

Linux 系统

——快速装机配置

详细教程

上官羽墨整理

QQ 交流群：274313939

E-mail:wangbobochan@gmail.com

注：本教程以 CentOS 7 为实例机

最后修订：2019 年 08 月 15 日

目 录

第一部分 软件分类介绍.....	1
负二、Linux 基础.....	1
1、联网.....	1
2、设置环境变量.....	1
3、无法绕开的文件权限.....	1
4、多线程编译.....	1
5、编译安装时指定安装目录的影响.....	2
6、挂载磁盘（或网络磁盘）分区.....	3
7、设置开机自动挂载某分区.....	3
8、开机自动启动某程序并后台运行.....	4
9、命令行操作中巧用管道命令“ ”.....	4
10、文本编辑器 vimx 的安装以及配置.....	4
11、配置 ssh 连接长时间保活.....	5
负一、软件包下载地址.....	6
零、Shadowsocks 翻墙.....	6
1、在 Debian 系统中安装：.....	6
2、配置命令行运行：.....	6
3、在 Debian 系统中安装图形界面：.....	7
4、在其他 Linux 发行版中安装：.....	7
一、编程.....	8
1、合集.....	8
2、QT.....	8
3、GTK+安装.....	8
4、代码编辑器 SublimeText.....	9
5、uml 图和 ER 图等的绘图工具.....	12
6、shell 中的 GUI 编程.....	12
二、浏览器（请务必阅读第四条）.....	13
1、firefox:.....	13
2、推荐的优秀火狐浏览器插件（必装）.....	13
3、其它好用的火狐浏览器插件：.....	14
三、种子文件解析下载.....	15
四、下载工具（请务必阅读第二条）.....	15
1、合集.....	15
2、优秀推荐.....	16
3.总结.....	16
五、影视设计/视频编辑.....	16
六、字幕编辑器.....	17
七、音频编辑.....	17
八、网页设计/网页开发.....	18

九、CAD 设计.....	18
十、平面设计/图像设计	18
十一、屏幕录像器.....	18
十二、管理.....	18
十三、命令行浏览器.....	18
十四、磁盘备份工具:.....	18
十五、乐谱编辑软件.....	19
十六、视频字幕提取器.....	19
十七、PDF 文档编辑器	19
十八、摄影.....	19
十九、三维绘图工具.....	19
二十、基于节点的合成工具.....	19
二十一、创作和编辑字体的工具.....	20
二十二、设计缝纫图案的应用.....	20
二十三、图表工具.....	20
二十四、数值分析.....	20
二十五、查看/转换文本文件编码方式的软件	20
二十六、磁盘管理工具.....	20
二十七、视频播放器.....	20
二十八、百度网盘官方 GUI 客户端.....	21
二十九、音乐播放器（网易云音乐）	23
三十、图片查看器.....	23
三十一、虚拟系统安装工具.....	23
三十二、FTP 连接工具.....	23
三十三、压缩解压软件.....	23
三十四、工具栏 docker 工具.....	24
三十五、输入法.....	24
三十六、光盘镜像管理工具.....	24
三十七、Linux 下能实时显示复制进度的软件	24
三十八、即时通讯工具.....	24
三十九、办公软件（WPS-office）	24
四十、服务器.....	25
四十一、远程登陆.....	25
四十二、递归目录列表命令 tree	25
四十三、安装网络管理器.....	25
四十四、安装和开启 SSH 服务.....	26
四十五、安装 Python3	26
四十六、安装微信 wechat.....	26
四十七、远程桌面管理同步软件 TeamViewer.....	27
四十八、轻量级办公软件 calligra 组件	27
四十九、PDF 阅读器	28
五十、CAJ 文件阅读器.....	28
五十一、安装网络加速器 BBR	28
五十二、安装 autossh	29

五十三、安装命令行邮件收发客户端.....	29
五十四、LaTeX 编辑器	30
1、texstudio 的安装.....	30
2、atom 的安装	30
3、其他编辑器的安装.....	30
五十五、开机启动项编辑软件.....	31
五十六、安装指定版本的 GCC.....	31
1、缘起.....	31
2、编译安装高版本的 GCC.....	32
五十七、vimx——可以使用系统粘贴板的 vim	36
五十八、终端模拟器和复用器——DomTerm.....	37
五十九、其他日常软件.....	37
1、DNS 缓存等管理工具 nscd	37
第二部分 redhat 系列系统的安装	38
一、获取 sudo 权限.....	38
二、修改开机启动项，增加 windows 开机引导项	38
三、给程序添加启动图标：	39
四、配置源.....	39
1、备份源.....	39
2、安装 yum 包管理器的基础插件.....	39
3、安装 epel 源.....	39
4、安装阿里源.....	40
5、添加 mosquito 源	40
6、添加 nux-dextop 源	40
7、添加 elrepo 源	40
8、添加 rpmfusion 源.....	41
9、添加 rpmforge 源.....	42
五、安装 dnf 包管理器	42
1、安装 dnf 方法如下	42
2、安装 dnf copr 插件：	42
3、解决 dnf metadata already locked 问题.....	43
六、安装 shadowsocks 图形界面	43
1、方法一.....	43
2、方法二.....	44
七、安装 Python3 和 Python2 共存.....	44
1、查看是否已经安装 Python	44
2、开始编译安装 python3	44
3、修改 yum 配置.....	45
4、关于 pip	45
5、安装另一个插件.....	46
6、替换 pip 的官方源.....	46
7、解决 python 无法找到 tkinter 图形模块的问题.....	47
8、解决 tkinter 中 tk.h 和 libtk.a 版本不匹配的问题.....	48
9、Python 虚拟环境——virtualenv.....	50

八、网易云音乐的安装.....	51
1、首先安装 epel 和 rpmfusion 源.....	51
2、下载网易云音乐的安装包.....	51
3、解压 deb 包.....	51
4、将解压得到的 usr 目录复制到系统中.....	52
5、安装依赖库.....	52
6、升级 glibc.....	53
7、升级 libstdc++.....	53
8、安装解码器.....	53
9、修改库文件权限（解决 abort 问题）.....	54
10、运行网易云音乐.....	54
11、创建快捷方式.....	55
九、搜狗拼音输入法的安装.....	55
1、用别人提供的安装包安装.....	55
2、在命令行直接安装所需基础包.....	55
3、安装图形输入法选择器.....	55
4、结束 ibus 守护进程.....	56
5、关闭 gnome-shell 对键盘的监听.....	56
6、切换输入法为 fcitx.....	56
7、重载 fcitx, 启动搜狗面板.....	56
8、在 Fcitx 配置里面选好搜狗输入法.....	56
9、对有些 WPS 里面无法输入中文的问题的解决.....	56
十、集成开发环境 monodevelop 的安装.....	57
十一、安装微信 wechat.....	57
十二、动画、视频编辑软件 Blender.....	57
第三部分 批量安装一些常用软件.....	59
一、预热：.....	59
二、常用：.....	59
三、系统/编程：.....	59
四、音视频：.....	59
五、CentOS7 没有声音解决办法.....	60
1、中文官网教程.....	60
2、转载自中文官网的教程.....	60
3、具体步骤.....	60
六、配置 CentOS7 睡眠 休眠 关机 电源等行为.....	61
1、官网关于该文件的详解页面（英文）.....	61
2、官方关于该文件的详解页面（中文）.....	61
3、普通设置教程.....	61
4、具体配置过程.....	61
七、其他.....	62
八、附录：我的 fedora 20 系统已经安装的软件.....	62
第四部分 Linux 字符界面日常系统的打造.....	63
一、进入纯字符界面.....	63
1、从字符界面转至字符界面.....	63

2、设置开机自动进入纯字符界面.....	63
3、纯字符界面下的终端切换.....	64
4、纯字符界面使用数字小键盘.....	64
二、中日韩字符系统&输入法——zhcon	64
三、纯字符界面播放音频.....	67
四、纯字符界面播放视频——MPlayer	68
1、安装 MPlayer（不包含皮肤）	68
2、黑白字符组成的大致的图像显示.....	69
3、彩色字符组成的大致图像的显示.....	70
4、完美的视频效果.....	70
5、关于帧缓冲（frame buffer）功能	73
五、纯字符界面调整音量.....	74
六、纯字符界面查看硬件信息.....	75
七、纯字符界面挂载 NAS 盘到本地.....	75
八、纯字符界面的 IM（即时通讯）工具	76
九、纯字符界面的图片查看器 fbi.....	76
十、纯字符界面的 PDF 查看器 fbgs.....	76
十一、纯字符界面下炫酷的屏保.....	77
十二、纯字符界面下的串口工具.....	78
十三、纯字符界面下的屏幕截图工具.....	79
十四、纯字符界面下的网页浏览器.....	79
十五、图片信息（exi）修改及图片旋转工具 exiftran.....	79
十六、纯字符界面的多窗口管理器.....	81
十七、纯字符界面的文件管理器.....	81
十八、纯字符界面的任务管理器.....	82
十九、纯字符界面的视频编辑工具 FFmpeg（命令行工具）	82
1、安装.....	82
2、视频剪切.....	82
3、视频拼接方法一	82
4、视频拼接方法二.....	83
5、关于 ffmpeg 的其他使用教程	83
二十、纯字符界面下的网易云音乐客户端.....	83
二十一、纯字符界面下的百度网盘命令行第三方客户端	86
二十二、纯字符界面下的 csv 文件处理器.....	86
二十三、纯字符界面下的 ftp 工具	87
二十四、纯字符界面下的标记语言文件转换工具 Pandoc.....	88
二十五、纯字符界面下的 word 等文件格式转换工具 LibreOffice	89
二十六、纯字符界面下的 PDF 处理工具.....	90
二十七、纯字符界面下的 Markdown 文件展示工具（相当于播放 PPT）	91
第五部分 Linux 系统运维	93
一、 文本编辑器 vim 和 vimx	93
二、 远程登录和控制工具（一）——ssh 工具	93
三、 远程登录和控制工具（二）——sshpass	94
四、 远程登录和控制工具（三）——telnet.....	94

五、	批处理和交互脚本语言（一）——shell	94
六、	批处理和交互脚本语言（二）——Tcl	94
1.	Tcl 的按安装	95
2.	Tk（Tcl 的扩展图形工具箱）的安装	95
七、	批处理和交互脚本语言（三）——expect	96
第六部分	其他不常用的软件的安装	98
一、	安装 kvm、xen、lxc 虚拟机和 docker:	98
1、	安装 docker 的准备工作	98
2、	安装虚拟机等	98
二、	安装 flash plugin:	99
1、	64 位系统:	99
三、	给 LibreOffice 安装中文字体	99
四、	解决 libstdc++.so.6 版本过旧问题	99
五、	安装 chrome	99
1、	64 位系统:	99
六、	配置 kde 桌面语言中文显示（十分不常用）	100
第七部分	一些常用功能的简化脚本	101
一、	递归修改目录中的子目录和文件名（全角——>半角）	101
二、	简化使用 mplayer 视频播放器	103

第一部分 软件分类介绍

负二、Linux 基础

1、联网

在没有网络管理器的情况下，使用 `network-manager` 可以管理配置有线、无线网络，可以配置 DSL 拨号上网。

命令行使用方法：`nmcli` [加参数命令进行]

图形界面配置方法：`nmtui` 即可进入界面（注：在界面中用空格进行选项的启用或禁用）。

有时刚刚安装系统，会发现双系统找不到 `win` 的开机启动项，可通过 [grub-customizer](#) 解决。

2、设置环境变量

在 Linux 系统的使用过程中，经常涉及到修改或者添加环境变量，因此添加环境变量的方如下（该方法添加环境变量后会对所有用户均生效）：

```
$echo 'export PATH=/usr/local/gcc-8.3.0/bin:$PATH' | sudo tee -ai /etc/bashrc
```

然后注销，重新登录系统就可以使用了。这里的示例是添加“`/usr/local/gcc-8.3.0/bin`”目录到环境变量中（事实上不应该这么做，而是对需要的命令建立软连接到 `/usr/bin` 目录中或者 `/usr/local/bin` 目录中。上面的这条命令中用到了管道命令，[关于管道命令请点击这里了解详情](#)。

3、无法绕开的文件权限

Linux 中很多时候程序无法执行、无法删除、安装失败等等的原因，都是因为没有权限所导致的。因此，关于权限，请参看以下文章，笔者觉得这五篇文章合起来，把 Linux 下文件的权限解释得十分透彻易懂：

<https://blog.csdn.net/lv8549510/article/details/85406215>

<https://blog.csdn.net/u013197629/article/details/73608613>

<https://www.cnblogs.com/garfieldcgf/p/8323489.html>

<https://www.cnblogs.com/webnote/p/5734714.html>

<https://www.cnblogs.com/songgi/p/8890710.html>

4、多线程编译

在从源码编译的时候，在 `make` 命令后加参数 `-j4` 可以开启多线程编译模式，这样就可以

加快编译速度了。即，在 `configure` 完了之后：

```
$make -j4
```

即可。

5、编译安装时指定安装目录的影响

这部分非常重要！

在 Linux 中，编译安装的时候，在 `configure` 步骤中，我们可以指定程序的安装路径，这样该程序的所有文件就都存在与该路径下。例如我安装 [mplayer](#)：

```
configure --prefix=/usr/local/mplayer-1.4 --codecsdir=/usr/local/lib/codecs/ --enable-gui  
--enable-freetype --language=zh_CN
```

这里 `--prefix=/usr/local/mplayer-1.4` 指定了程序将安装在 `/usr/local/mplayer-1.4` 目录下。

又如 [安装 zhcon 中文字符系统&中文输入法](#)：

```
configure --prefix=/usr/local/zhcon-0.2.6
```

这里指定程序将安装在 `/usr/local/zhcon-0.2.6` 目录下。

虽然指定安装目录可以让我们掌握程序安装在哪里，而且还可以让系统更“整洁”，但是有些时候安装完成后相关的可执行文件和系统文件并没有被放在系统变量所指定的目录下。例如我安装 `zhcon` 之后，其安装目录结构如图 1-2-5-1 所示：

```
[zhanghuoding@localhost ~]$ ls -alh /usr/local/zhcon-0.2.6/  
总用量 24K  
drwxr-xr-x.  6 root root 4.0K 7月  2 15:59 .  
drwxr-xr-x. 15 root root 4.0K 7月  2 15:59 ..  
drwxr-xr-x.  2 root root 4.0K 7月  2 16:03 bin  
drwxr-xr-x.  2 root root 4.0K 7月  2 16:03 etc  
drwxr-xr-x.  3 root root 4.0K 7月  2 15:59 lib  
drwxr-xr-x.  3 root root 4.0K 7月  2 15:59 man
```

图表 1-2-5-1 zhcon 安装后的目录结构

这里 `/usr/local/zhcon-0.2.6/bin` 目录下存放的是 `zhcon` 的可执行程序，如果不出现在系统变量所指向的目录中，这个软件就不能运行；

这里 `/usr/local/zhcon-0.2.6/etc` 这个目录存放的是该程序的配置文件，一般是要出现在 `/etc` 这个系统目录中才有效；

这里 `/usr/local/zhcon-0.2.6/lib/zhcon` 这是该程序带过来的库文件，本应该出现在系统目录 `/usr/share` 目录下，以便于依赖该软件部分库的程序可以找到这些库文件并顺利运行；

这里 `/usr/local/zhcon-0.2.6/man` 这个目录存放该程序的指导手册，在 Linux 中可以通过命令行 `$man 程序名` 来查看该软件的相关使用方法，例如 `$man zhcon`，该目录本应该出现在 `/usr/share/man/`；

以上这些文件或目录并没有如期出现在应该出现的地方，因为我们指定了程序安装目录。很多程序还有除了上面所列出来的其他一些目录或文件，也都没有出现在应该出现的地方（部分程序在安装之后会在对应的系统路径下生成相应目录或文件的软链接）。

因此在我们通过编译安装某些软件之后，会发现安装完成了，但是软件却不能运行，系统提示无法找到该命令，这就是因为可执行文件没有出现在系统变量指明的路径中，我们需要建立软连接以实现程序的正常运行。例如我在安装了 `zhcon` 之后，用以下命令为系统指明该文件的安装路径、库文件路径、帮助手册路径等：

```
$sudo ln -s /usr/local/zhcon-0.2.6/bin/zhcon /usr/bin/zhcon
```

```
$sudo ln -s /usr/local/zhcon-0.2.6/etc/zhcon.conf /etc/zhcon.conf
```

```
$sudo ln -s /usr/local/zhcon-0.2.6/lib/zhcon /usr/share/zhcon
```

```
$sudo ln -s /usr/local/zhcon-0.2.6/man/man1/zhcon.1 /usr/share/man/man1/zhcon.1.gz
```

经过上面的步骤，程序 zhcon 就能完美运行了，并且系统帮助手册也是可以正常使用的。

6、挂载磁盘（或网络磁盘）分区

- 挂载普通本地磁盘：

挂载 ext4 分区使用命令（必须使用 sudo 提权）：

```
$sudo mkdir -p /mnt/Amusement
```

```
$sudo chmod 777 /mnt/Amusement
```

```
$sudo mount /dev/sda7 /mnt/Amusement
```

这里是将分区/dev/sda7 挂载到我创建的目录/mnt/Amusement 中。Ext4 文件系统的挂载不需要其他参数。

- 挂载网络磁盘（群晖 NAS）的时候，首先要对群晖进行设置，设置方法见教程：

<https://blog.csdn.net/matengbing/article/details/80434435>

具体过程可以简述为：

- ① 在群晖的控制面板-->共享文件夹新增一个共享文件夹 NFSfile
- ② 授予相应用户权限
- ③ 在 NFS 权限的选项卡设置 NFS 权限，*代表所有主机可访问
- ④ 在控制面板-->文件服务下启用 NFS 文件服务
- ⑤ 在客户端 linux 系统中安装 NFS 客户端工具：

```
$sudo yum install nfs*
```

- ⑥ 在 linux 中检测开启 NFS 服务的群晖主机 IP：

```
$showmount -e 192.168.137.136
```

```
[zhanghuoding@node30 ~]$sudo showmount -e 192.168.137.136
```

```
Export list for 192.168.137.136:
```

```
/volume1/NFSfile *
```

然后就可以从命令行进行挂载了（创建一个目录，然后挂载到该目录）：

```
$sudo mount -t nfs 126.69.172.218:/volume1/video /mnt/video -o proto=tcp -o nolock
```

这里，我的 NAS 是放在内网的，IP 地址是 126.69.172.218，（如果又公网 IP 的话，直接写公网的 IP 就可以在全世界任意一个角落挂载使用了）。我需要挂载的是 NAS 上的根目录下的/volume1/video，将其挂载到本地目录/mnt/video/。

7、设置开机自动挂载某分区

每次都去挂载分区着实难受，因此可以配置使其开机自动挂载。如上第 6 小节中，我设置该分区自动挂载，则首先创建一个脚本文件（下面同颜色的行是一条命令，不可拆分）：

```
$tee -i ./my_auto_mount.sh <<-'EOF'
#!/bin/bash

sudo mkdir -p /mnt/Amusement
sudo chmod -R 777 /mnt/Amusement
sudo mount /dev/sda7 /mnt/Amusement
EOF
```

然后将该脚本复制到指定位置并添加开机启动：

```
$sudo mv ./my_auto_mount.sh /usr/bin/
$sudo sed -i 's@#!/bin/bash@#!/bin/bash\nnohup /usr/bin/my_auto_mount.sh 2>/dev/null
$@g' /etc/rc.local
$sudo sed -i 's@#!/bin/bash@#!/bin/bash\nnohup /usr/bin/my_auto_mount.sh 2>/dev/null
$@g' /etc/rc.d/rc.local
```

8、开机自动启动某程序并后台运行

这里的需求可以参看上一节[设置开机自动挂载某分区](#)，在脚本文件 my_auto_mount.sh 中添加自己想后台开机执行的命令就可以了。

9、命令行操作中巧用管道命令“|”

在命令行安装一些软件，有时候会需要在软件安装过程的中间某一时间进行一些确认行的输入，例如在[安装网易云音乐命令行客户端](#)的时候，需要先安装依赖 PyQt4，而 PyQt4 在进行 configure 得过程中，需要在中途对许可条款进行确认，输入“L”或者“no”或者“yes”分别表示查看许可条款内容、不同意许可条款、同意许可条款，在用脚本进行安装的时候，量大的情况下可能不会注意到这个输入需求，就可能造成安装卡壳，因此使用管道命令即可实现在脚本文件中预先设置好输入内容并到时候自动输入：

```
$echo "yes" | python3 ./configure.py -q /usr/lib64/qt4/bin/qmake
```

像这样就可以实现“yes”命令的自动输入了。

举一反三，我们可以将这种方式运用到 Linux 命令行操作的方方面面，以提高效率。关于 Linux 的管道命令，请参看文章：

<https://www.jianshu.com/p/9c0c2b57cb73>

<https://blog.csdn.net/sun8112133/article/details/81210844>

10、文本编辑器 vimx 的安装以及配置

Vim 是一个应用非常广泛的文本编辑程序，功能也是十分强大，高亮显示都是小菜一碟，其增强版 vimx 更是可以使用系统剪贴板，可以实现在不同终端之间打开的 vimx 编辑器之间

的内容复制共享。

其安装方法如下：

```
$sudo yum install vim-gtk vim-gnome vim-X11 vim* vim-X11*
```

为了使用方便，我们编辑配置文件如下（该配置文件如果不存在，则会新建一个，下面故意插入了一个空行，这个配置对所有用户都生效）：

```
$sudo tee -ai /etc/vimrc <<-"EOF"
```

```
set number
set cursorline
highlight CursorLine    cterm=NONE ctermbg=black ctermfg=green guibg=NONE guifg=NONE
set cursorcolumn
highlight CursorColumn  cterm=NONE ctermbg=black ctermfg=green guibg=NONE guifg=NONE
set autoindent
set smartindent
set laststatus=2
EOF
```

