**根据CentOS定制自己的发行版**

[cold's world](http://ju.outofmemory.cn/feed/1613/) 2012-01-18 **181** 阅读

本文使用的环境为CentOS5.5 32位. 首先要做的是最小化安装CentOS,就是安装过程中选择要安装包的时候使用自定义,然后去除所有要安装的包,系统就会默认最小化安装系统. 装完系统会再root的根目录下生成3个文件,我们用到两个: anaconda-ks.cfg : kisckstart脚本,记录安装过程的配置,包括选择语言,选择键盘,分区,root密码等等等等 install.log            : 记录安装过程中所安装的包 我们首先要做的就是建立我们的自己发行版的目录,在一个剩余大小大于4G的分区创建自己发行版的目录,比如我的发行版叫ColdOS,然后挂载DVD光盘,把光盘上的内容复制到自己发行版的目录:

mkdir /usr/cold # 创建发行版目录

mount /dev/hdc /mnt # 挂载IDE光驱

# 或

mount /dev/cdrom /mnt

# 如挂载本地ISO镜像使用下面命令

mount -o loop iso路径 /mnt

# 然后使用 tar命令把光盘的内容复制到/usr/cold

# 都说这个命令比cp快,在256内存的虚拟机测试也就快了几十秒,个人觉得最大的用处

# 就是解决了cp无法复制隐藏文件的问题

( cd /mnt/ &amp;&amp; tar -cf - . ) ¦ (cd /usr/cold &amp;&amp; tar -xvfp - ) # tar -p选项是保留原有权限

复制完成后发行版目录应该是:

ls -la /usr/cold

total 511

drwxr-xr-x 7 root root 6144 Apr 30 2010 .

drwxr-xr-x 21 root root 4096 Dec 13 18:07 ..

drwxrwxr-x 2 501 501 421888 Apr 30 2010 CentOS

-rw-r--r-- 1 root root 112 Apr 30 2010 .discinfo

-rw-r--r-- 1 root root 212 Jun 15 2008 EULA

-rw-r--r-- 1 root root 18009 Jun 15 2008 GPL

drwxr-xr-x 4 root root 2048 Apr 30 2010 images

drwxr-xr-x 2 root root 2048 Apr 30 2010 isolinux

drwxr-xr-x 2 root root 22528 Apr 27 2010 NOTES

-rw-r--r-- 1 root root 655 Apr 27 2010 RELEASE-NOTES-cs

-rw-r--r-- 1 root root 1401 Apr 27 2010 RELEASE-NOTES-cs.html

-rw-r--r-- 1 root root 839 Apr 27 2010 RELEASE-NOTES-de

-rw-r--r-- 1 root root 1571 Apr 27 2010 RELEASE-NOTES-de.html

-rw-r--r-- 1 root root 694 Apr 27 2010 RELEASE-NOTES-en

-rw-r--r-- 1 root root 1367 Apr 27 2010 RELEASE-NOTES-en.html

-rw-r--r-- 1 root root 694 Apr 27 2010 RELEASE-NOTES-en\_US

-rw-r--r-- 1 root root 1367 Apr 27 2010 RELEASE-NOTES-en\_US.html

-rw-r--r-- 1 root root 788 Apr 27 2010 RELEASE-NOTES-es

-rw-r--r-- 1 root root 1619 Apr 27 2010 RELEASE-NOTES-es.html

-rw-r--r-- 1 root root 852 Apr 27 2010 RELEASE-NOTES-fr

-rw-r--r-- 1 root root 1641 Apr 27 2010 RELEASE-NOTES-fr.html

-rw-r--r-- 1 root root 766 Apr 27 2010 RELEASE-NOTES-ja

-rw-r--r-- 1 root root 1565 Apr 27 2010 RELEASE-NOTES-ja.html

-rw-r--r-- 1 root root 706 Apr 27 2010 RELEASE-NOTES-nl

-rw-r--r-- 1 root root 1433 Apr 27 2010 RELEASE-NOTES-nl.html

-rw-r--r-- 1 root root 752 Apr 27 2010 RELEASE-NOTES-pt\_BR

-rw-r--r-- 1 root root 1480 Apr 27 2010 RELEASE-NOTES-pt\_BR.html

-rw-r--r-- 1 root root 801 Apr 27 2010 RELEASE-NOTES-ro

-rw-r--r-- 1 root root 1473 Apr 27 2010 RELEASE-NOTES-ro.html

drwxr-xr-x 2 root root 2048 Apr 30 2010 repodata

-rw-r--r-- 1 root root 1512 Jun 15 2008 RPM-GPG-KEY-beta

-rw-r--r-- 1 root root 1504 Jun 15 2008 RPM-GPG-KEY-CentOS-5

-r--r--r-- 1 root root 7048 Apr 30 2010 TRANS.TBL

-rw-r--r-- 1 root root 413 Apr 30 2010 .treeinfo

现在对系统进行精简:

