

如何在CentOS上安装和使用Wine

转载：<http://ghoulch.xninja.org/2016/08/09/install-and-use-wine-on-centos/>

最近想要把工作环境从Windows切换到Linux系统，但公司的IM软件（豆芽）只有Windows和MAC的版本，很多常用工具也只有Windows的版本，贸然切换环境的话会给工作带来很多不便，于是便想到使用Wine安装Windows应用软件。

Wine是“Wine Is Not an Emulator”的缩写，这表明Wine不是任何一种类型的模拟器。Wine不会模拟任何硬件环境，因此它的性能要比虚拟机和容器要好得多。实际上，Wine是一个系统调用的翻译层，负责将应用程序的Windows系统调用转换为Linux系统调用。这种方式会有一定的性能损失，但实际上较新版本的Windows要兼容较老的应用程序，也会在系统中引入类似的系统调用翻译层，所以性能损失几乎可以忽略不计。

虽然目前最流行的桌面版Linux系统是Ubuntu，但是本人平时使用CentOS较多，因此本文只会描述如何在虚拟环境下的CentOS系统中安装和使用Wine，过几天会再写一篇如何在Ubuntu系统中安装和使用Wine的文章。

一、环境描述

1. 虚拟机配置

- CPU：单核 2.4GHz
- 内存：DDR3 1333 2GB
- 硬盘：120GB
- IP地址：10.24.16.149
- 操作系统：CentOS 6.6 x86_64 Desktop

2. 编译环境

- GCC：4.4.7 20120313
- YUM源：官方Base源 + EPEL源 + RepoForge源

3. Wine

- 版本：wine-1.9.15.tar.bz2
- 安装方式：编译安装（本文内容）

二、添加YUM源

1. 添加EPEL源

在Shell中运行以下命令，添加EPEL源：

```
1. rpm -ivh http://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
```

1. 添加RepoForge源

在Shell中运行以下命令，添加RepoForge源：

```
1. rpm -ivh http://apt.sw.be/redhat/el6/en/x86_64/rpmsforge/RPMS/rpmsforge-release-0.5.3-1.el6.rf.x86_64.rpm
```

三、安装编译环境

在Shell中运行以下命令，安装GCC编译环境：

1. yum update -y
2. yum groupinstall -y "Development Tools"

四、安装依赖包

本文会同时编译和安装32位和64位的Wine，因此必须安装32位和64位的依赖包。

在Shell中运行以下命令，安装64位的依赖包：

```
1. yum install -y libX11-devel freetype-devel zlib-devel libxcb-devel libXcursor-devel libXi-devel libXext-devel libXxf86vm-devel libXrandr-devel libXinerama-devel libXcomposite-devel libGLU-devel mesa-libOSMesa-devel libpcap-devel hal-devel libv4l-devel pulseaudio-libs-devel gsm-devel lcms2-devel alsa-lib-devel audiofile-devel cups-devel dbus-devel fontconfig-devel giflib-devel lcms-devel libICE-devel libjpeg-turbo-devel libpng-devel libSM-devel libusb-devel libXau-devel libxml2-devel libXrender-devel libxslt-devel libXt-devel libXv-devel mesa-libGL-devel mesa-libGLU-devel ncurses-devel openldap-devel openssl-devel pkgconfig sane-backends-devel xorg-x11-proto-devel prelink fontforge flex bison openal-soft-devel samba-winbind
```

在Shell中运行以下命令，安装32位的依赖包：

```
1.yum install -y audiofile-devel.i686 dbus-devel.i686 fontconfig-devel.i686 freetype.i686 freetype-devel.i686 giflib-devel.i686 lcms-  
devel.i686 libICE-devel.i686 libjpeg-turbo-devel.i686 libpng-devel.i686 libSM-devel.i686 libusb-devel.i686 libX11-devel.i686 libXau-  
devel.i686 libXcomposite-devel.i686 libXcursor-devel.i686 libXext-devel.i686 libXi-devel.i686 libXinerama-devel.i686 libxml2-  
devel.i686 libXrandr-devel.i686 libXrender-devel.i686 libxslt-devel.i686 libXt-devel.i686 libXv-devel.i686 libXxf86vm-devel.i686  
mesa-libGL-devel.i686 mesa-libGLU-devel.i686 ncurses-devel.i686 openldap-devel.i686 openssl-devel.i686 zlib-devel.i686 sane-backends-  
devel.i686 glibc-devel.i686 libstdc++-devel.i686 pulseaudio-libs-devel.i686 gnutls-devel.i686 libgphoto2-devel.i686 opengl-soft-  
devel.i686 isdn4k-utils-devel.i686 gsm-devel.i686 libv4l-devel.i686 cups-devel.i686 libtiff-devel.i686 gstreamer-devel.i686  
gstreamer-plugins-base-devel.i686 gettext-devel.i686 libmpeg123-devel.i686 lcms2-devel.i686 mesa-libOSMesa-devel.i686 libpcap-  
devel.i686 hal-devel.i686
```