该配置实现光标所在行列的高亮显示、光标所在的行列号的显示，文档整体行号的显示等等。

关于 vimx 编辑器的使用命令详解请查看文章：

<https://www.cnblogs.com/softwaretesting/archive/2011/07/12/2104435.html>

<https://linuxtoy.org/archives/why-emacs-vim-good.html>

关于 vimx 的更多方便的功能配置请查看文章：

<https://linux.cn/article-10039-1.html>

<https://www.cnblogs.com/nklzj/p/6298421.html>

<https://www.cnblogs.com/bkylee/p/8403900.html>

<https://missall.iteye.com/blog/399411>

11、配置 ssh 连接长时间保活

ssh 连接在不使用的情况下，可能四五分钟就会自动断开，这种被动断开很可能会影响到自己的工作上下文，例如你运行了一个耗时较长的脚本，或用 vim 打开了一个文本文件，如果此时远程服务器断开了你的连接，你的工作就被中断了，且无法恢复，这种体验非常糟糕。

这里修改全局配置文件/etc/ssh/sshd_config 来实现 ssh 的自动长时间保活（该修改对所有用户都生效）。

修改方法为：找到文件/etc/ssh/sshd_config 中的 ClientAliveInterval 0、TCPKeepAlive yes 和 ClientAliveCountMax 3：

将这三行的注释符号（“#”）去掉；

将 ClientAliveInterval 对应的 0 改成 60，ClientAliveInterval 指定了服务器端向客户端请求消息的时间间隔（单位是秒），默认是 0，即不发送。ClientAliveInterval 60 表示每分钟发送一次，然后客户端响应，这样就保持长连接了；

ClientAliveCountMax，可以使用默认值 3，也可以自行指定。ClientAliveCountMax

表示服务器发出请求后客户端没有响应的次数达到该值（该值的单位是分钟），就自动断开 ssh 连接，正常情况下，客户端不会不响应；

TCPKeepAlive yes 表示允许系统保持连接的活动状态。

我们在命令行执行以下命令即可：

```
$sudo sed -i "s/#TCPKeepAlive yes/TCPKeepAlive yes/g" /etc/ssh/sshd_config
$sudo sed -i "s/#ClientAliveInterval 0/ClientAliveInterval 60/g" /etc/ssh/sshd_config
$sudo sed -i "s/#ClientAliveCountMax 3/ClientAliveCountMax 30/g" /etc/ssh/sshd_config
```

这里修改之后，是 30 分钟内如果主机没有收到客户机的回信，则会自动断开 ssh 连接。修改完成之后重启 sshd 服务或者重启系统，就可以达到 ssh 在设定时间内保活的目的了。

关于这部分内容可以查看文章：

<https://blog.csdn.net/Drongguang/article/details/78815530>

<https://blog.csdn.net/wbxaut/article/details/80966557>

<https://blog.csdn.net/Jalon2015/article/details/50222051>

负一、软件包下载地址

提供一些 rpm 包的下载网站：

下载地址 1、 <http://rpm.pbone.net/>

零、Shadowsocks 翻墙

在浏览器中，要使用代理设置软件。如火狐浏览器的 autoproxy FoxyProxy 等工具设置就好了。

1、在 Debian 系统中安装：

命令行 `install shadowsocks` 就可以了，然后就是编辑配置文件了。

2、配置命令行运行：

建立配置文件 `vi /etc/shadowsocks.json`

写入以下内容：

```
{
```

```

"server":"0.0.0.0",#####添上服务器地址
"server_ipv6": "::",
"server_port":8388,#####添上服务器的端口
"local_address": "127.0.0.1",
"local_port":1080,
"password":"mypassword",#####服务器的密码
"timeout":300,
"method":"aes-256-cfb",#####数据加密方式，由服务器决定
"obfs":"http_simple",
"fast_open": false,
"workers": 1
}

```

保存即可。然后用如下命令在终端运行：

```
$sslocal -c /etc/shadowsocks.json
```

如果要后台运行，可以用：

```
$nohup sslocal -c /etc/shadowsocks.json
```

然后就可以关闭终端了。

如果要在后台运行的同时输出日志：

```
$nohup sslocal -c /etc/shadowsocks.json >> name.log 2>&1
```

然后关闭终端就可以了。

3、在 Debian 系统中安装图形界面：

```
install shadowsocks
```

```
install qt5-qtbase-devel
```

```
install shadowsocks-qt5
```

4、在其他 Linux 发行版中安装：

[点击此处查看官方安装教程](#)

一、编程

1、合集

anjuta bluefish brasero codeblocks enca emacs monodevelop netbeans gcc
g++ kdevelop kdesdk mysql-server mysql-client libmysqlclient-dev *gdb* gdb valgrind valgrin*
安装完 mysql 后，还需要命令行安装 libmysqlclient-dev 这个东西，否则会在编译的时候提示
mysql/mysql.h: 没有那个文件或目录

valgrind 是一个内存检测工具，用法:\$valgrind ./test 将会检测并输出程序 test 运行中的内存泄露等问题。特别有用的工具。

2、QT

```
sudo apt-get install qt4-dev-tools #开发包
sudo apt-get install qtcreator #IDE
sudo apt-get install qt4-doc #开发帮助文档
sudo apt-get install qt4-qtconfig #配置工具
sudo apt-get install qt4-demos #DEMO 源码 QT
另一种
sudo apt-get install libqt4-dev qt4-designer qt4-doc #配置 qt4 开发环境
sudo apt-get install kde4-devel #配置 kde4 开发环境
sudo aptitude install kdevelop-kde4 #IDE
```

3、GTK+安装

Gtk+有 C++封装版本，名叫 GTKMM，这是一个 C++封装后的 API。切有图形界面设计工具 glade。
安装图形界面设计工具 `install glade glade*`

- 1、安装 gcc/g++/gdb/make 等基本编程工具
\$sudo apt-get install build-essential
- 2、安装 libgtk3.0-dev libglib3.0-dev 等开发相关的库文件
\$sudo apt-get install gnome-core-devel
- 3、用于在编译 GTK 程序时自动找出头文件及库文件位置
\$sudo apt-get install pkg-config
- 4、安装 devhelp GTK 文档查看程序

\$sudo apt-get install devhelp

5、安装 gtk/glib 的 API 参考手册及其它帮助文档

\$sudo apt-get install libglib2.0-doc libgtk2.0-doc

6、安装基于 GTK 的界面 GTK 是开发 Gnome 窗口的 c/c++语言图形库

\$sudo apt-get install glade libglade2-dev 或者 \$sudo apt-get install glade-gnome glade-common glade-doc

7、安装 gtk3.0 或者 将 gtk+3.0 所需的所有文件统统下载安装完毕

\$sudo apt-get install libgtk3-dev 或者 \$sudo apt-get install libgtk3*

8、总：

\$sudo apt-get install gnome-devel gnome-devel-docs

下面安装 gtkmm:

1、安装:

install gtkmm gtkmm* libgtkmm libgtkmm*

2、安装图形界面设计工具:

Install glade glade*

4、代码编辑器 SublimeText

(各个 linux 发行版安装教程 <https://www.tecmint.com/install-sublime-text-editor-in-linux/>)

1、安装过程:

sudo rpm -v --import <https://download.sublimetext.com/sublimehq-rpm-pub.gpg>

sudo yum-config-manager --add-repo

https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo

sudo yum install sublime-text

2、解决 ubuntu 系统中 sublime-text 不能输入中文的问题

原文链接 <http://www.cnblogs.com/wangkongming/p/4302642.html> 或者

<https://www.sinosky.org/linux-sublime-text-fcitx.html>

测试系统: Ubuntu 12.10 (理论上所有 Linux 发行版都通用)

输入法: Fcitx 4.2.6.1

Sublime Text 版本: Sublime Text 3 dev (Build 3026)

首先,保存下述代码为 sublime-imfix.c 文件

/*

sublime-imfix.c

Use LD_PRELOAD to interpose some function to fix sublime input method support for linux.

By Cjacker Huang


```

gcc -shared -o libsublime-imfix.so sublime-imfix.c `pkg-config --libs --cflags gtk+-2.0` -fPIC
LD_PRELOAD=./libsublime-imfix.so subl
*/
#include <gtk/gtk.h>
#include <gdk/gdkx.h>
typedef GdkSegment GdkRegionBox;

struct _GdkRegion
{
    long size;
    long numRects;
    GdkRegionBox *rects;
    GdkRegionBox extents;
};

GtkIMContext *local_context;

void
gdk_region_get_clipbox (const GdkRegion *region,
                        GdkRectangle    *rectangle)
{
    g_return_if_fail (region != NULL);
    g_return_if_fail (rectangle != NULL);

    rectangle->x = region->extents.x1;
    rectangle->y = region->extents.y1;
    rectangle->width = region->extents.x2 - region->extents.x1;
    rectangle->height = region->extents.y2 - region->extents.y1;
    GdkRectangle rect;
    rect.x = rectangle->x;
    rect.y = rectangle->y;
    rect.width = 0;
    rect.height = rectangle->height;
    //The caret width is 2;
    //Maybe sometimes we will make a mistake, but for most of the time, it should be the caret.
    if(rectangle->width == 2 && GTK_IS_IM_CONTEXT(local_context)) {
        gtk_im_context_set_cursor_location(local_context, rectangle);
    }
}

//this is needed, for example, if you input something in file dialog and return back the edit area
//context will lost, so here we set it again.

```

```
static GdkFilterReturn event_filter (GdkXEvent *xevent, GdkEvent *event, gpointer im_context)
{
    XEvent *xev = (XEvent *)xevent;
    if(xev->type == KeyRelease && GTK_IS_IM_CONTEXT(im_context)) {
        GdkWindow * win = g_object_get_data(G_OBJECT(im_context),"window");
        if(GDK_IS_WINDOW(win))
            gtk_im_context_set_client_window(im_context, win);
    }
    return GDK_FILTER_CONTINUE;
}
```

```
void gtk_im_context_set_client_window (GtkIMContext *context,
                                       GdkWindow      *window)
{
    GtkIMContextClass *klass;
    g_return_if_fail (GTK_IS_IM_CONTEXT (context));
    klass = GTK_IM_CONTEXT_GET_CLASS (context);
    if (klass->set_client_window)
        klass->set_client_window (context, window);

    if(!GDK_IS_WINDOW (window))
        return;
    g_object_set_data(G_OBJECT(context),"window",window);
    int width = gdk_window_get_width(window);
    int height = gdk_window_get_height(window);
    if(width != 0 && height !=0) {
        gtk_im_context_focus_in(context);
        local_context = context;
    }
    gdk_window_add_filter (window, event_filter, context);
}
```

然后， 安装 C/C++ 的编译环境和 gtk libgtk2.0-dev
install build-essential libgtk2.0-dev

然后， 编译共享库

```
gcc -shared -o libsublime-imfix.so sublime-imfix.c `pkg-config --libs --cflags gtk+-2.0` -fPIC
```

然后， 设置 LD_PRELOAD 并启动 Sublime Text

```
LD_PRELOAD=./libsublime-imfix.so subl
```

然后， 修改 /usr/share/applications/sublime_text.desktop 为
[Desktop Entry]

[...]

```
Exec=env LD_PRELOAD=/opt/sublime_text/libsublime-imfix.so  
/opt/sublime_text/sublime_text %F  
[...]
```

[Desktop Action Window]

```
[...]  
Exec=env LD_PRELOAD=/opt/sublime_text/libsublime-imfix.so /opt/sublime_text/sublime_text  
-n  
[...]
```

[Desktop Action Document]

```
[...]  
Exec=env LD_PRELOAD=/opt/sublime_text/libsublime-imfix.so /opt/sublime_text/sublime_text  
--command new_file  
[...]
```

然后，把 libsublime-imfix.so 放到 /opt/sublime_text/ 中
sudo cp ./libsublime-imfix.so /opt/sublime_text/

然后，修改 /usr/bin/subl 为
#!/bin/sh
export LD_PRELOAD=/opt/sublime_text/libsublime-imfix.so
exec /opt/sublime_text/sublime_text "\$@"

然后，重启电脑，可以看到 Sublime 能够输入中文了。

5、uml 图和 ER 图等的绘图工具

Umbrello、dia、umlgraph、calligra-flow

Calligra-flow 的安装见[四十八、轻量级办公软件 calligra 组件](#)

dia 比较好用，推荐使用。

```
sudo yum install dia umbrello umlgraph
```

6、shell 中的 GUI 编程

pinentry-gui、ncurses（这是一个库）、dialog

ncurses 库可以被 C 或者 python 程序导入，而用其提供的 API 进行终端的 GUI 编程。在 C 语言中，需要用#include<ncurses.h>，编译时用 gcc -o ./test ./test.c -lncurses 进行编译即可。

Python 中可以用 import 导入 ncurses 库。

dialog 可以直接使用，在 shell 中用相应的命令即可，例如：

```
dialog --title "hello" --backtitle "Dialog" --yesno "is everything  
okay" 20 60
```

可以执行并显示一个询问窗口。

如果期望每一步清晰可控，推荐使用 C 语言或 python 结合 ncurses 进行编程。但如果想要追求简洁，推荐使用 dialog。

```
sudo yum install pinentry-gui ncurses dialog
```

二、浏览器（请务必阅读[第四条](#)）

1、firefox:

安装火狐浏览器，为了满足文件下载需要，一定要安装 flashgot 插件。能够让 Firefox 直接调用外部下载工具下载单个，或批量的网页中的链接，支持主流下载工具。经测试，为了满足下载需要，最好将第四条下载工具中列出的所有下载器 aria2、curl、uget、axel、wget 等全部安装，在具体下载时根据需要启用合适的下载器。经测试，对不同网站、不同文件等，各个下载器速度不一致。对百度网盘提取的直链，uget 可以达到 2M/S 的速度。

一般情况火狐浏览器配合 uget 比较好用，有些情况下 aria2 更好用，极少数情况下 curl 和 axel、wget 好用！

2、推荐的优秀火狐浏览器插件（必装）

还有一款超级好用的多线程下载工具，是一个火狐浏览器的插件，DownThemAll! 多线程下载工具。超级好用。支持下载本页全部链接

PDF Print & Save，一款超级好用的将网页或者待打印内容保存为 PDF 的插件

图片编辑器 PaintMagick 用于照片和图片，在 web 浏览器中进行简单的图片编辑操作

“Google 学术搜索按钮”插件，此扩展程序会添加一个浏览器按钮，以便用户可以从任意网页轻松使用 Google 学术搜索。点击“学术搜索”按钮可执行以下三项操作：

- ① 在网络上或在您的大学图书馆中查找论文全文。在您正在阅览的页面上选择论文标题，然后点击“学术搜索”按钮，即可查找该论文的全文。
- ② 将您的查询从网页搜索转移到学术搜索。按下“学术搜索”按钮即可查看前三项搜索结果；点击弹出窗口左下角的“全屏”图标，可查看所有搜索结果。
- ③ 使用常用的引用格式对参考文献进行格式设置。按下弹出窗口中的“引用”按

钮，系统会显示设定好格式的参考文献，便于您将其复制到您正在撰写的论文中。

以上几款插件都是强力推荐安装的，在 Linux 这个生态不是很完整的系统中，这种火狐浏览器的插件十分有用。

3、其它好用的火狐浏览器插件：

- ① Adblock Plus 去广告，包括 youku 的开头广告
- ② All-in-One Sidebar 最大的好处就是省去书签栏
- ③ AutoProxy 用来“流量穿透” **必备**
- ④ FoxyProxy:流量代理工具 **必备**
- ⑤ Awesome screenshot 截图的
- ⑥ DragIt 这个是集成鼠标手势和超级拖曳的
- ⑦ Greasemonkey 脚本工具，用最多的是 GooglemonkeyR 和 purereader **好用**
- ⑧ Tampermonkey 脚本管理工具油猴 **必备**

Userscript+ (<https://greasyfork.org/zh-CN/scripts/24508-userscript-show-site-all-userjsa>)

是一款 Tampermonkey 脚本,作用是当你浏览网页的时候,从右下角自动为你推荐适用于当前页面的 Tampermonkey 脚本，并且可以一键安装指定的脚本。很多时候，我们并不知道一些网站是否有用户提供用来优化页面的脚本，而 Userscript+ 就能帮你自动寻找适用的 UserJS，并默认按照评分高低排序推荐给你,给你带来一种全新的 Tampermonkey 使用体验！

Userscript+ GitHub:<https://github.com/jae-jae/Userscript-Plus>

- ⑨ LastPass 密码管理工具
- ⑩ Xmarks 书签同步新同文堂 简繁转换
- 11 All-in-one sidebar:侧栏控制，率获大奖的用户体验！
- 12 Firebug:为你的 Firefox 集成了浏览网页的同时随手可得的丰富开发工具；
- 13 Xmarks:是排名第一的附加书签。保持你的书签，密码和备份和同步多台电脑和浏览器打开的标签；
- 14 popVIdéo: 可以让视频在一个独立的小窗口里播放；
- 15 firegesture: 鼠标手势；
- 16 Torrent Finder Toolbar:找 BT 种子
- 17 Webmail Notifier:Hotmail 和 Gmail 邮件；
- 18 Yonoo:上 Twitter 和 Buzz 用，感觉比 Echofon 要好点；
- 19 Reverse Image Search:找图片；
- 20 IE Tab+:有些网站必须使用 IE；

21 IETab2 (FF3.6+) : 给予 IETab 开发的增强版本, 除具备 IETab 的全部功能外, 还可在 IE 引擎中用 Adblocklilus 过滤广告, 以及同步 cookie, 使得切换到 IE 引擎是不需要重新登录;

22 Adblock Plus: 阻挡特定网址以及网页上的广告对象;

23 calculator: 增加一个计算器到状态栏;

24 ShowLocation: 显示服务器的地理位置。利用 IP 数据库, 在状态栏显示网页服务器的位置。鼠标右键复制 IP;

25 Tile Tabs 和 Fox Splitter 都是浏览器窗口拆分工具, 具体效果有所不同;

26 YOPmail: 临时、匿名的免费邮箱地址;

27 InFormEnter 和 AutoFill Form: 两款好用的火狐自动填表插件;

28 CacheViewer2: 特别好用的下载视频工具, 原理是提取缓存。国内国外的视频网站

都好缓存。安装了以后火狐工具里会有 cacheviewer, 点开就能看到缓存了 *可用*

29 NetVideoHunter: 视频下载插件, 从优酷土豆等提取视频, 解决视频需要分段提取的问题。

30 ScarBook: 剪贴簿

31 FoxTab: 为火狐分页浏览增加壮观的 3D 特效

32 FoxClocks: 无论何时何地都可以向用户提供准确的时间

33 FireFTP: 基于火狐平台的跨平台 FTP 客户端。

34 ImageZoom: 放大镜效果

35 RotatImage: 旋转网页上的图像

三、种子文件解析下载

ctorrent 第四条列出的部分下载工具也可以解析种子文件切速度很快

四、下载工具 (请务必阅读[第二条](#))

1、合集

multiget (类似于迅雷)

Wget

axel

aria2(命令行的下载工具, 功能非常强大, 能够类似于迅雷那样搜索相同资源, 为下载加速。

命令行安装用 `install aria2`, 但安装之后启动命令是 `aria2c`)

SteadyFlow

FlareGet(可作为 Youtube 下载工具)

kGet

xdm(Xtreme 下载管理器,命令行安装需要用 xdmn 的名字.可作为 Youtube 下载工具)
curl(利用 URL 语法在命令行方式下工作的开源文件传输工具)

2、优秀推荐

aria2

uget(图形界面下载工具,亲测很好用,可以下载磁力链接、解析种子文件等等。可以将 aria2 作为插件而加速下载)

uget 配置教程:

点击 uGet 菜单栏的 编辑--设置,进入设置选项页面。在设置页面,切换到“插件”标签,勾选“启用 aria2 插件”。aria2 是 Linux 下一个命令行的下载工具,功能非常强大,能够类似于迅雷那样搜索相同资源,为下载加速。我们在 uGet 中启用 aria2,就是为了使用“uGet 的简明外观+aria2 的加速能力”。当然,前提是本机已经安装了 aria2。如果你的系统没有安装,请先在命令行安装 aria2。

`sudo apt-get install aria2`

启用 aria2 后,还需要为其设置连接数。点击面板上的分类,选择“属性”。在默认分类的属性设置里,切换到“新下载的首选设置 1”标签,把“每台服务器连接数”设置为 12 比较合适。设置完毕,记得彻底退出 uGet 后,再次启动 uGet,使刚才的设置生效。对于 Linux 用户而言,可能 Firefox 火狐浏览器是大多数 Linux 发行版的默认浏览器。需要下载文件时,Firefox 火狐浏览器配合 FlashGot 扩展,可以非常完美地调用 uGet 进行下载。

3.总结

一般情况火狐浏览器配合 uget 比较好用,有些情况下 aria2 更好用,极少数情况下 curl 和 axel、wget 好用!