rm -f RELEASE-NOTES-\*

rm -rf NOTES/

rm -f RPM-GPG-KEY-\*

rm -f EULA

rm -f GPL

rm -f CentOS/\* # 删除所有rpm包,等会根据install.log复制过来,保证系统最小化

精简完后应该是这样子的:

s -la /usr/cold/

total 20

drwxr-xr-x 6 root root 4096 Dec 13 19:39 .

drwxr-xr-x 16 root root 4096 Dec 13 23:14 ..

drwxrwxr-x 3 501 501 172032 Dec 14 02:23 CentOS

-rw-r--r-- 1 root root 112 Apr 30 2010 .discinfo

drwxr-xr-x 4 root root 4096 Apr 30 2010 images

drwxr-xr-x 2 root root 4096 Dec 14 00:36 isolinux

drwxr-xr-x 2 root root 4096 Dec 14 01:42 repodata

-r--r--r-- 1 root root 7048 Apr 30 2010 TRANS.TBL

-rw-r--r-- 1 root root 413 Apr 30 2010 .treeinfo

现在根据install.log创建packages.list,用来安装所需要的rpm包:

# 根据install.log,提取其中的rpm包名,

cat install.log ¦ grep Installing ¦ awk '{print $2}' ¦ awk -F':' '{ if (NF==2){ print $2} else {print $1}}' &gt; packages.list

# 如果仅仅最小化安装就失去了定制自己的发行版的意义

# 向packages.list添加几个常用的工具,需要先用yum安装一遍记住包的依赖关系

# 把依赖关系的包也放入packages.list

# 比如man依赖bzip2 groff

vi packages.list

# 末尾添加如下内容:

setuptool

lszrz wget

kernel-devel

kernel-headers

libgomp

cpp

glibc-headers

glibc-devel

gcc

make

which

bzip2

groff

man

# 根据packages.list进行复制

for i in `cat packages.list `;do cp -p -f /mnt/CentOS/"$i"\* /usr/cold/CentOS/;done

复制完所需的RPM包之后我们如何来让系统安装的时候安装我们自定义添加的包呢? 首先我们需要编辑repodata/comps.xml,但是comps.xml文件内容太杂,大概 2万多行,所以我们需要对comps.xml进行一个预处理:

# comps.xml包含最多的是各国语言

# 我们先去除不需要的语言,这里我只需要英文,

sed -ri '/xml:lang/ {/en\_GB/!d}' comps.xml # 如果需要保留其他语言比如中文,在en\_GB后添加"¦zh\_CN"

去除了各国语言的comps.xml大概包含2000多行,现在我们要进一步处理,删除除了id为core的group的其他节点,删除完后comps.xml大概只剩下73行.现在我们要把我们自己添加的包顶一个group节点,在紧跟id为core的group节点(也就是</group>后面)添加如下内容:

&lt;group&gt;

&lt;id&gt;useful&lt;/id&gt;

&lt;name&gt;Useful&lt;/name&gt;

&lt;name xml:lang="en\_GB"&gt;Useful&lt;/name&gt;

&lt;description&gt;Useful tools for administartor &lt;/description&gt;

&lt;description xml:lang="en\_GB"&gt;Useful tools for administartor&lt;/description&gt;

&lt;default&gt;true&lt;/default&gt;

&lt;uservisible&gt;false&lt;/uservisible&gt;

&lt;packagelist&gt;

&lt;packagereq type="default"&gt;setuptool&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;lszrz&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;wget&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;kernel-headers&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;libgomp&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;cpp&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;glibc-headers&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;glibc-devel&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;gcc&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;make&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;which&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;bzip2&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;groff&lt;/packagereq&gt;

&lt;packagereq type="default"&gt;man&lt;/packagereq&gt;

&lt;/packagelist&gt;

&lt;/group&gt;

上面添加了一个id为useful的group节点,下面把这两个节点放到一个类别里:

&lt;category&gt;

&lt;id&gt;cold&lt;/id&gt;

&lt;name&gt;Cold&lt;/name&gt;

&lt;name xml:lang="en\_GB"&gt;Cold&lt;/name&gt;

&lt;description&gt;Cold Linux&lt;/description&gt;

&lt;description xml:lang="en\_GB"&gt;Cold Linux&lt;/description&gt;

&lt;display\_order&gt;92&lt;/display\_order&gt;