五、下载Wine源码包

在Shell中运行以下命令，下载Wine的源码包：

```
1.cd /root/Downloads/  
2.wget http://mirrors.ibiblio.org/wine/source/1.9/wine-1.9.15.tar.bz2
```

六、编译安装Wine

1. 解压缩源码包

在Shell中运行以下命令，解压缩源码包：

```
1.tar jxvf wine-1.9.15.tar.bz2  
2.cd wine-1.9.15
```

2. 编译安装

在Shell中运行以下命令，编译安装Wine（注意，编译耗时较长，两次编译分别需要1小时左右）：

```
1.mkdir build  
2.cd build  
3.mkdir wine64  
4.cd wine64  
5. ../../configure --prefix=/usr/local/Wine --enable-win64  
6.make -j4  
7.cd ..  
8.mkdir wine32  
9.cd wine32  
10. ../../configure --prefix=/usr/local/Wine --with-wine64=../wine64  
11.make -j4  
12.make install  
13.cd ../wine64  
14.make install
```

注意，在执行configure脚本进行编译前配置时，会找不到以下依赖关系（如果哪位读者能够补全这些依赖关系，请回复告知，谢谢！），如下图所示：

• 缺少的32位依赖关系

```
configure: OpenCL 32-bit development files not found, OpenCL won't be supported.  
configure: libgnutls 32-bit development files too old, no bcrypt hash support.  
configure: libgphoto2 32-bit development files not found, digital cameras won't be supported.  
configure: gstreamer-1.0 base plugins 32-bit development files not found, gstreamer support disabled  
configure: OSS sound system found but too old (OSSv4 needed), OSS won't be supported.  
configure: libmpeg123 32-bit development files not found (or too old), mp3 codec won't be supported.  
configure: opengl-soft 32-bit development files not found (or too old), XAudio2 won't be supported
```

• 缺少的64位依赖关系

```
configure: OpenCL 64-bit development files not found, OpenCL won't be supported.  
configure: libgnutls 64-bit development files too old, no bcrypt hash support.  
configure: gstreamer-1.0 base plugins 64-bit development files not found, gstreamer support disabled  
configure: OSS sound system found but too old (OSSv4 needed), OSS won't be supported.  
configure: libcap120 64-bit development files not found, ISDN won't be supported.  
configure: libmpeg123 64-bit development files not found (or too old), mp3 codec won't be supported.  
configure: opengl-soft 64-bit development files not found (or too old), XAudio2 won't be supported
```

3. 添加环境变量

在Shell中运行以下命令，配置全局环境变量：

```
1.vi /etc/profile
```

在上述文件的最后添加一行，内容为：

```
1.export PATH=$PATH:/usr/local/Wine/bin
```

使得全局环境变量生效：

```
1.source /etc/profile
```

4. 安装cabextract

若需要通过Wine安装以cab格式打包的软件，则还需要安装cabextract包：

```
1. yum install -y cabextract
```

5. 安装Mono

Mono是一种开源的、跨平台的.NET框架的实现。Wine可以通过在Windows平台上编译的Mono来运行.NET应用程序。

在Shell中运行以下命令，下载和安装Mono：

```
1. cd /root/Downloads
2. wget http://dl.winehq.org/wine/wine-mono/4.6.3/wine-mono-4.6.3.msi
3. mkdir -p /usr/local/Wine/share/wine/mono
4. mv wine-mono-4.6.3.msi /usr/local/Wine/share/wine/mono/
```

