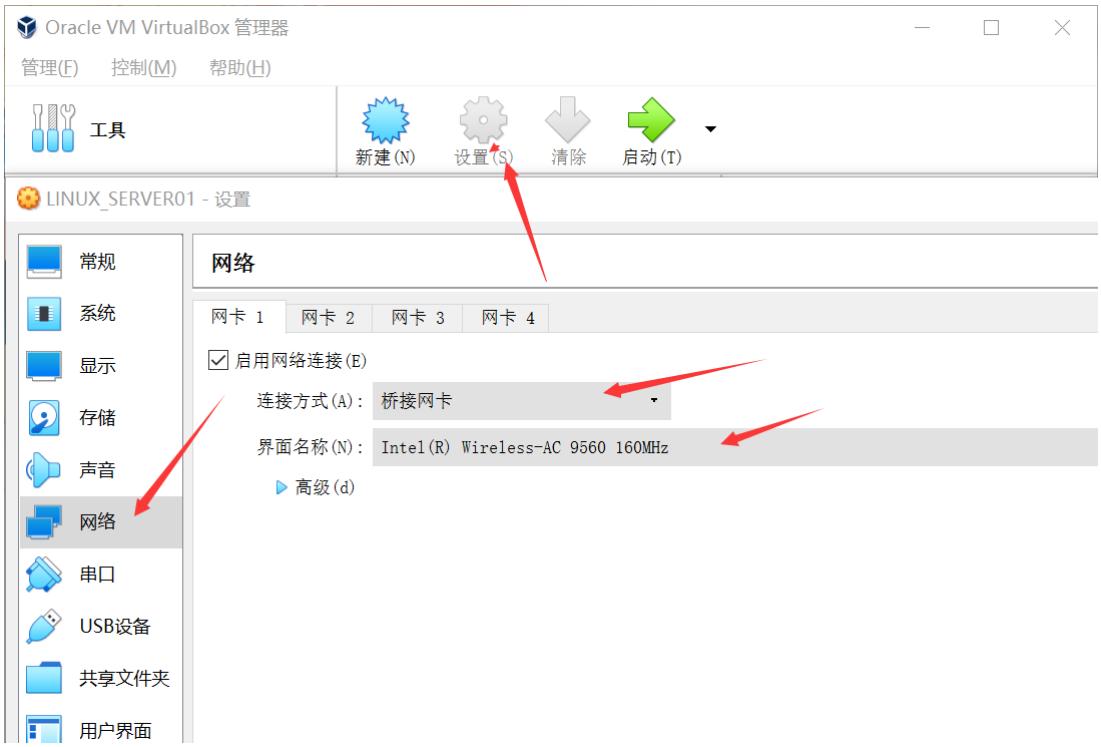
Linux下的软件安装

- 前言：

众所周知，一个机子不能失去软件，就像西方不能失去耶路撒冷。这么多节课的学习，总感觉Linux是不是少了些什么，难道我们一辈子都要对着已有的那些指令敲来敲去吗？并不，我们也可以像windows一样下载软件，美化增强自己的Linux系统。本节，让我们来学习学习如何快速安装软件吧。

- 正文：

1、先给虚拟机联网：要想让虚拟机连上网，要么插网线要么连WiFi，而且这两个玩意还不能离开网卡。虚拟机再怎么说也算一台机子，如果它没有网卡，那比不可能上网，故我们要用到virtualbox的桥接网卡。

