

Ubuntu 系统配置手册

Ubuntu Manual

李宝亮 编著

2014 年 6 月 15 日

清华大学出版社 · 北京

第一版前言

Ubuntu 操作系统非常好用，但是在系统安装之初的配置工作非常复杂，于是写此手册备忘。

李宝亮

happybaoliang@gmail.com

2014 年 6 月 15 日

目 录

0.1 装机必备软件及配置

0.1.1 修改默认的系统启动顺序

运行下面的命令并修改默认的启动序号：

```
$ sudo gedit /boot/grub/grub.cfg
```

0.1.2 配置系统更新源

(1) 运行下面的命令找开更新源文件：

```
$ sudo gedit /etc/apt/sources.list
```

(2) 增加下面的内容

```
#163源
deb http://mirrors.163.com/ubuntu/ precise main restricted
deb-src http://mirrors.163.com/ubuntu/ precise main restricted
deb http://mirrors.163.com/ubuntu/ precise-updates main restricted
deb-src http://mirrors.163.com/ubuntu/ precise-updates main restricted
deb http://mirrors.163.com/ubuntu/ precise universe
deb-src http://mirrors.163.com/ubuntu/ precise universe
deb http://mirrors.163.com/ubuntu/ precise-updates universe
deb-src http://mirrors.163.com/ubuntu/ precise-updates universe
deb http://mirrors.163.com/ubuntu/ precise multiverse
deb-src http://mirrors.163.com/ubuntu/ precise multiverse
deb http://mirrors.163.com/ubuntu/ precise-updates multiverse
deb-src http://mirrors.163.com/ubuntu/ precise-updates multiverse
deb http://mirrors.163.com/ubuntu/ precise-backports main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ precise-backports main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ precise-security main restricted
deb-src http://mirrors.163.com/ubuntu/ precise-security main restricted
deb http://mirrors.163.com/ubuntu/ precise-security universe
deb-src http://mirrors.163.com/ubuntu/ precise-security universe
deb http://mirrors.163.com/ubuntu/ precise-security multiverse
deb-src http://mirrors.163.com/ubuntu/ precise-security multiverse
deb http://extras.ubuntu.com/ubuntu precise main
deb-src http://extras.ubuntu.com/ubuntu precise main
#sohu源
deb http://mirrors.sohu.com/ubuntu/ precise main restricted
deb-src http://mirrors.sohu.com/ubuntu/ precise main restricted
deb http://mirrors.sohu.com/ubuntu/ precise-updates main restricted
deb-src http://mirrors.sohu.com/ubuntu/ precise-updates main restricted
deb http://mirrors.sohu.com/ubuntu/ precise universe
deb-src http://mirrors.sohu.com/ubuntu/ precise universe
deb http://mirrors.sohu.com/ubuntu/ precise-updates universe
deb-src http://mirrors.sohu.com/ubuntu/ precise-updates universe
deb http://mirrors.sohu.com/ubuntu/ precise multiverse
deb-src http://mirrors.sohu.com/ubuntu/ precise multiverse
deb http://mirrors.sohu.com/ubuntu/ precise-updates multiverse
deb-src http://mirrors.sohu.com/ubuntu/ precise-updates multiverse
deb http://mirrors.sohu.com/ubuntu/ precise-backports main restricted universe multiverse
```

```
deb-src http://mirrors.sohu.com/ubuntu/ precise-backports main restricted universe multiverse
deb http://mirrors.sohu.com/ubuntu/ precise-security main restricted
deb-src http://mirrors.sohu.com/ubuntu/ precise-security main restricted
deb http://mirrors.sohu.com/ubuntu/ precise-security universe
deb-src http://mirrors.sohu.com/ubuntu/ precise-security universe
deb http://mirrors.sohu.com/ubuntu/ precise-security multiverse
deb-src http://mirrors.sohu.com/ubuntu/ precise-security multiverse
deb http://extras.ubuntu.com/ubuntu precise main
deb-src http://extras.ubuntu.com/ubuntu precise main
```

(3) 然后运行下面的命令更新系统信息:

```
$ sudo apt-get update
```

0.1.3 安装 UCloner

0.1.4 安装基本编译环境

```
$ sudo apt-get install build-essential
```

0.1.5 64 位 Ubuntu 安装 32 位软件

(1) 首先安装 32 位库

```
$ sudo apt-get install ia32-libs*
```

```
$ sudo apt-get install getlibs
```

#如果没有, 到这个地址下<http://frozenfox.freehostia.com/cappy/>

(2) 然后就可以安装 32 位包, 安装时加 `-force-architecture`

(3) 然后用 `getlibs` 安装依赖库, 如

```
$ sudo dpkg -i --force-architecture cairo-dock*.deb
```

```
$ sudo getlibs cairo-dock
```

0.1.6 ICEauthority 无法更新的解决办法

(1) 开机过程中出现如下提示:

- 1.could not update ICEauthority file:/var/lib/gdm/.ICEauthority
- 2./usr/lib/libgconf2-4/gconif-sanity-check-2退出状态为256

(2) 解决办法是:

```
$ sudo dpkg-reconfigure gdm
```

0.1.7 解决 E: Sub-process /usr/bin/dpkg returned an error code (1)

(1) 先将原有的 info 目录移走

```
sudo mv /var/lib/dpkg/info /var/lib/dpkg/info.bak //现将info文件夹更名
```

```
sudo mkdir /var/lib/dpkg/info //再新建一个新的info文件夹
```

```
sudo apt-get update
```

(2) 重装出问题的软件

```
apt-get -f install xxx
```

```
sudo mv /var/lib/dpkg/info/* /var/lib/dpkg/info.bak
```

(3) 执行完上一步操作后会在新的 info 文件夹下生成一些文件，现将这些文件全部移到 info.bak 文件夹下

```
sudo rm -rf /var/lib/dpkg/info //把自己新建的info文件夹删掉
```

```
sudo mv /var/lib/dpkg/info.bak /var/lib/dpkg/info //把以前的info文件夹重新改回名字
```

0.1.8 建立 Ubuntu 本地源

(1) 用新立得查看 apt-mirror，我的机器已经安装，如果没有装，标记安装，应用。

(2) 配置 apt-mirror 软件，

```
$ sudo gedit /etc/apt/mirror.list
```

清空 mirror.list 文件，写入下列内容。

```
##### config #####
set base_path      /media/Software/mirror
set mirror_path    $base_path/mirror
set skel_path      $base_path/skel
set var_path       $base_path/var
set cleanscript    $var_path/clean.sh
set defaultarch    <running host architecture>
set postmirror_script $var_path/postmirror.sh
set run_postmirror 0
set nthreads       20
set _tilde 0
##### end config #####

#64bit deb
deb-amd64 http://mirrors.sohu.com/ubuntu/ precise main restricted universe multi
deb-amd64 http://mirrors.sohu.com/ubuntu/ precise-security main restricted universe
deb-amd64 http://mirrors.sohu.com/ubuntu/ precise-updates main restricted universe
deb-amd64 http://mirrors.sohu.com/ubuntu/ precise-proposed main restricted universe
deb-amd64 http://mirrors.sohu.com/ubuntu/ precise-backports main restricted universe

#32bit deb
deb-i386 http://mirrors.sohu.com/ubuntu/ precise main restricted universe multi
deb-i386 http://mirrors.sohu.com/ubuntu/ precise-security main restricted universe
deb-i386 http://mirrors.sohu.com/ubuntu/ precise-updates main restricted universe
deb-i386 http://mirrors.sohu.com/ubuntu/ precise-proposed main restricted universe
deb-i386 http://mirrors.sohu.com/ubuntu/ precise-backports main restricted universe

#src pack
deb-src http://mirrors.sohu.com/ubuntu/ precise main restricted universe multi
deb-src http://mirrors.sohu.com/ubuntu/ precise-security main restricted universe
deb-src http://mirrors.sohu.com/ubuntu/ precise-updates main restricted universe
deb-src http://mirrors.sohu.com/ubuntu/ precise-proposed main restricted universe
```



```
clean http://mirrors.sohu.com/ubuntu
```

说明：上面内容大意是使用 20 线程，把服务器上 precise 版的 ubuntu 32/64 位软件源的 main、restricted、universe、multiverse 几个部分镜像到本地。本地默认存放软件源的文件夹是/media/Software/mirror，请保证至少有 120G 的剩余空间。

(3) 运行 apt-mirror 元件，开始镜像。

```
$ sudo apt-mirror
```

之后你可以干的就是一段不太短的时间的等待。同步过程可以中断，关闭终端就可以。下次你想再同步的时候重新运行 `sudo apt-mirror` 就可以。

(4) 备份并打开软件源配置文件，

```
$ sudo cp /etc/apt/source.list /etc/apt/source.list_httpsourse
```

```
$ sudo gedit /etc/apt/source.list
```

清空文件，写入下列内容，

```
deb file:///media/Software/mirror/mirror/mirrors.sohu.com/ubuntu/ precise mai
deb file:///media/Software/mirror/mirror/mirrors.sohu.com/ubuntu/ precise-upd
deb file:///media/Software/mirror/mirror/mirrors.sohu.com/ubuntu/ precise uni
deb file:///media/Software/mirror/mirror/mirrors.sohu.com/ubuntu/ precise-upd
deb file:///media/Software/mirror/mirror/mirrors.sohu.com/ubuntu/ precise mul
deb file:///media/Software/mirror/mirror/mirrors.sohu.com/ubuntu/ precise-upd
```

保存，退出。好了，打开新立得，刷新。开始本地高速安装吧。

0.1.9 Ubuntu 无法进入系统的解决办法

ubuntu 12.04 进入恢复模式以后，文件系统是只读的，可以通过下面的方法修改。

```
# mount /dev/sda1 / -o rw,remount
```

如果在登陆界面输入密码后只是闪一下又回到登陆界面的，可以将本用户目录下的.Xauthority 文件删除。

0.1.10 ubuntu 清理系统垃圾

linux 不会产生无用垃圾文件，但是在升级缓存中，linux 不会自动删除这些文件。

(1) 非常有用的清理命令：

```
$ sudo apt-get autoclean
$ sudo apt-get clean
$ sudo apt-get autoremove
```

这三个命令主要清理升级缓存以及无用包的。

(2) 清理 opera firefox 的缓存文件：

```
$ ls ~/.opera/cache4
$ ls ~/.mozilla/firefox/*.default/Cache
```

(3) 清理 Linux 下孤立的包：

图形界面下我们可以用: gtkorphan\$ sudo apt-get install gtkorphan -y
终端命令下我们可以用: deborphan\$ sudo apt-get install deborphan -y