6. 安装Gecko

Wine实现了自己版本的IE浏览器，而这种实现基于Mozilla的Gecko布局引擎。Gecko具有32位和64位两种版本。

在Shell中运行以下命令，下载和安装Gecko：

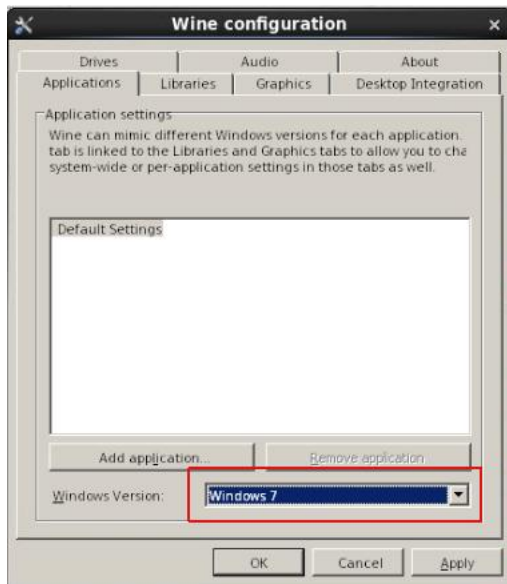
```
1. cd /root/Downloads
2. wget http://dl.winehq.org/wine/wine-gecko/2.47/wine_gecko-2.47-x86.msi
3. wget http://dl.winehq.org/wine/wine-gecko/2.47/wine_gecko-2.47-x86_64.msi
4. mkdir -p /usr/local/Wine/share/wine/gecko
5. mv wine_gecko-2.47-x86.msi /usr/local/Wine/share/wine/gecko/
6. mv wine_gecko-2.47-x86_64.msi /usr/local/Wine/share/wine/gecko/
```

7. 配置Wine

在Shell中运行以下命令，分别配置32位和64位的Wine：

```
1. WINEARCH=win32 WINEPREFIX=~/.wine32 winecfg
2. WINEARCH=win64 WINEPREFIX=~/.wine64 winecfg
```

执行上述两个命令时，会分别创建Wine的32位和64位的工作目录，并且会自动安装上文下载的Wine和Gecko组件，最后会打开Wine的配置窗口。在配置窗口中，最好将两个版本的Wine都配置为Windows 7的兼容性，如下图所示：



8. 安装Winetricks

Winetricks是一个很方便的工具脚本，当需要通过Wine运行某些程序时，这个脚本便可用于下载和安装各种必需的可再发行的运行时库。这些运行时库可能会使用一些闭源的库文件替换Wine的某些组件。注意，这个脚本下载和安装的软件全部都是32位平台的，即使是在Wine64的环境下，其安装的软件也都是32位的（一般安装在~/.wine64/drive_c/Program Files (x86)/目录中）。在Shell中运行以下命令，下载和安装Winetricks：

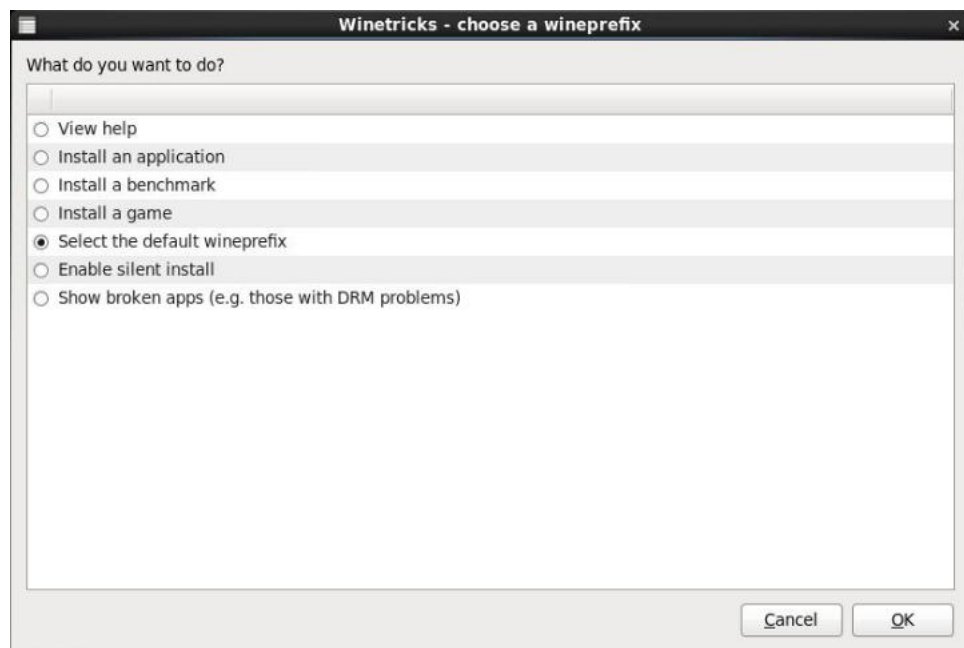
```
1. cd /root/Downloads
2. wget https://raw.githubusercontent.com/Winetricks/winetricks/master/src/winetricks
3. chmod 755 winetricks
4. mv winetricks /usr/local/Wine/bin
```