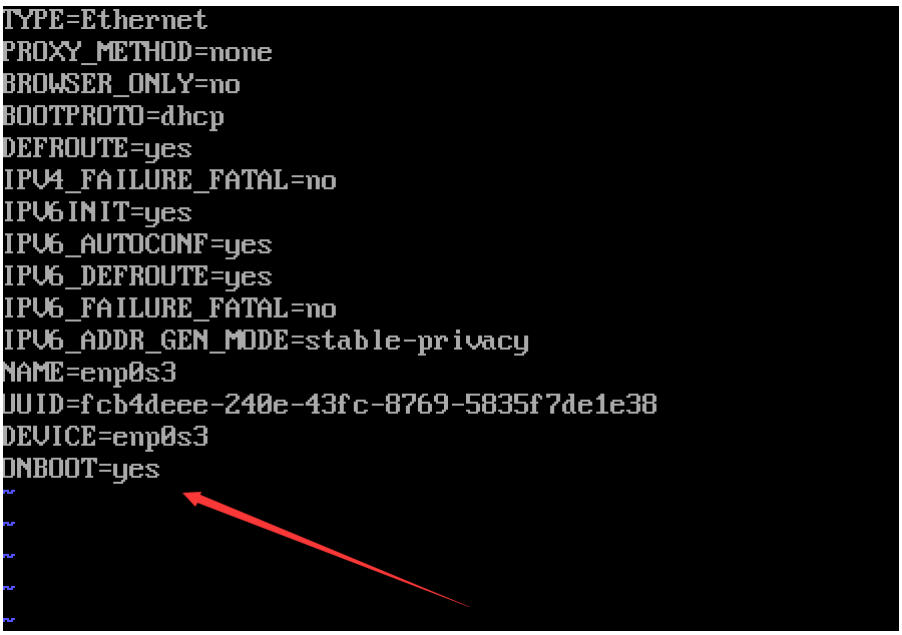


选择桥接网卡，其实就是让虚拟机连到物理机的网卡，一般情况下，界面名称第一个就是默认要找的。这里可以看到是AC 9560。以后要是像用Linux破解邻居的WiFi，也是要用到特定网卡，这里可以查看自己的网卡对不对头。设置好后，按确认保存然后重启即可。

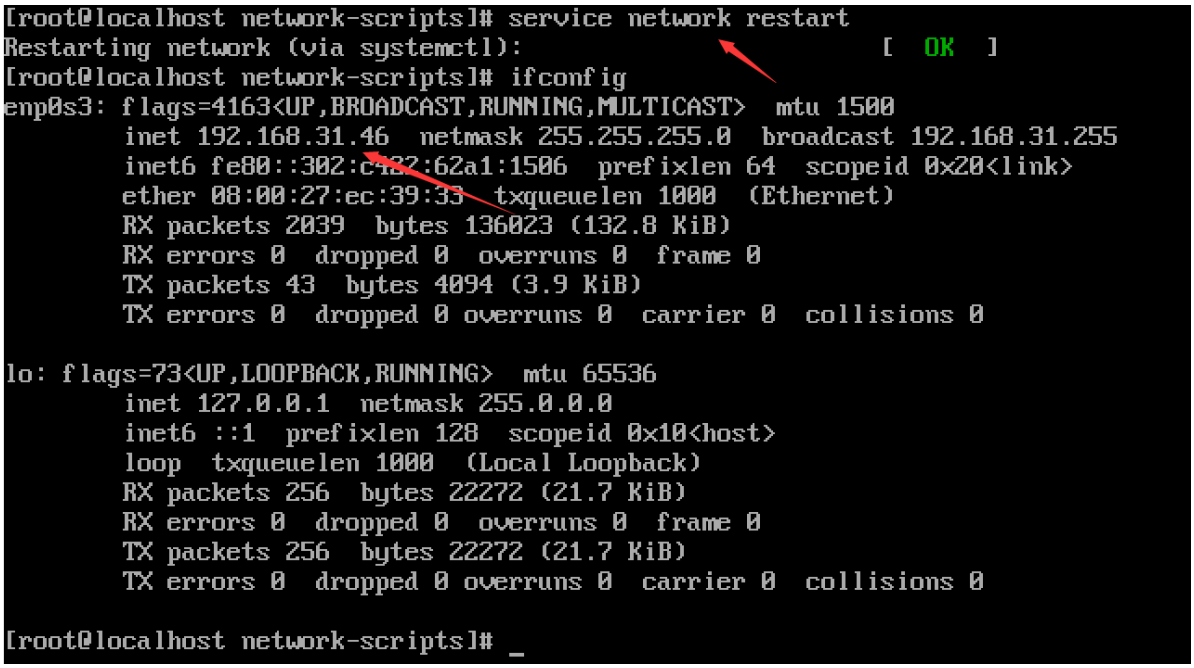
这就完了吗？没有，接下来要配置网卡获取IP，这一步要进Linux操作了。家用路由器有个DHCP功能，专门动态地给计算机分配IP地址，所以我们的目的就是一劳永逸地让Linux系统迎合DHCP自动获取IP。