(4) 卸载 tracker

这个东西一般我只要安装 ubuntu 就会第一删掉 tracker 他不仅会产生大量的 cache 文件而且还会影响开机速度。所以在新得利里面删掉就行。

(5) 删除多余的内核：

打开终端敲命令：# dpkg --get-selections|grep linux

有 image 的就是内核文件

删除老的内核文件：

sudo apt-get remove 内核文件名（例如：linux-image-2.6.27-2-generic）

内核删除，释放空间了，应该能释放 130 — 140M 空间。

最后不要忘了看看当前内核：uname -a

0.1.11 安装文件的默认打开方式

可以用 Ubuntu Tweak 中的“文件类型管理器”改

0.1.12 安装字体

0.1.13 调节 CPU 频率

#cpufrequtils——调节CPU频率

#监视cpu频率：右键单击面板，选择“添加到面板”，里面找到“cpu频率范围监视器”。

```
sudo apt-get install cpufrequtils --force-yes -y
```

```
sudo cpufreq-set -g ondemand
```

执行cpufreq-info可看到CPU所支持的模式，大致有如下几种：

powersave，是无论如何都只会保持最低频率的所谓“省电”模式；

userspace，是自定义频率时的模式，这个是当你设定特定频率时自动转变的；

ondemand，一有cpu计算任务，立即达到最大频率，执行完毕立即回到最低频率

conservative，保守模式（默认），一般选择这个，会自动在频率上下限调整；

performance，顾名思义只注重效率，无论如何一直保持以最大频率运行。

#监视CPU温度：

#添加到面板的项的名字叫“Hardware sensors monitor”

```
sudo apt-get install sensors-applet --force-yes -y
```

0.1.14 安装 NTFS 读写支持

安装完后，点击“应用程序”-“系统工具”-“NTFS 写入支持配置程序”输入密码，选中对内部设备的读写支持，和外部设备的读写支持，系统会自动扫描你硬盘上的 NTFS 分区，并重新挂载，这样，你的 NTFS 分区就能在 Feisty 下完美读写了！\$ sudo apt-get install ntfs-config --force-yes -y

0.1.15 安装 dkms 和 wine

```
$ sudo apt-get install dkms wine
```

0.1.16 安装 CCSM 以启动 Ubuntu 桌面特效和 3D 加速效果

```
$ sudo apt-get install compizconfig-settings-manager
```

0.1.17 安装 Ubuntu Restricted Extras

安装 Ubuntu Restricted Extras 软件包后，我们就可以播放 mp3,avi,Flash 等。

```
$ sudo apt-get install ubuntu-restricted-extras
```

0.1.18 解决在 virtualbox 中无法进入共享目录

```
$ sudo ln -f -s /opt/VBoxGuestAdditions-4.3.10/lib/VBoxGuestAdditions/moun
```

0.2 安装必备的应用软件

0.2.1 安装及配置 Chrome 浏览器

(1) 添加 chrome 的源

```
$ sudo vim /etc/apt/menu.lst
```

(2) 添加如下源到文件 menu.lst

```
deb http://ppa.launchpad.net/fta/ppa/ubuntu karmic main
deb-src http://ppa.launchpad.net/fta/ppa/ubuntu karmic main
```

(3) 导入密钥

```
gpg --keyserver keyserver.ubuntu.com --recv 0c713da6
gpg --export --armor 0c713da6 | sudo apt-key add -
```

(4) 更新源

```
$ sudo apt-get update
```

(5) 安装

```
$ sudo apt-get install chromium-browser
```

此时, chrome 浏览器安装完成, 但 flash 仍然无法播放, 需要安装 flashplayer 插件

(6) 如果系统还没安装 flashplayer plugin, 先安装其:

```
$ sudo apt-get install flashplugin-installer
```

(7) 将插件拷贝到 chrome 插件的目录下

```
$ sudo cp /usr/lib/flashplugin-installer/libflashplayer.so /usr/lib/chromium-browser/
```

(8) 设置 chrome 启动时加载插件, 修改快捷方式的启动命令为:

```
$ sudo chromium-browser %U --enable-extensions --enable-plugins
```

(9) 此时, 可以播放 flash 来, 但遇到中文字符是乱码, 解决方案如下:

```
$ sudo vim /etc/fonts/conf.d/49-sansserif.conf
```

将 edit 标签下的 sans-serif 修改为 sans, 保存并退出。

(10) Chrome 下载文件名出现乱码解决办法:

扳手 ——>Settings (设置) ——>ShowAdvance Settings (显示高级设置) ——>Web Content (网页内容) ——>Customizefonts (自定义字体) ——>Encoding (编码) ——> 会发现默认设置的是 ISO-8859-1——> 现在把它设置成 Chinese Simplified(GBK) (中文简体 GBK) ——> 你也可以设置自己喜欢的字体 ——> 问题解决。

0.2.2 安装 Adobe Reader

- (1) 双击下载的安装文件即可完成安装
- (2) 解决 PDF 阅读器的乱码问题

0.2.3 安装 China Union 3G 驱动

0.2.4 安装 EIOffice

0.2.5 安装 ibus

- (1) 只需要打开 console, 输入 `ibus-daemon -x -r -d` 就行
- (2) 另外, 建议将 ibus 设置为默认输入法:

```
$ im-switch -s ibus -z default
```

- (3) 并且默认启动:

系统设置 -高级 -自动启动, 添加程序, 输入 `ibus-daemon -x -r -d`, 确定, 最后注销重登。

- (4) 如果输入法没有输入框, 请检查是否已经安装了 `python-notify` 包:

```
$ sudo apt-get install python-notify
```

然后注销重新进入即可。

0.2.6 安装 INode 软件

- (1) 追加可执行权限

```
$ sudo chmod -R 777 iNodeClient
```

(2) 安装软件

```
$ sudo ./install.sh
```

(3) 对于 ubuntu 12.04 还需要安装以下软件:

```
# ln -
```

```
s /usr/lib/i386-linux-gnu/libtiff.so.4 /usr/lib/i386-linux-gnu/libtiff.so.3
```

```
# apt-get install libjpeg62
```

0.2.7 安装 Picasa

0.2.8 安装 ppstream

0.2.9 安装牛津高阶词典

最近一直在用 ubuntu 系统在看英文 pdf 书籍,可是在 linux 安装的词典,对词条解释的太麻烦,而且有不可发音。因为学习上的需要,所以自己要安装个牛津词典。之所以发表这个博客是想告诉大家如何简单的在 ubuntu 安装词典,自己在安装时候也看了过网上的安装方法,但是比较复杂,而且也不能发音,所以希望这篇博文能给安装 oald8 带来方便

1: 下载词典

牛津高阶词典下载地址: <http://www.verycd.com/topics/2808053/>

2: 提取文件

在 windows 下建立一个新的文件夹 oxford,然后将 oald8.iso 文件用虚拟光驱以文件形式的打开,打开后将里面的文件都复制的新建的 oxford 中。

3: 启动 ubuntu 系统

在 windows 下建立一个新的文件夹 oxford,然后将 oald8.iso 文件用虚拟光驱以文件形式的打开,打开后将里面的文件都复制的新建的 oxford 中。如图??所示:

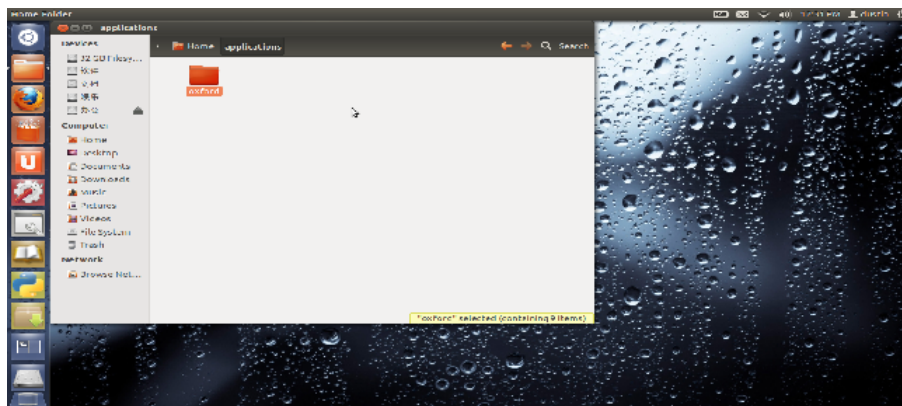


图 1: 安装词典

双击 oxford 文件夹，然后在打开 linux 文件夹，发现里面有一个 setup.sh，右击 setup.sh 选择属性，点击权限设置，选择 Allow excuting file as program。

4: 安装牛津高阶英语词典

Alt+Ctrl+T 打开终端，用 cd 命令打开 setup.sh 所在的文件夹。然后输入 ./setup.sh（安装词典的意思）

5: 安装过程设置

按回车就可以继续下一步

是否同意安装

安装结束

安装的过程可以自己设置安装路径，也可按照软件提供的路径。要是选择第二种话，安装过程一直按‘回车’就可以的。

6: 启动快捷方式设置

并不是安装结束后，这个软件就可以运行了，还需要下面关键两个步骤！

1: 启动设置：

将快捷方式设置成可执行程序，否则在双击这个快捷方式，是不能执行的，所以也就不能启动 oald8 软件了。右击桌面快捷方式，选择属性，权限设置，将这个快捷方式设置成可执行程序。

2: 词典发音设置：

若是没有这步骤，词典是不可发音的，具体过程如下：

command 设置，这个也是很关键一步，若是不设置这个 command，这个软件是不能发音的。同样是右击桌面的快捷方式，选择属性，你会发现有 command 里写着的是：home/dustin/oald8//oald8 将其改为如下命令：padsp '/home/dustin/oald8//oald8'，这个命令要根据安装路径的不同还有用户名字而定，写一个通用的命令吧，自己要根据自己的用户名，还有安装路径自己设置吧 padsp '/home/用户名字/安装路径//oald8'

最后大家就可以使用这个很棒的牛津词典了，perfect！

0.2.10 安装 Dropbox

1. Add Dropbox' s repository key

```
$ sudo apt-key adv --keyserver pgp.mit.edu --recv-keys 5044912E
```

2. Add Dropbox' s repository

```
$ sudo add-apt-repository "deb http://linux.dropbox.com/ubuntu $(lsb_release -
```

3. update and install Dropbox

```
$ sudo apt-get update"
```

```
$ sudo apt-get install dropbox nautilus-dropbox
```