安装 aria2 的方法是:

`$sudo yum install aria2`

然后为之创建软连接,以方便使用该工具:

`$ sudo ln -s /usr/bin/aria2c /usr/bin/aria2`

`$sudo ln -s /usr/bin/aria2c /usr/bin/aria`

五、影视设计/视频编辑

Blender (一款开源的跨平台全能三维动画制作软件,提供从建模、动画、材质、渲染、到音频处理、视频剪辑等一系列动画短片制作解决方案。极力推荐。点击查看[安装方法](#))

Flowblade（滤镜多且多为画中画滤镜和视频叠加透明度滤镜等，好用）
OpenShot(简单、界面十分友好、字幕模板较多、可编辑音频，好用)
Shotcut（简单、滤镜多切多为非透明度滤镜、界面友好、无字幕，好用）
aegisub（可以给视频加水印）

【以上三个可以配合使用】

Lightworks（强大但付费） Kdenlive（界面友好、没有使用过） Blender(强大：3D 建模、视频剪辑等)

Pitivi（先安装 flatpak 包管理器，再根据 Pitivi 官网说明进行安装。安装之后需要重启才能正常使用。滤镜数量可以，但软件容易卡顿。不推荐使用） Kino（一般）

Stopmotion（创建定格动画电影）

Cinelerra-CV（不太好） LiVES（似乎不好用） Avidemux（很不好用） Jahshaka（2D/3D 动画特效和视频合成）

mkvtoolnix 是 Matroska（MKV）的多媒体处理工具包。您可以编辑 MKV 视频文件，分割文件，视频同步录音。MKV 是 Matroska 的一种媒体文件，Matroska 是一种新的多媒体封装格式，也称多媒体容器 (Multimedia Container)。它可将多种不同编码的视频及 16 条以上不同格式的音频和不同语言的字幕流封装到一个 Matroska Media 文件当中。MKV 最大的特点就是能容纳多种不同类型编码的视频、音频及字幕流。它包含以下命令行工具：

mkvmerge - 从其他格式创建的 Matroska 文件的工具
mkvinfo - 用户能够获取有关 Matroska 文件曲目信息的工具
mkvextract - 即从 Matroska 文件提取曲目为其他格式的工具\

经过长久以来的使用以及结合软件安装过程的方便程度，我推荐使用 **Blender** 作为首选的视频编辑软件。

六、字幕编辑器

(gnome-subtitles) Aegisub Gaupol Jubler (Subtitle Composer) SubtitleEditor（一般）

七、音频编辑

Ardour Audacity Hydrogen Mixxx（Mixxx 是四仓 DJ 套件） Rosegarden MuseScore

八、网页设计/网页开发

AptanaStudio BlueGriffon Firebug（火狐浏览器的插件） AdobeBrackets（支持 Linux (Debian/Ubuntu) ）

九、CAD 设计

FreeCAD LibreCAD BRL-CAD

十、平面设计/图像设计

Krita MyPaint gimp Scribus Inkscape ImageMagick AlloyImage（由腾讯推出，由网页版）

十一、屏幕录像器

istanbul recordmydesktop qt-recordmydesktop

如果安装 recordmydesktop 的话，必须安装 pavucontrol，否则录制的视频可能没有声音（<https://www.linuxidc.com/Linux/2015-10/123918.htm>）。

十二、管理

nepomuk

十三、命令行浏览器

lynx w3m

十四、磁盘备份工具:

clonezilla partimage redobackup mondorescue fsarchiver partclone g4l doclone
安装磁盘备份工具如下：

```
sudo apt-get install clonezilla  
sudo apt-get install partimage  
sudo apt-get install fsarchiver  
sudo apt-get install partclone
```

运行磁盘备份工具需要 root 权限。在命令行输入工具名字即可。
其中 fsarchiver 可用。用法可见 <http://www.fsarchiver.org/Live-backup>

十五、乐谱编辑软件

musescore

十六、视频字幕提取器

（可将视频中的字幕提取出来，保存成文本或者字幕文件）
aegisub aegisub

十七、PDF 文档编辑器

Xournal （PDF Mod）

十八、摄影

Darktable Entangle Hugin

十九、三维绘图工具

MakeHuman（可以创造人型的真实模型）

二十、基于节点的合成工具

Natron（用于视频后期制作、动态图象和设计特效）

二十一、创作和编辑字体的工具

FontForge（允许你编辑某个字体中的字形，也能够使用这些字形生成字体）

二十二、设计缝纫图案的应用

Valentina

二十三、图表工具

Calligra Flow

二十四、数值分析

数值和符号计算，工程与科学绘图，控制系统的设计与仿真，数字图像处理，数字信号处理，通讯系统设计与仿真，财务与金融工程

MATLAB

二十五、查看/转换文本文件编码方式的软件

iconv enca

二十六、磁盘管理工具

gparted

二十七、视频播放器

kplayer smplayer vlc deepin-movie gnome-mplayer（该播放器在 CentOS7 系统中表现特别好，不卡顿，可以快进等等，完全可替代 smplayer 推荐安装。） totem（媒体播放器） vlc smplayer mplayer

安装可以用以下命令进行 kplayer* smplayer vlc deepin-movie gnome-mplayer totem mplayer kplayer* smplayer* vlc* deepin-movie* gnome-mplayer* totem* mplayer*

二十八、百度网盘官方 GUI 客户端

百度终于在 2019 年推出了百度网盘的 Linux 版，是和中标麒麟合作开发的。我们可以打开百度网盘网页版，然后点击下载客户端，转到客户端下载界面，选择下载 Linux 版本。这个版本截止 2019 年 6 月 19 号的时候是 v7.0 版，这个版本在软件仓库中是有的，可以通过命令行安装。

另外还有大神开发的第三方命令行客户端，也是非常好用，[百度网盘命令行客户端安装方法点这里](#)。

命令行安装方法：

```
#yum install baidunetdisk*
```

或者

```
#apt-get install baidunetdisk*
```

软件包安装方法（只有 rpm 的安装包）

7.0 版本的下载地址

http://issuedn.baidupcs.com/issue/netdisk/LinuxGuanjia/baidunetdisk_linux_2.0.1.rpm

下载之后进行安装，在安装包所在目录，打开终端，运行命令：

```
#yum localinstall ./baidunetdisk_linux_2.0.1.rpm
```

即可完成安装

至于下载速度就不知道了，这个客户端是百度官网授权开发的。

采用以上两种方法安装之后，百度网盘应用还没有 logo 图片，因此接下来建立为止设置快捷方式图片：

```
$sudo mkdir /opt/Baidu/Baidunetdisk -p
```

然后自己准备一张图片作为 Logo 图片，并放在该目录下。我准备的图片文件名是：

```
netdisk_logo.ico
```

接下来，移动图片：

```
$sudo cp ./netdisk_logo.ico /opt/Baidu/Baidunetdisk
```

修改快捷方式所在的用户组

```
$sudo chown YourUserGroup:YourUserName /usr/share/applications/baidunetdisk.desktop
```

然后添加一句用来确定 Logo 图片

```
$sudo echo 'Icon=/opt/Baidu/Baidunetdisk/netdisk_logo.ico' >> ./baidunetdisk.desktop
```

到这里就已经安装成功了。

可能有的系统中安装之后百度网盘并无法启动，从命令行运行 `$ baidunetdisk` 会报错：

```
A JavaScript error occurred in the main process
Uncaught Exception:
Error: Dynamic Linking Error: /lib64/libstdc++.so.6: version `CXXABI_1.3.8' not found (required by
/opt/baidunetdisk/.libkernel.so)
at new DynamicLibrary
```

```
(/opt/baidunetdisk/resources/app.asar/node_modules/ffi/lib/dynamic_library.js:74:11)
  at new Library (/opt/baidunetdisk/resources/app.asar/node_modules/ffi/lib/library.js:45:12)
  at Object.<anonymous>
(/opt/baidunetdisk/resources/app.asar/dist/electron/main.js:1:3434)
  at r (/opt/baidunetdisk/resources/app.asar/dist/electron/main.js:1:270)
  at Module.<anonymous>
(/opt/baidunetdisk/resources/app.asar/dist/electron/main.js:1:561527)
  at r (/opt/baidunetdisk/resources/app.asar/dist/electron/main.js:1:270)
  at /opt/baidunetdisk/resources/app.asar/dist/electron/main.js:1:1078
  at Object.<anonymous>
(/opt/baidunetdisk/resources/app.asar/dist/electron/main.js:1:1089)
  at Object.<anonymous> (/opt/baidunetdisk/resources/app.asar/dist/electron/main.js:2:3)
  at Module._compile (internal/modules/cjs/loader.js:693:30)
Fontconfig warning: "/etc/fonts/fonts.conf", line 86: unknown element "blank"
```

这是因为现有的库版本太低，解决办法有两种，一种是更新或者升级系统，一种是更新相关库。推荐首先更新或者升级系统：

```
$sudo yum update
```

```
$sudo yum upgrade
```

```
$sudo yum update
```

如果之后还是无法启动百度网盘，则再寻求相关库的高版本并安装：

从

ftp://ftp.pbone.net/mirror/ftp5.gwdg.de/pub/opensuse/repositories/home:/aevseev:/devel/CentOS7/x86_64/libstdc++-8.1.0-5.26.el7.x86_64.rpm 下载高版本的库进行安装：

```
$sudo yum install ./libstdc++-8.1.0-5.26.el7.x86_64.rpm
```

待安装完成后，如果还有其他错误，则再寻求解决。

小知识：

查看当前系统中的库 libstdc++ 支持的 “GLIBCXX” 版本：

```
$strings /lib64/libstdc++.so.6 |grep "GLIBC"
```

查看当前系统中的库 libstdc++ 支持的 “CXXABI” 版本：

```
$strings /lib64/libstdc++.so.6 |grep "CXXABI"
```

以上命令中，“/lib64/libstdc++.so.6”是我系统中该库的路径（实际是一个软连接）：

我的系统中查看情况如下：

```
$ strings /lib64/libstdc++.so.6 |grep "CXXABI"
```

```
CXXABI_1.3
```

```
CXXABI_1.3.1
```

```
CXXABI_1.3.2
```

```
CXXABI_1.3.3
```

```
CXXABI_1.3.4
```

```
CXXABI_1.3.5
```

```
CXXABI_1.3.6
```

```
CXXABI_1.3.7
```

```
CXXABI_TM_1
```

可以看出：百度云需要的是支持“CXXABI_1.3.8”，而我的系统最高只支持“CXXABI_1.3.7”，因此需要升级库。
下载需要的高版本软件库，可在 [Linux 软件包下载地址](#) 1 中下载，例如本软包 libstdc++ 则在网址中搜索“libstdc++-8.1”或者“libstdc++-8.2”，在搜索结果中找到对应本系统的版本，打开链接可以看到该版本库支持 CXXABI_1.3.8 和 CXXABI_1.3.9 那么下载安装即可。

二十九、音乐播放器（网易云音乐）

deepin-music netease-cloud-music(网易云音乐) moc（字符界面下的音乐播放器） cmus（基于终端的音乐播放器）

网易云音乐可从 <http://music.163.com/#/download/> 下载到安装包。暂时只提供 Debian Linux 发行版安装包下载。在 [Redhat 等系列系统上的安装过程](#) 见第二部分。

三十、图片查看器

shotwell feh（命令行/终端图片查看器）

三十一、虚拟系统安装工具

virtualbox

三十二、FTP 连接工具

filezilla

三十三、压缩解压软件

p7zip p7zip-full p7zip-rar rar unrar unar

Windows 下压缩的文件在 linux 中解压文件名会产生乱码。用 unar 工具可以解压.zip 文件并解决文件名的乱码问题。

wps-office_[中间这个里是版本号].tar.xz 包供下载，内涵 word excel ppt。十分好用
网址 <http://www.wps.cn/product>
或者网址 <http://community.wps.cn/download/>
均可下载到

解决 ubuntu 安装 wps 后，不能在其中输入中文的问题。（原文链接
<http://www.cnblogs.com/BlackStorm/p/5619882.html>）

如果你在基于 qt 的程序下不能使用基于 fcitx 的中文输入法，请检安装以下软件包：
fcitx-frontend-qt4 fcitx-frontend-qt5 fcitx-libs-qt fcitx-libs-qt5 libfcitx-qt0 libfcitx-qt5-1
将它们安装好后，注销再登录即可在 qt 程序下使用 fcitx 的中文输入法，不需要修改环境变量等。

四十、服务器

局域网 ftp-yum 服务器测试 samba
apache 服务器 httpd
ntp 服务器 nfs-utils nfs4
telnet 远程登录 telnet-server xinetd
ssh 远程登录 openssh-server openssh-client openssh

四十一、远程登陆

openssh-server openssh-client openssh samba telnet-server telnet ssh *ssh*

四十二、递归目录列表命令 tree

tree 是一种递归目录列表命令，产生一个深度缩进列表文件
sudo yum install tree

四十三、安装网络管理器

sudo yum install network-manager *network-manager*

四十四、安装和开启 SSH 服务

原文连接 <http://jingyan.baidu.com/article/3ea51489f9efbf52e61bba05.html>

查看 SSH 服务是否安装

```
rpm -qa | grep ssh
```

安装 SSH 服务

```
yum install *ssh* ssh*
```

启动 SSH 服务

```
service sshd restart
```

设置 SSH 服务开机启动

```
chkconfig sshd on
```

禁止 SSH 服务开机启动

```
chkconfig sshd off
```

查看 22 端口是否启动

```
netstat -antp | grep sshd
```

四十五、安装 Python3

由于 Python 已经升级到了 3，且其中变化很大，所以 python3 是必须要安装的。

CentOS7 系统安装方法见后文（[第二部分 第七章 安装 Python3 和 Python2 共存](#)），Debian 系列安装方法自行百度。

四十六、安装微信 wechat

原文链接 <http://blog.csdn.net/gatieme/article/details/52800015>

（注：以上教程中安装的是较老的版本，可至项目主页下下载最新版的进行安装，安装步骤相同）

项目主页 <https://github.com/geeeeeeeek/electronic-wechat/releases/>

下载后解压后目录中的 electronic-wechat 文件双击即可直接运行。将解压后的文件夹重命名为 electronic-wechat-linux-x64，然后拷贝到/opt/目录下，然后创建启动器，即可从快捷方式启动。

创建启动器：

```
sudo touch /usr/share/applications/Wechat.desktop
```

```
sudo gedit /usr/share/applications/Wechat.desktop
```

将下面文本粘贴到/usr/share/applications/Wechat.desktop 文件中，保存。

```
[Desktop Entry]
```

```
Name=Electronic Wechat
```

```
Name[zh_CN]=微信电脑版
```

```
Name[zh_TW]=微信电脑版
```

```
Exec=/opt/electronic-wechat-linux-x64/electronic-wechat
```

```
Icon=/opt/electronic-wechat-linux-x64/Wechat-logo.png
```

```
Terminal=false
```

```
X-MultipleArgs=false
```

```
Type=Application
```

```
Encoding=UTF-8
```

```
Categories=Application;Utility;Network;InstantMessaging;
```

```
StartupNotify=true
```

然后找一张想作为启动器图标图片，命名为 Wechat-logo.png，并复制到 /opt/electronic-wechat-linux-x64/ 目录下。

安装完成。

四十七、远程桌面管理同步软件 TeamViewer

该软件支持真正的全平台，甚至支持黑莓系统，所以是一款特别好用的远程桌面分享和管理软件。

百度中即可搜索到其官网，下载对应自己系统的版本，

rhel 在命令行即可安装：`sudo yum localinstall ./teamviewer_12.0.85001.i686.rpm`

Debian 可以双击 deb 安装包在图形界面进行安装。

四十八、轻量级办公软件 calligra 组件

Calligra 是一个轻量级的办公组件，包括好几部分其中 flow，word 等是比较常用的。Flow 可以用来绘制很多中图文件。

安装：

```
sudo yum install calligra calligra*
```

四十九、PDF 阅读器

其中，最好用是 okular 和 evince，okular 和 adobe reader 功能几乎相同。推荐使用 okular。

在下面的几款 PDF 阅读器中，有的可能由于一些未知原因无法安装成功。

```
sudo yum install mendeley mendeleydesktop mendeley* okular evince xPDF gv
```

五十、CAJ 文件阅读器

CAJ 文档是从中国知网上下载下来的论文等的文件格式，在 windows 系统下，能打开其的软件不多，切多数情况下只能用 CAJviewer 打开。在 Linux 系统下，可以用 okular 和 evince 打开 CAJ 文件。

可以在 Linux 下，用 evince 将 CAJ 文件打印为 PDF 文件，以方便传阅和打印等。

Linux 系统一般自带一款中文名叫“文档查看器”的软件，其英文名叫 evince。该文档查看器目前已知可以阅读 PDF 文档、CAJ 文档（在 windows 系统下，只能用 CAJ 阅读器才能打开，且有些情况下无法打开本没有损坏的文件，不知道原因。但在 Linux 下可以用 Evince 文档查看器查看那些在 windows 下无法打开的本没有损坏的文件）等。且可以利用 evince 中的打印功能（可以用 Ctrl+P 快捷键激活）将 CAJ 文件打印（另存）为 PDF 文档，这样就可以方便得在其他电脑上打开这些文件了。

一般情况下，evince 由 Linux 系统默认自带，在没有自带的情况下，可用以下命令安装：

```
sudo yum install evince
```

安装 okular 的命令如下：

```
sudo yum install okular
```

五十一、安装网络加速器 BBR

BBR 是由谷歌公司在 2016 年开发的一个 TCP 加速算法，该算法已经在 4.9.0 的内核中自动集成了，所以可以自动起到网络加速的作用。对于普通的低于 4.9.0 的内核中，我们需要自己手动安装 BBR 以达到网络加速作用。据说比收费的网络加速器锐速还好用。谷歌已经将该算法部署到 youtube 的网站上了。

安装方法可参照文章 <https://51.ruyo.net/2783.html> 进行手动安装。

也可以参照文章 <http://blog.csdn.net/fang8682/article/details/77727472>，使用 github 上的全自动安装脚本安装。脚本下载地址是 <https://raw.githubusercontent.com/teddysun/across/master/bbr.sh>

该项目在 github 上的地址是 <https://github.com/google/bbr>

注意：按照网上的提示，有可能在安装该算法之后会导致系统无法启动，此时可以根据网上的教程，尝试修改启动项以解决问题，或者，可以重装系统以解决问题。

五十二、安装 autossh

Linux 用户都需要经常访问 SSH 服务器，因此，为了方便我们的使用，不少用户会在自己的电脑上安装 autossh，它可以帮助我们管理 SSH 会话、自动重连和停止转发流量。

安装过程也可以参考 <http://www.xitongzhijia.net/xtjc/20150518/48611.html> 进行。

在安装该工具之前，首先需要安装一些下载编译等的工具：

```
install wget gcc make
```

下载安装包：

```
$ wget http://www.harding.motd.ca/autossh/autossh-1.4e.tgz
```

解压安装：

```
$ tar -xf autossh-1.4e.tgz
```

```
$ cd autossh-1.4e
```

```
$ ./configure
```

```
$ make
```

```
$ sudo make install
```

到这里，autossh 就安装成功了。试用命令 `$autossh` 就可以看见有很多提示信息，说明该工具已经安装成功了。

五十三、安装命令行邮件收发客户端

```
$sudo yum install mutt swaks mailx sharutils sendEmail mailutils
```

五十四、LaTeX 编辑器

【文章链接】<https://www.zhihu.com/question/19954023> 推荐了几款好用的 LaTeX 编辑器。

LaTeX (LATEX, 音译“拉泰赫”)是一种基于 T E X 的排版系统,由美国计算机学家莱斯利·兰伯特 (Leslie Lamport) 在 20 世纪 80 年代初期开发,利用这种格式,即使使用者没有排版和程序设计的知识也可以充分发挥由 TeX 所提供的强大功能,能在几天,甚至几小时内生成很多具有书籍质量的印刷品。对于生成复杂表格和数学公式,这一点表现得尤为突出。因此它非常适用于生成高印刷质量的科技和数学类文档。这个系统同样适用于生成从简单的信件到完整书籍的所有其他种类的文档。

有很多编辑器可以进行 LaTeX 编辑。如: lyx texworks texstudio emacs atom texmaker

其中 texmaker 的界面是最好看的, texstudio 的界面次之。Texworks 和 Lyx 的界面真的有点简陋,但是功能还可以。

据说好用和依赖程度是

初学者: Texmaker > TeXworks > TeXstudio

高级用户: TeXstudio > Texmaker > TeXworks

用起来似乎真的是 TeXstudio 最方便好用(只是个人感受)!