&lt;grouplist&gt;

&lt;groupid&gt;core&lt;/groupid&gt;

&lt;groupid&gt;useful&lt;/groupid&gt;

&lt;/grouplist&gt;

&lt;/category&gt;

然后根据我们的comps.xml创建源:

# 安装所需要的工具

yum -y install createrepo anaconda anaconda-runtime

# 创建源

createrepo -g /usr/cold/repodata/comps.xml /usr/cold/CentOS

# 完成后会有如下提示:

/224 - kudzu-1.2.57.1.24-1.el5.centos.i386.rpm

Saving Primary metadata

Saving file lists metadata

Saving other metadata

由于我们编辑了comps.xml,所以comps.xml的sha值会改变,这样就会导致跟repomd.xml中所记录的不同,安装的时候会报错:An error occurred umounting the CD. Please make sure you'are not accessing  /mnt/source from the shell on tty2 an the click OK retry. 所以我们更改完comps.xml要计算comps.xml的sha值

# 计算comps.xml的sha值

sha1sum repodata/comps.xml

c1d304cae50f969370a72d95e3cd2f71087fc73a repodata/comps.xml

然后更新到repomd.xml中编辑repodata/repomd.xml找到location href="http://www.linuxzen.com/repodata/comps.xml"/的一个data节点把sha值更新为刚刚计算的

&lt;data type="group"&gt;

&lt;location href="http://www.linuxzen.com/"repodata/comps.xml"/&gt;

&lt;checksum type="sha"&gt;c1d304cae50f969370a72d95e3cd2f71087fc73a&lt;/checksum&gt;

&lt;timestamp&gt;1272586365&lt;/timestamp&gt;

&lt;/data&gt;

comps.xml里新加了一个咱们的useful 组,怎么使系统安装我们定义的包呢?怎么自定义安装过程呢?下面将讲解根据anaconda-ks.cfg文件定义安装过程: 首先复制anaconda-ks.cfg到我们的发行版目录:

cp ~/anaconda-ks.cfg /usr/cold/isolinux/ks.cfg # 复制到isolinux下并命名为ks.cfg

cd /usr/cold

# 编辑kickstart脚本

vi isolinux/ks.cfg

install # 定义安装

cdrom # 从光盘安装

lang en\_US.UTF-8 # 安装语言为英文

keyboard us # 定义键盘布局

rootpw --iscrypted $1$OKNHES6P$tPdz9HxIp6.QUvulqxNwa. # 定义root密码(你安装的时候提供的密码)

firewall --disable # 禁用防火墙

authconfig --enableshadow --enablemd5 # 使用md5加密

selinux --disabled # 禁用selinux

timezone --utc Asia/Chongqing # 定义时区

bootloader --location=mbr --driveorder=sda # 在一块硬盘上安装mbr

# The following is the partition information you requested

# Note that any partitions you deleted are not expressed

# here so unless you clear all partitions first, this is

# not guaranteed to work

# 如果想安装过程中手动分区就把下面几行注释掉

clearpart --linux --drives=sda # 格式化sda

part / --fstype ext3 --size=1 --grow --maxsize=4096 # 创建/分区大小为4个G

part /usr --fstype ext3 --size=1 --grow # 创建/usr分区,大小为剩余空间

part swap --size=1 --grow --maxsize=512 # 创建swap大小为512M

%packages # 定义安装时安装的包

@core # 最小化安装的包

@useful # 自定义的包

然后就要修改配置文件使安装时使用ks.cfg的配置来安装,修改isolinux.cfg:

vi isolinux/isolinux.cfg

default linux # 默认启动的label

prompt 1

timeout 10 # 等待时间为1秒钟

display boot.msg

F1 boot.msg

F2 options.msg

F3 general.msg

F4 param.msg

F5 rescue.msg

label linux

kernel vmlinuz

append ks=cdrom:/isolinux/ks.cfg initrd=initrd.img text #修改默认linux的label,ks使用自定义的ks,并文本启动

label text

kernel vmlinuz

append initrd=initrd.img text

label ks

kernel vmlinuz

append ks initrd=initrd.img

label local

localboot 1

label memtest86

kernel memtest

append -

到这里配置就基本完成,下面就是制作iso镜像进行安装测试:

# 首先安装工具

yum -y install mkisofs

# 创建iso镜像:

cd /usr/cold

mkisofs -R -J -T -r -l -d -allow-multidot -allow-leading-dots -no-bak -o /usr/ColdOS-0.1-i386.iso -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table .

然后就可以把镜像下载下来进行安装测试了...下一篇就会介绍如何把之前一篇文章介绍的cfengine集成到自己的发行版

"