4. Follow the steps When dropbox prompted with the screen

0.2.11 安装 Skype

```
sudo apt-add-repository "deb http://archive.canonical.com/ $(lsb_release -sc)
sudo apt-get update && sudo apt-get install skype
```

```

jerry@jerry-linux ~ $ youtube-dl -F http://www.youtube.com/watch?v=7pKHsPcQot4
[youtube] Setting language
[youtube] 7pKHsPcQot4: Downloading video webpage
[youtube] 7pKHsPcQot4: Downloading video info webpage
[youtube] 7pKHsPcQot4: Extracting video information
[youtube] 7pKHsPcQot4: Encrypted signatures detected.
Available formats:
22 : mp4 [720x1280]
18 : mp4 [360x640]
5 : flv [240x400]
17 : mp4 [144x176]
137 : mp4 [1080p] (DASH Video)
136 : mp4 [720p] (DASH Video)
135 : mp4 [480p] (DASH Video)
134 : mp4 [360p] (DASH Video)
133 : mp4 [240p] (DASH Video)
160 : mp4 [192p] (DASH Video)
140 : mp4 [128k] (DASH Audio)

```

图 2: youtube 可用参数

0.2.12 smaplyer

```
$ sudo apt-get install smplayer subdownloader
```

播放文件时标题栏经常会出现乱码，解决办法：

首选项—> 高级—> 在窗口标题上显示标签信息这一项去掉，smplayer 标题栏显示的就是你的文件名了。

0.2.13 ubuntu one

0.2.14 下载 youtube 视频

(1) 用下列指令安装 youtube-dl

```
$ sudo apt-get install youtube-dl
```

(2) 用浏览器浏览 <http://youtube.com>，搜寻到需要的影片后複製其網址

輸入下列指令先找出能下载的影片格式，我打算下载成 1080p 的高清格式，因此格式代码是 137（後記：137、136 下载的档案都无法播放，改用 22 就可以）：

```
$ youtube-dl -F http://www.youtube.com/watch?v=7pKHsPcQot4
```

 (3) 以 -f 137 下载成 1080p 的格式：

```
$ youtube-dl -f 137 http://www.youtube.com/watch?v=7pKHsPcQot4
```

```
jerry@jerry-linux ~/影片 $ youtube-dl -f 137 http://www.youtube.com/watch?v=7pKHsPcQot4
[youtube] Setting language
[youtube] 7pKHsPcQot4: Downloading video webpage
[youtube] 7pKHsPcQot4: Downloading video info webpage
[youtube] 7pKHsPcQot4: Extracting video information
[download] Resuming download at byte 3673519529
[download] Destination: 2014央视马年春晚 1080P超清 (265分钟终极完整版) -7pKHsPcQot4.mp4
[download] 83.5% of 7.38GiB at 3.96MiB/s ETA 05:13
```

图 3: youtube 使用举例

0.2.15 一个 Linux 脚本搞定常用软件的安装

```
#Bittorrent
sudo apt-get remove bittorrent -y
#安装StarDict翻译词典
sudo apt-get install stardict stardict-common --force-yes -y

sudo apt-get install stardict-cdict-gb stardict-cedict-gb stardict-hanzim star
#安装浏览器Flash插件:
sudo mkdir -p /usr/lib/X11/fonts/Type1
sudo apt-get install flashplugin-nonfree --force-yes -y
#安装FTP工具
sudo apt-get install gftp --force-yes -y
#安装进入终端的右键快捷菜单
sudo apt-get install nautilus-open-terminal --force-yes -y
#alien--把rpm包转换成deb包。使用命令: alien abc.rpm
sudo apt-get install alien --force-yes -y
#安装视频播放软件和相应解码器

sudo apt-get install mplayer mozilla-mplayer libxine-extracodecs w32codecs --f

sudo apt-get install gstreamer0.10-plugins-ugly gstreamer0.10-pitfdll gstreame
```

```
sudo apt-get install im-switch fcitx libapt-pkg-perl --force-yes -y
#切换输入法
```

```
sudo im-switch -s fcitx
```

#阅读CHM文件，chmsee对某些不规范的chm文件支持效好， gnochm支持搜索

```
sudo apt-get install chmsee gnochm --force-yes -y
```

#桌面搜索，功能类似于GOOGLE的那个桌面搜索。安装后在“附件”菜单可找到一个“搜索”项

```
sudo apt-get install beagle --force-yes -y
```

```
sudo apt-get install gnuplot vim automake autoconf gwenview kchmviewer alien na
```

0.2.16 安装压缩解压缩

```
$ sudo apt-get install unace unrar zip unzip p7zip-full p7zip-rar sharutils
```

0.2.17 安装 vlc 视频播放器

(1) 音视频播放:

```
$ sudo apt-get install vlc
```

(2) 安装编解码器:

```
$ sudo apt-get install non-free-codecs libxine1-ffmpeg gxine mencoder libmp
```

0.2.18 安装 WPS 办公软件

(1) 将下载的 WPS 和 symbol font 的 deb 文件双击安装即可。

(2) 将下载的字体文件解压缩到 /.fonts 目录即可解决字体丢失问题

0.2.19 修改系统的默认字体

- (1) 左键点击右上角的齿轮, 找到其中的系统设置选项.
- (2) 点击 universal access, 在 contrast 下拉列表中选择 small 即可.

0.2.20 修改系统输入法

- (1) 按一下 windows 键然后在出现的对话框中输入 keyboard 命令并找到图标点击.
- (2) 点击 OK 后出现的对话框中运行相应的设置即可.
- (3) 添加五笔输入法: `$ sudo apt-get install ibus-table-wubi`
- (4) 点击右上角的齿轮, 选择 startup applications. 在弹出来的对话框的 command 里填写 `/usr/bin/ibus-daemon -d`, Name 和 comment 可乱写.

0.2.21 BT 下载工具

```
$ sudo apt-get install deluge
```

0.2.22 系统配置工具

- (1) 安装编辑器配置工具

```
$ sudo apt-get install myunity gconf-editor dconf-tools
```

- (2) 安装 ubuntu-tweak

```
# apt-add-repository ppa:tualatrix/ppa
```

```
# apt-get update
```

```
# apt-get install ubuntu-tweak
```

0.2.23 安装 QQ 软件

- (1) 安装 QQ 软件

```
$ sudo dpkg -I WineQQ2013.deb
```

(2) 配置 ibus 输入法

在/etc/profile 文件最后添加

```
XMODIFIERS="@im=ibus"
```

```
XIM="ibus"
```

```
GTK_IM_MODULE="xim"
```

```
QT_IM_MODULE="xim"
```

```
ibus-daemon -d -x
```

把/etc/X11/xinit/xinput.d/ibus 文件中的 XIM_ARGS="--xim" 改成

```
XIM_ARGS="-d -x"
```

0.2.24 安装 dock 工具

```
$ sudo apt-get install docky cairo-dock conky screenlets
```

0.2.25 gmail 邮件通知工具

```
$ sudo apt-get install gm-notify
```

0.2.26 Liberooffice 的美化工具

```
$ sudo apt-get install lo-menubar
```

0.2.27 解除 gedit 乱码

(1) apt-get install dconf-tools

(2) 在终端输入 dconf-editor

(3) 找到 org-gnome-gedit-preferences-encodings 在其中 UTF-8 前面添加 'GBK' 就可以了.

也可以命令行下输入:

```
gsettings set org.gnome.gedit.preferences.encodings \
auto-detected "['UTF-8','GB18030','GB2312','GBK','BIG5','CURRENT','UTF-16']"
```

0.2.28 安装 **assaultcube** 游戏软件

- (1) 下载地址 <http://assault.cubers.net/>
- (2) 下载解压后点击 **assaultcube.sh** 就可以玩咯

0.2.29 **PDF Xchange**

0.2.30 安装 **thunderbird** 邮件客户端

```
$ sudo apt-get install thunderbird
```

在 tool-> add-on 中搜索 'new mail attention' 插件并安装即可收到邮件提醒。

0.2.31 安装 **115** 网盘

0.3 重要学术软件安装

0.3.1 安装及配置 matlab

(1) 配置 Matlab 使用的 Java 环境

```
$ sudo update-alternatives --config java
```

```
$ export MATLAB_JAVA=/usr/lib/jvm/java-6-sun-1.6.0.20/jre/
```

(2) 挂载 iso 文件

```
$ sudo mount -o loop Mathworks.Matlab.R2012a.UNIX.iso /mnt
```

(3) 跳转到挂载目录

```
$ cd /mnt
```

(4) 安装

```
$ sudo ./install
```

(5) 安装中选择“不使用 Internet 安装”

(6) 窗口界面默认安装位置为 /usr/local/MATLAB/R2012a

(7) 接受许可协议

(8) 输入安装密钥:

37176-43568-09521-61284-60764-48411-11831-17282-31342-18748-48552-26727-

(9) 安装类型选择“自定义”

(10) 点击“安装”进行安装

(11) 倒入许可协议 (/mnt/crack/lic_s_tandalone.dat)

(12) 等待安装结束

(13) 设置快捷方式

1. 将附件里的 matlab.desktop 文件放在 /usr/share/applications 下, 图片 matlab.png 放在 /usr/share/icons

2. 建立软链接 `sudo ln -s /usr/local/MATLAB/R2012a/bin/matlab /usr/bin/matlab`

(14) 解决中文乱码问题

1. 字体显示美化

进入 Matlab, 从菜单打开: Files->preferences, 打开 Fonts 页, 把右边最下面的复选框 Use antialiasing to smooth desktop fonts 选中, 重启 MATLAB, 字体显示的效果就很好了.