1、texstudio 的安装

从 <http://texstudio.sourceforge.net/> 下载对应的版本安装包,然后进行安装。下载下来的是二进制安装包,用命令 `localinstall` 命令进行安装即可。

2、atom 的安装

从 <https://atom.io/> 可以下载到跨平台的 atom。下载二进制文件之后,可在命令行用命令 `localinstall` 来安装。

3、其他编辑器的安装

用命令行安装工具:

```
install lyx texworks texstudio emacs atom texmaker
```

```
install *texworks* texworks* *texworks *texmaker* texmaker* *texmaker
```

即可完成安装。

五十五、开机启动项编辑软件

我们经常需要对启动项进行编辑，但是命令行的修改文件，经常因为不同的系统、不同的 grub 版本导致很多问题。我们可以使用 grub-customizer 这个软件进行启动项的编辑。具体的使用方法，网上可以找到很多教程。

install grub-customizer

即可完成安装，之后可以从软件列表中找到并启动。

五十六、安装指定版本的 GCC

1、缘起

注：mplayerde 的编译安装、网易云音乐，有道字典等在 CentOS Linux release 7.2.1511 都会遇到该问题。

本方法来自教程【<https://segmentfault.com/a/1190000006863994>】

将 ubuntu 下的网易云音乐安装到 redhat 系统下后，高版本的网易云音乐将无法打开。按照[第二部分中第八节](#)安装网易云音乐后，在终端运行命令\$netease-cloud-music 网易云音乐无法正常启动，且会提示：

```
netease-cloud-music: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.21' not found (required by
/usr/bin/./lib/netease-cloud-music/lib/libqcef.so.1)
netease-cloud-music: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.21' not found (required by
/usr/bin/./lib/netease-cloud-music/lib/libcef.so)
netease-cloud-music: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.20' not found (required by
/usr/bin/./lib/netease-cloud-music/lib/libcef.so)
```

这三条错误。错误原因是开发人员使用的更高版本的 gcc，而本机上面使用的 gcc 和对应的动态链接库版本过低（笔者系统中版本是 gcc4.8.0）。

我们可以通过命令\$strings /lib64/libstdc++.so.6 | grep GLIBCXX 来查看当前的 gcc 所支持的库的版本。例如我的 gcc (GCC) 4.8.5 支持的库列出来如下：

```
$ strings /lib64/libstdc++.so.6 | grep GLIBCXX
```

```
GLIBCXX_3.4
GLIBCXX_3.4.1
GLIBCXX_3.4.2
GLIBCXX_3.4.3
```

GLIBCXX_3.4.4
GLIBCXX_3.4.5
GLIBCXX_3.4.6
GLIBCXX_3.4.7
GLIBCXX_3.4.8
GLIBCXX_3.4.9
GLIBCXX_3.4.10
GLIBCXX_3.4.11
GLIBCXX_3.4.12
GLIBCXX_3.4.13
GLIBCXX_3.4.14
GLIBCXX_3.4.15
GLIBCXX_3.4.16
GLIBCXX_3.4.17
GLIBCXX_3.4.18
GLIBCXX_3.4.19
GLIBCXX_DEBUG_MESSAGE_LENGTH

2、编译安装高版本的 GCC

这里如此安装之后，并不能使系统默认 GCC 更改为高级版本：

在华中科技大学的官方源中下载需要的 gcc 版本：<http://mirrors.hust.edu.cn/gnu/gcc/>

笔者下载 gcc-8.1.0.tar.gz

然后解压该文件：

```
$tar -zxvf ./gcc-8.1.0.tar.gz
```

```
$cd ./gcc-8.1.0
```

```
$/contrib/download_prerequisites
```

```
$/configure
```

```
$make -j4
```

```
$sudo make install
```

然后将刚才编译生成的文件 libstdc++.so.6.0.25 拷贝到/lib64 目录下：

```
$sudo cp ./x86_64-pc-linux-gnu/libstdc++-v3/src/.libs/libstdc++.so.6.0.25 /lib64
```

接下来建立这个库的软链接：

首先备份原来的软链接：

```
$sudo mv /lib64/libstdc++.so.6 /lib64/libstdc++.so.6.backup
```

```
$sudo ln -s /lib64/libstdc++.so.6.0.25 /lib64/libstdc++.so.6
```

到这里我们就已经安装完成了。在执行一下\$netease-cloud-music 命令，发现刚才的问题已经解决了。如果没有其他问题的话，网易云音乐等就可以成功运行了！

那么我们就可以删除刚才安装 GCC 时编译的文件了：

```
$rm -rf ./gcc-8.1.0
```

以上方法并不能解决默认 GCC 版本问题，并且在 GCC 编译过程中可能会出现很多错误。

下面的过程安装并提高默认 GCC 的版本，并解决在编译过程中遇到的部分错误：

这里以 gcc-8.3.0 为例，GCC 的下载地址为：

<http://mirrors.hust.edu.cn/gnu/gcc/gcc-8.3.0/gcc-8.3.0.tar.gz>

编译 GCC 之前，需要先分别顺序安装 gmp, mpfr 和 mpc：

安装 gmp：

```
$wget https://gmplib.org/download/gmp/gmp-6.1.2.tar.bz2
$tar -jxvf ./gmp-6.1.2.tar.bz2
$mkdir ./gmp-6.1.2/build
$cd ./gmp-6.1.2/build
$./ configure --prefix=/usr/local/gmp-6.1.2 --build=x86_64-linux-gnu
$make
$sudo make install
$sudo ln -s /usr/local/gmp-6.1.2/lib/libgmp.so.10.3.2 /usr/lib/libgmp.so.10
$sudo ln -s /usr/local/gmp-6.1.2/lib/libgmp.so.10.3.2 /lib/libgmp.so.10
```

安装 mpfr：

```
$wget https://www.mpfr.org/mpfr-current/mpfr-4.0.2.tar.gz
$tar -zxvf ./mpfr-4.0.2.tar.gz && mkdir ./mpfr-4.0.2/build && cd ./mpfr-4.0.2/build
$./ configure --build=x86_64-linux-gnu --prefix=/usr/local/mpfr-4.0.2
--with-gmp=/usr/local/gmp-6.1.2
$make
$sudo make install
$sudo ln -s /usr/local/mpfr-4.0.2/lib/libmpfr.so.6.0.2 /usr/lib/libmpfr.so.6
$sudo ln -s /usr/local/mpfr-4.0.2/lib/libmpfr.so.6.0.2 /lib/libmpfr.so.6
```

安装 mpc：

```
$wget http://ftp.gnu.org/gnu/mpc/mpc-1.1.0.tar.gz
$tar -zxvf ./mpc-1.1.0.tar.gz && mkdir ./mpc-1.1.0/build && cd ./mpc-1.1.0/build
$./ mpc-1.1.0/configure --build=x86_64-linux-gnu --prefix=/usr/local/mpc-1.1.0
--with-gmp=/usr/local/gmp-6.1.2 --with-mpfr=/usr/local/mpfr-4.0.2
$make
$sudo make install
$sudo ln -s /usr/local/mpc-1.1.0/lib/libmpc.so.3.1.0 /usr/lib/libmpc.so.3
$sudo ln -s /usr/local/mpc-1.1.0/lib/libmpc.so.3.1.0 /lib/libmpc.so.3
```

安装 gcc：

准备工作：

```
$ export LIBRARY_PATH=/usr/lib/x86_64-linux-gnu/
$ export C_INCLUDE_PATH=/usr/include/x86_64-linux-gnu
$ export CPLUS_INCLUDE_PATH=/usr/include/x86_64-linux-gnu
$sudo ln -s /usr/lib/x86_64-redhat-linux6E/lib64/crt1.o /usr/lib/crt1.o
$sudo ln -s /usr/lib/x86_64-redhat-linux6E/lib64/crti.o /usr/lib/crti.o
$sudo yum install gcc-* glibc-devel.i686 glibc-devel libgcc.i686 gcc gcc-c++
$sudo ln -s /usr/lib/x86_64-linux-gnu /usr/lib64
```

下面开始编译 GCC


```

$wget http://mirrors.hust.edu.cn/gnu/gcc/gcc-8.3.0/gcc-8.3.0.tar.gz
$tar -zxvf ./gcc-8.3.0.tar.gz && mkdir ./gcc-8.3.0/build && cd ./gcc-8.3.0
$./contrib/download_prerequisites
$cd ./build
$./configure --build=x86_64-linux-gnu --prefix=/usr/local/gcc-8.3.0
--with-gmp=/usr/local/gmp-6.1.2 --with-mpfr=/usr/local/mpfr-4.0.2
--with-mpc=/usr/local/mpc-1.1.0 --enable-checking=release --disable-multilib
--program-suffix=-8.3.0
$make -j4
$sudo make install && mkdir -p /opt/gcc-8.3.0_has-been-installed

```

编译过程相当漫长，需要耐心等待好几个小时……

编译过程中的错误解决：

如果出现形如 “..../mpc/src/mul.c:error: conflicting types for ‘mpfr_fmma’ 错误：与 ‘mpfr_fmma’ 类型冲突” 的错误，则：

```
$sudo sed -i 's/mpfr_fmma/mpfr_fmma_mp/g' /usr/local/mpfr/include/mpfr.h
```

如果出现形如 “make[2]: *** [configure-stage1-zlib] 错误 1” 的错误，则：

```
$sudo ntpdate cn.pool.ntp.org
```

如果出现形如 “/cc1: error while loading shared libraries: libmpfr.so.6: cannot open shared object file: No such file or directory” 的错误，则设置环境变量：

```
$ export LIBRARY_PATH=/usr/lib/x86_64-linux-gnu/
```

```
$ export C_INCLUDE_PATH=/usr/include/x86_64-linux-gnu
```

```
$ export CPLUS_INCLUDE_PATH=/usr/include/x86_64-linux-gnu
```

关于 GCC 编译安装过程中出现的问题，您可以访问以下文章寻求解决办法：

https://tieba.baidu.com/p/5538320119?red_tag=0424682863

<https://blog.csdn.net/renhaofan/article/details/79836016>

<https://www.cnblogs.com/yuqianwen/p/5025843.html>

<https://www.cnblogs.com/pugna/p/4070407.html>

https://www.oschina.net/question/12_49423

<https://blog.csdn.net/winycg/article/details/80572735>

<https://www.cnblogs.com/kevingrace/p/9456563.html>

安装成功之后要将 gcc 的默认链接更改成新的版本：

```
$sudo mv /usr/bin/gcc /usr/bin/gcc-4.8
```

```
$sudo ln -s /usr/local/gcc-8.3.0/bin/gcc-8.3 /usr/bin/gcc
```

```
$sudo mv /usr/bin/gcc-ar /usr/bin/gcc-ar-4.8
```

```
$sudo ln -s /usr/local/gcc-8.3.0/bin/gcc-ar-8.3 /usr/bin/gcc-ar
```

```
$sudo mv /usr/bin/gcc-nm /usr/bin/gcc-nm-4.8
```

```
$sudo ln -s /usr/local/gcc-8.3.0/bin/gcc-nm-8.3 /usr/bin/gcc-nm
```

```
$sudo mv /usr/bin/gcc-ranlib /usr/bin/gcc-ranlib-4.8
```

```
$sudo ln -s /usr/local/gcc-8.3.0/bin/gcc-ranlib-8.3 /usr/bin/gcc-ranlib
```

```
$sudo mv /usr/bin/gcov /usr/bin/gcov-4.8
```

```

$ sudo ln -s /usr/local/gcc-8.3.0/bin/gcov-8.3 /usr/bin/gcov
$ sudo mv /usr/bin/c++ /usr/bin/c++-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/c++-8.3 /usr/bin/c++
$ sudo mv /usr/bin/cpp /usr/bin/cpp-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/cpp-8.3 /usr/bin/cpp
$ sudo mv /usr/bin/g++ /usr/bin/g++-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/g++-8.3 /usr/bin/g++
$ sudo mv /usr/bin/gfortran /usr/bin/gfortran-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/gfortran-8.3 /usr/bin/gfortran
$ sudo mv /usr/bin/gcov-dump /usr/bin/gcov-dump-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/gcov-dump-8.3 /usr/bin/gcov-dump
$ sudo mv /usr/bin/gcov-tool /usr/bin/gcov-tool-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/gcov-tool-8.3 /usr/bin/gcov-tool
$ sudo mv /usr/bin/x86_64-linux-gnu-gcc-nm /usr/bin/x86_64-linux-gnu-gcc-nm-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/x86_64-linux-gnu-gcc-nm-8.3
/usr/bin/x86_64-linux-gnu-gcc-nm
$ sudo mv /usr/bin/x86_64-linux-gnu-gfortran /usr/bin/x86_64-linux-gnu-gfortran-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/x86_64-linux-gnu-gfortran-8.3
/usr/bin/x86_64-linux-gnu-gfortran
$ sudo mv /usr/bin/linux-gnu-gcc-ranlib /usr/bin/linux-gnu-gcc-ranlib-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/linux-gnu-gcc-ranlib-8.3 /usr/bin/linux-gnu-gcc-ranlib
$ sudo mv /usr/bin/x86_64-linux-gnu-gcc-ar /usr/bin/x86_64-linux-gnu-gcc-ar-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/x86_64-linux-gnu-gcc-ar-8.3
/usr/bin/x86_64-linux-gnu-gcc-ar
$ sudo mv /usr/bin/x86_64-linux-gnu-c++ /usr/bin/x86_64-linux-gnu-c++-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/x86_64-linux-gnu-c++-8.3 /usr/bin/x86_64-linux-gnu-c++
$ sudo mv /usr/bin/x86_64-linux-gnu-gcc /usr/bin/x86_64-linux-gnu-gcc-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/x86_64-linux-gnu-gcc-8.3 /usr/bin/x86_64-linux-gnu-gcc
$ sudo mv /usr/bin/x86_64-linux-gnu-g++ /usr/bin/x86_64-linux-gnu-g++-4.8
$ sudo ln -s /usr/local/gcc-8.3.0/bin/x86_64-linux-gnu-g++-8.3 /usr/bin/x86_64-linux-gnu-g++
$ sudo mv /usr/lib64/libstdc++.so.6 /usr/lib64/libstdc++.so.6.old
$ sudo ln -s /usr/local/gcc-8.3.0/lib64/libstdc++.so.6.0.25 /usr/lib64/libstdc++.so.6
$ sudo mv /usr/lib64/libstdc++.so.6 /usr/lib64/libstdc++.so.6.old
$ sudo ln -s /usr/local/gcc-8.3.0/lib64/libstdc++.so.6.0.25 /usr/lib64/libstdc++.so.6

```

到这里新版本的 gcc 就安装成功了。用如下方法就可以进行测试：

```
[zhanghuoding@localhost ~]$ gcc --version
```

```
gcc (GCC) 8.3.0
```

```
Copyright © 2018 Free Software Foundation, Inc.
```

本程序是自由软件；请参看源代码的版权声明。本软件没有任何担保；

包括没有适销性和某一专用目的下的适用性担保。

```
[zhanghuoding@localhost ~]$ g++ --version
```

```
g++ (GCC) 8.3.0
```

```
Copyright © 2018 Free Software Foundation, Inc.
```

本程序是自由软件；请参看源代码的版权声明。本软件没有任何担保；
包括没有适销性和某一专用目的下的适用性担保。

```
[zhanghuoding@localhost ~]$ gcc-4.8 --version
```

```
gcc-4.8 (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
```

```
Copyright © 2015 Free Software Foundation, Inc.
```

本程序是自由软件；请参看源代码的版权声明。本软件没有任何担保；
包括没有适销性和某一专用目的下的适用性担保。

五十七、vimx——可以使用系统粘贴板的 vim

在我们安装系统之后，系统默认安装的 vim 编辑器本身很好用，但是它没有使用系统粘贴板的功能。

使用系统粘贴板，我们就可以在不同的应用之间、不同的 vim 窗口之间、vim 和浏览器之间等互相复制粘贴内容了。关于 vim 粘贴板请参看

<https://blog.csdn.net/yujinan1990/article/details/52553407>，这里有详细的描述。

在 debian 系统中，安装 vim-gtk 和 vim-gnome 如下：

```
$sudo apt-get install vim-gtk vim-gnome
```

在 redhat 系统中，安装 vim-X11 如下：

```
$sudo yum install vim-X11
```

为了方便日常的使用，我们编辑配置文件，以实现行列高亮等功能，在文件 `/etc/vimrc` 后面追加一些配置信息（这里是笔者习惯和觉得足够的配置信息，你也可以自己增加），命令如下：

```
$sudo tee -ai /etc/vimrc <<-"EOF"
```

```
set number
```

```
set cursorline
```

```
highlight CursorLine    cterm=NONE ctermbg=black ctermfg=green guibg=NONE guifg=NONE
```

```
set cursorcolumn
```

```
highlight CursorColumn cterm=NONE ctermbg=black ctermfg=green guibg=NONE guifg=NONE
```

```
set autoindent
```

```
set smartindent
```

```
set laststatus=2
```

```
EOF
```

使用方法如下：

```
$vimx filename
```

这样就可以了，其他的控制命令都和 vim 是相同的。

五十八、终端模拟器和复用器——DomTerm

这是一款为 Linux 打造的终端模拟器和复用器，带有 HTML 图形和其他不多见的功能。它使用浏览器引擎作为“GUI 工具包”，这就支持了一些相关的特性，例如可以嵌入图像和链接、HTML 富文本以及可折叠（显示/隐藏）命令，除此之外，他看起来感觉就像一个功能完整、独立的终端模拟器，有着出色的 xterm 兼容性。

介绍链接（<https://mp.weixin.qq.com/s/WpgKenK1yYr8h7bTMHnfvA>）

关于该终端的安装，请移步 GitHub 下载并按照提示进行安装。

GitHub 仓库链接（<https://github.com/PerBothner/DomTerm>）

DomTerm 项目主页链接（<https://domterm.org/>）

五十九、其他日常软件

1、DNS 缓存等管理工具 nscd

install nscd

刷新 DNS 缓存等，需要刷新时：

对 Centos7 及以上版本，用 `$/etc/rc.d/init.d/nscd start`

对其他发行版（ubuntu、Centos）等，用：`$systemctl restart nscd`

第二部分 redhat 系列系统的安装

以下是 CentOS7 系统安装完成之后，需要做的一些基础级的个性化配置。Redhat 系列其他系统，可以根据本文档相关原理，更改系统版本、发行版本等之后，同理安装。

提示：rpm 命令安装 rpm 包时，不会自动解决依赖问题，只是将缺失的依赖列出。然后打断。yum 安装 rpm 包时，会自动检查依赖，并安装相关的包。cd 到 rpm 包所在目录，运行 yum localinstall -y aabbcc.rpm 会安装 aabbcc.rpm 并且自动解决依赖问题。如果遇到依赖冲突的话，yum 也会打断。

如果双系统安装后找不到 win 的开机启动项，可以尝试通过 [grub-customizer](#) 解决。

一、获取 sudo 权限

安装 CentOS 系统后，首先需要将普通用户添加到/etc/sudoers 文件中，以获得 sudo 能力。操作如下（这里需要获取 root 权限）

首先用 \$su 命令获取权限，然后如下操作：

```
#gedit /etc/sudoers
```

在文档结尾插入一行 username ALL=(ALL) ALL

这里 username 指的是你的用户名。之后保存退出即可。

二、修改开机启动项，增加 windows 开机引导项

由于 centos7 不能识别 ntfs 分区，所以添加开机启动项之前，似乎要首先安装 ntfs-3g。然后添加之后才可以。

编辑/boot/grub2/grub.cfg 文件，sudo gedit /boot/grub2/grub.cfg

添加以下语句保存即可（hd0 后面的 1 指的是你的 win7 安装位置，你的有可能是 2 或 3 或.....，请根据具体情况多次试验设置）：

```
menuentry 'windows 7' {
    insmod ntfs
    set root=(hd0,1)
    chainloader +1
}
```

三、给程序添加启动图标：

以火狐浏览器为例进行介绍：

```
sudo touch /usr/share/applications/Firefox.desktop
```

```
sudo gedit /usr/share/applications/Firefox.desktop
```

将以下文本写入 Firefox.desktop 文件：

```
[Desktop Entry]
```

```
Name=Firefox
```

```
Name[zh_CN]=Firefox
```

```
Comment=Firefox Client
```

```
Exec=/opt/firefox/firefox
```

```
Icon=/opt/firefox/browser-firefox_2.png
```

```
Terminal=false
```

```
Type=Application
```

```
Categories=Application;
```

```
Encoding=UTF-8
```

```
StartupNotify=true
```

四、配置源

1、备份源

```
sudo mkdir /etc/yum.repos.d.backup
```

```
sudo cp /etc/yum.repos.d/* /etc/yum.repos.d.backup/
```

2、安装 yum 包管理器的基础插件

这将有利于我们进行后面的安装，以及加快安装过程中的下载等的速度（这一步不是必须的，事实证明如果用 yum-* 安装所有 yum 插件的话，程序安装速度反而会变慢，所以可以酌情不安装这些插件或者直接禁用这些插件）：

```
yum install yum-*
```

3、安装 epel 源

```
sudo yum install epel-release
```

到 http://dl.fedoraproject.org/pub/epel/7/x86_64/e/ 去下载对应版本的源安装包，例如

```
epel-release-7-10.noarch.rpm
```

```
sudo rpm -i ./epel-release-7-10.noarch.rpm
```

之后/etc/yum.repos.d/文件夹下会多出两个文件 epel.repo 和 epel-testing.repo，表示安装成功

例如（用以下命令即可安装）：

```
curl -O https://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-9.noarch.rpm
```

```
sudo rpm -ivh epel-release-7-9.noarch.rpm
```

4、安装阿里源

在 <http://mirrors.aliyun.com/repo/> 下载对应版本的源文件，并至于/etc/yum.repo.d/下即可

5、添加 mosquito 源

```
sudo touch /etc/yum.repos.d/mosquito-myrepo.repo
```

```
sudo gedit /etc/yum.repos.d/mosquito-myrepo.repo
```

在文件 mosquito-myrepo.repo 中写入以下文本：

```
[mosquito-myrepo]
```

```
name=Copr repo for myrepo owned by mosquito
```

```
baseurl=http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-testing/epel-7-$basearch/
```

```
skip_if_unavailable=True
```

```
gpgcheck=0
```

```
enabled=1
```

6、添加 nux-dextop 源

（或者官网 <http://li.nux.ro/repos.html> 有安装教程，不用导入公钥）

在 http://li.nux.ro/download/nux/dextop/el7/x86_64/中下载形如

nux-dextop-release-0-5.el7.nux.noarch.rpm 名称的文件，并安装。**注意：**一定是

nux-dextop-release 开头的安装包

然后安装 `sudo rpm -i ./nux-dextop-release-0-5.el7.nux.noarch.rpm`

7、添加 elrepo 源

（该源包含 NVIDIA 显卡驱动，除非特殊需要，否则，不要安装或者启用这个源）

首先导入公钥 `rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org`

然后在 http://elrepo.repos.mirrors.net/elrepo/el7/x86_64/RPMS/中下载形如 `elrepo-release-7.0-3.el7.elrepo.noarch.rpm` 名称的安装包文件。

然后安装 `sudo rpm -i elrepo-release-7.0-3.el7.elrepo.noarch.rpm`

安装显卡驱动后会有几个问题，可在 <http://blog.csdn.net/lockvictor/article/details/49301183> 找到解决方案，或者干脆不安装显卡驱动，一般也没有关系。
然后查看本机需要的的显卡驱动型号 `sudo yum install nvidia-detect`，型号是形如 `kmod-nvidia-340xx` 的
然后安装 `yum install kmod-nvidia-340xx`

8、添加 rpmfusion 源

（这个源比较重要）

方法一：下载以下两个二进制包

<http://download1.rpmfusion.org/free/Fedora/rpmfusion-free-release-stable.noarch.rpm>
<http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-stable.noarch.rpm>
用命令 `sudo rpm -i rpmfusion-free-release-stable.noarch.rpm`
`rpmfusion-nonfree-release-stable.noarch.rpm` 安装该源

方法二：（亲测可行）

`sudo yum localinstall --nogpgcheck`
[http://download1.rpmfusion.org/free/el/updates/7/x86_64/r/rpmfusion-free-release-7-1.noarch](http://download1.rpmfusion.org/free/el/updates/7/x86_64/r/rpmfusion-free-release-7-1.noarch.rpm)
`.rpm`
http://download1.rpmfusion.org/nonfree/el/updates/7/x86_64/r/rpmfusion-nonfree-release-7-1.noarch.rpm

方法三：（亲测可行）

`curl -O http://download1.rpmfusion.org/free/el/rpmfusion-free-release-7.noarch.rpm`
`curl -O http://download1.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-7.noarch.rpm`
`sudo rpm -ivh rpmfusion-free-release-7.noarch.rpm rpmfusion-nonfree-release-7.noarch.rpm`