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# ls ifcfg-*
ifcfg-enp0s3  ifcfg-lo
[root@localhost network-scripts]# _
```

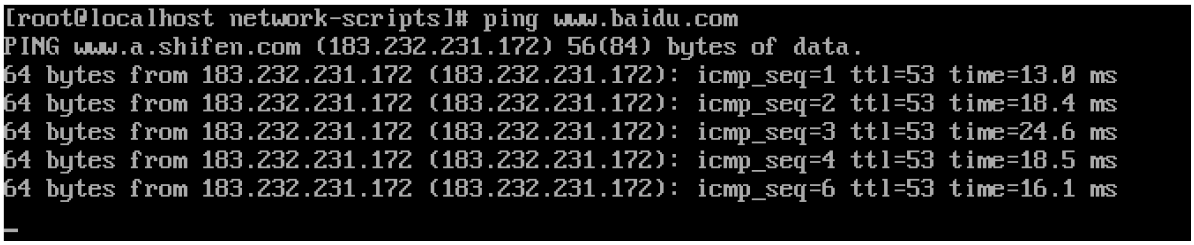
打开配置网络的文件夹，然后找到这个ifcfg-enp0s3，我电脑里叫这个，也有可能是类似ifcfg-eth0这样的名字，反正就是网卡配置文件。



接着用Vim打开它，将上面一栏修改成DHCP，但是可以看到这个文件的默认已经是dhcp了，那么我们只用改下面的onboot为yes即可，意思就是开机自动获取动态IP。



service network restart重启网卡，可以看到获取了一个IP，联网成功啦！



ping一下百度服务器，可以看到ping通了，之前我们示范重定向时是ping不通的，现在是真连上网了。当然这里也可以curl -I 网站来访问url，curl命令相当于简易浏览器，它会输出一部分网页信息。它常用来测试某个网站是否可以正常打开。另外，还有wget指令，后接下载地址可以直接下载文件而不用登录网站。

2、认识RPM及其应用：

Linux下的软件安装和windows大同小异，都是把别人打包好并配置好的文件放在指定的文件夹路径里，然后启动可执行文件，就用上了。就连我们平时使用的ls等指令，都可以算是软件。我们whereis ls，看到它在/usr/bin/ls里，那么我们再ls -l查看它的属性，发现是个文件，是文件那好办，那就可以查看它的内容——head /usr/bin/ls，发现全是乱码，看个🐂！那就file命令查看吧：

```
[root@localhost ~]# whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz
[root@localhost ~]# ls -l /usr/bin/ls
-rwxr-xr-x. 1 root root 117608 Aug 20  2019 /usr/bin/ls
(reverse-i-search)`': head /usr/bin/ls^C
[root@localhost ~]# file /usr/bin/ls
/usr/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (u
bs), for GNU/Linux 2.6.32, BuildID[sha1]=aaf05615b6c91d3cbb076af81aeff531c5d7dfd9, str
[root@localhost ~]# _
```

如图，executable，可执行的，意思是说这是个可执行文件。相当于我们Windows下的.exe，这我们再熟悉不过了。

那么，到底什么是可执行文件呢？说白了，一堆人写的代码编译成计算机看的懂的二进制文件，然后计算机就可以按照我们的代码行事了，这个二进制文件就是执行文件。我原来的用户上有GCC，Linux最常用的编译软件，但这里没有，就不演示了，如果有兴趣的小伙伴可以yum -y install gcc 来安装gcc，自己编个hello,world，至于gcc的编译命令，建议百度。

好，言归正传，Linux下的软件安装有三种方式：

①网上找源码或自己写，然后编译代码得到可执行文件；

②使用RPM包：说白了就是安装包，它把可执行软件都打包在一个个RPM包里，然后我们下载安装即可，不过这样会实去很多灵活性，例如我们不能DIY软件；

③使用Yum指令安装：yum的底层也是使用RPM包，但它自动把RPM包间的联系处理了，相当于懒人式下载，它会自动在软件仓库（应用商城，笑）找相应的软件一键安装，若没有，那就开摆。