2.MATLAB 使用自带的 Java 运行环境, 根据 CPU 架构的不同, 相对应的字体配置文件路径为:

32 位版本 /usr/local/matlab/sys/java/jre/glnx86/jre/lib/fontconfig.properties

64 位版本 /usr/local/matlab/sys/java/jre/glnxa64/jre/lib/fontconfig.properties

下面以 32 位版本为例

3. 进入字体配置文件目录 `cd /usr/local/MATLAB/R2012a/sys/java/jre/glnx86/jre/lib`

如果 fontconfig.properties 文件不存在, 可以从 fontconfig.properties.src 复制一个

```
sudo cp fontconfig.properties.src fontconfig.properties
```

4. 字体可直接用系统自带的文泉驿

修改 JRE 的字体配置文件, 打开配置文件: `sudo gedit fontconfig.properties`

加入中文字体定义, 在 `version=1` 下面一行输入

```
allfonts.chinese-arphic1=-misc-simsun-medium-r-normal-0-0-0-0-p-0-iso10646-1
```

如果文件已有 `allfonts.chinese-arphic1` 这行, 就直接把它们改成上面那样。

指明中文字体路径, 在 `allfonts.chinese-arphic1` 行后回车另起一行, 输入中文字体文件的完整路径: `filename.-misc-simsun-medium-r-normal-0-0-0-0-p-0-iso10646-1=/usr/share/fonts/truetype/wqy/wqy-microhei.ttc`

5. 修改字体搜索, 在配置文件中查找 `sequence.allfonts`, 如果其后的 `sequence` 开头的行中有: `chinese-arphics1`, 可以略过此步, 否则在其后面加入一行: `sequence.fallback=chinese-arphic1`

(15) 解决/usr/bin/matlab: 1: /usr/local/MATLAB/R2012a/bin/util/oscheck.sh: /lib/libc.so.6: not found

对于 3 2 位系统:

```
$ sudo ln -s /lib/i386-linux-gnu/libc.so.6 /lib/libc.so.6
```

对于 6 4 位系统:

```
$ sudo ln -s /lib/x86_64-linux-gnu/libc.so.6 /lib64/libc.so.6
```

0.3.2 Vmware 安装及配置

0.3.3 Virtualbox 安装及配置

- (1) 双击 deb 文件进行安装
- (2) 选择对应版本的 extension 包进行安装

0.3.4 Cuda 模拟器 Ocelot 的安装及配置

- (1) 建议使用 svn checkout 最新的 trunk

```
$ sudo svn checkout http://gpuocelot.google
```
- (2) 安装各种依赖包和库

```
sudo apt-get install flex bison autoconf automake libtool g++
sudo apt-get install libboost1.40-all-dev
sudo apt-get install libglu1-mesa-dev freeglut3-dev mesa-common-dev
```

上面一行是安装图形库 GL，这个在后面 cuda sdk 编译的时候会用到，反正安着也没坏处。我开始在安装 ocelot 的同时，手贱也去玩 barra，安 barra 得注意几个包，因为需要直接和 cuda sdk 肉搏，常被告知 /usr/bin/ld 找不到 lxxx，直接 `sudo apt-get install libxxx-dev` 就是了。

(3) 这些其实就可以进行正常编译了，当然，如果还想安装 hydrazine 也是可以的。

```
$ svn checkout http://hydrazine.googlecode.com/svn/trunk/ hydrazine
$ cd hydrazine
$ libtoolize; aclocal; autoconf; automake
$ ./configure;make;make check
$ sudo make install
```

- (4) 编译 ocelot

```
$ cd gpuocelot/ocelot
$ libtoolize; aclocal; autoconf; automake
$ ./configure;make
```

建议进行安装，要不在后续进行 regression test 时，它会要你回来安装的

```
$ sudo make install
```

但是这里注意，还需要安装 cuda toolkit, 最好版本匹配，就是说，你要进行 cuda2.2 的 sdk 做测试，就最好安装 2.2 版本的 toolkit。

(5) 安装 cuda 2.2 toolkit

```
$ gedit ~/.bashrc
```

添加：

```
PATH=/usr/local/cuda/bin: $PATH
LD_LIBRARY_PATH=/usr/local/cuda/lib:$LD_LIBRARY_PATH
```

(6) 进行 regression test

```
$ cd ../gpuocelot/tests/cuda2.2
$ libtoolize;aclocal; autoconf; automake
$ ./configure; make; make check
```

在 make check 这一步常会出错，错误大致为：“/usr/include/c++/4.4/new:91: error: ‘operator new’ takes type ‘size_t’ (‘unsigned int’) as first parameter,” 这个错误我在测试 cuda2.3 和 cuda3.2 的时候都出现过，很是无语，gcc.gnu.org 说这是一个 bug, 我不知道如何应对，但是对于 cuda2.2 没有出现这个错误。

(7) 测试安装

```
$ make test
```

0.3.5 Cadence, Allegro 和 MMSIM 的安装

0.3.6 Mathematica 的安装

0.3.7 MPICH2 的安装

(1) 下载 MPICH2 并使用下面命令解压

```
tar xzf mpich2-1.3.2.tar.gz
cd mpich2-1.3.2
```

(2) 选择安装目录/home/<USERNAME>/mpich2-install, <USERNAME> 改为您的用户名, 并确保安装目录为空或不存在。

(3) 将 MPICH2 配置到指定的安装目录:

```
$ sudo mkdir /usr/local/mpich2
$ ./configure --prefix=/usr/local/mpich2 2>&1 | tee c.txt
```

(4) 编译 MPICH2:

```
$ make 2>&1 | tee m.txt
```

(5) 安装 MPICH2 命令:

```
$ make install 2>&1 | tee mi.txt
```

(6) 将安装目录中 bin 子目录添加到你的启动脚本中 (.bashrc for bash):

```
PATH=/usr/local/mpich2/bin:$PATH ; export PATH
```

用以下命令进行命令的检查:

```
which mpicc
which mpiexec
```

这些命令应该显示出你安装目录的 bin 子路经.

(7) 使用下面命令测试安装是否成功 <number> 为要使用的 cpu 数目:

```
$ mpiexec -n <number> ./examples/cpi
```

0.3.8 Modelsim 的安装

(1) 直接运行

```
$ ./install.linux
```

如果权限不够，添加权限

```
$ sudo chmod a+x install.linux
```

由于是图形界面，很 easy。

如果出现下面的错误：

```
Exception in thread "main" java.lang.UnsatisfiedLinkError:
```

```
/home/happy/mgc/install.ixl/JRE/lib/i386/xawt/libmawt.so: libXtst.so.6:  
cannot open shared object file: No such file or directory
```

说明是在 64 位系统上运行了 32 位的 Java，因此还需要安装以下软件包：

```
$ sudo apt-get install libxtst6:i386 libxi6 libxrender1
```

(2) 破解

修改 license.src 和 mentor 文件，将前两行的 SERVER 和 VENDER 的信息按照 Linux 系统进行修改。

```
SERVER HostName MACADDR 27001
```

```
VENDOR mgcld /path-to-modelsim/modeltech/linux/mgcld
```

安装 wine aptitude install wine

运行 wine MentorKG.exe 生成 LIENCE.TXT

添加 license 到 path 里面

```
vim ~/.bashrc
```

```
export LM_LICENSE_FILE=[license存放目录]/LICENSE.TXT
```

```
export PATH=$PATH:[modelsim安装目录]/bin
```

```
source ~/.bashrc
```

在 ubuntu 12.04 上需要修改 modeltech/bin/vsim 文件的 204 行，增加

```
3.[1-9].[0-9]*) vco="linux" ;;
```

将 crack/linux 中的三个文件拷贝到 Modelsim 的安装目录 Modelsim/modeltech/linux/mgls/lib 目录中，然后运行 patch_2013 中的命令：

```
./sfk6 rep -yes -pat -bin /5589E557565381ECD00000008B5508/31C0C357565381EC
```

如果在输出的信息中出现

```
[total hits/matching patterns/non-matching patterns]
```

```
error: failed to read+write: sfk6 - skipping
```

```
[001/1/0] mgcld
```

```
[001/1/0] mgls_asynch
```

```
5 files checked, 2 changed.
```

```
1 errors occurred.
```

说明破解是成功的。如果是 0 files checked, 在 ubuntu 下加 sudo 试试。

(3) 运行

```
$ vsim
```

如果不能创建文件, 考虑权限问题. 如果在 64 位系统中出现下面的错误

```
error while loading shared libraries: libXft.so.2
```

则运行下面的命令

```
$ sudo apt-get install ia32-libs
```

0.3.9 Nero 的安装

(1) 双击 deb 文件进行安装

0.3.10 NS2 的安装

(1) 首先需要安装的是:

```
sudo apt-get install build-essential
```

```
sudo apt-get install tcl8.4 tcl8.4-dev tk8.4 tk8.4-dev
```

```
sduo apt-get install libxmu-dev libxmu-headers
```

(2) 把解压缩后的资料夹移动到你想安装的位置去

```
$ tar xvfz ns-allinone-2.31.tar.gz
$ sudo mv ns-allinone-2.31 /usr/local/NS2
$ sudo chmod 777 -R NS2
$ cd /usr/local/NS2
$ ./install
```

经过一些时间的等待，就会看到他显示一串要你修改.bashrc 或.cshrc 的讯息，依照提示信息加入。如果是.bashrc 的话就会是：

```
export PATH=$PATH:/usr/local/NS2/bin:/usr/local/NS2/tcl8.5.10/unix:/usr/local/NS2/tcl8.5.10/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/NS2/otcl-1.14:/usr/local/NS2/tcl8.5.10/library
export TCL_LIBRARY=$TCL_LIBRARY:/usr/local/NS2/tcl8.5.10/library
```

(3) 接着依照最后几行的讯息，去做验证，例如：

```
$ cd ns-2.31; ./validate
```

当 ns 回车出现

0.3.11 OMNet++ 的安装

(1) 安装必要的软件：

```
$ sudo apt-get install build-essential gcc g++ bison flex perl \
    tcl-dev tk-dev blt libxml2-dev zlib1g-dev openjdk-6-jre \
    doxygen graphviz openmpi-bin libopenmpi-dev libpcap-dev
```

(2) 添加环境变量：

在.bashrc 中增加如下代码：

```
export PATH=/home/happy/omnet++-4.4.1/bin:$PATH
export TCL_LIBRARY=/usr/share/tcltk/tcl8.5
```

(3) 配置、编译和安装:

```
$ ./configure
$ make
```

0.3.12 SystemC-2.2 的安装 (SystemC-2.3 可以直接安装)

1. Either install a Linux system natively or install a Virtual machine (VirtualBox) (recommended).

2. Download systemc-2.2.0.tgz

3. tar -xvf systemc-2.2.0.tgz

4. cd systemc-2.2.0

5. configure --prefix=/usr/local/systemc-2.2

6. sudo mkdir /usr/local/systemc-2.2

7.0. add '#include <cstdlib>' and '#include <string.h>' before '#include "sysc/utls/sc_report.h"' in systemc-2.2.0/src/sysc/utls/sc_utils_ids.cpp

7.1 error: reference 'm_obj' cannot be declared 'mutable' [-fpermissive]

这个错误信息有好几行连续的, 找到 sc_bit_proxies.h, 将那几行中的 mutable 都删去即可。

8. make (ignore the compilation error of example code)

9. sudo make install

10. add newline with expression:

```
SYSTEMC_HOME="/usr/local/systemc-2.2/" in /etc/environment
export SYSTEMC_HOME=/usr/local/systemc-2.2/
```

11. To compile a systemC program simply use this expression:

```
g++ -I. -ISYSTEMC_HOME/include -L. -LSYSTEMC_HOME/lib-linux
-o sim hello.cpp -lsystemc -lm
```