9、添加 rpmforge 源

```
wget -O ./
ftp://195.220.108.108/linux/dag/redhat/el7/en/x86\_64/dag/RPMS/rpmforge-release-0.5.3-1.el7.rf.x86\_64.rpm
sudo yum localinstall ./rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
sudo yum makecache
```

五、安装 dnf 包管理器

对于 CentOS，有 centos7 - centos 7.4 can not install dnffrom epel - Server Fault。即其不能通过 yum 安装 dnf。

在谷歌通过 package dnf available 搜索，可见网址

<https://serverfault.com/questions/874471/centos-7-4-can-not-install-dnffrom-epel> 有详细说明。

1、安装 dnf 方法如下

```
wget
http://springdale.math.ias.edu/data/puias/unsupported/7/x86\_64/dnf-conf-0.6.4-2.sdl7.noarch.rpm
```

```
wget
http://springdale.math.ias.edu/data/puias/unsupported/7/x86\_64/dnf-0.6.4-2.sdl7.noarch.rpm
```

```
wget
http://springdale.math.ias.edu/data/puias/unsupported/7/x86\_64/python-dnf-0.6.4-2.sdl7.noarch.rpm
```

```
sudo yum install python-dnf-0.6.4-2.sdl7.noarch.rpm dnf-0.6.4-2.sdl7.noarch.rpm
dnf-conf-0.6.4-2.sdl7.noarch.rpm *dnf*
```

(注：该条安装命令没有错误。确是用 yum 安装)

2、安装 dnf copr 插件：

```
wget -P /etc/yum.repos.d/
https://copr.fedoraproject.org/coprs/jkastner/dnf-plugins-core/repo/epel-7/jkastner-dnf-plugins-core-epel-7.repo
```

```
sudo yum install dnf-plugins-core
```

3、解决 dnf metadata already locked 问题

如果在使用 dnf 命令进行安装的过程中因为某些原因终止了安装过程,当再次使用 dnf install 命令的时候会提示如下错误:

```
metadata already locked by 1570
```

```
PID 为 1570 的应用程序是: dnf
```

```
内存: 35 M RSS (537 MB VSZ)
```

```
已启动: Tue Oct 24 09:16:54 2017 - 02:50 之前
```

```
状态 : 跟踪/停止
```

解决办法:

```
sudo rm -f /var/cache/dnf/x86_64/7/x86_64/7/metadata_lock.pid
```

六、安装 shadowsocks 图形界面

(官方安装教程 <https://github.com/madeye/shadowsocks-libev#fedora--rhel>)

1、方法一

在/etc/yum.repo.d/目录下新建文件 librehat-shadowsocks-epel-7.repo 并编辑。

即:

```
sudo touch /etc/yum.repos.d/librehat-shadowsocks-epel-7.repo
```

```
sudo gedit /etc/yum.repos.d/librehat-shadowsocks-epel-7.repo
```

在 librehat-shadowsocks-epel-7.repo 文件中写入以下文本并保存

```
[librehat-shadowsocks]
```

```
name=Copr repo for shadowsocks owned by librehat
```

```
baseurl=https://copr-be.cloud.fedoraproject.org/results/librehat/shadowsocks/epel-7-$basearch  
/
```

```
type=rpm-md
```

```
skip_if_unavailable=True
```

```
gpgcheck=1
```

```
gpgkey=https://copr-be.cloud.fedoraproject.org/results/librehat/shadowsocks/pubkey.gpg
```

```
repo_gpgcheck=0
```

```
enabled=1
```

```
enabled_metadata=1
```

然后 `sudo yum update`

然后 `sudo yum install gcc gettext autoconf libtool automake make pcre-devel asciidoc xmlto udns-devel libev-devel`

2、方法二

或者可以不建立上面的 `librehat-shadowsocks-epel-7.repo` 文件，而使用命令进行安装。

`sudo wget`

`https://copr.fedorainfracloud.org/coprs/librehat/shadowsocks/repo/epel-7/librehat-shadowsocks-epel-7.repo -P /etc/yum.repos.d/`

然后执行 `sudo yum update`

`sudo yum makecache`

`sudo yum install shadowsocks`

`sudo yum install qt5-qtbase-devel`

`sudo yum install shadowsocks-qt5`

七、安装 Python3 和 Python2 共存

原文连接 <http://www.cnblogs.com/JahanGu/p/7452527.html>

1、查看是否已经安装 Python

CentOS 7.2 默认安装了 `python2.7.5`，因为一些命令要用它比如 `yum` 它使用的是 `python2.7.5`。使用 `python -V` 命令查看一下是否安装 Python。然后使用命令 `which python` 查看一下 Python 可执行文件的位置，可见执行文件在 `/usr/bin/` 目录下，执行 `ll /usr/bin/python*` 命令查看软链接，`python` 指向的是 `python2.7`。

2、开始编译安装 python3

【在安装之前，为了以后能正常使用 `tkinter` 图形模块，建议首先阅读 [7、解决 python 无法找到 tkinter 图形模块的问题](#)。当然也可以直接安装 `python`，待以后遇到问题再行解决】

先安装相关包

`yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel`

gcc make

一定不能忽略以上相关包。

去官网下载编译安装包或者直接执行以下命令下载

wget -P ./ https://www.python.org/ftp/python/3.6.3/Python-3.6.3.tar.xz

解压

tar -xvJf ./Python-3.6.3.tar.xz

切换进入

cd ./Python-3.6.3

编译安装

./configure --prefix=/usr/local/python3

make

make install

安装完毕，/usr/local/目录下就会有 python3 了

因此我们可以添加软链到执行目录下/usr/bin

ln -s /usr/local/python3/bin/python3 /usr/bin/python3.6

ln -s /usr/bin/python3.6 /usr/bin/python3

ll /usr/bin/python* 命令查看软链接，可以看到软链创建完成

测试安装成功了没，执行

python3 -V 看看输出的是不是 python3 的版本

执行 python2 -V 看到的的就是 python2 的版本

3、修改 yum 配置

（如果将/usr/bin/目录下的 python 软链接修改成了指向 python3 及以上版本，则需要进行这步，否则，跳过这步。）

因为执行 yum 需要 python2 版本，所以我们还要修改 yum 的配置，执行：

vi /usr/bin/yum

把

#!/usr/bin/python

修改为

#!/usr/bin/python2

同理 vi /usr/libexec/urlgrabber-ext-down 文件里面的#!/usr/bin/python 也要修改为#!/usr/bin/python2

/usr/bin/python2

这样 python3 版本就安装完成；同时 python2 也存在

python -V 版本 3

python2 -V 版本 2

4、关于 pip

在同一个系统中，多个 pip 不能共存，因此可以手动下载安装 setuptools 这个工具软件就可

以了。

为 pip 创建软链接：

```
$sudo ln -s /usr/local/python3/bin/pip3 /usr/bin/pip3
```

然后就可以直接在终端中运行 pip3 为 python3 安装模块了

5、安装另一个插件

在用 pip 安装模块的过程中，有的模块安装不上，可以先安装 python-tools 这个软件，然后就可以了。

```
$sudo yum install python-tools
```

还有另一个软件 setuptools。也可以自行查阅其用途和用法。

6、替换 pip 的官方源

【关于 pip 源的相关问题，可以参看博客 <https://www.cnblogs.com/sunnydou/p/5801760.html>】

Pip 的官方源速度不仅慢，而且在大多数情况不能成功。因此需要在安装模块的时候手动指定源，或者配置永久的源

例如临时指定源：

```
$sudo pip3 install networkx -i http://mirrors.aliyun.com/pypi/simple/
```

以上这种方式就是为 pip 临时指定了阿里的源，有时会报错，可以以如下方式，添加信任就可以了：

```
$sudo pip3 install tkinter -i http://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com
```

也可以永久替换官方的源：

需要创建或修改配置文件（一般都是创建），linux 的文件在 ~/.pip/pip.conf，windows 在 %HOMEPATH%\pip\pip.ini，

修改内容为：

```
[global]
```

```
index-url = http://pypi.douban.com/simple
```

```
[install]
```

```
trusted-host=pypi.douban.com
```

这样在使用 pip 来安装时，会默认调用该镜像（示例中使用的是豆瓣源）。

另外，有几个值得使用的不错的国内镜像源如下：

阿里云 <http://mirrors.aliyun.com/pypi/simple/>

中国科技大学 <https://pypi.mirrors.ustc.edu.cn/simple/>

豆瓣(douban) <http://pypi.douban.com/simple/>

清华大学 <https://pypi.tuna.tsinghua.edu.cn/simple/>

中国科学技术大学 <http://pypi.mirrors.ustc.edu.cn/simple/>

7、解决 python 无法找到 tkinter 图形模块的问题

Tkinter 是 python 中广泛使用的图形模块。但是通常在安装了 tkinter 之后，只有系统自带的 python 版本能找到 tkinter，我们自己安装的 python 版本无法找到 tkinter 这个模块。但是系统自带的 python 版本太低，很多情况下不能满足开发需求。

tkinter 是由 tcl 和 tk 两个部分组成的，所以下载 tkinter 的时候，要下载 tcl 和 tk 两部分。下载完之后分别解压，然后分别使用 `./configure && make && sudo make install` 命令来安装。

具体的安装教程也可以参看文章：

<https://www.zhihu.com/question/42162071/answer/95441732>

或者

<https://www.douban.com/note/524197380/?type=like>

或者参看本教程[第五部分 第六章 批处理和交互脚本语言（二）——tcl](#)，里面有详细的安装步骤以及其他教程可供参考。

重新编译安装我们需要的 python 版本的安装包：

① 下载并使用 `tar -xvJf` 命令解压 python 的 tar.xz 包，然后进入解压后的源文件目录，编辑源文件根目录下的 Modules 目录下的 Setup.dist 文件。例如：

```
vim ./Modules/Setup.dist
```

② 在文件中找到并修改以下五行：

```
第一行：_tkinter _tkinter.c tkappinit.c -DWITH_APPINIT \
```

```
第二行：-L/usr/local/lib \
```

```
第三行：-I/usr/local/include \
```

或者

```
-I/usr/X11R6/include \
```

```
第四行：-ltk8.2 -ltcl8.2 \
```

```
第五行：-lX11
```

修改方法：将以上 5 行前面的注释符号“#”删除，然后按照你系统中安装的 tk 和 tcl 的版本修改，第四行中的版本号。然后保存就可以了。查看 tk 和 tcl 版本号的方法：

```
$rpm -qa | grep ^tk
```

```
[root@li250-193 Python-2.7.4]# rpm -qa | grep ^tk
```

```
tk-8.5.7-5.el6.x86_64
```

```
tkinter-2.6.6-36.el6.x86_64
```

```
[root@li250-193 Python-2.7.4]# rpm -qa | grep ^tcl
```

```
tcl-8.5.7-6.el6.x86_64
```

例如，查询到如上信息，则修改第四行为：`-ltk8.5 -ltcl8.5 \`

如下图：

```

# *** Always uncomment this (leave the leading underscore in!):
_tkinter _tkinter.c tkappinit.c -DWITH_APPINIT \
# *** Uncomment and edit to reflect where your Tcl/Tk libraries are:
-L/usr/local/lib \
# *** Uncomment and edit to reflect where your Tcl/Tk headers are:
-I/usr/local/include \
# *** Uncomment and edit to reflect where your X11 header files are:
-I/usr/X11R6/include \
# *** Or uncomment this for Solaris:
-I/usr/openwin/include \
# *** Uncomment and edit for Tix extension only:
-DWITH_TIX -ltix8.1.8.2 \
# *** Uncomment and edit for BLT extension only:
-DWITH_BLT -I/usr/local/blt/blt8.0-unoff/include -lBLT8.0 \
# *** Uncomment and edit for PIL (TkImaging) extension only:
(See http://www.pythonware.com/products/pil/ for more info)
-DWITH_PIL -I../Extensions/Imaging/libImaging tkImaging.c \
# *** Uncomment and edit for TOGL extension only:
-DWITH_TOGL togl.c \
# *** Uncomment and edit to reflect your Tcl/Tk versions:
4 -ltk8.6 -ltcl8.6 \
# *** Uncomment and edit to reflect where your X11 libraries are:
-L/usr/X11R6/lib \
# *** Or uncomment this for Solaris:
-L/usr/openwin/lib \
# *** Uncomment these for TOGL extension only:
-lGL -lGLU -lXext -lXmu \
# *** Uncomment for AIX:
-lld \
# *** Always uncomment this; X11 libraries to link with:
-lX11 \
# Lance Ellinghaus's syslog module
#syslog syslogmodule.c # syslog daemon interface
264,24

```

③ 修改完成后，保存退出即可。

④ 然后在 python 源码目录中，运行 `./configure && make && sudo make install` 重新安装我们需要的高版本的 python 即可。

安装完成后，运行 python，可以发现 `import tkinter` 已经不会报错了。

8、解决 tkinter 中 tk.h 和 libtk.a 版本不匹配的问题

关于这部分，也可以参看文章 <https://blog.csdn.net/sarahpanee/article/details/9837723> 但是，要根据自己的机器上的提示信息，修改对应文件就好了。

经过上面第 7 步，已经可以正常导入 tkinter 这个包了，但是，有可能在使用该包的过程中，还会有提示如下：

Traceback (most recent call last):

File "`./__init__.py`", line 40, in `<module>`

`mf.draw_network_structure(['a','b','c','e'],test_edges)`

File

`"/mnt/Work/Programs/BN-create-incremental-learning-PY/BN-structure-learning/mine_function.py`", line 259, in `draw_network_structure`

`nx.draw(gra,with_labels = True)` #绘制网络

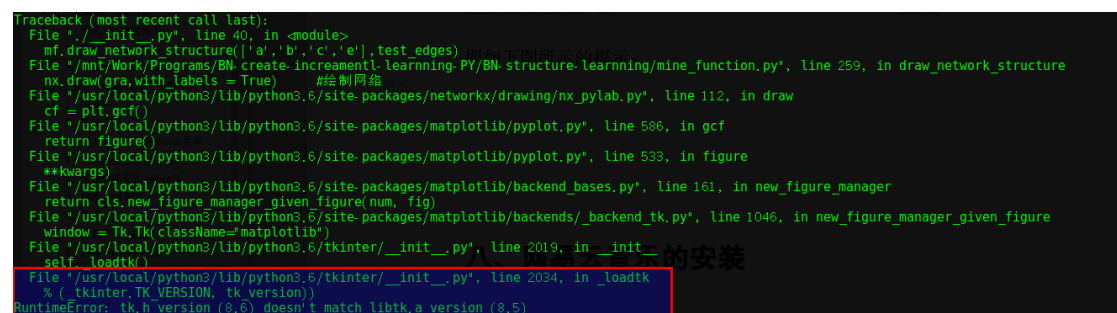
File "`/usr/local/python3/lib/python3.6/site-packages/networkx/drawing/nx_pylab.py`", line

```

112, in draw
    cf = plt.gcf()
File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/pyplot.py", line 586, in gcf
    return figure()
File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/pyplot.py", line 533, in figure
    **kwargs)
File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/backend_bases.py", line 161,
in new_figure_manager
    return cls.new_figure_manager_given_figure(num, fig)
File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/backends/_backend_tk.py",
line 1046, in new_figure_manager_given_figure
    window = Tk.Tk(className="matplotlib")
File "/usr/local/python3/lib/python3.6/tkinter/__init__.py", line 2019, in __init__
    self._loadtk()
File "/usr/local/python3/lib/python3.6/tkinter/__init__.py", line 2034, in _loadtk
    % (_tkinter.TK_VERSION, tk_version))
RuntimeError: tk.h version (8.6) doesn't match libtk.a version (8.5)

```

即如下图所示的提示：



```

Traceback (most recent call last):
  File "./__init__.py", line 40, in <module>
    mf.draw_network_structure(['a','b','c','e'], test_edges)
  File "/mnt/Work/Programs/BN-create-incremental-learning-PY/BN-structure-learning/mine_function.py", line 259, in draw_network_structure
    nx.draw(g, with_labels=True) #绘制网络
  File "/usr/local/python3/lib/python3.6/site-packages/networkx/drawing/nx_pyplot.py", line 112, in draw
    cf = plt.gcf()
  File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/pyplot.py", line 586, in gcf
    return figure()
  File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/pyplot.py", line 533, in figure
    **kwargs)
  File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/backend_bases.py", line 161, in new_figure_manager
    return cls.new_figure_manager_given_figure(num, fig)
  File "/usr/local/python3/lib/python3.6/site-packages/matplotlib/backends/_backend_tk.py", line 1046, in new_figure_manager_given_figure
    window = Tk.Tk(className="matplotlib")
  File "/usr/local/python3/lib/python3.6/tkinter/__init__.py", line 2019, in __init__
    self._loadtk()
  File "/usr/local/python3/lib/python3.6/tkinter/__init__.py", line 2034, in _loadtk
    % (_tkinter.TK_VERSION, tk_version))
RuntimeError: tk.h version (8.6) doesn't match libtk.a version (8.5)

```

也就是说会提示文件 `tk.h` 和文件 `libtk.a` 的版本不一致。

这个问题根本得怎么解决我本人也不知道，不过网上一致的解决办法就是注释点掉版本检测的代码。

按照出错提示信息的倒数第三行，我们需要修改文件：

`/usr/local/python3/lib/python3.6/tkinter/__init__.py`

（当然这是我系统中的文件，在网上所说的教程里面，修改的文件应该是 `/usr/local/lib/python2.6/lib-tk/Tkinter.py` 的第 1665 行。这个因个人安装的系统以及软件版本而异。我们只需要按照错误提示的倒数第三行，修改对应文件的对应部分就可以了。）

我这里的做法是：

- ① 进入 `/usr/local/python3/lib/python3.6/tkinter/` 目录，然后备份原文件（以防出错）。执行：`$cd /usr/local/python3/lib/python3.6/tkinter/`
`$sudo cp ./__init__.py ./__init__.py.backup`
- ② 然后打开文件进行修改：
`$sudo vim ./__init__.py`

根据上面的错误提示，是在 2034 行报错的，但是我们可以发现，出错的原因是 if 条件判断

语句:

```
2028     self.tkloaded = 1
2029     global _default_root
2030     # Version sanity checks
2031     tk_version = self.tk.getvar('tk_version')
2032     if tk_version != _tkinter.TK_VERSION:
2033         raise RuntimeError('tk.h version (%s) doesn't match libtk.a version (%s)'
2034                             % (_tkinter.TK_VERSION, tk_version))
2035     # Under unknown circumstances, tcl_version gets coerced to float
2036     tcl_version = str(self.tk.getvar('tcl_version'))
2037     if tcl_version != _tkinter.TCL_VERSION:
2038         raise RuntimeError('tcl.h version (%s) doesn't match libtcl.a version (%s)'
2039                             % (_tkinter.TCL_VERSION, tcl_version))
2040     # Create and register the tkerror and exit commands
2041     # We need to inline parts of _register here, _register
2042     # would register differently-named commands.
2043     if self._tclCommands is None:
2044         self._tclCommands = []
```

因此，我们可以注释掉它们，我们在两个 tk_version 和 tcl_version 版本判断的 if 条件语句的前后分别加三个英文单引号或者分别加一个英文双引号就可以了。如下图四个地方：

```
2030     # Version sanity checks
2031     tk_version = self.tk.getvar('tk_version')
2032     '''if tk_version != _tkinter.TK_VERSION:
2033         raise RuntimeError('tk.h version (%s) doesn't match libtk.a version (%s)'
2034                             % (_tkinter.TK_VERSION, tk_version))'''
2035     # Under unknown circumstances, tcl_version gets coerced to float
2036     tcl_version = str(self.tk.getvar('tcl_version'))
2037     '''if tcl_version != _tkinter.TCL_VERSION:
2038         raise RuntimeError('tcl.h version (%s) doesn't match libtcl.a version (%s)'
2039                             % (_tkinter.TCL_VERSION, tcl_version))'''
2040     # Create and register the tkerror and exit commands
2041     # We need to inline parts of _register here, _register
2042     # would register differently-named commands.
```

③ 然后我们保存退出，再次运行我们的程序就不会报版本错误的信息了。

9、Python 虚拟环境——virtualenv

在开发 Python 应用程序的时候，系统安装的 Python3 只有一个版本：3.4。所有第三方的包都会被 pip 安装到 Python3 的 site-packages 目录下。如果我们要同时开发多个应用程序，那这些应用程序都会共用一个 Python，就是安装在系统的 Python 3。如果应用 A 需要 jinja 2.7，而应用 B 需要 jinja 2.6 怎么办？这种情况下，每个应用可能需要各自拥有一套“独立”的 Python 运行环境。virtualenv 就是用来为一个应用创建一套“隔离”的 Python 运行环境。

此处只做提示，说明有这么一个 Python 的虚拟环境工具，可以帮助我们创建相互隔离的 python 环境。具体的安装以及使用，可以查看以下几篇文章：

- ① <https://www.cnblogs.com/technologylife/p/6635631.html>
- ② <https://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000/001432712108300322c61f256c74803b43bfd65c6f8d0d0000>
- ③ <https://ft.wupo.info/category/ways/linux-ways/page/3/>

或者也可以在网上搜索其他文章和教程以安装和学习使用该虚拟工具。

八、网易云音乐的安装

网易在其官网上只放出了 Debian 系列安装包。在 Redhat 系列系统上安装就会比较麻烦。详细教程有以下三个，可以结合起来参考和运行三个教程中的全部命令进行安装。

有大神开发了网易云音乐的纯字符界面客户端，功能很全，[安装方法](#)见后文。

<https://github.com/caojx-git/learn/blob/master/notes/linux/CentOS7%E4%B8%AD%E5%AE%89%E8%A3%85%E7%BD%91%E6%98%93%E4%BA%91%E9%9F%B3%E4%B9%90.md>

http://blog.csdn.net/hong_sir1988/article/details/72716690

<http://blog.csdn.net/u010526125/article/details/72526694>

总结安装步骤如下：

1、首先安装 epel 和 rpmfusion 源

（前文[安装 epel 源](#)和[添加 rpmfusion 源](#)讲述了安装方法）。

2、下载网易云音乐的安装包

到网易官网下载网易云音乐 ubuntu14.04 版本 这里下载的是 ubuntu14.04(64 位)安装包，下载其他的版本有些依赖包找不到。

3、解压 deb 包

使用如下解压命令解压 netease-cloud-music_1.0.0-2_amd64_ubuntu14.04.deb

```
ar -vx netease-cloud-music_1.0.0-2_amd64_ubuntu14.04.deb
```

