1. **安装源码包依赖的步骤**

软件安装命令介绍参考资料：

鸟哥私房菜（第四版）第22章

rpm命令介绍:https://blog.csdn.net/qq\_18312025/article/details/78080112

编译源码包我需要知道一下几个步骤，就是先找到SPRM源码包的链接使用wget将源码包下载放到本地再执行rpm -ivh (srpm源码包)生成rpmbuild文件，当然也可以使用rpm -ivh （源码包链接）的方式直接生成（建议不使用后者），这时我们需要安装编译源码包的命令（rpmbuild）环境先使用

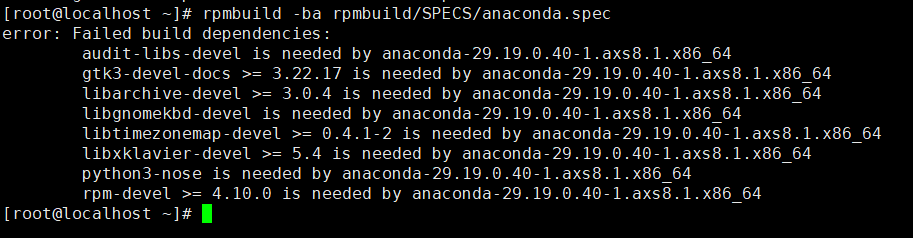
sudo yum install rpm-build安装命令

然后再sudo yum install rpmdevtools

当然直接装后者也行，这样会同时把rpm-build命令也装上。

然后进入rpmbuild/SPECS/文件夹执行rpmbuild -ba .spec文件。

这是可能出现缺依赖的情况比如



就需要我们去解决依赖问题；下面是*常用的查找rpm包的网站:*

*https://pkgs.org/*

*http://rpmfind.net/*

*http://rpm.pbone.net/*

*https://developer.aliyun.com/mirror/*

可以再这几个网站里找相关的依赖包，解决依赖后就可以编译了。

大多数依赖都可以自动安装是用dnf或yum命令建议使用dnf命令。

### **以下是使用脚本dnf自动解决一些依赖包的脚本命令：**

1、 普通用户下使用脚本1

#!/bin/bash

rpmbuild -ba ~/rpmbuild/SPECS/\* 2>tmp.txt

grep -v "error" tmp.txt|awk '{print $1}'>dnf\_file.txt

rm -f tmp.txt

会在家目录生成dnf\_file.txt文件

2、切换到root用户下使用脚本2

#!/bin/bash

cat $1|while read name

do

dnf --enablerepo=PowerTools install -y $name || dnf install -y $name

done

rm -f $1

会自动读取dnf\_file.txt文件中的内容进行安装

使用以上脚本可解决大部分依赖问题,其次 遇到没有对应版本的依赖项,可使用Fedora或centos7对应版本的src包进行重新编译,如使用此方法直接安装其他版本的RPM包,其依赖包,会和当前环境有冲突问题,或者会对当前环境造成破坏

1. **其他解决依赖的方式方法以及错误处理的方式**

有的包的依赖与系统的一些依赖包冲突不能再普通环境下进行编译，那就得使用mock环境进行编译了。

**首先：**

### 构建mock环境编译rpm包 (优点不会破坏当前环境,生成一个独立的环境)

方法:

1、安装 epel-release, mock

yum -y install epel-release

yum -y install mock

2 、root用户

mock -r /etc/mock/epel-8-x86\_64.cfg --clean

mock -r /etc/mock/epel-8-x86\_64.cfg --init

3 、将生成的root拷贝到桌面

cp -r /var/lib/mock/epel-8-x86\_64/root 你想放的地方

4 、chroot root/进去到环境中

**在chroot环境中可能会遇到缺少依赖的问题 可通过本机环境定向安装rpm包**

命令为:

rpm -ivh --prefix=绝对路径 +对应rpm包名

Yum: dnf --installroot=绝对路径 install +rpm包名

使用以上命令可以使用本机环境的yum源,定向安装

或者使用mock -r ax8\_x86-64 加需要编译的源码包

### **三、再编译的过程中一些比较大的包可能会遇到内存不够用的问题，设置交换分区可以解决一定的问题**

### 设置swap交换分区:

在编译较大的源码包时会遇到速度过慢,卡死,或者编译失败的问题,此时要考虑是否是本机交换分区过小的原因(查看本机交换分区大小命令为:free -h)

设置方法:

*sudo mkdir swapfile*

*cd /swapfile*

*sudo dd if=/dev/zero of=swap bs=1024 count=20000000 (此处设置为20G)*

*sudo mkswap -f swap*

*sudo swapon swap*

1. **再编译的过程中可能会出现报错，这时候就得找官方的补丁来打上，下面是打补丁的步骤**

*有些时候需要对源码包进行打补丁操作*

*使用命令 diff -Nur 旧包 新包 > 补丁名字.patch*

*Spec文件中写法:补丁位置卸载Source标签下*

*例如:*

*Source1: \*\*\*\*.tar.xz*

*#补丁位置 一般从0开始 要备注好你所打补丁的介绍*

*Patch0:补丁名字.patch*

*%prep*

*# 补丁位置*

*%patch0 -p0*