9. 验证测试

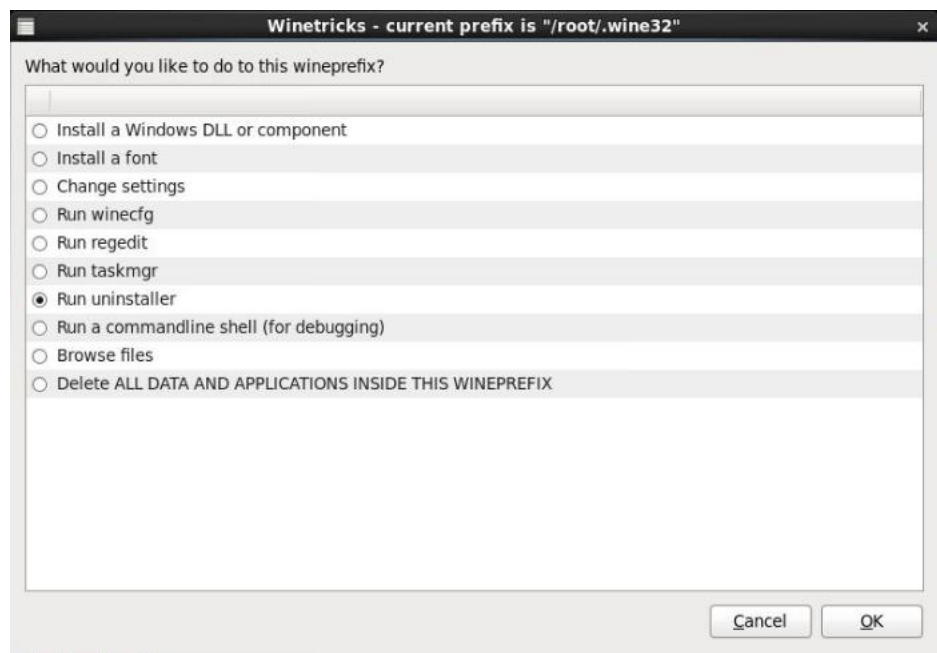
在Shell中运行以下命令，运行winetricks脚本，验证Wine32是否安装成功：