得到如下三个文件

x - debian-binary

x - control.tar.gz

x - data.tar.xz

继续解压 data.tar.xz

```
xz -dk data.tar.xz
```

```
tar -xvf data.tar
```

得到一个 usr 目录

4、将解压得到的 usr 目录复制到系统中

将上面得到的 usr 目录其放到 Centos7 的根目录下（两个 usr 目录会自动合并）。

```
sudo cp -r usr /
```

5、安装依赖库

用 ldd 命令查看需要的库文件。

```
ldd /usr/lib/netease-cloud-music/netease-cloud-music
```

终端会列出很多依赖的库文件。用如下两条命令，即可全部安装需要的库文件（根据终端提示，必要时可用--skip-broken 安装这些库文件）。

```
sudo yum --skip-broken install linux-vdso.so.1 libcef.so libX11.so.6 libQt5Widgets.so.5
libQt5X11Extras.so.5 libQt5DBus.so.5 libQt5Gui.so.5 libQt5Network.so.5
libQt5Multimedia.so.5 libQt5Xml.so.5 libXext.so.6 libXtst.so.6 libfontconfig.so.1
libglib-2.0.so.0 libz.so.1 libQt5Core.so.5 libdl.so.2 libstdc++.so.6 libm.so.6
libgcc_s.so.1 libpthread.so.0 libc.so.6 librt.so.1 libgobject-2.0.so.0 libfreetype.so.6
libpangocairo-1.0.so.0 libcairo.so.2 libpango-1.0.so.0 libXi.so.6 libnss3.so libnssutil3.so
libsmime3.so libnspr4.so libasound.so.2 libXfixes.so.3 libgio-2.0.so.0 libatk-1.0.so.0
libXcursor.so.1 libXrender.so.1 libXss.so.1 libXrandr.so.2 libdbus-1.so.3 libexpat.so.1
libcups.so.2 libgtk-x11-2.0.so.0 libgdk-x11-2.0.so.0 libgdk_pixbuf-2.0.so.0
ld-linux-x86-64.so.2 libxcb.so.1 libGL.so.1 libpng15.so.15 libproxy.so.1 libssl.so.10
libcrypto.so.10 libpulse.so.0 libpcrc.so.1 libicu18n.so.50 libicuuc.so.50 libicudata.so.50
libpcrc16.so.0 libgthread-2.0.so.0
```

```
sudo yum --skip-broken install libsystemd.so.0 libffi.so.6 libpangoft2-1.0.so.0 libthai.so.0
libharfbuzz.so.0 libpixmap-1.so.0 libEGL.so.1 libxcb-shm.so.0 libxcb-render.so.0
libplc4.so libplds4.so libgmodule-2.0.so.0 libsdl.so.1 libresolv.so.2 libmount.so.1
libgssapi_krb5.so.2 libkrb5.so.3 libk5crypto.so.3 libcom_err.so.2 libavahi-common.so.3
libavahi-client.so.3 libcrypt.so.1 libXinerama.so.1 libXcomposite.so.1 libXdamage.so.1
libXau.so.6 libxcb-dri3.so.0 libxcb-present.so.0 libxcb-sync.so.1 libxshmfence.so.1
libglapi.so.0 libX11-xcb.so.1 libxcb-glx.so.0 libxcb-dri2.so.0 libXxf86vm.so.1 libdrm.so.2
libmodman.so.1 libpulsecommon-10.0.so libcap.so.2 liblzma.so.5 libgrypt.so.11
libgpg-error.so.0 libdw.so.1 libgraphite2.so.3 libxcb-xfixes.so.0 libgbm.so.1 libblkid.so.1
libuuid.so.1 libkrb5support.so.0 libkeyutils.so.1 libfreebl.so libICE.so.6 libSM.so.6
libwrap.so.0 libsndfile.so.1 libasyncns.so.0 libattr.so.1 libelf.so.1 libbz2.so.1 libnsl.so.1
libgsm.so.1
```

如果安装过程提示“没有可用软件包 xxxx”，则可以在 <http://www.rpmfind.net/> 搜索可用的软件包并下载，然后用命令 `yum localinstall -y ./xxxxx.rpm` 安装该包。（注：如果在网站上找不到相应包，或者下载的包由于依赖冲突安装失败，则可以先跳过这步，先不安装这些包，看后来云音乐实在运行不起来，再来想办法解决这些问题。）

继续安装其他依赖库

```
sudo yum install qt5-qtbase-gui qt5-qtxmlextras qt5-qtmultimedia  
libXScrnSaver
```

6、升级 glibc

（这个不是一定要安装，经测试，可以选择不进行安装）

```
curl -O http://ftp.gnu.org/gnu/glibc/glibc-2.26.tar.gz
```

```
tar xzf glibc-2.26.tar.gz; cd glibc-2.26/; mkdir build; cd build; ../configure --prefix=/usr; make -j16  
&& sudo make install
```

7、升级 libstdc++

（这一步似乎不会成功，因为找不到要下载的文件。我跳过了这步，但是没有问题。）

```
curl -O  
ftp://rpmfind.net/linux/fedora/linux/updates/22/x86\_64/l/libstdc++-5.3.1-6.fc22.x86\_64.rpm
```

```
rpm2cpio ./libstdc++-5.3.1-6.fc22.x86_64.rpm | cpio -id
```

```
sudo cp usr/lib64/libstdc++.so.6.0.21 /usr/lib64/; sudo ln -sf /usr/lib64/libstdc++.so.6.0.21  
/usr/lib64/libstdc++.so.6
```

8、安装解码器

```
sudo yum install gstreamer1-libav gstreamer1-plugins-bad-freeworld  
gstreamer1-plugins-base-tools gstreamer1-plugins-ugly gstreamer1-plugins-bad-free  
gstreamer1-plugins-good gstreamer1-plugins-base gstreamer1 gstreamer-plugins-good
```

gststreamer-plugins-bad gstreamer-plugins-ugly gstreamer-ffmpeg ffmpeg libvdpau mpg123
mplayer mplayer-gui

9、修改库文件权限（解决 abort 问题）

此时运行程序/usr/bin/netease-cloud-music

会发现报错如下：

The SUID sandbox helper binary was found, but is not configured correctly. Rather than run without sandboxing I'm aborting now. You need to make sure that /usr/lib/netease-cloud-music/chrome-sandbox is owned by root and has mode 4755.

是因为文件权限问题，根据提示，赋予/usr/lib/netease-cloud-music/chrome-sandbox 文件 4755 权限（可参看 [Linux 中的文件权限](#) 来了解原理）请参即可：

```
sudo chmod 4755 /usr/lib/netease-cloud-music/chrome-sandbox
```

用 ls -al 命令可看到，/usr/lib/netease-cloud-music/chrome-sandbox 文件信息如下：

```
-rwsr-xr-x. 1 root root 14464 10月 6 13:11 chrome-sandbox
```

这个权限值都是以往没有见过的。

10、运行网易云音乐

运行程序/usr/bin/netease-cloud-music 发现客户端能启动了，并且所有功能都正常。

但是如果安装的是新版本的网易云音乐客户端，则可能会出现网易云音乐无法启动的情况，从命令行启动\$netease-cloud-music，会报错如下：

```
netease-cloud-music: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.21' not found (required by /usr/bin/./lib/netease-cloud-music/lib/libqcef.so.1)
netease-cloud-music: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.21' not found (required by /usr/bin/./lib/netease-cloud-music/lib/libcef.so)
netease-cloud-music: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.20' not found (required by /usr/bin/./lib/netease-cloud-music/lib/libcef.so)
```

这个问题可以通过[升级安装高版本的 gcc](#) 来解决。

11、创建快捷方式

编辑 `/usr/share/applications/netease-cloud-music.desktop` 文件，添加 logo。

```
sudo gedit /usr/share/applications/netease-cloud-music.desktop
```

九、搜狗拼音输入法的安装

（以下安装步骤中 1 和 2 只需执行其中一个就行）

注意：安装前不需要卸载 **ibus**，卸载 **ibus** 桌面可能会出错。

原文链接 <http://www.cnblogs.com/Yiutto/p/6204085.html>

1、用别人提供的安装包安装

下载解压 centos7（sougou）fcitx.tar.gz，下载地址：<http://pan.baidu.com/s/1gfNugyZ> 密码：
cz3v

解压命令：`tar -zxvf ./fcitx.tar.gz`

进入解压后的文件夹，安装目录下的所有二进制文件：

```
sudo yum localinstall ./*.rpm --skip-broken
```

2、在命令行直接安装所需基础包

（在这一步中，如果遇到因为依赖问题无法安装某些包，则可以多次尝试。如果多次尝试还是无法解决，则用第一步中的方法进行安装。）

```
sudo yum install --skip-broken --force fcitx*
```

```
sudo yum install --skip-broken --force sogou-pinyin*
```

```
sudo yum install --skip-broken --force sysconftool*
```

3、安装图形输入法选择器

```
sudo yum install im-chooser
```

4、结束 ibus 守护进程

```
sudo pkill ibus-daemon
```

5、关闭 gnome-shell 对键盘的监听

```
gsettings set org.gnome.settings-daemon.plugins.keyboard active false
```

6、切换输入法为 fcitx

```
imsettings-switch fcitx
```

7、重载 fcitx，启动搜狗面板

```
fcitx -r
```

8、在 Fcitx 配置里面选好搜狗输入法

分别在命令运行以下命令，之后会弹出相应的设置面板，在里面进行输入法的相关个人喜好设置后，就可以了。

```
fcitx-configtool  
sogou-qimpanel
```

这里完成以后搜狗输入法就可以正常使用了。
如果输入法面板错误无法打开，则重新运行第 7 步，重启一下就可以了！

9、对有些 WPS 里面无法输入中文的问题的解决

（注意：这一步是针对有相关问题出现时需要做的，不出现的情况下，这步时没有必要做的。）

分别修改/usr/bin/wps， /usr/bin/et， /usr/bin/wpp 内容，添加黑体部分变量，如下：

```
#!/bin/bash  
exportXMODIFIERS="@im=fcitx"
```

```
exportQT_IM_MODULE="fcitx"
gOpt=
#gOptExt=-multiply
.....
```

十、集成开发环境 monodevelop 的安装

（教程链接：<http://blog.csdn.net/aoshilang2249/article/details/50084515>）

```
rpm --import
"http://keyserver.ubuntu.com/pks/lookup?op=get&search=0x3FA7E0328081BFF6A14DA29AA6A
19B38D3D831EF"
```

```
yum-config-manager --add-repo http://download.mono-project.com/repo/centos/
```

```
yum install mono-complete
```

```
yum install monodevelop
```

到这里就安装成功了。

十一、安装微信 wechat

安装步骤请移步第一部分的[四十六、安装微信 wechat](#)，安装过程完全相同。

十二、动画、视频编辑软件 Blender

Blender 是一款开源的跨平台全能三维动画制作软件，提供从建模、动画、材质、渲染、到音频处理、视频剪辑等一系列动画短片制作解决方案，linux 环境下，该软件解压即用。

官方发布很多版本，其中稳定的历史版本下载地址为：<https://download.blender.org/release/> 这里面也包含最新发布的稳定版。

下载需要的版本，解压即可运行，然后将其放到自己希望的目录，创建一个桌面启动器即可。下面以我的安装过程为例，进行示范。

首先，我们确定需要的版本已经下载到本地，例如我下载的是（2.79b 的版本）：

```
blender-2.79b-linux-glibc219-x86_64.tar.bz2
```


然后解压：\$tar -jxvf ./blender-2.79b-linux-glibc219-x86_64.tar.bz2
 得到一个 blender-2.79b-linux-glibc219-x86_64 目录。
 我打算将其安装在 opt 目录下，并新建 Blender 目录中。做如下工作；
 \$sudo mkdir -p /opt/Blender/blender-2.79b
 修改新建文件夹的权限：
 \$sudo chmod 777 -R /opt/Blender/*
 将解压后的文件拷贝到目标文件夹中：
 \$sudo cp ./blender-2.79b-linux-glibc219-x86_64/* /opt/Blender/blender-2.79b -r
 然后创建一个 launcher：
 \$sudo touch /usr/share/applications/blender-2.79b.desktop
 编辑刚刚创建的文件：
 \$sudo gedit /usr/share/applications/blender-2.79b.desktop
 写入以下内容即可（注意：每行结尾不能有多余的空格、tab 等的，文件最后也不要有空行）：
 [Desktop Entry]
 Name=Blender-2.79b
 Name[zh_CN]=Blender-2.79b
 Comment=Blender-2.79b Client
 Exec=/opt/Blender/blender-2.79b/blender
 Icon=/opt/Blender/blender-2.79b/blender.svg
 Terminal=false
 Type=Application
 Categories=Application;
 Encoding=UTF-8
 StartupNotify=true
 GenericName=3D modeller
 GenericName[es]=modelador 3D
 GenericName[de]=3D-Modellierer
 GenericName[fr]=modeleur 3D
 GenericName[ru]=Редактор 3D-моделей
 Comment=3D modeling, animation, rendering and post-production
 Comment[es]=modelado 3D, animación, renderizado y post-producción
 Comment[de]=3D-Modellierung, Animation, Rendering und Nachbearbeitung
 Categories=Graphics;3DGraphics;
 MimeType=application/x-blender;

这样，Blender 就安装好了，这款软件视频编辑功能十分强大，具体怎么使用，可以自己找教程。

第三部分 批量安装一些常用软件

以下是粗粒度的安装方法，尤其适合 CentOS 等系统中软件的批量安装。(CentOS 安装以后缺少音频解码器声卡驱动等，没有声音，需要安装以下音视频中的音频解码器和声卡驱动等等，重启之后会正常)

一、预热：

```
gcc gcc-++ g++ ntfs-3g
```

二、常用：

```
createrepo cairo-dock thunderbird gimp evince feh zgv pho chromium p7zip p7zip-plugins  
p7zip-full p7zip-rar rar unrar grub-customizer
```

三、系统/编程：

```
gcc gcc-++ g++ ntfs-3g adb stardict fuse-ntfs-3g dict gnome-utils gnome-system-monitor axel  
wget filezilla docky azureus deluge shutter stardict supertux zsnes acroread kdevelop anjuta  
netbeans codeblocks monodevelop glibc.i686 glibc-devel.i686 zlib-devel.i686  
ncurses-devel.i686 anjuta cscope Insight flawfinder crosstool indent flashgot wxwidgets  
xulrunner.i686 libXtst.i686 libjpeg* libpng* freetype* openssh-server openssh-client openssh  
samba telnet-server telnet ssh xinetd fcitx fcitx* makehuman darktable entangle hugin python  
python3 python-pip sublime-text xournal tree network-manager *network-manager* *gdb*  
gdb autoss
```

四、音视频：

```
qt-recordmydesktop mvgather screenfetch pointdownload gparted k3b unetbootin ms-sys  
win32codecs mplayer mplayer* smplayer smplayer* gstreamer* kmplayer vlc potplayer  
xine quicktime vebvbox itunes totem-xine realplayer clenmentine audacious  
audacious-plugins-freeworld gstreamer-plugins-good gstreamer-plugins-bad  
gstreamer-plugins-ugly libtunepimp-extras-freeworld xine-lib-extras-freeworld ffmpeg
```

```
ffmpeg-libs gstreamer-ffmpeg xvidcore libdvdread libdvdnav lsdvd gstreamer-plugins-good
gstreamer-plugins-bad gstreamer-plugins-ugly  istanbul wink xvidcap pyvnc2swf
recordmydesktop gtk-recordmydesktop kplayer smplayer  vlc deepin-movie  gnome-mplayer
totem* kplayer* smplayer*  vlc* deepin-movie  gnome-mplayer* totem*
```

其他音视频解码器

```
gstreamer1-libav gstreamer1-plugins-bad-freeworld gstreamer1-plugins-base-tools
gstreamer1-plugins-ugly gstreamer1-plugins-bad-free gstreamer1-plugins-good
gstreamer1-plugins-base gstreamer1 gstreamer-plugins-good gstreamer-plugins-bad
gstreamer-plugins-ugly  gstreamer-ffmpeg ffmpeg libvdpau mpg123 mplayer mplayer-gui
```

五、CentOS7 没有声音解决办法

需要安装音频解码器、声卡驱动等等。可以按如下网址教程解决：

1、中文官网教程

<https://wiki.centos.org/zh/TipsAndTricks/MultimediaOnCentOS7>

2、转载自中文官网的教程

<http://www.cnblogs.com/wecnlove/p/4435520.html>

3、具体步骤

```
rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm
```

一般情况下，请不要执行下面两条命令：

```
sudo yum --enablerepo=elrepo-kernel install kernel-ml
sudo yum --enablerepo=elrepo-kernel install kernel-ml-devel
```

```
yum -y install
```

http://li.nux.ro/download/nux/dextop/el7/x86_64/nux-dextop-release-0-5.el7.nux.noarch.rpm

yum -y install

http://linuxdownload.adobe.com/adobe-release/adobe-release-x86_64-1.0-1.noarch.rpm

yum install flash-plugin

yum install icedtea-web

yum install vlc smplayer ffmpeg HandBrake-{gui,cli}

yum install libdvdcss gstreamer{,1}-plugins-ugly gstreamer-plugins-bad-nonfree
gstreamer1-plugins-bad-freeworld

六、配置 CentOS7 睡眠 休眠 关机 电源等行为

1、官网关于该文件的详解页面（英文）

<https://www.freedesktop.org/software/systemd/man/logind.conf.html>

2、官方关于该文件的详解页面（中文）

<http://www.jinbuguo.com/systemd/logind.conf.html>

3、普通设置教程

<https://www.bbsmax.com/A/amd01Lizge/>

4、具体配置过程

参看以上三个页面进行设置

七、其他

freecad librecad brl-cad krita mypaint gimp scribus inkscape imagemagick

八、附录：我的 fedora 20 系统已经安装的软件

gcc gcc-c++ g++ ntfs-3g bluefish juffed anjuta asymptote audacious azureus boxes brasero
cmyktool codeblocks darktable deluge devhelp dia docky emacs empathy evolution feh filezilla
fontforge fontmanager geary gedit gnomebaker gnusmalltalk gnu gparted gpick gthumb kdevelop
kmpc musique monodevelop okteta pan pdf-shuffler pidgin qt qt5 scribus shotwell shutter
supertuxkart supertux smplayer thunderbird transmission vlc xine zsnes kvm xen virtualbox samba
openssh-server openssh-client openssh samba telnet-server telnet ssh xinetd openssh-server
openssh-client openssh fcitx fcitx* makehuman darktable entangle hugin sublime-text
xournal kplayer* smplayer* vlc* deepin-movie* gnome-mplayer* totem* tree
network-manager *network-manager* *gdb* gdb

第四部分 Linux 字符界面日常系统的打造

注：1、本字符界面日常系统打造的教程，是以 CentO S 7 为例进行的，其他发行版（包括 FreeBSD 可以用同样的方法甚至命令进行。

2、本部分内容中虽然是针对村字符界面的，但是在字符界面的终端中也可进行同样的尝试和体验。

一、进入纯字符界面

进入纯字符界面有以下两种方法：

1、从字符界面转至字符界面

Linux 系统的运行级别分为 7 种，分别用从 0 到 5 的数字进行标记……，关于这部分内容，请参看博文：<https://www.jianshu.com/p/635e8480a75e>，我们平时使用的图形界面的运行级别是 5（在命令行输入命令 `$runlevel` 即可查看系统当前的运行级别）。

我们要转换至春字符界面，一般是转换到运行级别 3：

```
$sudo init 3
```

这样回车就可以开启纯字符模式了，然后输入用户名和密码登录系统。

2、设置开机自动进入纯字符界面

我们也可以设置内核启动参数，使得每次系统启动均自动进入纯字符模式（一般是设置时期使其进入运行级别 3。

在 CentO S 7 中使用命令：

```
$sudo systemctl set-default multi-user.target
```

然后，在下次重启和以后每次开机的时候，系统就会自动进入纯字符模式，其运行级别为 3。

在不清楚当前系统启动时的默认运行级别的时候，可以用 `$systemctl get-default` 命令查看，如果输出显示为 `multi-user.target`，说明当前为多用户字符界面，如果输出为 `graphical.target`，说明当前系统默认启动会进入图形界面。

如果想设置默认运行级别为图形界面，则运行：

```
$sudo systemctl set-default graphical.target
```

即可完成切换。

SysV 运行级别	Systemd 目标	注释
0	runlevel0.target, poweroff.target	中断系统 (halt)
1, s, single	runlevel1.target, rescue.target	单用户模式
2, 4	runlevel2.target, runlevel4.target, multi-user.target	用户自定义运行级别，通常识别为级别3。
3	runlevel3.target, multi-user.target	多用户，无图形界面。用户可以通过终端或网络登录。
5	runlevel5.target, graphical.target	多用户，图形界面。继承级别3的服务，并启动图形界面服务。
6	runlevel6.target, reboot.target	重启
emergency	emergency.target	急救模式 (Emergency shell)

