务客户端的控制文件
/etc/hosts	# 完成主机名映射为IP地址的功能
/etc/resolv.conf	# 域名服务客户端的配置文件,用于指定域名服务器的位置
/etc/sysconfig/network	# 包含了主机最基本的网络信息,用于系统启动
/etc/sysconfig/network-script/	#系统启动时初始化网络的一些信息
/etc/xinetd.conf	# 定义了由超级进程xinetd启动的网络服务
/etc/networks	# 完成域名与网络地址的映射
/etc/protocols	# 设定了主机使用的协议以及各个协议的协议号
/etc/services	# 设定主机的不同端口的网络服务

开启, 关闭 eth0 网卡, 注意网卡的名字可能不一样

```
ifup eth0
```

```
ifdown eth0
```

CentOS 查询 IP 地址，输入下面的命令：

```
ip -4 add
```

CentOS 打开/关闭网络连接，输入下面的命令：

```
ifup ens192 ifdw ens192
```

CentOS 重启网络服务，输入下面的命令：

```
systemctl retart network.service
```

或

```
systemctl restart network
```

CentOS 启动网络服务，输入下面的命令：

```
systemctl start network.service
```

或

```
systemctl start network
```

CentOS 停止网络服务，输入命令：

```
systemctl stop network.service
```

或

```
systemctl stop network
```

下面逐个文件说明配置方法：

- /etc/host.conf

文件的默认信息如下：

```
multi on          # 允许主机拥有多个IP地址
order hosts,bind  # 主机名解析顺序，即本地解析,DNS域名解析的顺序
```

这个文件一般不需要我们修改，默认的解析顺序是本地解析,DNS 服务器解析，也就是说在本系统里对于一个主机名首先进行本地解析，如果本地解析没有，然后进行 DNS 服务器解析。

- /etc/hosts

文件默认的内容大概如下：

```
127.0.0.1 butbueatiful localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
```

可见，默认的情况是本机 ip 和本机一些主机名的对应关系，第一行是 ipv4 信息，第二行是 ipv6 信息，如果用不上 ipv6 本机解析，一般把该行注释掉。第一行的解析效果是，butbueatiful localhost.localdomain localhost 都会被解析成 127.0.0.1，我们可以用 ping 试试。

```
[root@butbueatiful ~]# ping -c 3 butbueatiful
PING butbueatiful (127.0.0.1) 56(84) bytes of data.
64 bytes from butbueatiful (127.0.0.1): icmp_seq=1 ttl=64 time=0.061 ms
64 bytes from butbueatiful (127.0.0.1): icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from butbueatiful (127.0.0.1): icmp_seq=3 ttl=64 time=0.051 ms
```

```

--- butbueatiful ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.051/0.054/0.061/0.009 ms

[root@butbueatiful ~]# ping -c 3 localhost.localdomain
PING butbueatiful (127.0.0.1) 56(84) bytes of data.
64 bytes from butbueatiful (127.0.0.1): icmp_seq=1 ttl=64 time=0.055 ms
64 bytes from butbueatiful (127.0.0.1): icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from butbueatiful (127.0.0.1): icmp_seq=3 ttl=64 time=0.050 ms

--- butbueatiful ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.035/0.046/0.055/0.011 ms

```

看到上面的结果，你可能会问为什么 ping localhost.localdomain 的时候，下面显示的是却是 butbueatiful，这是因为第一个主机名 butbueatiful 后面的那些主机名其实都是 butbueatiful 的主机别名。

如果我们要追加新的本地解析，比如我们希望在我们的机器里把 yyy.com 和 www.yyy.com 都解析成 192.168.0.100，那么就追加如下一句即可：  
192.168.0.100 yyy.com www.yyy.com

同样，在这里，www.yyy.com 是 yyy.com 的主机别名。

如果你仔细一想，会发现，其实这个文件是很危险的，如果有人恶意修改了你这个文件，比如把淘宝的网站域名解析到了他的钓鱼网站，那你就中招了。

- /etc/resolv.conf

指定域名解析的 DNS 服务器 IP 等信息，配置参数一般接触到的有 4 个：

nameserver	# 指定DNS服务器的IP地址
domain	# 定义本地域名信息
search	# 定义域名的搜索列表
sortlist	# 对gethostbyname返回的地址进行排序

但是最常用的配置参数是 `nameserver`，其他的可以不设置，这个参数指定了 DNS 服务器的 IP 地址，如果设置不正确，就无法进行正常的域名解析。

一般来说，推荐设置 2 个 DNS 服务器，比如我们用 google 的免费 DNS 服务器，那么该文件的设置内容如下：

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

同样，这个文件也是危险的，如果被人恶意改成了他自己的 DNS 服务器，他就可以为所欲为的控制你通过域名访问的每个目的地了，这就是常说的 DNS 劫持。

- `/etc/sysconfig/network`

典型的配置如下：

```
NETWORKING=yes
```

```
NETWORKING_IPV6=no
```

```
HOSTNAME=butbueatiful
```

```
GATEWAY=192.168.0.1
```