`g++ -I. -ISYSTEMC_HOME/include -L. -LSYSTEMC_HOME/lib-linux64
-o sim hello.cpp -lsystemc -lm` the example code:

```
// All systemc modules should include systemc.h header file
#include "systemc.h"
// Hello_world is module name
SC_MODULE (hello_world) {
    SC_CTOR (hello_world) {
        // Nothing in constructor
    }
    void say_hello() {
        //Print "Hello World" to the console.
        cout << "Hello World.\n";
    }
};
// sc_main in top level function like in C++ main
int sc_main(int argc, char* argv[]) {
    hello_world hello("HELLO");
    // Print the hello world
    hello.say_hello();
    return(0);
}
```

0.3.13 SoCLib 的安装

0.3.14 TeX Live 2011 安装步骤

第一步：准备

我是使用光盘镜像安装 TeX Live 2011, 所以在安装前需要准备如下材料:

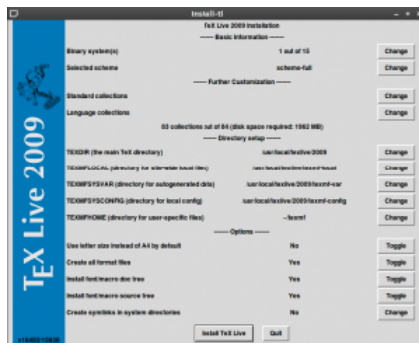


图 4: Texlive 安装界面

TeX Live 2011 光盘镜像，Windows 字体（从 Windows 系统拷贝），Adobe 字体（网络下载）

第二步：开始安装 (采用 GUI 安装方式)

首先安装 perl-tk

```
$ sudo aptitude install perl-tk
```

挂载 TeX Live 2011 镜像

```
sudo mkdir /mnt/iso
```

```
sudo mount -o loop texlive2011.iso /mnt/iso
```

安装

```
$ sudo /mnt/iso/install-tl --gui
```

第三步：安装 TeX Live 2011

信息读取完毕后，弹出一个界面如下： 我把最后一项“自动创建链接”修改外，其他保持原样。点击“安装 TeX Live”。

第四步：配置环境变量

我的默认 shell 是 bash. 一次对照安装指南。打开终端，输入：

```
vi /.profile 或者 vim /etc/bash.bashrc
```

然后把以下代码添加进去 (注意 path 一定要将 texlive 放在前面)

```
PATH=/usr/local/texlive/2011/bin/i386-linux:$PATH; export PATH
```

```
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH; export MANPATH
```

```
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH; export INF
```

接着, 修改/etc/manpath.config

```
$ sudo vi /etc/manpath.config
```

在#set up PATH to MANPATH mapping 下输入 MANPATH_MAP /usr/local/texlive/20

第五步: 配置中文环境和中文字体安装

首先: 创建 Windows 字体目录 WinFonts 和 Adobe 字体目录 AdobeFonts

```
sudo mkdir /usr/share/fonts/WinFonts
```

```
sudo mkdir /usr/share/fonts/AdobeFonts
```

第二复制上述准备的字体到各自目录. 这里需要注意: 修改上面拷贝字体的权限

```
sudo chmod 644 /usr/share/fonts/WinFonts/*
```

```
sudo chmod 644 /usr/share/fonts/AdobeFonts/*
```

如果没有这一步, 会在编译 tex 文件时出现下面类似的错误: SimSun at 17.28pt not 1

第三刷新字体缓存

```
sudo mkfontscale
```

```
sudo mkfontdir
```

```
sudo fc-cache -fsv
```

第六步: 安装中文字体后的配置

首先, 查看系统中安装的中文字体的名字。

```
$ fc-list :lang=zh | sort
```

第二, 查看并根据生成的 fonts 编辑 ctex-xecjk-winfonds.def

```
$ sudo vi /usr/local/texlive/2011/texmf-dist/tex/latex/ctex/fontset/ctex-
```

编辑前 ctex-xecjk-winfonds.def 如下:

% ctex-xecjk-winfonds.def: Windows 的 xeCJK 字体设置,默认为六种中易字体

```
%vim:ft=tex
\setCJKmainfont[BoldFont={SimHei},ItalicFont={[simkai.ttf]]
    {SimSun}
\setCJKsansfont{SimHei}
\setCJKmonofont{[simfang.ttf]}
\setCJKfamilyfont{zhsong}{SimSun}
\setCJKfamilyfont{zhhei}{SimHei}
\setCJKfamilyfont{zhkai}{[simkai.ttf]}
\setCJKfamilyfont{zhfs}{[simfang.ttf]}
\newcommand*\songti{\CJKfamily{zhsong}} % 宋体
\newcommand*\heiti{\CJKfamily{zhhei}} % 黑体
\newcommand*\kaishu{\CJKfamily{zhkai}} % 楷书
\newcommand*\fangsong{\CJKfamily{zhfs}} % 仿宋
\newcommand*\lishu{\CJKfamily{zhli}} % 隶书
\newcommand*\youyuan{\CJKfamily{zhyou}} % 幼圆
\endinput
```

其中带中括号的字体名都是需要修改的,这时需运行

```
$ fc-list :lang=zh-cn
```

来查看系统中的中文字体,记下楷体和仿宋对应的名称,即显示信息中第一个英文在我的系统中楷体是 KaiTi, 仿宋是 FangSong 不过会因为安装的字体版本不同而有所差异.

接下来只要将对应的字体修改即可, 即把 [SIMKAI.TTF] 修改为 KaiTi, 把 [SIMFANG.TTF] 修改为 FangSong, 编辑后 ctex-xecjk-winfonds.def 的内容:

% ctex-xecjk-winfonds.def: Windows 的 xeCJK 字体,默认六种中易字体

```
% vim:ft=tex
```

```

\setCJKmainfont[BoldFont={SimHei},ItalicFont={KaiTi}] {SimSun}
\setCJKsansfont{SimHei}
\setCJKmonofont{FangSong}
\setCJKfamilyfont{zhsong}{SimSun}
\setCJKfamilyfont{zhhei}{SimHei}
\setCJKfamilyfont{zhkai}{KaiTi}
\setCJKfamilyfont{zhfs}{FangSong}
\setCJKfamilyfont{zhli}{LiSu}
\setCJKfamilyfont{zhyou}{YouYuan}
\newcommand*\songti{\CJKfamily{zhsong}} % 宋体
\newcommand*\heiti{\CJKfamily{zhhei}} % 黑体
\newcommand*\kaishu{\CJKfamily{zhkai}} % 楷书
\newcommand*\fangsong{\CJKfamily{zhfs}} % 仿宋
\newcommand*\lishu{\CJKfamily{zhli}} % 隶书
\newcommand*\youyuan{\CJKfamily{zhyou}} % 幼圆
\endinput

```

第三 ctex-xecjk-adobefonts.def 不用改。

第四.sudo tlmgr install xeCJK ctex

第八步：测试

输入一个典型的中文支持例子测试，用 xelatex 或 pdflatex 命令编译

```

\documentclass[UTF8]{ctexart}

\begin{document}
这是我的第一个\TeX{}文件
\end{document}

```

第九步：安装 texmaker

```
$ sudo apt-get install texmaker
```

0.3.15 安装 Cuda

0.3.16 安装 Android 开发环境

0.3.17 安装 Blender 软件

0.3.18 安装 Bochs 软件

0.3.19 安装 Docear

0.3.20 安装 glimpse

0.3.21 安装 gephi

0.3.22 安装 IDA Pro

0.3.23 安装 Jabref

0.3.24 安装 Ruby

```
$ sudo apt-get install ruby irb rdoc
```

0.3.25 安装 lxr

0.3.26 安装 Maple

0.3.27 安装 Tomcat

(1) 下载连接 <http://java.sun.com/javase/downloads/index.jsp>

选择 jdk-6u3-linux-i586.bin 下载 (可以下载最新的, 但要注意一定不能是....i586-rpm.bin, 一般 Ubuntu 下没有 rpm 工具), 将 jdk-6u3-linux-i586.bin 放置于任意目录下如: /home/test

(2) 更改文件权限为可执行、解压:

```
cd /home/test
```

```
chmod u+x jdk-6u3-linux-i586.bin
```

```
sudo ./jdk-6u3-linux-i586.bin yes/no选择yes, 执行完之后边可
```

以在 test 目录下面看到文件夹 jdk1.6.0_03

(3) 设置环境变量

```
sudo vi /etc/profile 在profile文件最后添加
```

```
JAVA_HOME=/home/test/jdk1.6.0_03
```

```
export JRE_HOME=$JAVA_HOME/jre
```

```
export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
```

```
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

保存并关闭

(4) 重启系统 (也可以不用重启系统先 logout 然后 login)

(5) 查看 java 版本

在终端输入 java -version 将会显示 java 版本的相关信息, jdk 安装成功

(6) 下载 tomcat 6 <http://apache.etoak.com/tomcat/tomcat-6/v6.0.20/bin/apache-tomcat-6.0.20.tar.gz>

(7) 解压: tar zxvf apache-tomcat-6.0.20.tar.gz 就会在同一目录下产生 apache-tomcat-6.0.20 文件夹; 可以把 apache-tomcat-6.0.20 拷贝到任意目录;

```
sudo cp -r apache-tomcat-6.0.20 /var/tomcat6
```

```
cd /var/tomcat6/bin
```

```
sudo ./startup.sh 或sudo ./catalina.sh run
```

在浏览器输入 <http://主机地址:8080/>就能看到 tomcat 界面了

(8) tomcat 设置