图表 4-1-2 Linux 系统的运行级别

关于 CentOS 7 的切换默认运行级别以及其 systemctl 命令的使用，请查阅文章：

<https://www.jb51.net/article/136559.htm>

和文章：

https://blog.csdn.net/weixin_42665279/article/details/82732829

关于 CentOS 6 修改默认运行级别请参照文章：

<https://www.cnblogs.com/rouqinglangzi/p/11077443.html>

或者文章：

https://blog.csdn.net/qq_16234613/article/details/53924545

关于 systemctl 命令请参看文章：<https://www.jianshu.com/p/c498327f39d4>

3、纯字符界面下的终端切换

在纯字符界面下，Linux 拥有 12 个终端可以供切换，分别是 F1 到 F12，切换方法是：Alt+Fn，其中“n”是对应的编号。

4、纯字符界面使用数字小键盘

当系统运行级别在 3 级（纯字符界面）的时候，数字小键盘会失去作用。此时用 Alt+Num 键可关闭或启用数字小键盘。

二、中日韩字符系统&输入法——zhcon

中文字符系统的安装是十分重要的，如果不安装中文字符系统，则所有的中文文件名、文件内容、所有软件的字符输出等都会是乱码，有了这个，还可以输入中文。

关于 zhcon 中文、日文、韩文系统的安装也可以参看文章：

https://blog.csdn.net/f_066/article/details/41940999

<https://www.linuxidc.com/Linux/2011-02/31916.htm>

<https://blog.51cto.com/itchenyi/1024752>

首先在网站：<https://sourceforge.net/projects/zhcon/files/zhcon/> 查找目前最新或者任何时候的版本的目录（笔者这里最新的版本是 0.2.6），并进入目录，下载 0.2.5 的文件 [zhcon-0.2.5.tar.gz](#) 以及其补丁文件 [zhcon-0.2.5-to-0.2.6.diff.gz](#)（这个补丁文件可以将上面的 0.2.5 版源码包升级为 0.2.6 的源码包）。

或者可以通过命令行进行下载：

```
$wget -O zhcon-0.2.5.tar.gz
https://sourceforge.net/projects/zhcon/files/zhcon/0.2.6/zhcon-0.2.5.tar.gz/download
$wget -O zhcon-0.2.5-to-0.2.6.diff.gz
https://sourceforge.net/projects/zhcon/files/zhcon/0.2.6/zhcon-0.2.5-to-0.2.6.diff.gz/download
```

然后分别解压源码包：

```
$tar -zxvf ./zhcon-0.2.5.tar.gz
```

```
$gunzip ./zhcon-0.2.5-to-0.2.6.diff.gz
```

然后对 0.2.5 版源码打补丁，升级到 0.2.6 版：

```
$cd ./zhcon-0.2.5/
```

```
$patch -p1 < ../zhcon-0.2.5-to-0.2.6.diff
```

然后进行编译操作：

```
$mkdir -p ./build && cd ./build
```

有可能会在编译的过程中出现错误，提前预防一下：

```
$export LIBS="-lncurses"
```

```
$../configure
```

然后要在这里要修改几个错误，不然编译的过程中会遇到其他错误：

```
$sed -i 's/#include "fbdev.h"/#include "fbdev.h"\n#include
<string.h>/g' ../src/display/fblinear4.h
```

```
$sed -i 's/#include "fbdev.h"/#include "fbdev.h"\n#include
<string.h>/g' ../src/display/fblinear8.h
```

```
$sed -i 's/#include <string>/#include <string>\n#include <string.h>/g' ../src/basefont.h
```

```
$sed -i 's/ *p = (unsigned int) p + mpText;/ p = (unsigned long int) p +
mpText;/g' ../src/winime.cpp
```

```
$sed -i 's/#include <vector>/#include <vector>\n#include
<sys/select.h>/g' ../src/inputmanager.h
```

```
$sed -i 's/#include <string>/#include <string>\n#include
<stdlib.h>/g' ../src/inputclient.h ../src/configfile.h ../src/zhcon.h
```

```
$sed -i 's/#include <vector>/#include <vector>\n#include <stdlib.h>/g' ../src/inputmanager.h
```

```
$sed -i 's/#include "cmdline.h"/#include "cmdline.h"\n#include <stdlib.h>/g' ../src/cmdline.c
```

然后进行编译并安装：

```
$make -j4
```

如果编译过程中出现了其他错误，请参看

<https://blog.csdn.net/smtong/article/details/8730235> 或百度寻求解决。

```
$sudo make install
```

因为是在安装的时候指定了目录，因此现在要将 zhcon 软件的相关程序创建软连接：


```

$sudo ln -s /usr/local/zhcon-0.2.6/bin/zhcon /usr/bin/zhcon
$sudo ln -s /usr/local/zhcon-0.2.6/etc/zhcon.conf /etc/zhcon.conf
$sudo ln -s /usr/local/zhcon-0.2.6/lib/zhcon /usr/share/zhcon
$sudo ln -s /usr/local/zhcon-0.2.6/man/man1/zhcon.1 /usr/share/man/man1/zhcon.1.gz
$sudo ln -s /usr/local/etc/zhcon.conf /etc/zhcon.conf
$sudo ln -s /usr/local/bin/zhcon /usr/bin/zhcon

```

中文字符系统&中文输入法 zhcon 安装完成之后，启用它的方式是：

```
$zhcon --utf8
```

接着，运行了该命令的终端就能正常显示中文了。快捷键 **Ctrl+space**（也就是同时按下 **Ctrl+空格**）键来启用/停用中文输入法。快捷键 **Ctrl+Shift** 键实现不同输入法之间的切换，例如从双拼到全拼到五笔等。其他用法如下：

使用 **Ctrl+.**（句号）键切换中英文标点状态；

使用 **CTRL+,**（逗号）键切换全角/半角状态；

系统运行时可随时使用 **CTRL-ALT-H** 键来激活帮助窗口查看热键定义。

可能在运行该程序时会报错：

```
zhcon: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.20' not found (required by zhcon)
```

```
zhcon: /lib64/libstdc++.so.6: version `CXXABI_1.3.9' not found (required by zhcon)
```

```
zhcon: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.21' not found (required by zhcon)
```

原因是需要高版本的 GCC，[安装高版本 GCC 请看这里](#)，如果你已经安装过了高版本的 GCC，则首先查看其安装目录在哪里，然后创建对应的软连接即可（我的 GCC 的安装目录是 /usr/local/gcc-8.3.0）：

```
$sudo mv /usr/lib64/libstdc++.so.6 /usr/lib64/libstdc++.so.6.old
```

```
$sudo ln -s /usr/local/gcc-8.3.0/lib64/libstdc++.so.6.0.25 /usr/lib64/libstdc++.so.6
```

```
$sudo mv /lib64/libstdc++.so.6 /lib64/libstdc++.so.6.old
```

```
$sudo ln -s /usr/local/gcc-8.3.0/lib64/libstdc++.so.6.0.25 /lib64/libstdc++.so.6
```

该程序支持的输入法种类如图 4-2-1 所示（因为很多的输入法我的系统支持不完美，因此我将其禁用了，在前面加了“#”将其从配置文件中注释了）：

```

#ime = 智能拼音,modules/cce/cce_pinyin.so,modules/cce/dict,gb2312,unicon
ime = 全拼,,input/winy.py.mb,gb2312,native
ime = 五笔,,input/wb.mb,gb2312,native
ime = 双拼,,input/winsp.mb,gb2312,native
ime = ㄟ 30,,input/big5-ary30.mb,big5,native
ime = ㄟ,,input/big5-cj.mb,big5,native
ime = ㄟ,,input/big5-phone.mb,big5,native
ime = ㄟ,,input/big5-liu5.mb,big5,native
ime = GBK拼音,modules/turbo/TL_hzinput.so,modules/turbo/dict/gbk/gbkpy_mb.tab,gbk,unicon
ime = 自然码,modules/turbo/TL_hzinput.so,modules/turbo/dict/gb/zrm-2.tab,gb2312,unicon
ime = ㄟ ㄟ,modules/turbo/TL_hzinput.so,modules/turbo/dict/big5/pinyin.tab,big5,unicon
ime = ㄟ ㄟ,modules/turbo/TL_hzinput.so,modules/turbo/dict/big5/cj.tab,big5,unicon
ime = ㄟ ㄟ,modules/turbo/TL_hzinput.so,modules/turbo/dict/big5/simplex.tab,big5,unicon
ime = 全拼2,,input/py.mb,gb2312,native
ime = 双拼2,,input/py.mb,gb2312,native
ime = 大众,,input/dzm.mb,gb2312,native
ime = 英中,,input/ed.mb,gb2312,native
ime = 简拼,,input/jp.mb,gb2312,native
ime = 普通,,input/pt.mb,gb2312,native
ime = 五笔二维,,input/wbew.mb,gb2312,native
ime = 五笔划,,input/wbhb.mb,gb2312,native
ime = 繁体仓颉,,input/cjf.mb,gb2312,native
ime = 简体仓颉,,input/cjj.mb,gb2312,native

```

图表 4-2-1 zhcon 支持的输入法

因此这里的有些输入法是需要禁用掉的，我只留了全拼 2 和双拼 2。其他的都在前面加

了“#”注释掉了。像我一样的禁用方法是：

```
$sudo sed -i '/#type := native | unicon/,+23d' /etc/zhcon.conf
$sudo tee -ai /etc/zhcon.conf <<-'EOF'
#type := native | unicon
#ime = 智能拼音,modules/cce/cce_pinyin.so,modules/cce/dict,gb2312,unicon
#ime = 全拼,,input/winpy.mb,gb2312,native
#ime = 五笔,,input/wb.mb,gb2312,native
#ime = 双拼,,input/winsp.mb,gb2312,native
#ime = ㄣ 30,,input/big5-ary30.mb,big5,native
#ime = 縷,,input/big5-cj.mb,big5,native
#ime = 粽 ,,input/big5-phone.mb,big5,native
#ime = 礪郊 μ ,,input/big5-liu5.mb,big5,native
#ime = = GBK 拼
音,modules/turbo/TL_hzinput.so,modules/turbo/dict/gbk/gbkpy_mb.tab,gbk,unicon
#ime = 自然码,modules/turbo/TL_hzinput.so,modules/turbo/dict/gb/zrm-2.tab,gb2312,unicon
#ime = 惧块な,modules/turbo/TL_hzinput.so,modules/turbo/dict/big5/pinyin.tab,big5,unicon
#ime = 縷块 ,modules/turbo/TL_hzinput.so,modules/turbo/dict/big5/cj.tab,big5,unicon
#ime = 虜 块
,modules/turbo/TL_hzinput.so,modules/turbo/dict/big5/simplex.tab,big5,unicon
ime = 全拼 2,,input/py.mb,gb2312,native
ime = 双拼 2,,input/py.mb,gb2312,native
#ime = 大众,,input/dzm.mb,gb2312,native
#ime = 英中,,input/ed.mb,gb2312,native
#ime = 简拼,,input/jp.mb,gb2312,native
#ime = 普通,,input/pt.mb,gb2312,native
#ime = 五笔二维,,input/wbew.mb,gb2312,native
#ime = 五笔划,,input/wbh.mb,gb2312,native
#ime = 繁体仓颉,,input/cjf.mb,gb2312,native
#ime = 简体仓颉,,input/cjj.mb,gb2312,native
EOF
```

三、纯字符界面播放音频

笔者现整理网络上提供的几款字符界面的音乐播放器：

- 1、 cmus
- 2、 mp3blaster 其主页地址为： <http://mp3blaster.sourceforge.net/>
- 3、 moc 其主页地址为： <http://moc.daper.net/>
- 4、 herrie 其主页地址为： <http://herrie.info/>
- 5、 mplayer 其主页地址为： <http://www.mplayerhq.hu/design7/news.html>
- 6、 sox 其主页地址为： <http://sox.sourceforge.net/>
- 7、 pytone 其主页地址为： <http://www.luga.de/pytone/>
- 8、 pyradio 其主页地址为： <http://www.coderholic.com/pyradio/>
- 9、 ogg123 其主页地址为： <http://www.vorbis.com/>

10、 mpg123 其主页地址为: <http://www.mpg123.de/>

```
$sudo yum install cmus* mp3blaster* moc herrie sox pytone pyradio ogg123 mpg123
```

关于音乐播放器,笔者没有进行安装,总之[安装 mplayer](#)之后,就可以用它来播放音频了, mplayer 非常强大。

四、纯字符界面播放视频——MPlayer

1、安装 MPlayer (不包含皮肤)

网站: <http://www.mplayerhq.hu/MPlayer/releases> 包含 MPlayer 安装需要的各种组件(皮肤、源代码、编解码器等)。

首先需要[安装新版本的 gcc gcc-c++](#)。

然后安装音视频的编解码库以及 MPlayer (不需要 GUI) 界面。先尝试通过软件仓库安装:

```
$sudo yum install mpalyer mplayer*
```

如果软件仓库中没有这个软件,则需要从源码编译安装:

搜索并下载官网的 mplayer 源码包。官网下载 mplayer 源码以及编解码库,把编解码库解压并把里面的所有内容(注意不是文件夹本身,而是解压后文件夹里面的内容)放到 /usr/local/lib/codecs/下。

编译 mplayer 前的准备工作如下:

安装 yasm, 因为编译 mplayer 时需要使用安装方法:

```
$sudo yum install yasm*
```

然后还要安装 asla 开发库, 让 mplayer 支持 alsa 声卡驱动输出:

```
$sudo yum install alsa* madplay* libmad*
```

还需要提前安装 GTK+以及 GTK2 相关的东西:

```
$sudo yum install gtk+* gtk2*
```

源码下载:

```
$wget http://www.mplayerhq.hu/MPlayer/releases/MPlayer-1.4.tar.gz
```

下载需要的编解码库:

```
$wget http://www.mplayerhq.hu/MPlayer/releases/codecs/all-20110131.tar.bz2
```

解压编解码库 all-20110131.tar.bz2 并将其移动到/usr/local/lib/codecs/目录下:

```
$sudo mkdir -p /usr/local/lib/codecs
```

```
$tar -jxvf ./all-20110131.tar.bz2
```

```
$ sudo cp -r ./all-20110131/* /usr/local/lib/codecs/
```

然后解压 MPlayer 的源码, 并进行安装:

```
$ tar -zxvf ./MPlayer-1.4.tar.gz
```

```
$ cd ./MPlayer-1.4/
```

```
$/configure --prefix=/usr/local/mplayer-1.4 --codecsdir=/usr/local/lib/codecs/ --enable-gui  
--enable-freetype --language=zh_CN
```

```
$make -j4
```

```
$sudo make install && mkdir -p /opt/MPlayer-1.4_has-been-installed
```

至此 MPlayer 就安装成功了。

纯字符界面的视频播放简单来说有三种显示效果。分别是：

2、黑白字符组成的大致的图像显示

该视频播放方式可以将视频图像在远程终端中播放。

这需要库 libaa 的支持，首先安装这个库：`$sudo yum install libaa libaa*`

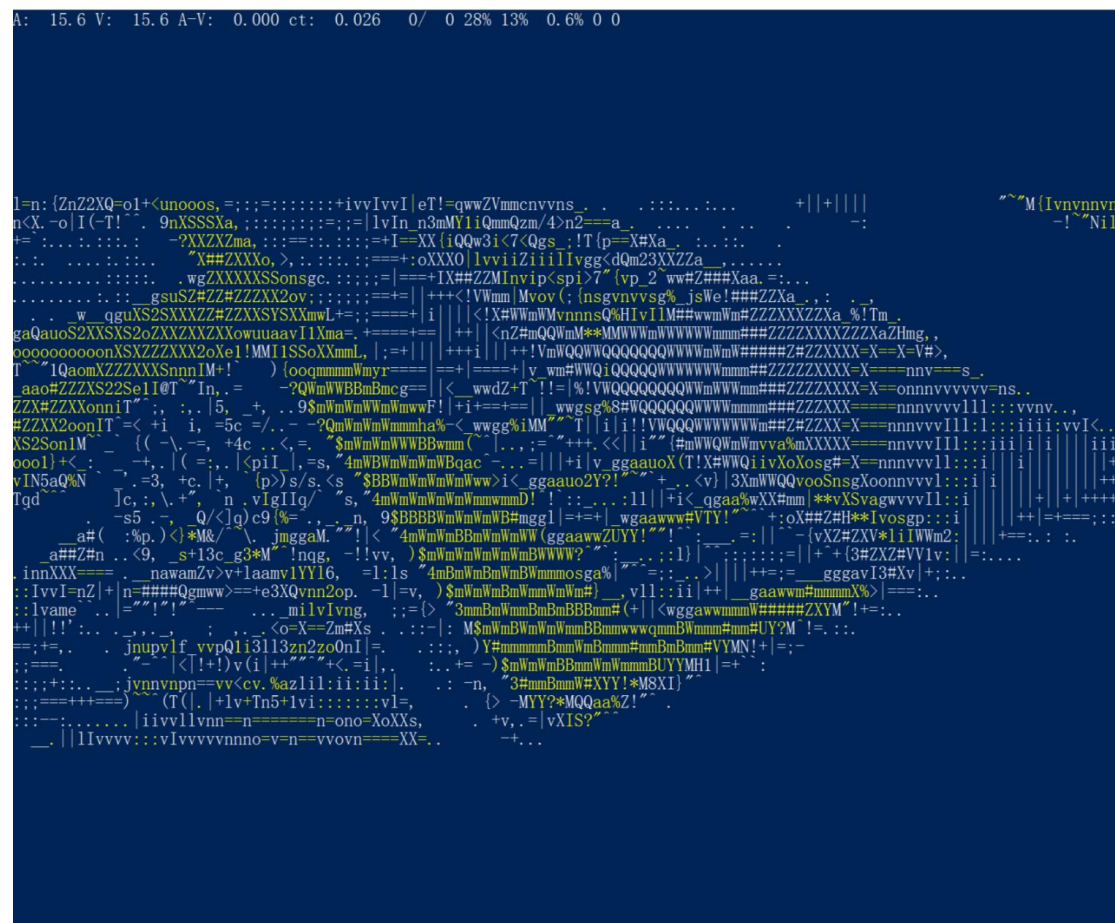
安装完成之后就可以试试播放效果了：

命令：`$ mplayer -vo aa filename`

在这个过程中，可以添加参数 `-aspect` 来指定视频显示比例。比如：

`$ mplayer -vo aa -aspect 4:1 filename`

效果如图表 4-4-2-1 所示。



图表 4-4-2-1 黑白字符效果

3、彩色字符组成的大致图像的显示

该视频播放方式可以将视频图像在远程终端中播放。

这需要库 libcac 的支持，首先安装这个库：`$sudo yum install libcaca libcaca*`

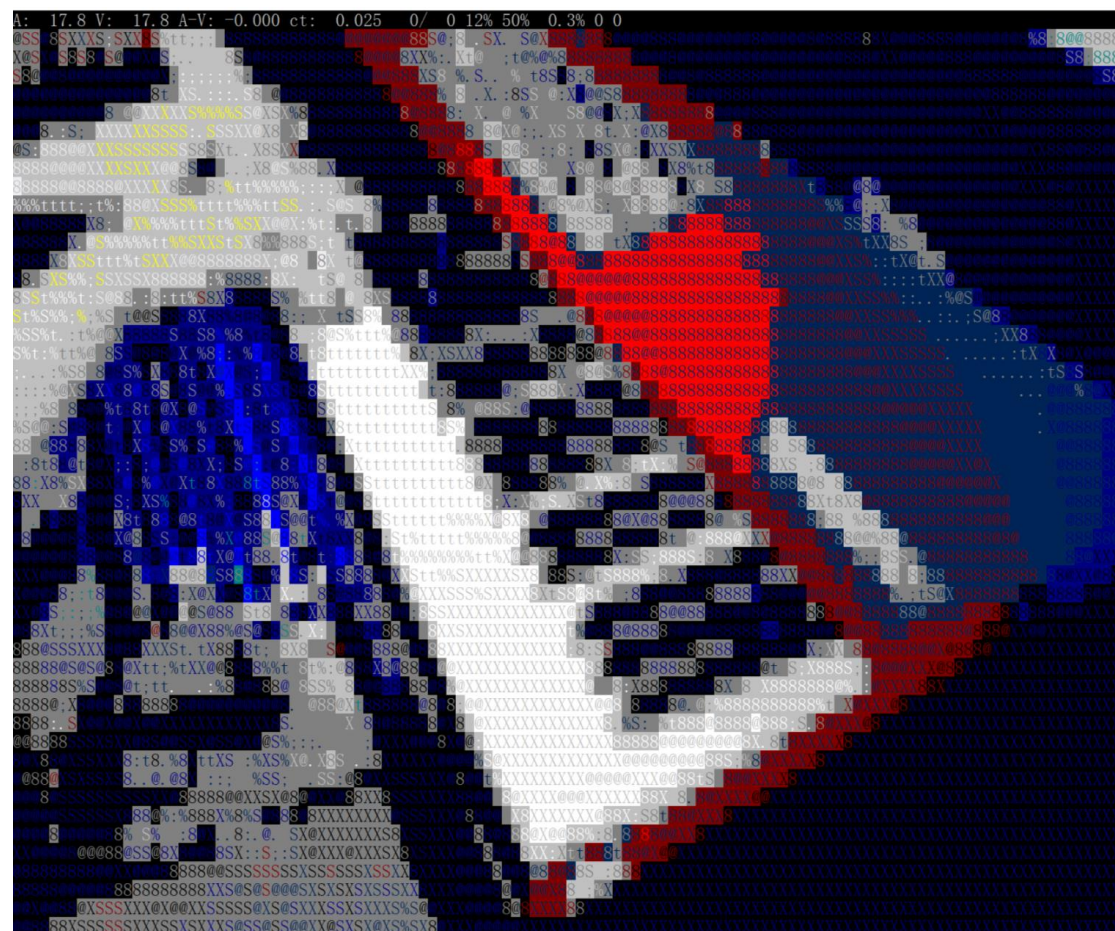
安装完成之后就可以试试播放效果了：

命令：`$ mplayer -vo caca filename`

在这个过程中，可以添加参数 `-aspect` 来指定视频显示比例。比如：

`$ mplayer -vo caca -aspect 4:1 filename`

效果如图表 4-4-3-1 所示。



图表 4-4-3-1 彩色字符效果

4、完美的视频效果

该视频播放方式不可将视频图像在远程终端中播放。

在这一步中，首先我们要开启系统的帧缓冲功能。

然后安装声卡驱动 `alsa-utils`：

`$sudo yum install alsa-utils*`

运用命令：`$alsamixer` 调整音量大小(向上箭头增大音量，向下箭头减小音量，Esc 退出)。

运行命令：\$aplay audioFileName 播放一个 wav 格式的音频文件，发出声音说明声卡工作正常了。

然后再修改一下 MPlayer 的权限（关于权限，可以参看 [Linux 文件权限详解](#)）：

```
$sudo chmod 4755 /usr/bin/mplayer
```