```
[root@localhost ~]# rpm -qa | wc -l
493
[root@localhost ~]# rpm -qf /usr/bin/ls
coreutils-8.22-24.el7.x86_64
[root@localhost ~]# rpm -qa | grep coreutil
coreutils-8.22-24.el7.x86_64
policycoreutils-2.5-34.el7.x86_64
[root@localhost ~]# _
```

我们rpm -qa可以看到现在系统有多少个rpm包（为什么能查看，不是安装包吗？理解成Windows下控制面板下的卸载程序那里的程序一览，装过的包有登记的），但太多了，所以我管道符统计一下就作罢了，否则截图截不过来。rpm -qf对软件操作，可以看到它来自哪个rpm包。当然这个包里肯定不止是ls这个软件，还有别的大把，例如su那些指令。尴尬的是，我用rpm -ql + 完整的包名称，系统提醒我没安装，所以我没啥能截图的力。另外，也可以rpm -qi + 完整的包名称，查看包的详细信息。

下面的关于下载和安装rpm的实操内容我就不示范了，演示这个还是有点麻烦的，还不能保证不出问题：

①wget + 下载网址，下载rpm包；

②rpm -ivh 完整的包名称（带后缀.rpm），安装rpm包；

③若是提示已存在，rpm -e，即可卸载原rpm包，然后再安装。

④有时候是想通过rpm更新软件（旧版的存在使得新版的安装出现conflict），但是下载好包由于依赖关系处理不清楚，删不明白原文件，那就使用 -Uvh 升级参数。

上面我们提到了一个依赖关系，那什么是rpm的依赖关系呢？我们要知道，由于一些底层原因，就像我们编软件都常用多文件编程一样，一些应用的安装可能是多个rpm包合一起安装的，失去它们其中一个，另外的包都不能发挥作用，这就是它们之间的依赖关系。别看coreutil好像是个小小的基础包，但是它的删除特别麻烦，因为它被成百个包关联了，这意味着我们删除它要上百个包一起删除，虽然它的安装只要两个包一起安装就好了= =

这也是为什么Yum横空出世，因为它给我们解决了这些繁杂的依赖关系。

3、快速入门rpm包的制作：

话不多说，先上大致制作流程：

①安装rpm-build命令；②创建一个rpmbuild目录以及6个下面的子目录；③创建一个配置文件让rpmbuild目录生效；④准备好原文件并且打包；⑤编写一个Spec文件（rpm打包配置文件）；⑥使用rpm-build命令生成rpm包。

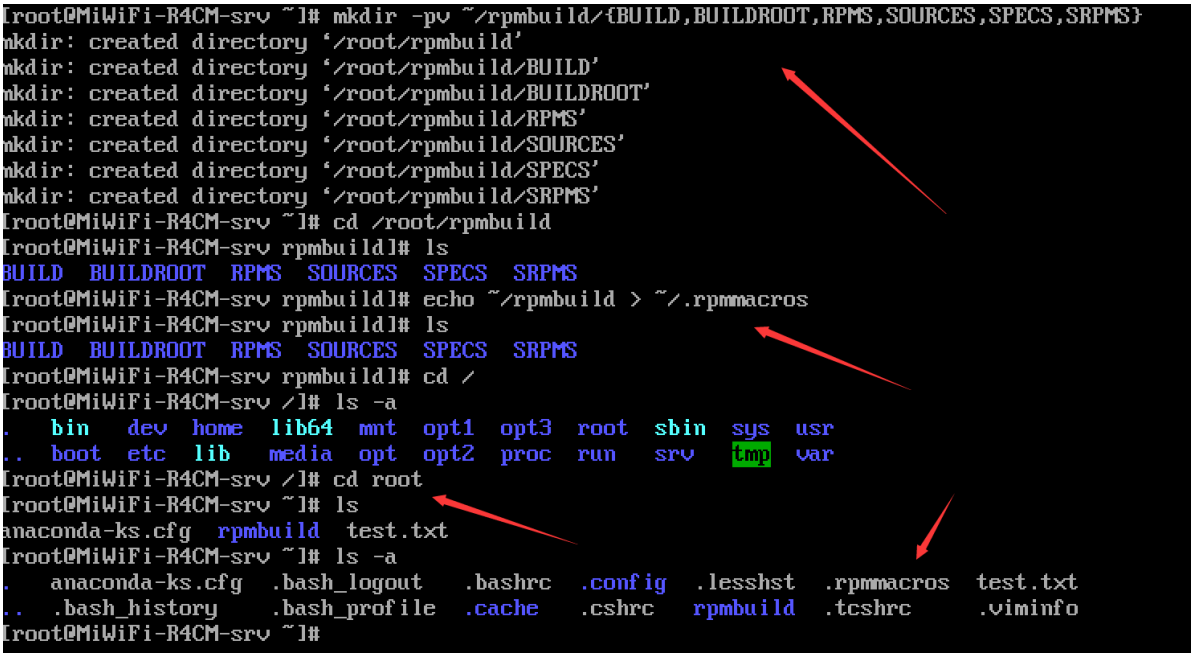
然后我们细细道来：

①使用yum -y install rpm-build命令下载制作软件。

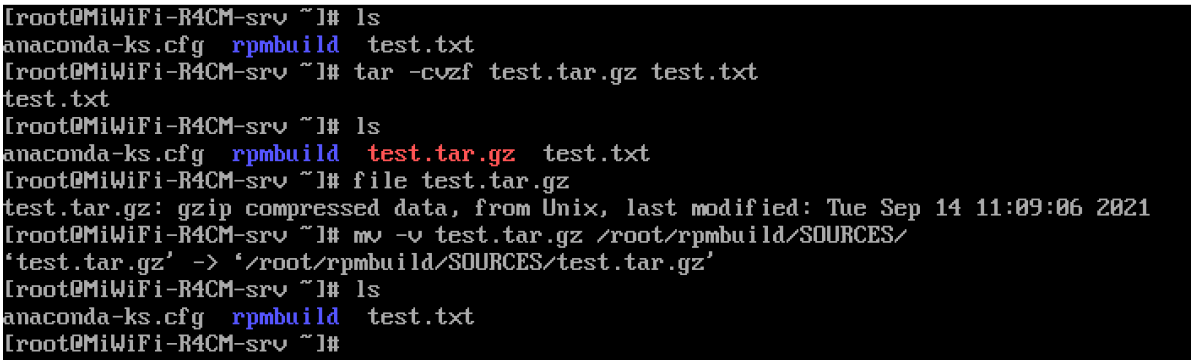
```
--------------------------------------------------------------------------------
Total                                              424 kB/s | 470 kB  00:00:01
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
 Userid     : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
 Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
 Package    : centos-release-7-9.2009.0.el7.centos.x86_64 (@anaconda)
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python-srpm-macros-3-34.el7.noarch                           1/7
  Installing : dwz-0.11-3.el7.x86_64                                        2/7
  Installing : perl-Thread-Queue-3.02-2.el7.noarch                          3/7
  Installing : perl-srpm-macros-1-8.el7.noarch                              4/7
  Installing : redhat-rpm-config-9.1.0-88.el7.centos.noarch                 5/7
  Installing : patch-2.7.1-12.el7_7.x86_64                                  6/7
  Installing : rpm-build-4.11.3-45.el7.x86_64                               7/7
  Verifying  : patch-2.7.1-12.el7_7.x86_64                                  1/7
  Verifying  : perl-srpm-macros-1-8.el7.noarch                              2/7
  Verifying  : perl-Thread-Queue-3.02-2.el7.noarch                          3/7
  Verifying  : rpm-build-4.11.3-45.el7.x86_64                               4/7
  Verifying  : dwz-0.11-3.el7.x86_64                                        5/7
  Verifying  : python-srpm-macros-3-34.el7.noarch                           6/7
  Verifying  : redhat-rpm-config-9.1.0-88.el7.centos.noarch                 7/7