```
cd /var/tomcat6/conf
```

```
sudo vi/tomcat-users.xml
```

```
<user username="tomcat" password="tomcat" roles="admin,manager"/>
```

增加进入文件内文中保存退出;

(9) 重新启动 tomcat

回到 bin 目录找到 shutdown.sh

运行命令 `sudo ./shutdown.sh`

在运行命令 `sudo ./startup.sh`

等待提示后就说明启动好了。

(10) 在“tomcat 界面” — “manager ” 输入上面的用户名和密码后就能管理站点了

0.3.28 安装 GTK

`apt-get install build-essential` #这将安装gcc/g++/gdb/make 等基本编程工具

`apt-get install gnome-core-devel` #这将安装
`libgtk2.0-dev libglib2.0-dev` 等开发相关的库文件

`apt-get install pkg-config` #用于在编译GTK程序时自动找出头文件及库文件位置

`apt-get install devhelp` #这将安装 devhelp GTK文档查看程序

`apt-get install libglib2.0-doc libgtk2.0-doc` #这将安装
gtk/glib 的API参考手册及其它帮助文档

`apt-get install glade libglade2-dev` #这将安装基于GTK的界面GTK是开发Gnome窗口的c/c++语言图形库。

`apt-get install libgtk2.0*`, gtk+2.0所需的所有文件统统下载安装完毕。

应用程序编译命令: `gcc test.c `pkg-config --cflags --libs gtk+-2.0``, 编译通过, 运行正常。

`pkg-config` 是一个用来管理包的程序, 在控制台输入 `pkg-config --cflags --libs gtk+-2.0`, 可以发现输出的文本包括了 gcc 编译 gtk+2.0 所需要的所有选项 (头文件目录和库文件)。

这里有一点需要注意, `gcc test.c 'pkg-config --cflags --libs gtk+-2.0', pkg-config --cflags --libs gtk+-2.0` 两侧的引号并不是真正的引号, 而是键盘数字件那一行, 最左边的那个字符。如果错用了单引号, `gcc` 无法使用 `pkg-config --cflags --libs gtk+-2.0` 产生的文本作为编译选项。

0.3.29 安装 PHP

安装平台基于 Ubuntu 9.04. 使用 `apt-get` 简单安装. 在安装之前你要准备好源. 还有安装库 `g++ vim ssh links` 因为你要用到这些工具.

(1) Install Tools

```
$ sudo apt-get install g++ vim links ssh
```

(2) 安装 MySQL 5.0

```
$ sudo apt-get install mysql-server mysql-client
```

在安装这个过程中会提示让你输入 MySQL 数据库的密码:

New password for the MySQL “root” user: <- yourrootsqlpassword 你的 MYSQL 密码

Repeat password for the MySQL “root” user: <- yourrootsqlpassword 你的 MYSQL 密码

(3) 安装 Nginx

```
$ sudo apt-get install nginx
```

启动 nginx:

```
$ sudo /etc/init.d/nginx start
```

在 IE 浏览器输入你的 IP 地址:`http://myip`

```
$ sudo links ls.ptubuntu.com
```

看到 `welcome to nginx` 说明你已安装上了 `nginx` 了. 接下来我们要来配置它. 设置启动系统时会自动启动它.

```
$ sudo update-rc.d nginx defaults
```

提示: `System startup links for /etc/init.d/nginx already exist.`

(4) 安装 PHP5

```
$ sudo apt-get install php5-cgi php5-mysql php5-curl php5-gd php5-idn php-p
```

接下来要配置 `php.ini` 这个文件, 在做一些配置文件之前最好你要做一个备份.

```
root@ptUbuntu:~# cd /etc/php5/cgi/
root@ptUbuntu:/etc/php5/cgi# ls
conf.d  php.ini
root@ptUbuntu:/etc/php5/cgi# cp php.ini php.ini.bak
root@ptUbuntu:/etc/php5/cgi# vi php.ini
在php.ini这个文件里添加下一行
cgi.fix_pathinfo = 1
```

(5) 安装 lighttpd

```
$ sudo apt-get install lighttpd
```

安装完接下来要移除它的自动启动程序让它不自动启动.

```
$sudo update-rc.d -f lighttpd remove
```

```
Removing any system startup links for /etc/init.d/lighttpd ...
/etc/rc0.d/K09lighttpd
/etc/rc1.d/K09lighttpd
/etc/rc2.d/S91lighttpd
/etc/rc3.d/S91lighttpd
/etc/rc4.d/S91lighttpd
/etc/rc5.d/S91lighttpd
/etc/rc6.d/K09lighttpd
```

开启 PHP FastCGI 设置听的端口 9000 上运行的本地用户和 `www-data`, 运行下面程序:

```
$ sudo /usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u www-data -g www-data -f /
显示 spawn-fcgi.c.197: child spawned successfully: PID: 29470
```

修改 rc.local 这个文件. 先备份一个.

```
sudo cp /etc/rc.local .
```

```
sudo vi /etc/rc.local
```

添加

```
/usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u www-data -g www-data -f /usr/bin/php
```

配置 sites-available/default

```
sudo cp default default.bak
```

```
sudo vi default
```

```
# You may add here your
# server {
#     ...
# }
# statements for each of your virtual hosts
server {
    listen    80;
    server_name  ls.ptUbuntu.com localhost;
    access_log  /var/log/nginx/localhost.access.log;
    location / {
        root    /var/www/nginx-default;
        index   index.php index.html index.htm;
    }
    location /doc {
        root    /usr/share;
        autoindex on;
        allow 127.0.0.1;
        deny all;
    }
}
```

```
location /images {
    root    /usr/share;
    autoindex on;
}

#error_page 404 /404.html;
# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root    /var/www/nginx-default;
}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80
#
#location ~ /\.php$ {
#    proxy_pass http://127.0.0.1;
#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
location ~ /\.php$ {
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;

    fastcgi_param SCRIPT_FILENAME /var/www/nginx-default$fastcgi_script_name;
    include fastcgi_params;
}

#
#location ~ /\.php$ {
#    fastcgi_pass 127.0.0.1:9000;
#    fastcgi_index index.php;
```

```
#fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
    #includefastcgi_params;
    #}

# deny access to .htaccess files, if Apache' s document root
# concurs with nginx' s one
#
#location ~ /\.ht {
#deny all;
#}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#listen 8000;
#listen somename:8080;
#server_name somename alias another.alias;
#location / {
#root html;
#index index.html index.htm;
#}
#}

# HTTPS server
#
#server {
#listen 443;
#server_name localhost;
#ssl on;
```

```

#ssl_certificate cert.pem;
#ssl_certificate_key cert.key;
#ssl_session_timeout 5m;
#ssl_protocols SSLv2 SSLv3 TLSv1;

#ssl_ciphers ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;
#ssl_prefer_server_ciphers on;
#location / {
#root html;
#index index.html index.htm;
#}
#}

```

创建一个info.php页面.

```

#vi /var/www/nginx-default/info.php
<?php phpinfo(); ?>

```

重启nginx

```

sudo /etc/init.d/nginx restart
sudo /etc/init.d/lighttpd stop

```

这 lighttpd 要关了. 要不然会网页显示会给跑到这里来. 因为 nginx & lighttpd 两个同时打开也会发生冲突的. 而这里我们只是用到 lighttpd 的插件所以没有必要开启.

(6) 接下来要安装的是支持 PHP mysql

```
wget http://nchc.dl.sourceforge.net/sourceforge/phpmyadmin/phpMyAdmin-2.11.9
```

```
root@ptUbuntu:/usr/local/src#cp phpMyAdmin-2.11.9.5-all-languages.tar.bz2 /va
```

```
root@ptUbuntu:/usr/local/src#cd /var/www/nginx-default/
```

```
root@ptUbuntu:/usr/local/src#tar xvf phpMyAdmin-2.11.9.5-all-languages.tar.bz2
```

```
root@ptUbuntu:/usr/local/src#mv phpMyAdmin-2.11.9.5-all-languages phpmyadmin/
```

```
root@ptUbuntu:/usr/local/src#cd phpmyadmin/
```

接着修改配置文档。

```
root@ptUbuntu:/usr/local/src#cp config.sample.inc.php config.inc.php
```

```
*/
```

```
$cfg['blowfish_secret'] = 'ptUbuntu';      ptubuntu 改为  
你的mysql密码
```

```
/*
```

其他地方也就不用改了就可以使用了。

还有下面的软件包：

```
php5-cgi php5-mysql php5-curl php5-gd php5-idn php-pear php5-imagick php5-  
imap php5-mcrypt php5-memcache php5-mhash php5-ming php5-pspell php5-recode  
php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl
```

0.3.30 安装 QT

Linux 下安装 Qt4 有两大问题，一是环境变量，二是 IDE（集成开发环境）。安装 Qt4 也有两种方法，一种是 apt-get，一种是下载源码包，而后一种方法已经人证实是最有可能不好使的方法。所以我最终采用了 apt-get 的方式。而 apt-get 也有两种方式（这就是 Free OS 之不爽之处）：新立得与命令行。这里强烈建议大家使用命令行方式！因为新立得里面的东西太乱，你很可能下载了一大堆东西却没一个是我们真正需要的，而且下载完成后要自己去配置环境变量。

提到环境变量，我不得不多说两句。这真的是一个可恶的东西！到现在我也没弄明白到底是在/etc/profile 中改还是在/.profile 中改。

关于 IDE，网上有人通过设置 KDevelop 跑起来 Qt，但也不是非常的好使，关键时刻还是有找不到的头文件。QDevelop 是 Qt 的官方 IDE，据说跟 Qt4 配

合得更好一些，所以我选用这个。

利用 apt-get 安装 Qt4 过程如下：

Ubuntu Linux 下配置 Qt4 的步骤（我的 Ubuntu 是 8.04 版）：

- 1、请在你的电脑里或虚拟机里安装好 Ubuntu 8.04 版。
- 2、改源并更新，详细操作请参考 wiki.ubuntu.org.cn 上面的“快速配置指南”。

- 3、请不要按捺不住热切的心情安装任何软件更新。

- 4、启动终端，命令：`sudo apt-get install build-essential`

- 5、等待。

- 6、`sudo apt-get install qt4-dev-tools qt4-doc qt4-qtconfig qt4-demos qt4-designer`

注意在这个版本的软件包中，qt4-dev-tools 包含了 Qt Assistant 及 Qt Linguist 等工具，因此不需要单独安装这两个工具。其它的，qt4-doc 是帮助文档，包含了 Qt 中各个类库的详细说明以及丰富的例子程序，可以使用 Qt Assistant 工具来打开阅读。qt4-qtconfig 是配置 Qt 环境的一个对话框，一般默认就行了，很少有必要去更改。qt4-demos 包含很多可以运行起来的可执行文件以及源代码。qt4-designer 是用来设计 GUI 界面的设计器。

- 7、继续等待并祈祷。

- 8、你要用 QDevelop 的话就 `sudo apt-get install qdevelop` 吧。

- 9、如果你用 QDevelop 的话，就直接启动它，你可能会发现提示“Qt 文件夹不存在”之类的提示，这是因为有些工具还没有被安装，如 qmake, ctags 之类，不要被小红叉吓倒，执行以下语句就可以：`sudo apt-get install libqt4-dev`。

- 10、有可能到这儿还有一个 ctags 的红叉，可以执行：`apt-get install ctags`，他会自动帮你查出来并装上，真是方便。然后环境变量不再提示出错，你可以进入 Qdevelop，尽情地 coding 吧！

附：网上查资料过程中看到的也许以后有用：

- 1、为了连接 MySQL 数据库，需要安装连接 MySQL 的驱动程序：`sudo apt-get install libqt4-sql-mysql`

比起在 Windows 下安装和配置 Qt 的 MySQL 驱动来说，简直太方便了。如果还需要其它的没有默认安装的 Qt 库，可以在命令行输入 `sudo apt-get install`

libqt4- 然后按 `tab` 键自动补全，就会列出所有以 `libqt4-` 开头的软件包。这些都可以使用一个命令搞定，而不需要自己从源码开始编译。在记不准或不知道名字的情况下，使用 `tab` 键列出所有可选的软件包是一个很实用的小技巧。

2、在我的项目中，还需要画一些数据曲线和统计图表等，而第三方的 QWT 库提供了这些功能。同样，只需要一个命令即可完成安装：`sudo apt-get install libqwt5-qt4 libqwt5-qt4-dev`，这时，打开 Qt Designer，就会发现左边的 Widget 列表里面多了“Qwt Widget”这一组。

3、关于集成开发环境我觉得 QDevelop 很不错，它跟 Qt Designer 结合的很好，而且有提示类成员函数的功能。使用 Qdevelop 编写代码和编译、调试，使用 Qt Designer 设计界面，开发效率较高。

0.3.31 Java 安装

(1) 开发 JAVA 程序的 JDK 环境 (如果仅是运行 Java 程序，可用 `sun-java6-jre`)