1. WINEPREFIX=~/.wine32 winetricks

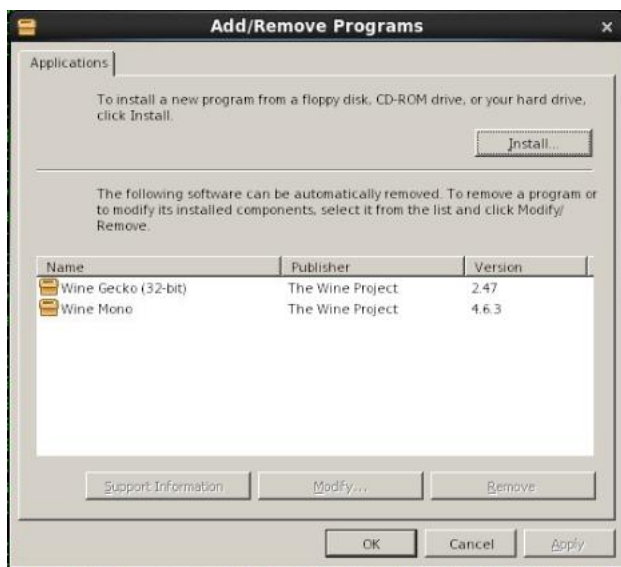
winetricks脚本运行时会打开一个窗口，如下图所示：



在上图中，选择Select the default wineprefix，然后点击OK按钮，进入Wine32的配置窗口，如下图所示：



在上图中，选择Run uninstaller，然后点击OK按钮，进入Wine32的添加/删除程序窗口，如下图所示：



由上图可知，Mono和Gecko已经在Wine32环境中成功安装了。

验证Wine64是否安装成功的步骤和Wine32几乎完全相同，需要在Shell中运行以下命令：

```
1. WINEPREFIX=~/.wine64 winetricks
```

七、安装软件

现在，以豆芽（32位）和灵格斯字典（64位）为例，使用Wine安装这两个软件，并且分别设置它们的快捷键。这样便能分别验证Wine32和Wine64是否能够正常工作。

1. 设置字体

由于Wine默认不支持中文字体，因此需要将宿主机（Windows 7）的微软雅黑和宋体字体文件拷贝至Wine的相关目录中，否则运行中文软件将会出现乱码，如下所示：

- 传输软件：Xftp 5
- 微软雅黑源路径：C:\Windows\Fonts\msyh.ttc
- 宋体源路径：C:\Windows\Fonts\simsun.ttc
- Wine32目标路径：~/.wine32/drive_c/windows/Fonts/
- Wine64目标路径：~/.wine64/drive_c/windows/Fonts/

2. 安装软件

在Shell中执行以下命令，下载豆芽和灵格斯字典的安装程序：

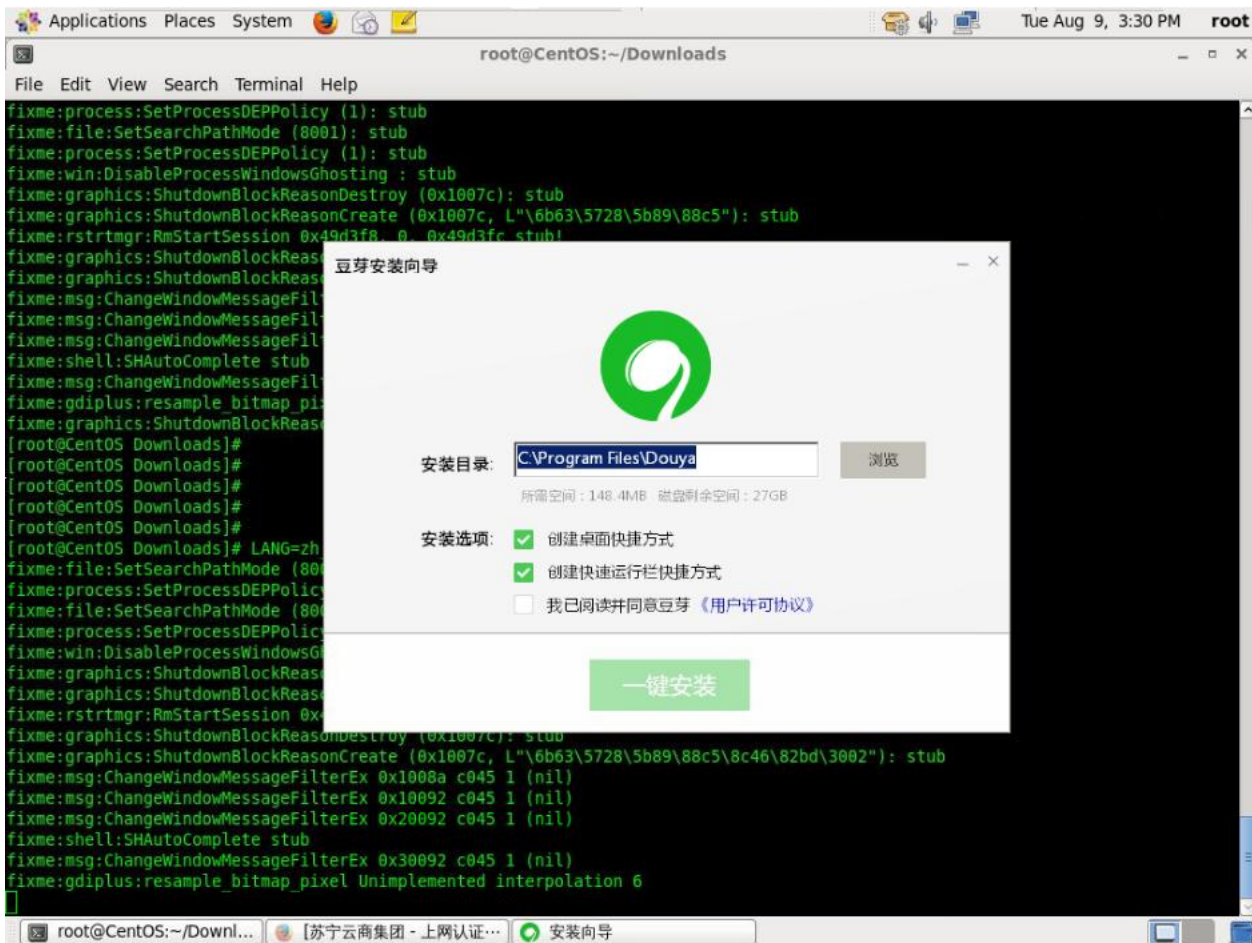
```
1. cd /root/Downloads
2. wget http://im.suning.com/updates/PC/3.5.0.2/SuningIM_PC.exe
3. wget http://www.lingoes.cn/download/lingoes_2.9.2_x64_cn.exe
```

