1.工具安装:

#yum install rpmbuild

#yum install rpmdevtools

2.创建工作目录:

#rpmdev-setuptree

(rpmdev-setuptree命令默认将再当前用户主目录下创建一个RPM工作目录，如果需要改变默认位置，可以修改配置文件:~/.rpmmacros中变量\_topdir对应的值)

目录结构:

(建议用命令行构建)

--BUILD #编译之前，如解压包后存放的路径(./configure)  
--BUILDROOT #编译后存放的路径(cp&&make install)  
--SOURCES #源包所放置的路径(tar)  
--SPECS #spec文档放置的路径

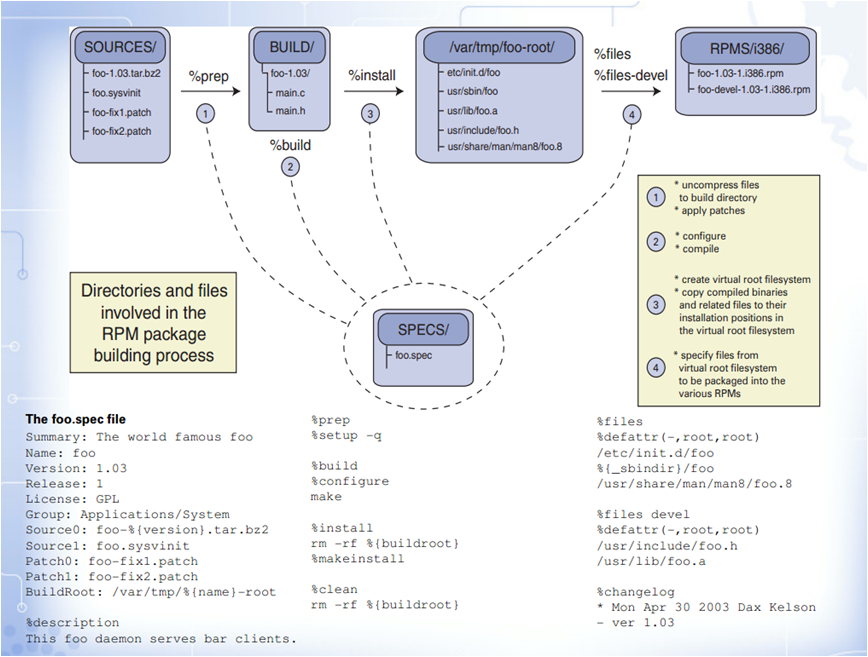
--RPMS #打包完成后二进制rpm包存放的路径  
--SPRMS #打包完成后源码rpm包存放的路径

3.将需要打成rpm包的源包放到SOURCES目录下

4. 撰写.spec文件 .spec撰写是打包RPM的核心, rpmbuild会根据这个.spec文件来打包，

几乎就是编写一个脚本

一个.spec文件的例子



%prep (关键字)预处理脚本

%setup –q(宏命令)把SOURCES下源码包解压并放到BUILD目录下  
%build 在BUILD开始构建包

#调用源码目录中的configure命令

% configure

#需要指定路径时，和一般的Linux源码安装一样，

#./configure --prefix=$RPM\_BUILD\_ROOT/指定路径(替换% configure),%install时再到#cd $RPM\_BUILD\_DIR/解压后文件夹下make [prefix=指定目录] install

#在源码目录中执行自动构建命令make(compile)

make

%install 开始把软件安装到虚拟的根目录(BUILDROOT)中

#先删除安装路径下的所有

rm –rf %{BuildRoot}

#安装

#下面的宏命令相当于make DESTDIR=$RPM\_BUILD\_ROOT install

%makeinstall

%clean

#清理命令

rm –rf %{BuildRoot}

%files

#需要打包的文件 在寻找这些文件时,rpmbuild会在之前加上$RPM\_BUILD\_ROOT路##径，比如/etc/init.d/foo会在$RPM\_BUILD\_ROOT/etc/init.d/foo下找

#可能会用到的关键字

%changelog 变更日志

%pre rpm安装前执行的脚本

%post rpm安装后执行的脚本

可以执行测试(java -version)，添加环境变量等

%preun rpm卸载前执行的脚本

%postun rpm卸载后执行的脚本

$RPM\_BUILD\_DIR-------------BUILD目录

$RPM\_BUILD\_ROOT---------BUILDROOT目录

5 构建

#cd ~/rpmbuild/

#rpmbuild -ba SPECS/nginx.spec

(-ba表示build all，即生成包括二进制包和源代码包的所有RPM包，如果正常的话，rpmbuild将正常退出，同时在RPMS目录和SRPMS目录中将生成对应的RPM包)

遇到的问题

1.%files下指定的文件找不到

打包时，所有的操作都在~/rpmbuild目录下,安装时的操作在$RPM\_BUILD\_ROOT即

~/rpmbuild/BUILDROOT文件夹下,确保该路径下可以找到;

注意： 用户在安装时的安装路径为根目录/指定路径

打包时的安装路径为$RPM\_BUILD\_ROOT/指定路径

2举个例子

%build

cd nginx-1.2.1 #调用源码目录中的configure命令

./configure --prefix=/usr/local/nginx #在源码目录中执行自动构建命令make

make

%install

cd nginx-1.2.1 #调用源码中安装执行脚本

make install

3而一般的解压一个文件,配置环境变量就可以运行的，如jdk

%prep #解压前卸载自带jdk

%install #解压到$RPM\_BUILD\_ROOT/指定路径下

%post #写入/etc/profile,测试(java -version)等