```
sudo apt-get install sun-java6-jdk -force-yes -y
```

(2) 安装浏览器的 JAVA Plugin

```
sudo apt-get install sun-java6-plugin -force-yes -y
```

0.3.32 安装 MySQL

```
sudo apt-get install mysql-server mysql-client -force-yes -y
```

(1) root 原密码为空，给它加个密码

```
mysqladmin -uroot password 123456
```

(2) 重启动 mysql 服务（此步可省）

```
mysqladmin -uroot -p123456 shutdown
```

```
sudo mysqld&
```

0.3.33 安装绘图工具

//和 Visio 类似的 dia（默认只能在命令行启动）

```
sudo apt-get install dia -force-yes -y
```

//画 UML 图的 umbrello

```
sudo apt-get install umbrello -force-yes -y
```

0.3.34 安装 eclipse

```
sudo apt-get install eclipse
```

0.3.35 安装 vim

```
sudo apt-get install vim
```

0.3.36 安装 Emacs

(1) 安装 emacs

(2) 修改 emacs 配置文件，将下面的内容拷贝到/home 目录下的.emacs.conf 文件中。

0.3.37 视频编辑

```
sudo apt-get install openshot
```

```
sudo apt-get install cheese
```

0.3.38 游戏工具箱

```
$ sudo apt-get install playdeb
```

0.3.39 安装 Python

```
$ sudo apt-get install python python-dev
```

0.3.40 安装版本管理工具

```
$ sudo apt-get install subversion cvs git git-core git-doc git-svn git-email
```

0.3.41 安装 perl

```
$ sudo apt-get install perl
```

0.3.42 安装 GEM5

(1) 将下载的 gem5-stable-aaf017eaa7d.tar 安装包解压，文件夹重命名为 gem5_stable。

(2) 因为 SCons 是用 Python 编写的，所以你必须在使用 SCons 之前安装好 Python (2.7.5)。

(3) 安装 scons (2.1 以上)

```
$ sudo apt-get install scons
```

(4) 安装 swig (2.07 以上)

```
$ sudo apt-get install swig
```

(5) 安装 zlib (1.2.8 以上)

```
$ sudo apt-get install zlib
```

(6) 安装 M4

先将下载的 m4-1.4.17.tar.gz 解压:tar -xzvf m4-1.4.17.tar.gz 解压之后的文件夹 m4-1.4.17 放到 gem5_stable 目录下 cd m4-1.4.17 执行命令:

```
$ sudo ./configure sudo make install
```

(7) 安装 protobuf

将下载的安装包解压后进入源代码目录

```
./configure
```

```
sudo make install
```

(8) 安装 libprotobuf-dev

```
$ sudo apt-get install libprotobuf-dev
```

(9) 安装 libgoogle-perftools-dev

```
$ sudo apt-get install libgoogle-perftools-dev
```

(10) 编译 gem5: cd gem5-stable

```
$ sudo mkdir build
```

指定编译的选项及目标文件, 例如:

```
scons build/ALPHA/gem5.opt
```

如果出现如下错误:

错误: can't find Python.h header in ['/usr/include/python2.7'] 解决: sudo apt-get install python-dev

(11) 测试 SE 模式下的 Hello World

在 gem5 目录下输入命令

```
./build/ALPHA/gem5.opt ./configs/example/se.py -c tests/test-progs/hello/bin/  
gem5 Simulator System. http://gem5.org  
gem5 is copyrighted software; use the --copyright option for details.
```

```
gem5 compiled Nov 22 2013 21:02:57
```

```
gem5 started Nov 22 2013 21:06:19
```

```
gem5 executing on ubuntu
```

```
command line: ./build/ALPHA/gem5.opt ./configs/example/se.py -c tests/test-pro  
/home/hu/gem5-stable/configs/common/CacheConfig.py:48: SyntaxWarning: import  
def config_cache(options, system):
```

```
Global frequency set at 1000000000000 ticks per second
```

```
warn: CoherentBus system.membus has no snooping ports attached!
```

```
0: system.remote_gdb.listener: listening for remote gdb #0 on port 7000
```

```

**** REAL SIMULATION ****
info: Entering event queue @ 0. Starting simulation...
info: Increasing stack size by one page.
Hello world!
hack: be nice to actually delete the event here
Exiting @ tick 3233000 because target called exit()

```

安装成功!

(12) 在 full system 下模式下运行 alpha 编译的测试程序

在 gem5-stable 根目录下创建 dist 目录，并在该目录中创建 alpha 目录，并将下载的 m5-system-2.03.tar.bz2 解压，将其中的 binaries 和 disks 目录放在 dist/alpha 目录中。

修改 GEM5/config/common/SysPath.py 文件：

把 except KeyError: path = ['/dist/m5/system', '/n/poolfs/z/dist/m5/system

修改成 except KeyError: path = ['/dist/m5/system', '/home/happy/gem5-stable/dist/alpha']

然后编译 gem5

```
/gem5-stable $ scons ./build/ALPHA/gem5.opt
```

可以通过 GEM5/m5out/system.terminal 查看启动 linux 内核的 monitor 进程。

运行模拟的 linux 系统

```
./build/ALPHA/gem5.opt ./configs/example/fs.py
```

将看到如下界面

```
gem5 Simulator System. http://gem5.org
```

```
gem5 is copyrighted software; use the --copyright option for details.
```

```
gem5 compiled Jul 13 2013 15:50:46
```

```
gem5 started Jul 13 2013 15:53:18
```

```
gem5 executing on js1-desktop
```

```
command line: ./build/ALPHA/gem5.opt ./configs/example/fs.py
```

```
Global frequency set at 1000000000000 ticks per second

info: kernel located at: /home/wyj2/gem5-stable/dist/binaries/vmlinux
Listening for system connection on port 3456

0: system.tsunami.io.rtc:Real-time clock set to Thu Jan 100:00:00 2009
warn: CoherentBus system.membus has no snooping ports attached!

0: system.remote_gdb.listener: listening for remote gdb #0 on port 7000
**** REAL SIMULATION ****
info: Entering event queue @ 0. Starting simulation...
warn: Prefetch instructions in Alpha do not do anything
warn: Prefetch instructions in Alpha do not do anything

开启另外一个 ssh 界面，使用 M5Term 来与 simulatedsystem 进行交互

~/gem5-stable$cd ./util/term
~/gem5-stable$make
~/gem5-stable$sudo make install
~/gem5-stable$m5term localhost 3456
```

这样就进入了模拟出来的系统：

#ls后就看到如下：

```
# ls
benchmarks  etc          linuxrc      modules      sys          var
bin          iscsi        lost+found  proc         tmp
dev          lib          mnt         sbin         usr
```

5、在模拟系统中运行一个测试程序试试：

```
#cd benchmarks
#ls
```

将看到如下几个测试程序：

```
aio-bench          netperf-bin          surge
micros             pthread_mutex_test
#./pthread_mutex_test2 2
```

运行结果如下：

```
Using 2 threadsfor 2 iters
Counter value is 4
#
```

现在以将 GEM/tests/test-progs/hello/bin/alpha/linux/hello，mount 进模拟的系统为例，讲述如何将编译好的程序 mount 进被模拟的系统。

1、将 hello 文件拷贝到当前路径/gem5-stable\$ cp./tests/test-progs/hello/bin/alpha/linux/hello

2、/gem5-stable\$ sudo mount -o,loop,offset=32256 ./dist/disks/linux-latest.img /mnt

关于偏移量 32256 请参考链接：<http://my.oschina.net/toyandong/blog/65002>

3、显示一下/mnt，可以看到挂载好的操作系统

```
~/gem5-stable$ ls/mnt
benchmarks  dev iscsi  linuxrc    mnt        proc  sys  usr
bin         etc  lib    lost+found modules sbin  tmp  var
```

4、 在使用 linux 的 image 文件之前，应该执行 umount 操作。

```
/gem5-stable$ sudo umount /mnt
```

5、 重新开启模拟的 linux，进入模拟的 linux (参考本文 “运行” 中的 2 和 3)

```
#ls
```

可以看到我们添加的testGem5目录

```
benchmarks  etc          linuxrc      modules      sys          usr
bin         iscsi       lost+found  proc         testGem5     var
dev         lib         mnt         sbin         tmp
```

6、cd testGem5

```
$ sudo ./hello
```

执行结果:

```
Hello world!
```

(13) 在 full system 模式下运行 x86 程序的方法可以参考下面的网页。

<http://blog.csdn.net/wyj7260/article/details/9320113>

0.3.43 安装 Design Compiler

1. 在 linux 的根目录下建立/usr/synopsys 文件夹。
2. 在目录下创立以下的目录结构

```
/usr/synopsys  
|---installer  
|---10.9.3  
|---license  
|---B-2008.09
```

3. 安装 installer。将 install_v2.0.rar.Z 解压到/usr/synopsys/installer 目录即可。
4. 解压 scl.rar 到/usr/synopsys 目录中。
5. 在终端中以管理员账户下运行下面的命令:

```
#cd /usr/synopsys/installer/  
#./installer -gui
```

便可以调出安装界面, 如果运行出错说找不到 setup.sh 文件, 则运行下面的命令安装 csh。\$ sudo apt-get install csh

6. 选中 scl_v10.9.3_linux 文件后点击下一步, 如下图所示: 然后一直 next, 过程中有些选项都不需要管, 直到选中安装目标路径/usr/synopsys/10.9.3。软件包每次安装完以后, 点击 finish 以后都会出现一个对话框, 不管他直接点击 dismiss 就可以了。 7. 以同样的方法把 scl_v10.9.3_common.tar 文件安装到 10.9.3 目录。在安装过程中可能会出现下面的错误:

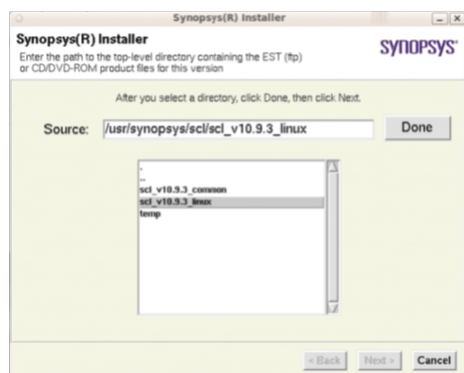


图 5: synopsys 安装界面

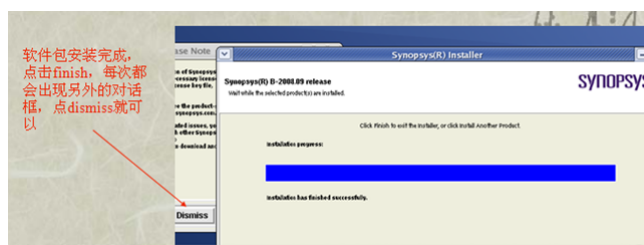


图 6: synopsys 安装界面

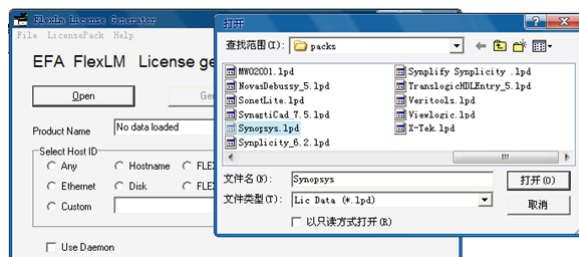


图 7: synopsys 安装界面

cannot find any platform files for product in /usr/synopsys/scl/scl_v10.9.3_common platform files.

不用理它，选择 No，然后在弹出的对话框中选择 next。然后即可完成安装。

8. 将 Design_compiler_2008.09_linux.rar 和 Design_Compiler_2008.09_common.rar 解压到/usr/synopsys 目录下。

9. 以同样的方法将 common 包和 linux 包安装好，这两个包的时候最好分开安装，先安装 linux 包。这两个包都放在/usr/synopsys/B-2008.09 下。

10. 以同样的方法将 vcs-mx_vD-2009[1].12_linux.rar 和 vcs-mx_vD-2009[1].12_common.rar 解压缩并安装好，这两个包的时候最好分开安装，先安装 linux 包。这两个包都放在/usr/synopsys/D-2009.12 下。

11. 在 Windows 环境下制作 DC2008—license 和启动配置文件 （1）解压 dc_license.rar

（2）进入 EFA LicGen 0.4b 文件夹，双击里面的 licGen.exe，打开 packs 中的 synopsys.lpd 文件。

（3）输入 mac 号，然后点击 generate 生成 synopsys.dat 文件。然后点击 save 将这个文件保存。

（4）（5）运行 KeyGen.exe, 在 Secretdata 栏中输出上面的 secret data 码，在 host id 中填入你的 mac 地址。然后点击 generate 命令即可以得到 lincense.dat 文件。

（6）用记事本打开”synopsys.dat”，将第一行修改为: SERVER 主机名 MAC 地址 27000。其中主机名是 Linux 系统下的主机名，可在 Linux 的终端中用”uname

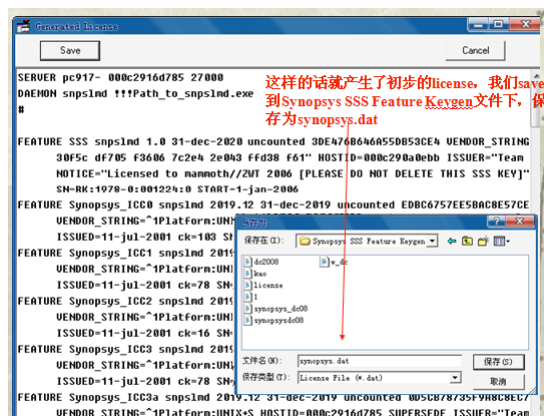


图 8: synopsys 安装界面

-a”命令查看，一般情况下就是 @ 后面的名字；MAC 地址就是网卡地址，后面的 27000 是默认需要的。将”synopsys.dat”第二行改为：

```
DAEMON snpslmd /usr/synopsys/10.9.3/linux/bin/snpslmd
```

下图就是我得到的 license：

(7) 修改 FEATURE SSS 部分

打开之前生成的 license.dat(license

Synopsys SSS Feature Keygen

license.dat)，复制其中的中的 FEATURE SSS 部分，覆盖掉 synopsys.dat 中的 Feature SSS 部分。

(8) 至此 license 的制作完成，将 synopsys.dat 拷贝的/usr/synopsys/license 文件夹下。

(9) 打开用户目录下的.bashrc 文件，在末尾加上如下内容：

```
export SYNOPSYS=/usr/synopsys
export SNPSLMD_LICENSE_FILE=27000@happy-ThinkPad_R400
export LM_LICENSE_FILE=$SYNOPSYS/license/synopsys.dat
```

```
export PATH=$SYNOPSYS/B-2008.09/bin:$SYNOPSYS/D-2009.12/bin:$PATH
```

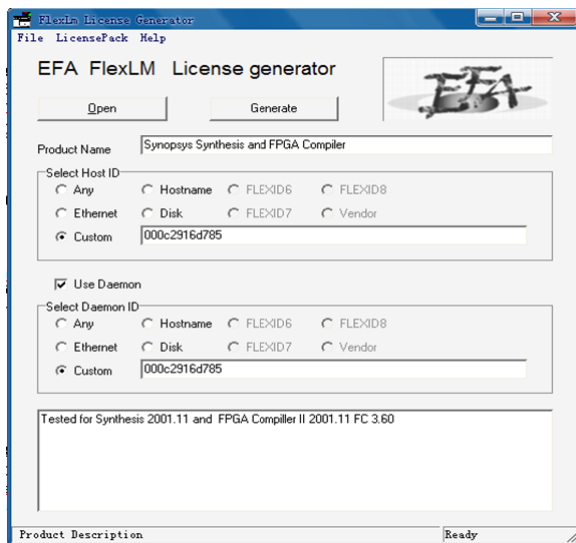


图 9: synopsys 安装界面

```
export VCS_HOME=$SYNOPSYS/D-2009.12
```

```
alias lmli2="/usr/synopsys/10.9.3/linux/bin/lmgrd -c /usr/synopsys/license/sy
```

(10) 更新.bashrc 文件

```
$ sudo source .bashrc
```

(11) 如果 vcs 启动过程中出现下面的错误:

```
bin sh illegal option
```

那么运行下面的命令修改/bin/sh 的链接

```
#rm -f /bin/sh
```

```
#ln -s /bin/bash /bin/sh
```

(15) 如果在运行 vcs 时出现下面的错误:

```
/usr/synopsys/D-2009.12/bin/vcsMsgReport: line 332: /bin/basename: No such
file or directory
```

将 Makefile 文件中的 -full64 选项去掉，同时在.bashrc 文件中增加下面的代码

```
export VCS_ARCH_OVERRIDE=linux
```

即可。

(16) 如果在 64 位系统中编译过程中遇到下面的错误：

/usr/include/features.h:324:26: fatal error: bits/predefs.h: No such file or directory

输入下面的命令解决：

```
$ sudo apt-get install gcc-multilib g++-multilib
```

(17) 在编译过程中出现的链接错误是因为 gcc 版本太高，可以通过下面的命令：

```
#cd /usr/bin
#ls -l gcc*
#mv gcc gcc.bak
#apt-get install gcc-4.4 g++-4.4
#ln -s gcc-4.4 gcc
#ls -l g++*
#mv g++ g++.bak
#ln -s g++-4.4 g++
```

0.3.44 安装 Source Insight

(1) 安装 wine

(2) 安装 source insight.exe

```
$ sudo wine source insight.exe
```

(3) 破解

0.3.45 安装 Codeblocks

(1) 安装基本编译环境

```
$sudo apt-get install build-essential  
$sudo apt-get install gdb
```

(2) 安装 codeblock

```
$sudo apt-get install codeblocks  
$sudo apt-get install codeblocks-dbg  
$sudo apt-get install wxformbuilder
```

(3) 安装 wxWidgets

```
$sudo apt-get install libwxbase2.8  
$sudo apt-get install libwxbase2.8-dev  
$sudo apt-get install libwxgtk2.8-0  
$sudo apt-get install libwxgtk2.8-dev  
$sudo apt-get install libwxgtk2.8-dbg  
$sudo apt-get install wx-common  
$sudo apt-get install wx2.8-headers  
$sudo apt-get install wx2.8-i18n  
($sudo apt-get install wx2.8-examples  
$sudo apt-get install wx2.8-doc
```

(4) 安装完之后，打开 Code::Blocks 就能直接使用了。我没有进行编译器路径的设置，只是把编译器选择为 GCC 而已，使用 `#include` 时要用到的一些头文件还是能找到的。在最后的第一个参考文章中说要进行基本编译运行环境的配置，否则工程编译无法通过。就我们前的学习还用不到工程文件，所以就没有配置。

0.3.46 安装 gtwave

```
$ sudo apt-get install gtwave
```

0.3.47 安装 kscope

- (1) sudo add-apt-repository ppa:fbirlik/kscope
- (2) apt-get update
- (3) sudo apt-get install kscope-trinity

The following package was automatically installed and is no longer required:

```
gir1.2-unique-3.0
```

Use 'apt-get autoremove' to remove them.

The following extra packages will be installed:

```
cscope kdelibs-data-trinity kdelibs4c2a-trinity libarts1c2a-trinity libartsc0
```

Suggested packages:

```
kscope-el fam perl-suid libarts1-akode-trinity
```

The following NEW packages will be installed:

```
cscope kdelibs-data-trinity kdelibs4c2a-trinity kscope-trinitylibarts1c2a-tri
```

0 upgraded, 8 newly installed, 0 to remove and 6 not upgraded.

Need to get 24.1 MB of archives.

After this operation, 61.8 MB of additional disk space will be used.

0.3.48 安装 meld

```
$ sudo apt-get install meld
```