参数简要解释：

```
NETWORK          # 设置网络是否有效，yes有效，no无效
```

```
NETWORKING_IPV6  # 设置ipv6网络是否有效，yes有效，no无效
```

```
HOSTNAME         # 设置服务器的主机名，最好和/etc/hosts里设置一样，否则在使用
```

```
GATEWAY          # 指定默认网关IP
```

- `ifcfg-ethX`

设置对应网口的 IP 等信息，比如第一个网口，那么就是 `/etc/sysconfig/network-scripts/ifcfg-eth0`，配置例子：

参数简要解释：\\

```
TYPE=Ethernet      #类型=以太网\\
```

```
PROXY_METHOD=none  #代理模式\\
```

```
BROWSER_ONLY=no\\
```

```
BOOTPROTO=none     #开机协议\\
```

```

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=ens32\\
UUID=0e6d72a7-8a6c-43ac-aef2-25d165562fd0\\
DEVICE=ens32          #设备\\
ONBOOT=yes            #启动或者重启网络时，是否启动该设备，yes是启动，no是不启动\\
IPADDR=192.168.1.15    #IP地址\\
PREFIX=24              #子网掩码\\
GATEWAY=192.168.1.1    #网关\\
DNS1=192.168.1.2       #DNS服务器地址\\
IPV6_PRIVACY=no        #IPV6协议\\

```

说明如下:\\

```

DEVICE      设备名，不要自己乱改，和文件ifcfg-ethX里的ethX要一致\\
BROADCAST   广播地址\\
HWADDR      物理地址，这个你不要乱改\\
IPADDR      IP地址\\
NETMASK     子网掩码\\
ONBOOT      启动或者重启网络时，是否启动该设备，yes是启动，no是不启动\\
BOOTPROTO   开机协议，最常见的三个参数如下：\\
static(静态IP)\\
none(不指定,设置固定ip的情况，这个也行，但是如果要设定多网口绑定bond的时候，必须\\
dhcp(动态获得IP相关信息)

```

- route-ethX

比如第一个网口 eth0 的路由信息，那么就是/etc/sysconfig/network-scripts/route-eth0:

比如我们现在有这样一个需求，通过 eth0 去网络 172.17.27.0/24 不走默认路由，需要走 192.168.0.254，那么我们第一反应，肯定是用 route

命令追加路由信息:

```
[root@butbueatiful ~]# route add -net 172.17.27.0 netmask 255.255.255.0 gw 192.168.0.254  
可是, 你没意识到的是, 这样只是动态追加的而已, 重启网络后, 路由信息就消失了, 所以  
[root@butbueatiful ~]# vi /etc/sysconfig/network-scripts/route-eth0
```

#追加

```
172.17.27.0/24via 192.168.0.254
```

这下即使重启网络, 重启系统, 该路由也会自动加载, 当然了, 如果你没有这样的需要, 那

## 4.2 CentOS 拨号上网

### 4.3 yum

### 4.4 安装 32 位的 lib

### 4.5 NSF 服务

### 4.6 远程安装系统

## 第五章 Ubuntu 相关

## 第六章 on L<sup>A</sup>T<sub>E</sub>X

### 6.1 L<sup>A</sup>T<sub>E</sub>X 基本命令

#### 6.1.1 使用 L<sup>A</sup>T<sub>E</sub>X 抄录代码

抄录代码可以使用 `lstlisting` 环境，抄录 C 代码的办法如下：

```
\begin{lstlisting}[language=C]
int main(int argc, char ** argv)
{
printf("Hello world!\n");
return 0;
}
\end{lstlisting}
```

其显示效果如下：

```
int main(int argc, char ** argv)
{
printf("Hello□world!\n");
return 0;
}
```

抄录 Python 如下：

```
\begin{lstlisting}[language=Python]
from pyx import *
g = graph.graphxy(width=8)
g.plot(graph.data.function("y(x)=sin(x)/x", min=-15, max=15))
```



```
g.writePDFfile("function")
print r'\includegraphics{function}'
\end{lstlisting}
```

其显示效果如下:

```
from pyx import *
g = graph.graphxy(width=8)
g.plot(graph.data.function("y(x)=sin(x)/x", min=-15, max=15))
g.writePDFfile("function")
print r'\includegraphics{function}'
```

## 6.2 字体

## 6.3 版面设计

## 6.4 标题

## 6.5 表格

## 6.6 列表

## 6.7 数学式

## 6.8 插图

## 6.9 正文工具

## 6.10 浮动体

## 6.11 幻灯片: beamer

### 6.11.1 tableofcontents

## 第七章 on Python

## 第八章 on C

## 第九章 on Haskell