Installed:
  rpm-build.x86_64 0:4.11.3-45.el7

Dependency Installed:
  dwz.x86_64 0:0.11-3.el7                   patch.x86_64 0:2.7.1-12.el7_7
  perl-Thread-Queue.noarch 0:3.02-2.el7     perl-srpm-macros.noarch 0:1-8.el7
  python-srpm-macros.noarch 0:3-34.el7      redhat-rpm-config.noarch 0:9.1.0-88.el7.centos

Complete!
[root@MiWiFi-R4CM-srv lib]#
```

我在图书馆的时候用图书馆的网，执行这条命令说没找对应的源仓库，我还以为真没这个仓库。我测了一下网，发现根本没连上网。后来回到宿舍才分配到宿舍WiFi的动态IP，图书馆的网要验证账户的，问题应该就是这里，没有验证就没有DHCP服务。如图，原本Linux不带这个build软件包的，要自己下载安装，yum一键解决。

②在/root/下，创建一个rpmbuild目录，然后在它下面分别建6个子目录，并创建一个.rpmmacros文件使得/root/rpmbuild目录生效。

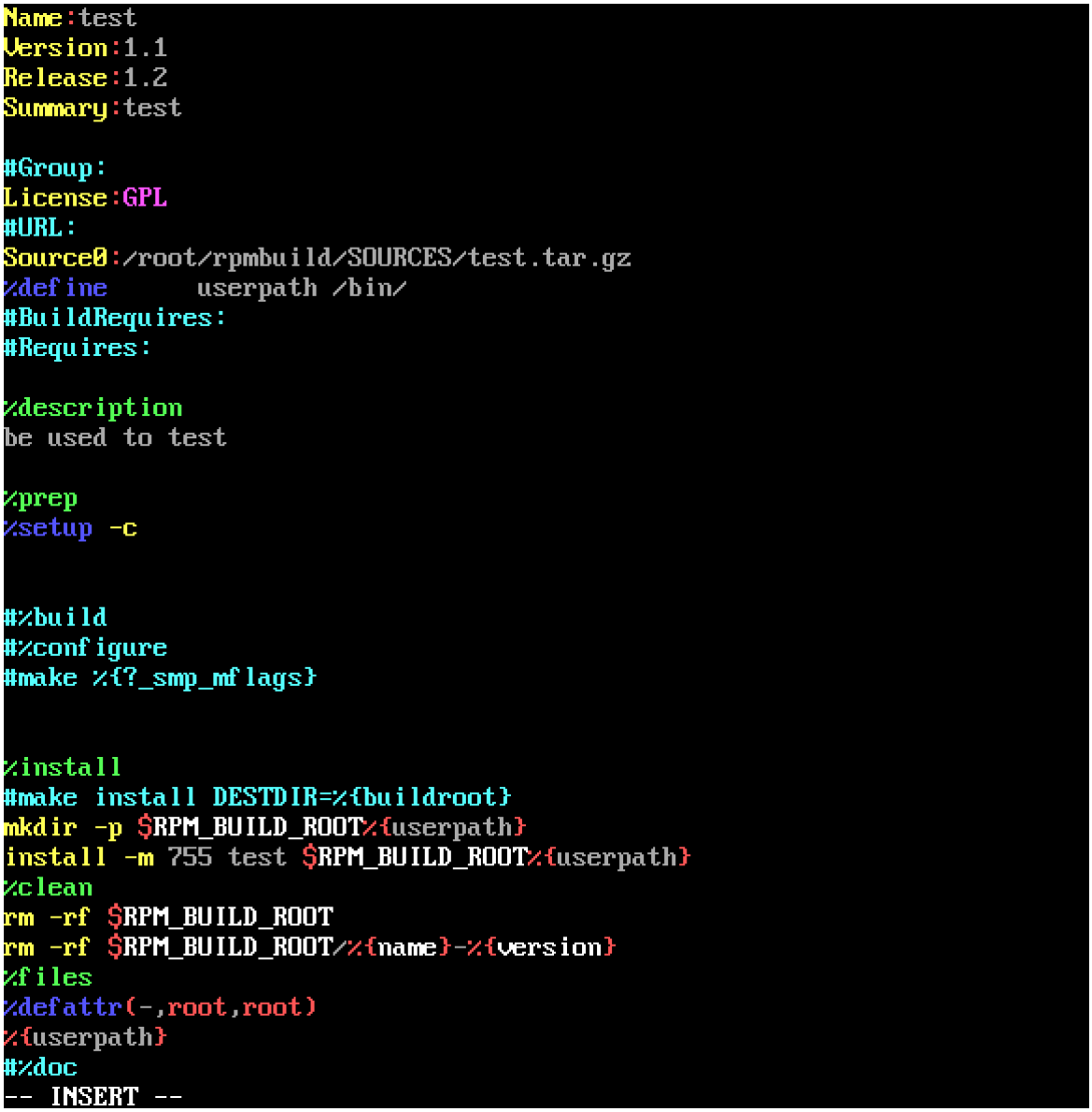


③打包压缩要安装的文件，并移到/root/rpmbuild/SOURCES中。



tar -cvf etc.tar /etc/表示把/etc/下所有东西（包括etc目录）打包成一个文件，且起名etc.tar，若要还原，则是tar -xvf etc.tar命令。此外，tar还支持压缩文件，tar -cvzf etc.tar.gz /etc/，表示打包后压缩一下，成为.tar.gz，注意这是个标准后缀。

④准备RPM包制作的Spec文件，即配置文件，以下是一个模板demo.spec。



其实不用照着打啦，因为这个是失败的，我根本不会写这些信息，如果你有需求的话，自己上网百度看看这些信息怎么填吧。反正spec文件就是放在rpmbuild/SPECS/里的，定义了这个安装包文件来源、相关信息和安装的具体操作。

⑤制作模板保存在SPECS/里后，执行rpmbuild -ba /root/rpmbuild/SPECS/demo.spec