在Shell中分别执行以下两个命令，安装豆芽和灵格斯字典：

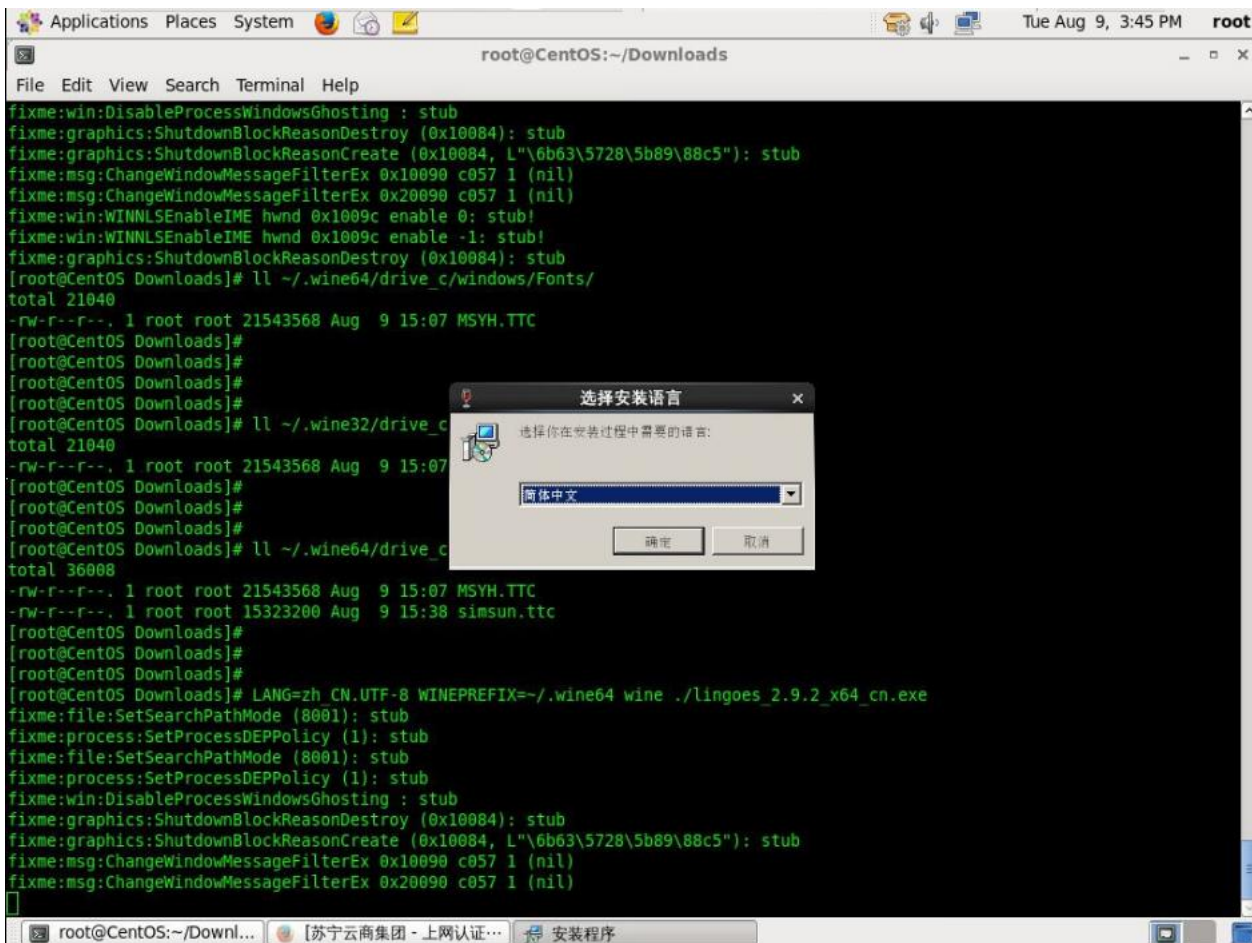
```
1. # 安装豆芽 (Wine32)
2. LANG=zh_CN.UTF-8 WINEPREFIX=~/.wine32 wine ./SuningIM_PC.exe
3. # 安装灵格斯字典 (Wine64)
4. LANG=zh_CN.UTF-8 WINEPREFIX=~/.wine64 wine ./lingoes_2.9.2_x64_cn.exe
```

