Linux软件包安装（源码安装）

软件包的安装可以分为两大类，一类是二进制形式的，比如说rpm，另外一类就是源代码，源代码稍微复杂一些，但是使用源代码可以使用最新的版本，并且可以使用最新的功能，而且可以自定制，让它更加符合自身的习惯和需求。并且源代码一旦在使用的时候可以在不同的平台上实现。但是rpm包是有平台限定的，源代码可以在任意平台上配置、编译并运行。

源码就是用特定的语言来写出文本。

拆解----》源码-----》配置（./configure）----》（Makefile）编译-----》（可执行头文件 手册）-----》安装

编译器 gcc

如何知道应该具备哪些编译环境呢?

只要在系统当中安装了下列四类软件组，那么编译的环境应该已经具备了。

Installed Groups:

Development Libraries　开发库

Development Tools　开发工具

Legacy Software Development　传统的软件开发

X Software Development　图形界面下的软件开发

在实现源代码编译的时候，先查看一下是否具备了这么四类组。

库文件:

系统能自动找到的目录：/lib　/usr/lib

为了找到需要用到的一些库，必须把那些库放到 /lib 或 /usr/lib 下

动态库文件 .so（system object）

静态库文件 .a

头文件:

/include　/usr/include　建立符号链接，让系统能找到

手册:

编辑/etc/man.config文件 添加MANPATH

拆解位置　/usr/local/src

安装位置　/usr/local/

Configure选项

--prefix 指明安装目录

--sysconfdir　指明配置文件放在哪里

--enable*

--disable*

示例：安装比较新版本的Apache httpd 2.4

需要用到的软件包

apr-1.4.6.tar.gz

apr-util-1.5.1.tar.gz

httpd-2.4.3.tar.gz

pcre-8.32.tar.gz

拆解httpd软件包至/usr/local/src/目录下

tar -zxvf httpd-2.4.3.tar.gz -C /usr/local/src/

查看INSTALL或README说明文件：

注意：configure是一个脚本，这个脚本的最终目的是为了生成Makefile文件的。但是在生成Makefile文件之前，由于每个人的机器环境不一样，所以它要做各种各样的检测看当前的环境是否满足运行这个软件的需求，所以会看到各种各样的checking。

可以看到，找不到APR。

发现有装apr。可能有些库文件没有，这些库文件并不包含在apr-1.2.7-11.el5_3.1这个主程序中。比如apr往往会包含在apr-devel，会把一些共享的库放在devel这个文件中，所以可能是devel没装。

安装了apr-devel之后，编译的时候提示apr版本至少要是1.4或更高，说明版本太低。

安装 apr-1.4.6.tar.gz

```
[root@Device-8C324C apr-1.4.6]# cd /usr/local/apr/
[root@Device-8C324C apr]# ll
total 16
drwxr-xr-x 2 root root 4096 Jan 25 12:54 bin       可执行文件
drwxr-xr-x 2 root root 4096 Jan 25 12:54 build-1
drwxr-xr-x 3 root root 4096 Jan 25 12:54 include       头文件
drwxr-xr-x 3 root root 4096 Jan 25 12:54 lib       库文件
```

头文件的处理：

库文件的处理：

ldconfig：刷新缓存

ldconfig -pv |grep apr 查看链接库是否已经加载进去（-p显示路径，-v显示详细信息）

安装apr-util-1.5.1.tar.gz

上面的准备工作都做好之后，再重新安装Apache：

pcre是描述正则表达式的一个扩展库。显示找不到pcre。

说明pcre装了，但是库找不到，库往往会出现在devel这个包里面。

最后再执行make ， make install。

头文件的处理：

```
[root@localhost httpd-2.4.3]# cd /usr/include/
[root@localhost include]# ln -s /usr/local/apache/include/* .
```

库文件的处理：

[root@localhost ld.so.conf.d]# pwd

/etc/ld.so.conf.d

[root@localhost ld.so.conf.d]# vim apache.conf

/usr/local/apache/modules

[root@localhost ld.so.conf.d]# ldconfig

[root@localhost ld.so.conf.d]# ldconfig -pv | grep apache

[root@localhost ld.so.conf.d]#

没有加载成功，不过没关系，这些库都是Apache自己使用，而且库的目录名称module也正常lib的不一样。

[root@localhost bin]# file apachectl

apachectl: Bourne shell script text executable

[root@localhost bin]# file httpd

表示它是一个bash的脚本，只是一个可执行脚本

httpd: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared libs), for GNU/Linux 2.6.9, not stripped

表示他是一个服务器的程序

将服务挂接配置文件来执行：

[root@localhost bin]# ./httpd -h

Options:

-f file : specify an alternate ServerConfigFile

[root@localhost bin]# ./httpd -f /etc/apache/httpd.conf

一般情况下对一个服务的控制：

service 名称（控制脚本） start 控制脚本一般放在/etc/init.d/目录下。

源码一般不会提供控制脚本，这时候要写控制脚本。

提供控制脚本，一般放在/etc/init.d/目录下

要做到对服务的自动控制

[root@localhost init.d]# chkconfig --list（查看系统中所有安装的服务）

[root@localhost init.d]# chkconfig --list| grep sshd

sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@localhost init.d]# chkconfig sshd off

将所有级别设置为关闭

服务启动的顺序号

[root@localhost init.d]# ll /etc/rc.d/rc3.d/ | less

total 292

lrwxrwxrwx 1 root root 17 Dec 21 04:52 K01dnsmasq -> ../init.d/dnsmasq

lrwxrwxrwx 1 root root 24 Dec 21 04:56 K02NetworkManager -> ../init.d/NetworkManager

lrwxrwxrwx 1 root root 24 Dec 21 04:55 K02avahi-dnsconfd -> ../init.d/avahi-dnsconfd

lrwxrwxrwx 1 root root 15 Dec 21 04:56 S05kudzu -> ../init.d/kudzu

lrwxrwxrwx 1 root root 18 Dec 21 04:51 S06cpuspeed -> ../init.d/cpuspeed

lrwxrwxrwx 1 root root 19 Dec 21 04:51 S08ip6tables -> ../init.d/ip6tables

lrwxrwxrwx 1 root root 18 Dec 21 04:51 S08iptables -> ../init.d/iptables

S表示进入这个级别的时候要启动哪些服务，后面的数字表示启动的顺

K表示进入这个级别的时候要杀死的服务，

[root@localhost init.d]# ll /etc/rc.d/rc3.d/ | grep sshd

lrwxrwxrwx 1 root root 14 Dec 21 04:55 S55sshd -> ../init.d/sshd

[root@localhost init.d]# chkconfig sshd off

[root@localhost init.d]# ll /etc/rc.d/rc3.d/ | grep sshd

lrwxrwxrwx 1 root root 14 Jan 6 23:11 K25sshd -> ../init.d/sshd

[root@localhost init.d]# vim /etc/init.d/sshd

#!/bin/bash

#

Init file for OpenSSH server daemon

#

chkconfig: 2345 55 25 表示可以使用chkconfig来管理它，在2345级别自动为on，启动的序号是55，杀死的序号是25

description: OpenSSH server daemon

期望用chkconfig来管理我们自己写的httpd脚本。

[root@localhost init.d]# chkconfig --add httpd

service httpd does not support chkconfig 不支持

在控制脚本中，添加下面两条信息即可支持。

chkconfig: 2345 56 26

description: httpd server daemon 如果没有描述信息，chkconfig加不进去

[root@localhost init.d]# chkconfig --add httpd

[root@localhost init.d]# chkconfig --list | grep httpd

httpd   0:off 1:off 2:off 3:off 4:off 5:off 6:off

[root@localhost init.d]# chkconfig httpd on

[root@localhost init.d]# chkconfig --list | grep httpd

httpd   0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@localhost init.d]# ll /etc/rc.d/rc3.d/ | grep httpd

lrwxrwxrwx 1 root root 15 Jan 6 23:20 S56httpd -> ../init.d/httpd

[root@localhost init.d]# chkconfig httpd off

[root@localhost init.d]# ll /etc/rc.d/rc3.d/ | grep httpd

lrwxrwxrwx 1 root root 15 Jan 6 23:21 K26httpd -> ../init.d/httpd

man手册的处理：

httpd脚本:

```
#!/bin/bash
# chkconfig: 2345 80 88
# description: OpenSSH server daemon
prog=/usr/local/apache/bin/httpd
configfile=/etc/apache/httpd.conf
lockfile=/var/lock/subsys/httpd
. /etc/init.d/functions

start() {
if [ -e $lockfile ];then
echo "the program `basename $prog` is started"
else
echo -n -e "the program `basename $prog` is startting...."
sleep 2
fi
}

stop() {
if [ -e $lockfile ];then
echo -n -e"the program `basename $prog` is stopping...."
sleep 2
killproc httpd && echo "ok" && rm -rf $lockfile || echo "fail"
else
echo "the program `basename $prog` is stoped"
fi
}

status() {
if [ -e $lockfile ];then
echo "the program `basename $prog` is running"
else
echo "the program `basename $prog` is stop"
fi
}

case "$1" in
start)
start
;;
stop)
stop
;;
status)
status
;;
restart)
stop
start
;;
*)
echo "USAGE: start|stop|restart|status"
esac
```