```
+ /usr/bin/mkdir -p test-1.1
+ cd test-1.1
+ /usr/bin/tar -xvvf -
+ /usr/bin/gzip -dc /root/rpmbuild/SOURCES/test.tar.gz
-rw-r--r-- root/root        48 2021-09-12 09:39 test.txt
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ /usr/bin/chmod -Rf a+rX,u+w,g-w,o-w .
+ CFLAGS='-O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=s
sp-buffer-size=4 -grecord-gcc-switches   -m64 -mtune=generic'
+ export CFLAGS
+ CXXFLAGS='-O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param
=ssp-buffer-size=4 -grecord-gcc-switches   -m64 -mtune=generic'
+ export CXXFLAGS
+ FFLAGS='-O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=s
sp-buffer-size=4 -grecord-gcc-switches   -m64 -mtune=generic -I/usr/lib64/gfortran/modules'
+ export FFLAGS
+ FCFLAGS='-O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=
ssp-buffer-size=4 -grecord-gcc-switches   -m64 -mtune=generic -I/usr/lib64/gfortran/modules'
+ export FCFLAGS
+ LDFLAGS='-Wl,-z,relro '
+ export LDFLAGS
+ '[' 1 == 1 ']'
+ '[' x86_64 == ppc64le ']'
++ find . -name config.guess -o -name config.sub
+ ./configure --build=x86_64-redhat-linux-gnu --host=x86_64-redhat-linux-gnu --program-prefix= --dis
able-dependency-tracking --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin --sy
sconfdir=/etc --datadir=/usr/share --includedir=/usr/include --libdir=/usr/lib64 --libexecdir=/usr/l
ibexec --localstatedir=/var --sharedstatedir=/var/lib --mandir=/usr/share/man --infodir=/usr/share/i
nfo
/var/tmp/rpm-tmp.batXgq: line 51: ./configure: No such file or directory
error: Bad exit status from /var/tmp/rpm-tmp.batXgq (%prep)


RPM build errors:
    Bad exit status from /var/tmp/rpm-tmp.batXgq (%prep)
[root@MiWiFi-R4CM-srv ~]# _
```

果然失败了，我不会写啦🤣🤣🤣如果最后是exit 0则表示执行成功，然后RPM包生成在/root/rpmbuild/RPMS/里

4、我是懒狗，只用yum：

①查看要下载的软件的相关信息——yum install + 软件名