注意，上述命令中的LANG=zh_CN.UTF-8环境变量表示需要使用UTF-8编码的中文，否则运行中文程序时便会出现乱码。

豆芽的安装窗口如下图所示（安装过程不再赘述），可以看出使用了微软雅黑字体：



灵格斯字典的安装窗口如下图所示（安装过程不再赘述），可以看出使用了宋体字体：



3. 设置快捷键

Windows的应用程序通常具有很多快捷键，可以大大提高操作效率，但是在Wine环境中安装的应用程序，默认却不能使用快捷键。此时，需要安装xdotool和xbindkeys，前者负责将按键转发给指定的应用程序窗口，后者负责将按键绑定至指定的Shell命令。在Shell中安装xdotool和xbindkeys：

```
1. yum install -y xdotool xbindkeys
```

豆芽的常用快捷键是ctrl+alt+z，这个快捷键可以最小化豆芽窗口，或者将豆芽窗口呼出至前台。灵格斯的常用快捷键是ctrl+q，这个快捷键可以最小化灵格斯窗口，或者将灵格斯窗口呼出至前台。在Shell中运行以下命令，设置豆芽的快捷键：

```
1. xbindkeys --defaults > /root/.xbindkeysrc
2. vi ~/.xbindkeysrc
```

在这个文件的最后添加如下的内容：

```
1. "xdotool key --window $(xdotool search --name '豆芽2016' | head -n1) ctrl+alt+z"
2. Control+Alt + z
3.
4. "xdotool key --window $(xdotool search --name 'Lingoes 灵格斯' | head -n1) ctrl+q"
5. Control + q
```

上述的两项配置都会将快捷键绑定至xdotool命令，xdotool会根据名称找到相应的应用程序窗口，然后将快捷键转发给这个窗口。保存这个配置文件之后，在Shell中运行以下命令，使得XBindKeys重新载入配置文件：

