

## Contents

### 目录

1. 常用的网络工具: .....	1
1.1 IP 命令的使用.....	1
1.2 ss 与 netstat .....	1
1.3 ping 命令.....	2
2. 源码编译安装: .....	2
2.1 源码编译安装的优势 .....	2
2.2 源码编译安装的基本实现过程 .....	3
2.2 源码编译安装的步骤 .....	3
3. 自定义 yum 仓库: .....	5
3.1 yum 仓库的定义.....	5
3.2 自定义 yum 仓库的配置.....	5

## 1. 常用的网络工具:

### 1.1 IP 命令的使用

查看 IP 地址

```
[root@svr7 ~]# ip address show
```

添加 IP 地址

```
[root@svr7 ~]# ip address add 192.168.8.1/24 dev ens33
```

#会有8.1的IP, 但是是临时的, 只能通过ip address show查看, ifconfig查看不到

```
[root@svr7 ~]# ip addr show
```

```
[root@svr7 ~]# ping 192.168.8.1 #可以ping通
```

如果想要每次开机生效, 可以写/etc/rc.d/rc.local 文件, 给个执行的权限

```
[root@svr7 ~]# vim /etc/rc.d/rc.local ip address add 192.168.8.1/24 dev ens33
```

添加路由

```
[root@svr7 ~]# ip route add 10.0.0.0/24 via 192.168.8.100 dev eth0 #via是下一跳
```

```
[root@svr7 ~]# ip route show #查看路由表
```

删除路由

```
[root@svr7 ~]# ip route del 10.0.0.0/24
```

```
[root@svr7 ~]# ip route show
```

### 1.2 ss 与 netstat

-a 显示所有端口的信息

-n 以数字格式显示端口号

-t 显示 TCP 连接的端口

-u 显示 UDP 连接的端口

-l 显示服务正在监听的端口信息

-p 显示监听端口的服务名称是什么（也就是程序名称）

```
[root@svr7 ~]# netstat -anptu
[root@svr7 ~]# ss -anptu
[root@svr7 ~]# netstat -anptu | grep :22
[root@svr7 ~]# ss -anptu | grep :22
```

## 1.3 ping 命令

常用选项：-c 测试包个数

```
[root@svr7 ~]# ping -c 2 192.168.4.207 #ping2次结束
```

## 2. 源码编译安装：

### 2.1 源码编译安装的优势

RPM 软件包：rpm -ivh yum -y install

源码包---->开发工具----->可以执行的程序----->运行安装

优势：

- 1) 灵活度较高，自定义较高，可以指定安装位置
- 2) 获得软件的最新版,及时修复 bug
- 3) 软件功能可按需选择/定制,有更多软件可供选择
- 4) 源码包适用各种平台

## 2.2 源码编译安装的基本实现过程



## 2.2 源码编译安装的步骤

步骤一: 安装开发工具 **gcc** 与 **make**

```
[root@svr7 ~]# yum -y install gcc make #可能会报错, 报错参考下面步骤修复
[root@svr7 ~]# mount /dev/cdrom /dvd/
[root@svr7 ~]# vim /etc/yum.repos.d/dvd.repo
[dvd]
name=CentOS7.5
baseurl=file:///dvd
enabled=1
gpgcheck=0
[root@svr7 ~]# yum -y install gcc make
```

查看是否安装成功

```
[root@svr7 ~]# rpm -q gcc make
gcc-4.8.5-28.el7.x86_64
```

```
make-3.82-23.el7.x86_64
```

步骤二: tar 解包,释放源代码至指定目录

```
[root@svr7 ~]# tar -xf tools.tar.gz -C /opt
[root@svr7 ~]# ls /opt/tools/
[root@svr7 ~]# tar -xf /opt/tools/inotify-tools-3.13.tar.gz -C /opt/
[root@svr7 ~]# ls /opt/
[root@svr7 ~]# cd /opt/inotify-tools-3.13/
```

步骤三: ./configure 配置,指定安装目录/功能模块等选项, 检测系统是否安装了 gcc 软件

--prefix=安装路径

```
[root@svr7 inotify-tools-3.13]# ./configure --prefix=/opt/myrpm #指定安装目录, 目录可以不存在
```

常见的错误提示, 没有安装 gcc

```
checking for gcc... no
checking for cc... no
checking for cl.exe... no
configure: error: no acceptable C compiler found in $PATH
See `config.log' for more details.
```

步骤四: make 编译,生成可执行的二进制程序文件

```
[root@svr7 inotify-tools-3.13]# make
```

步骤五: make install 安装,将编译好的文件复制到安装目录

```
[root@svr7 inotify-tools-3.13]# make install
[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/
[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/bin/
```

## 3. 自定义 yum 仓库:

### 3.1 yum 仓库的定义

自定义 Yum 仓库: 将自己准备好的软件包,进行 Yum 机制的管理

### 3.2 自定义 yum 仓库的配置

步骤一: 准备软件包, 这个已经提前准备好, 在 tools 压缩包里

```
[root@svr7 ~]# ls /opt/tools/other/ #注: tools包释放到哪个目录就写哪里
```

步骤二: 通过命令生成仓库数据文件

```
[root@svr7 ~]# createrepo /opt/tools/other/  
[root@svr7 ~]# ls /opt/tools/other/      #再次查看有repodata目录
```

步骤三: 书写客户端配置文件

```
[root@svr7 ~]# vim /etc/yum.repos.d/dvd.repo  
[dvd]  
name=CentOS7.5  
baseurl=file:///dvd  
enabled=1  
gpgcheck=0  
[nsd]  
name=myrpm  
baseurl=file:///opt/tools/other  
enabled=1  
gpgcheck=0  
[root@svr7 ~]# yum clean all  
[root@svr7 ~]# yum repolist
```

步骤四: 安装软件包

```
[root@svr7 ~]# yum -y install sl
[root@svr7 ~]# sl
[root@svr7 /]# yum -y install cmatrix
[root@svr7 /]# cmatrix
[root@svr7 /]# yum -y install oneko
[root@svr7 /]# oneko
[root@svr7 /]# oneko & #将程序放入后台运行
[root@svr7 /]# killall oneko #杀死所有oneko程序
[root@svr7 /]# yum -y install boxes
[root@svr7 /]# echo "I am dog" | boxes -d dog -a c
```