```
 libthai                        x86_64      0.1.14-9.el7                  base         187 k
 libtiff                        x86_64      4.0.3-35.el7                  base         172 k
 libwayland-client              x86_64      1.15.0-1.el7                  base          33 k
 libwayland-server              x86_64      1.15.0-1.el7                  base          39 k
 libxcb                         x86_64      1.13-1.el7                    base         214 k
 libxshmfence                   x86_64      1.2-1.el7                     base         7.2 k
 libxslt                        x86_64      1.1.28-6.el7                  base         242 k
 lksctp-tools                   x86_64      1.0.17-2.el7                  base          88 k
 log4j                          noarch      1.2.17-16.el7_4               base         444 k
 mesa-libEGL                    x86_64      18.3.4-12.el7_9               updates      110 k
 mesa-libGL                     x86_64      18.3.4-12.el7_9               updates      166 k
 mesa-libgbm                    x86_64      18.3.4-12.el7_9               updates       39 k
 mesa-libglapi                  x86_64      18.3.4-12.el7_9               updates       46 k
 pango                          x86_64      1.42.4-4.el7_7                base         280 k
 pcsc-lite-libs                 x86_64      1.8.8-8.el7                   base          34 k
 python-javapackages            noarch      3.4.1-11.el7                  base          31 k
 python-lxml                    x86_64      3.2.1-4.el7                   base         758 k
 tomcat-el-2.2-api              noarch      7.0.76-16.el7_9               updates       83 k
 tomcat-jsp-2.2-api             noarch      7.0.76-16.el7_9               updates       96 k
 tomcat-lib                     noarch      7.0.76-16.el7_9               updates      3.9 M
 tomcat-servlet-3.0-api         noarch      7.0.76-16.el7_9               updates      214 k
 ttmkfdir                       x86_64      3.0.9-42.el7                  base          48 k
 tzdata-java                    noarch      2021a-1.el7                   updates      191 k
 xalan-j2                       noarch      2.7.1-23.el7                  base         1.9 M
 xerces-j2                      noarch      2.11.0-17.el7_0               base         1.1 M
 xml-commons-apis               noarch      1.4.01-16.el7                 base         227 k
 xml-commons-resolver           noarch      1.2-15.el7                    base         108 k
 xorg-x11-font-utils            x86_64      1:7.5-21.el7                  base         104 k
 xorg-x11-fonts-Type1           noarch      7.5-9.el7                     base         521 k

Transaction Summary
================================================================================================
Install  1 Package (+84 Dependent packages)

Total download size: 58 M
Installed size: 167 M
Is this ok [y/d/N]:
```

如图，yum install tomcat后，打印出了大量消息，罗列的那些项其实就是这个RPM包的依赖关系，+84说明要下载84个包才能成功安装这个软件，最后还提示了下载大小和安装大小，并问我们要不要下，那我当然是N了。

②Yum也可只是单纯地查看软件——yum search + 软件名，支持模糊搜索

```
[root@MiWiFi-R4CM-srv ~]# yum search emacs
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.ustc.edu.cn
 * extras: mirrors.dgut.edu.cn
 * updates: mirrors.ustc.edu.cn
================================== N/S matched: emacs ==================================
emacs.x86_64 : GNU Emacs text editor
emacs-a2ps.x86_64 : Emacs bindings for a2ps files
emacs-a2ps-el.x86_64 : Elisp source files for emacs-a2ps under GNU Emacs
emacs-auctex.noarch : Enhanced TeX modes for Emacs
emacs-common.x86_64 : Emacs common files
emacs-el.noarch : Lisp source files included with GNU Emacs
emacs-filesystem.noarch : Emacs filesystem layout
emacs-gettext.noarch : Support for editing po files within GNU Emacs
emacs-git.noarch : Git version control system support for Emacs
emacs-git-el.noarch : Elisp source files for git version control system support for Emacs
emacs-gnuplot.noarch : Emacs bindings for the gnuplot main application
emacs-gnuplot-el.noarch : Emacs bindings for the gnuplot main application
emacs-libidn.noarch : GNU Emacs libidn support files
emacs-mercurial.x86_64 : Mercurial version control system support for Emacs
emacs-mercurial-el.x86_64 : Elisp source files for mercurial under GNU Emacs
emacs-nox.x86_64 : GNU Emacs text editor without X support
emacs-php-mode.noarch : Major GNU Emacs mode for editing PHP code
emacs-terminal.noarch : A desktop menu item for GNU Emacs terminal.
ocaml-emacs.x86_64 : Emacs mode for OCaml
protobuf-emacs.x86_64 : Emacs mode for Google Protocol Buffers descriptions
protobuf-emacs-el.x86_64 : Elisp source files for Google protobuf Emacs mode
ctags-etags.x86_64 : Exuberant Ctags for emacs tag format
emacs-auctex-doc.noarch : Documentation in various formats for AUCTeX