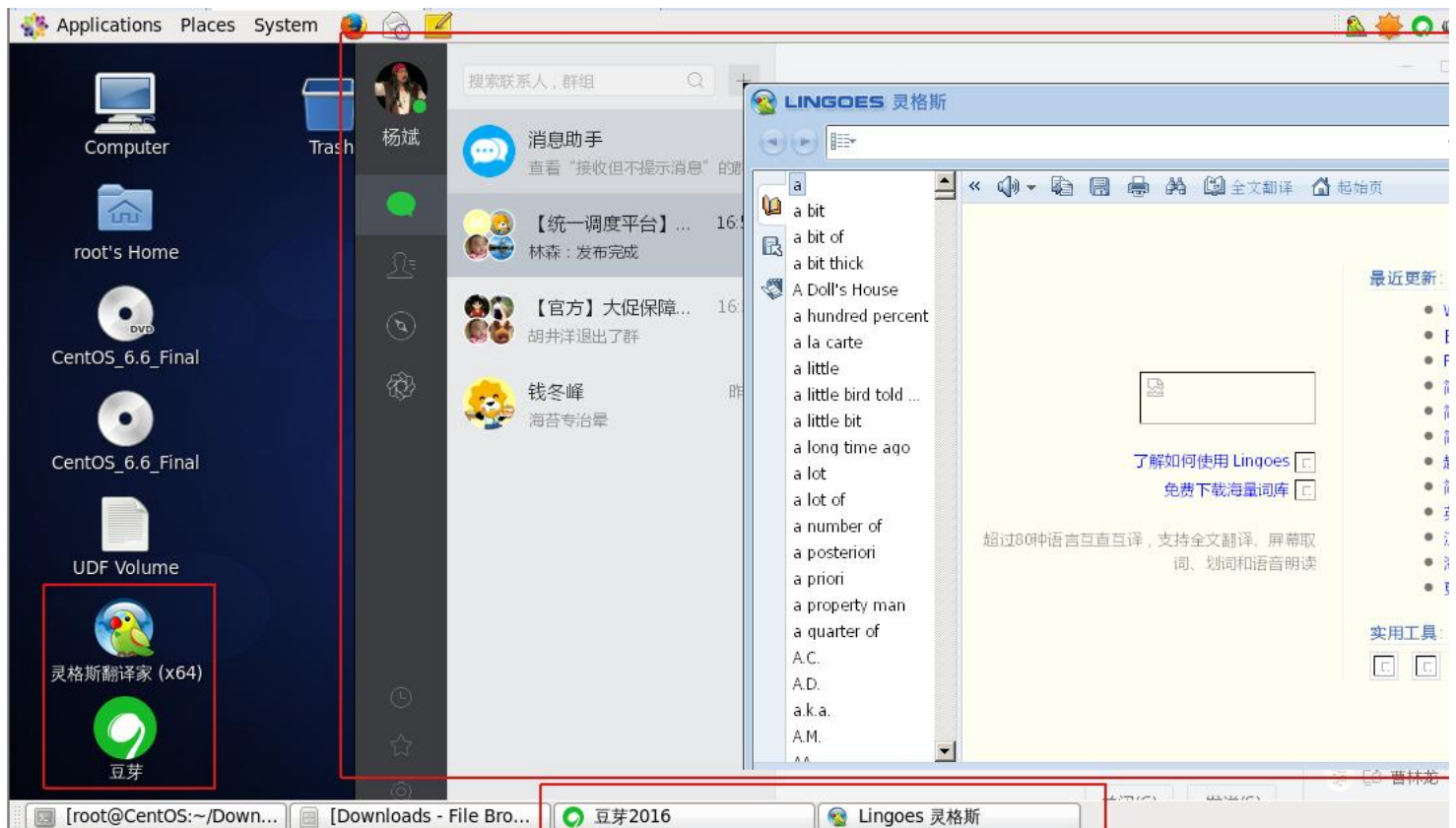
```
1. pkill xbindkeys && xbindkeys &
```

4. 验证测试

在Shell中运行以下命令，确保XBindKeys正在运行：

```
1. ps aux | grep xbindkeys
```

在CentOS的桌面上双击运行豆芽和灵格斯的快捷方式，豆芽还需要输入工号和密码登录，运行后的图形界面如下图所示：



然后，按下ctrl+alt+z和ctrl+q这两个快捷键，便可以分别最小化和呼出豆芽和灵格斯的应用程序窗口了。

八、注意事项

- Wine64现在对于32位的应用程序的兼容性不是很好，所以尽量将32位的软件安装在Wine32中，将64位的软件安装在Wine64中。
- 尽管本文使用root权限执行所有的安装和配置操作，但建议尽量不要使用root账户，否则Windows应用程序获得的权限过大，很有可能对系统造成安全隐患。

- 目前Wine兼容的Windows软件非常多，Wine的官网有一个APPDB数据库，包含了所有兼容软件的信息，APPDB的URL为：

1. <http://appdb.winehq.org/>