  Name and summary matches only, use "search all" for everything.
[root@MiWiFi-R4CM-srv ~]# _
```

如图，一堆包，顺带注解。

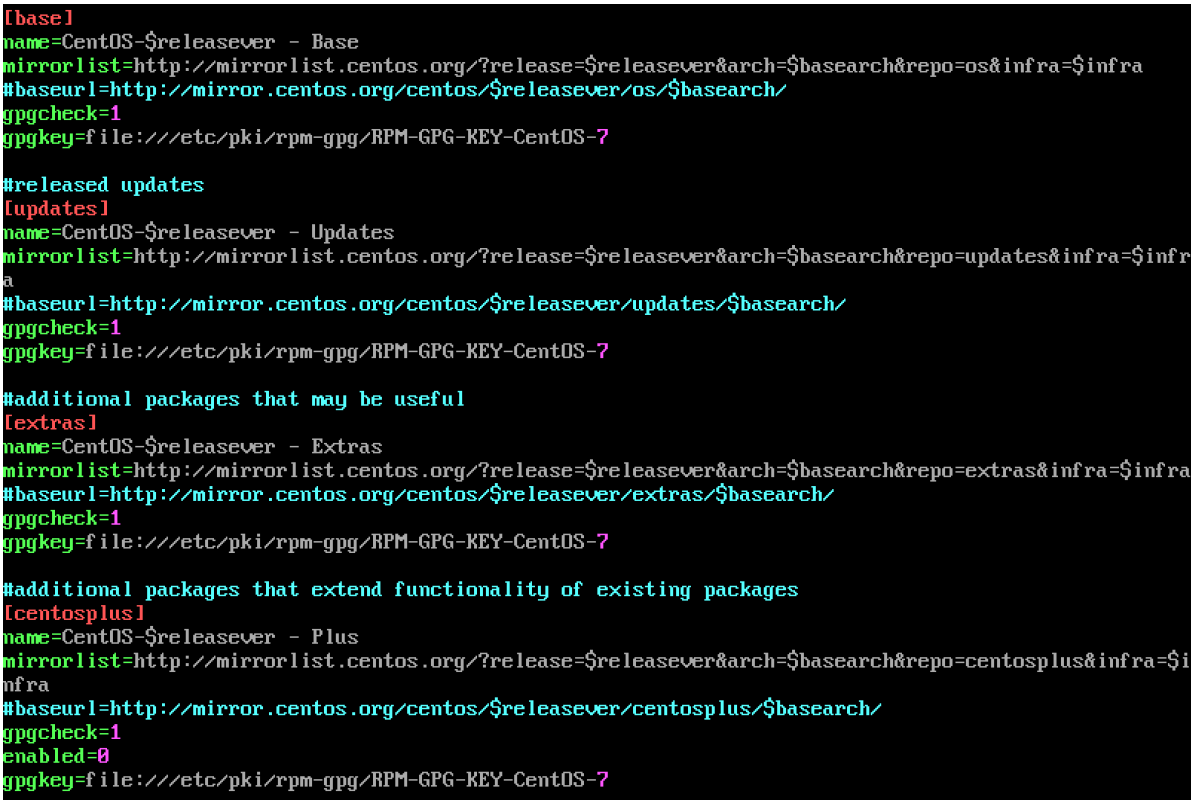
③删除命令——yum remove + 软件名，相当于rpm -e + 具体的包，这里就不演示了，没东西删力。

④更新命令——yum upgrade + 软件名，只升级包不升级系统内核；yum update + 软件名，既升级包有升级内核。

⑤为什么yum总能下到软件呢？这是因为yum是从已有的仓库里寻找对应的软件下的，这些仓库就好比Windows上的应用商店，当系统装好后，自带了很多yum源，统一放在了/etc/yum.repos.d/里。

```
[root@MiWiFi-R4CM-srv ~]# ls /etc/yum.repos.d/
CentOS-Base.repo  CentOS-Debuginfo.repo  CentOS-Media.repo    CentOS-Vault.repo
CentOS-CR.repo    CentOS-fasttrack.repo  CentOS-Sources.repo  CentOS-x86_64-kernel.repo
[root@MiWiFi-R4CM-srv ~]# _
```

打开一个看看，我们选Base吧



```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates&infra=$infr
a
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus&infra=$i
nfra
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

【base】：表示一段配置的开始，下面的行都属于这一段配置，它也叫容器。里面的名字自己定义，当不能存在两个相同的【】。

name：这个仓库的注解。

mirrorlist：可以使用的镜像站点。

baseurl：后面的网址就是Yum真正去到的地址。

enable：是否启用

如果要添加Yum源，直接网上复制粘贴到repo文件再放到这里即可。

- 后记：

这节我们要学会上网，Linux下软件安装的原理和流程，以及使用yum快捷安装。