命令：

```
$ mplayer -vo fbdev2 -geometry 1200:0 -zoom -vf scale -x 400 -y 225 -ao alsa -loop 0 ./testVideo.flv
```

参数-vo fbdev2 的作用是播放视频的同时，可以通过 ALT+F2 等切换终端，-vo fbdev 则无法切换；

参数-geometry 1200:0 是指定视频播放时图像的左上角的坐标，即 x 坐标为 1200 像素处，y 坐标为 0 像素处；

参数-zoom -vf scale -x 400 -y 225 是指定视频播放视频的画面比例的，如果没有-zoom 选项-vf scale 将忽略-x / -y / -xy / -fs / -aspect 等选项，这里视频宽高比是 400:225，因为我是在写代码的时候看视频的，因此这个配置参数刚好将视频放在窗口右上角且大小比例合适（现在的视频比例一般为 16:9，因此如果希望视频宽度是 400，则视频高度是 $(400*9)/16$ 约等于 225。

参数-x 1024 指定视频画面的横向的宽度是 1024 像素；

参数-ao alsa 指定使用的音频解码器；

参数-loop 3 指定就当前视频循环播放 3 次，如果数字是 0，则无限循环播放；

Mplayer 除了可以播放单个视频，还可以进行列表播放，只需要加上参数 -playlist listText 就可以了。例如我在一个全是视频的目录下，生成列表的方式也很简单：

```
$ls /* > ./list
```

然后就可以播放这个列表了：

```
$ mplayer -vo fbdev2 -geometry 1200:0 -zoom scale -x 1024 -ao alsa -loop 0 ./list
```

这里因为有参数 -loop 0，所以会循环播放这个列表。

排错：

如果再播放的过程中，没有图像只有声音，或者只有图像没有声音，可能是因为 mplayer 权限的问题，执行命令：

```
$sudo chmod 4755 /usr/bin/mplayer
```

即可解决问题。

Mplayer 的其他用途：

1、用 mplayer 我们还可以边下载边播放视频：

```
$wget ftp://micorsops.com/something.avi -O - | mplayer -vo fbdev2 -geometry 1200:0 -zoom scale -x 1024 -ao alsa
```

2、还可以用其中带的工具将一个目录下的所有图片转换成视频：

```
$mencoder /*.jpg -mf on:fps=25 -ovc divx4 -o output.avi
```

3、可以用来录音：

```
$mplayer mms://202.***.***.***/test.asf -dumpstream -dumpfile MyMovie.asf
```

4、把 mplayer 可播放的视频进行格式转换：

```
mencoder yourfile.rm -ovc copy -oac copy -o testOut.avi
```

5、将视频分离出 MP3 文件：

```
mplayer -vo null -ao pcm music.wmv lame -h audiodump.wav music.mp3
```

mpplayer 的控制热键

基本播放：

→	前进 10 秒
←	后退 10 秒
↑	前进 60 秒
↓	后退 60 秒
PageUP	前进 10 分钟
PageDown	后退 10 分钟
Enter	全屏开关
Space	暂停开关
Esc	退出
q	退出

音量：

9	降低音量
0	增大音量
/	降低音量
*	增大音量
a	切换声道：若有多种语言音轨，在多音轨间切换； 否则，在左右声道与立体声间切换；

字幕：

o	切换屏幕显示键盘控制（比如音量）和时间的格式
v	切换是否显示字幕
j	在不同语言的字幕之间切换
z	字幕推迟显示 0.1 秒
x	字幕提前显示 0.1 秒
{	字幕提前显示 1 秒
}	字幕提前显示 10 秒
r	字幕位置向上 1%
t	字幕位置向下 1%

画面：

1	对比度减
2	对比度加
3	亮度减
4	亮度加
7	饱和度减，到最后就是黑白了
8	饱和度加
d	切换是否降低帧数，显卡跟不上去的时候使用，以避免影音不同步
s	截屏，截取的图片(png 格式)存放在视频所在文件夹下

其他非常用控制

<	暂停状态下，后退一帧
>	暂停状态下，前进一帧
F7	将播放速度减慢 0.25 倍速
F8	将播放速度加快 0.25 倍速
F6	将播放速度重设为 1.0 倍速

鼠标控制：

单击	暂停开关
双击	全屏切换
右键单击	切换前端显示-总是 播放时 总不
按滚轮	切换 OSD（同 o）
滚动滚轮	调节音量

MPlayer 的其他详细用法参看文章：<https://blog.csdn.net/u011857683/article/details/78378032>

Mplayer 的其他参数或用法参看文章：

<https://blog.csdn.net/xdw1985829/article/details/6675756>

<https://blog.csdn.net/futurepeter/article/details/5316014>

5、关于帧缓冲（frame buffer）功能

帧缓冲功能（就是 frame buffer，关于帧缓冲功能的相关知识请参看文章：<https://www.cnblogs.com/createyuan/p/4437912.html>），修改系统配置文件 /etc/sysconfig/grub，在其中“GRUB_CMDLINE_LINUX=”crashkernel=”行的最后的英文后引号前面插入字段：vga=795 或者插入：nomodeset，注意与其前面的内容（之前的内容一般都是参数“quiet”）之间要保留最少一个空格（这种开启帧缓冲功能，其实是设置一定的系统屏幕分辨率。关于 Linux 各个主流分支中如何开启帧缓冲功能的介绍，请参看文章：<https://blog.csdn.net/youngstonep/article/details/45010867>）。

关于 vga 后面的参数，可以安装软件 hwinfo 来查看系统支持的数值（参考文章：<https://blog.csdn.net/cscrazybing/article/details/19910285>）：

```
$sudo yum install hwinfo*
```

查看方式：

```
$sudo hwinfo -framebuffer
```

输出如图 4-4-5-1 所示：


```

[zhanghuoding@localhost ~]$ sudo hwinfg --framebuffer
[sudo] zhanghuoding 的密码:
02: None 00.0: 11001 VESA Framebuffer
[Created at bios.459]
Unique ID: rdCR.ku_DuShewhl
Hardware Class: framebuffer
Model: "Intel(R)Sandybridge Mobile Graphics Controller"
Vendor: "Intel Corporation"
Device: "Intel(R)Sandybridge Mobile Graphics Controller"
SubVendor: "Intel(R)Sandybridge Mobile Graphics Chipset Accelerated VGA BIOS"
SubDevice:
Revision: "Hardware Version 0.0"
Memory Size: 63 MB + 960 kB
Memory Range: 0x00000000-0x03ffffff (rw)
Mode 0x0360: 960x600 (+960), 8 bits
Mode 0x0361: 960x600 (+1920), 16 bits
Mode 0x0362: 960x600 (+3840), 24 bits
Mode 0x033c: 1920x1440 (+1920), 8 bits
Mode 0x034d: 1920x1440 (+3840), 16 bits
Mode 0x035c: 1920x1440 (+7680), 24 bits
Mode 0x033a: 1600x1200 (+1600), 8 bits
Mode 0x034b: 1600x1200 (+3200), 16 bits
Mode 0x035a: 1600x1200 (+6400), 24 bits
Mode 0x0307: 1280x1024 (+1280), 8 bits
Mode 0x031a: 1280x1024 (+2560), 16 bits
Mode 0x031b: 1280x1024 (+5120), 24 bits
Mode 0x0305: 1024x768 (+1024), 8 bits
Mode 0x0317: 1024x768 (+2048), 16 bits
Mode 0x0318: 1024x768 (+4096), 24 bits
Mode 0x0312: 640x480 (+2560), 24 bits
Mode 0x0314: 800x600 (+1600), 16 bits
Mode 0x0315: 800x600 (+3200), 24 bits
Mode 0x0301: 640x480 (+640), 8 bits
Mode 0x0303: 800x600 (+832), 8 bits
Mode 0x0311: 640x480 (+1280), 16 bits
Config Status: cfg=new, avail=yes, need=no, active=unknown
[zhanghuoding@localhost ~]$

```

图表 4-4-5-1 hwinfgode 查看的信息

即:

```
$sudo vimx /etc/sysconfig/grub
```

然后编辑/etc/sysconfig/grub 文件并保存。也可以直接用下面的命令直接修改,不用打开文件/etc/sysconfig/grub:

直接进行修改(插入 nomodeset):

```
$sudo sed -i "s/GRUB_CMDLINE_LINUX=\"crashkernel=auto rhgb
quiet\"/GRUB_CMDLINE_LINUX=\"crashkernel=auto rhgb quiet nomodeset\"/g"
/etc/sysconfig/grub
```

或者进行修改(插入 vga=795):

```
$ sudo sed -i "s/GRUB_CMDLINE_LINUX=\"crashkernel=auto rhgb
quiet\"/GRUB_CMDLINE_LINUX=\"crashkernel=auto rhgb quiet vga=795\"/g"
/etc/sysconfig/grub
```

修改完成之后,执行命令(如果想要修改生效,则一定要执行该命令):

```
$sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

将修改写入到系统启动文件中,运行这条命令之后(重启就自动开启帧缓冲功能了),系统内核启动配置文件中 linux16 字段显示如图 4-4-5-2 所示:

```

dos6 --hint='hd0,msdos6' '93b40b50-842b-4f17-95b6-cealba7f9a70
else
search --no-floppy --fs-uuid --set=root 93b40b50-842b-4f17-95b6-cealba7f9a70
fi
linux16 /boot/vmlinuz-3.10.0-957.21.3.el7.x86_64 root=UUID=93b40b50-842b-4f17-95b6-cealba7f9a70 ro crashkernel=a
uto rhgb quiet vga=795
initrd16 /boot/initramfs-3.10.0-957.21.3.el7.x86_64.img

```

图表 4-4-5-2 修改后的/boot/grub2/grub.cfg 配置文件

五、纯字符界面调整音量

安装软件 alsas 系列,就可以在字符界面调整音量了:

```
$sudo yum install alsa*
```

然后就可以使用命令\$ alsamixer 打开音量控制页面，并使用上下键来调整系统音量。

六、纯字符界面查看硬件信息

- 1、命令 lscpu 查看 cpu 信息：

```
$lscpu
```

- 2、命令 lshw 显示硬件信息列表。这个命令用途广泛，它可以根据你的需求以详细或概要的形式列出许多个不同硬件的参数信息，其中包括 cpu，内存，磁盘，usb 控制器，网卡等等，lshw 实质上是从/proc 文件目录下的不同文件中提取对应的硬件信息。

```
$ sudo lshw -short
```

- 3、命令 hwdm 是一个使用非常广泛的硬件信息查询命令，类似于 lshw，它同样可以详细或概要地报告多个部件的硬件信息，但不同的是，它有时可以给出比 lshw 更多的信息。

```
$sudo hwdm --short
```

- 4、命令 lspci - 显示 pci 总线相关信息,可以列出 pci 总线信息，以及所有连接到 pci 总线上的设备信息，比如 VGA 适配器，显卡，网卡，usb 端口，sata 控制器等等。

```
$lspci
```

```
$lspci -v | grep "VGA" -A 12
```

- 5、命令 lsccsi 列出 scsi/sata 设备信息，比如硬盘驱动器，光盘驱动器。

```
$ lsccsi
```

```
[3:0:0:0] disk ATA ST3500418AS CC38 /dev/sda
```

```
[4:0:0:0] cd/dvd SONY DVD RW DRU-190A 1.63 /dev/sr0
```

- 6、命令 lsusb - 列出 usb 总线信息，及设备信息。该命令可以显示 USB 控制器及连接到控制的设备信息。默认情况下，只显示概要信息，可以通过-v 选项指示其显示更多的硬件信息。

```
$ lsusb
```

- 7、命令 lxi 是一个大的 bash 脚本文件，其通过从不同的信息来源和命令中获取硬件信息，最后能给出一个颜色高亮的，适合于非专业人员阅读的硬件信息报告。

```
$lxi -Fx
```

- 8、命令 lsblk - 列出块设备信息：

```
$lsblk
```

七、纯字符界面挂载 NAS 盘到本地

在字符界面下，我们想打开 NAS 上的文件，就不能通过浏览器或者资源管理器等的了，因此就需要手动命令行挂载，挂载方法见 [Linux 挂载磁盘](#)。

八、纯字符界面的 IM（即时通讯）工具

其实对于中国来说，Linux 上的即时通讯工具几乎不能用，且不友好，因此，笔者只简单列出两个（笔者也没有进行尝试）：

```
$sudo yum install finch* centerim*
```

其中 CenterIM 的主页地址为：http://www.centerim.org/index.php/Main_Page

九、纯字符界面的图片查看器 fbi

该工具是 fbida 工具包中的一员（该工具包共携带四个工具，分别是 fbi, fbgs, ida, exiftran）。

一般图片使用 fbida，可以查看图片，安装方法：

```
$sudo yum install fbida*
```

然后给该软件修改权限（否则只有在 root [权限](#)下才能查看图片）：

```
$sudo chmod 4755 /usr/bin/fbi
```

然后就可以正常使用了，使用方法为：

```
$fbi filename
```

也可以通过一个文本文件列出所有图片的文件名或者路径，进行批量查看：

```
$fbi -list picList
```

可以使用方向键来在大图片中移动视野，使用 + 和 - 来缩放，或者使用 r 或 l 来向右或向左旋转 90 度。Escape 键则可以关闭查看的图片。fbi 还支持自动缩放。还可以使用 -a 选项来控制缩放比例。--autoup 和 --autodown 则是用于告知 fbi 只进行放大或者缩小。要调整图片切换时淡入淡出的时间则可以使用 --blend [时间] 来指定一个以毫秒为单位的时间长度。使用 k 和 j 键则可以切换文件列表中的上一张或下一张图片，使用 Enter 键来切换播放列表中的下一张图片。fbi 还提供了命令来为你浏览过的文件创建文件列表，或者将你的命令导出到文件中，以及一系列其它很棒的选项。你可以通过 man fbi 来查阅完整的选项列表。

由于该工具显示的图片一般都是默认放大的，因此加参数--autodown 会显示正常的图片：

```
$fbi --autodown fileName
```

十、纯字符界面的 PDF 查看器 fbgs

笔者已成功安装该工具但无法正常使用。

fbgs 是 fbida（[安装方法](#)）包中提供的一个 PDF 阅读器（该工具包共携带四个工具，分别是 fbi, fbgs, ida, exiftran）。它可以设置页面大小、分辨率、指定页码以及绝大部分 fbi 所提供的选项，当然除了一些在 man fbgs 中列举出来的不可用选项。我主要用到的选项是页面大小，你可以选择 -l、xl 或者 xxl：

```
$sudo yum install fbida
```

```
$sudo chmod 4755 /usr/bin/fbgs
```

```
$fbgs -xl filename.pdf
```

十一、纯字符界面下炫酷的屏保

纯字符界面下也有一些炫酷的屏保程序：cmatrix（官网已经无法下载到目标文件了似乎），其他途径的老版本的源码包为：

<https://jaist.dl.sourceforge.net/project/cmatrix/cmatrix/1.2a/cmatrix-1.2a.tar.gz>

有人在 GitHub 上进行该软件的维护以及版本更新工作（推荐这个版本，下图 4-5-11-1 所示，就是该程序 GitHub 仓库中放出来的示例），地址是：

<https://github.com/abishekvashok/cmatrix>

安装步骤：

可能在安装之前有些准备工作需要完成：

```
$sudo yum install ncurses* gcc gcc-*
```

```
$wget
```

<https://github.com/abishekvashok/cmatrix/releases/download/v2.0/cmatrix-v2.0-Butterscotch.tar>

```
$tar -xvf ./cmatrix-v2.0-Butterscotch.tar
```

```
$cd ./cmatrix && ./configure && make && sudo make install
```

只能装过程特别简单也特别顺利。一下子就装好了。

一些常用参数（也可以使用命令 `$man cmatrix` 来查看帮助手册）：

-a: 异步滚动

-b: 启用粗体字符

-b: 所有粗体字符（覆盖 -b）

-f: 强制启用 linux\$term 类型

-l: linux 模式（使用矩阵控制台字体）

-o: 使用旧式滚动

-h: 打印使用和退出

-n: 无粗体字符（覆盖 -b 和 -b，默认）

-s: “屏幕保护程序”模式，在第一次按键时退出

-x: 窗口模式，如果您的 xterm 使用 mt.x.pcf，则使用

-v: 打印版本信息并退出

-u 延迟（0-10，默认 4）：屏幕更新延迟

-C [颜色]: 将此颜色用于矩阵（默认为绿色），支持 green (默认), red, blue, white, yellow, cyan, magenta 和 black;

例如: 使用红色

```
cmatrix -b -C red
```

使用蓝色

```
cmatrix -b -C blue
```

在运行状态下,使用 0-9 数字,可以改变运行速度快慢.

据说在运行的时候按下面这些键会有意外的收获哦!

a B b n

1~9

! @ # \$

使用-s 参数运行时以上按键无效。

关于该程序的安装和使用可以参考以下文章:

<https://blog.5lcto.com/5232821/2146115?source=dra>

<https://blog.csdn.net/springyh/article/details/79204695>

<https://www.jianshu.com/p/3fe67d85850b>



图表 4-5-11-1 CMatrix 的屏保效果图

十二、纯字符界面下的串口工具

这部分内容网络工程师的需求会比较多一些。

命令\$lsusb 可以显示 usb 设备的信息。

在串口工具中, minicom 是一个超级终端, 安装:

```
$sudo yum install minicom
```

运行\$minicom -s 设备名、速率等参数。

十三、纯字符界面下的屏幕截图工具

笔者已成功安装该工具但无法正常使用。

该工具的主页地址：<https://fbgrab.monells.se/>

首先安装一些依赖：

```
$sudo yum install splint libpng libpng* zlib splint* libpng*  
zlib* libjpeg libjpeg*
```

在纯字符界面下，屏幕截图工具还是实分有必要的：

```
$wget https://fbgrab.monells.se/fbgrab-1.3.tar.gz  
$tar -zxvf ./fbgrab-1.3.tar.gz && cd ./fbgrab-1.3  
$make  
$sudo make install
```

即可安装成功，但是还需要修改权限：

```
$sudo chmod 4755 /usr/bin/fbgrab
```

然后该工具就可以使用了，使用方法是：

```
$fbgrab -z 9 path/fileName.png
```

即可完成截屏并保存成 png 文件。为了不破坏图像显示，可以使用 fbgrab 的 -s 参数，或者使用 ssh 远程登陆操作。

使用 -c 参数可以实现为指定终端截图。这一操作需要 root 权限。如

```
$sudo fbgrab -c 2 ./b.png
```

即可对 Ctrl-Alt-F2 进入的终端截图。

也可同时使用 -C 参数延时，如：

```
$sudo fbgrab -c 2 -C 3 ./c.png
```

使用 -z 参数制定图片压缩率，0 为最高比率的压缩，9 为不压缩，保存原始最好效果的图像。

十四、纯字符界面下的网页浏览器

纯字符界面下的网页浏览器有：lynx elinks links w3m 等，但都十分不推荐，因为显示是在太乱了，各种问题。

十五、图片信息 (exi) 修改及图片旋转工具 exiftran

该工具是 fbida 工具包中的一员（该工具包共携带四个工具，分别是 fbi, fbgs, ida, exiftran），其中的 exiftran 可以修改图片的信息以及旋转图片。[这里是安装方法](#)。

关于该工具的使用，可以查看网站（<https://linux.die.net/man/1/exiftran>）的说明：

exiftran is a command line utility to transform digital image jpeg images. It can do lossless rotations like jpegtran, but unlike jpegtran it cares about the EXIF data: It can rotate images automatically by checking the exif orientation tag, it updates the exif informaton if needed (image dimension, orientation), it also rotates the exif thumbnail. It can process multiple images

at once.

Transform Options

-a

automatic (using exif orientation tag)

-9

rotate by 90 degrees

-1

rotate by 180 degrees

-2

rotate by 270 degrees

-f

flip vertical

-F

flip horizontal

-t

transpose

-T

transverse

-nt

Don't rotate exif thumbnail.

-ni

Don't rotate jpeg image. You might need this or or the -nt option to fixup things in case you rotated the image with some utility which ignores the exif thumbnail. Just generating a new thumbnail with -g is another way to fix it.

-no

Don't update the orientation tag. By default exiftran sets the orientation to "1" (no transformation needed) to avoid other exif-aware applications try to rotate the already-rotated image again.

Other Options

-h

print a short help text

-d

Dump exif data for the file(s).

-c <text>

Set jpeg comment tag to <text>.

-g

(re)generate EXIF thumbnail.

-o <file>

Specify output file. Only one input file is allowed in this mode.

-i

Enable inplace editing of the images. Exiftran allows multiple input files then. You must specify either this option or a output file with -o for all operations which modify the image (i.e. everything but -d right now).

-b

Create a backup file when doing inplace editing.

-p

Preserve timestamps (atime + mtime) when doing inplace editing.

Examples

Autorotate all jpeg files in the current directory:

```
$exiftran -ai *.jpeg
```

十六、纯字符界面的多窗口管理器

多窗口管理器推荐 `tmux` 和 `screen`。`tmux` 应用程序的名称来源于终端（terminal）复用器（muxer）或多路复用器（multiplexer）。换句话说，它可以将您的单终端会话分成多个会话。它管理窗口和窗格：

1. 窗口（window）是一个单一的视图 - 也就是终端中显示的各种东西。
2. 窗格（pane）是该视图的一部分，通常是一个终端会话。

据网上的帖子都比较推荐 `tmux`。`Tmux` 的使用可以参考：

<https://www.linuxprobe.com/better-screen-tmux.html>

官网手册：<http://man.openbsd.org/OpenBSD-current/man1/tmux.1>

这两个工具安装方法都很简单：

```
$sudo yum install tmux* screen*
```

两个工具的参考文章：

<https://www.cnblogs.com/bamanzi/p/switch-tmux-to-gnu-screen.html>

<https://blog.csdn.net/hongchangfirst/article/details/37818947>

`tmux` 使用：

```
$tmux
```

即可进入程序，使用 `exit` 命令即可退出程序。

十七、纯字符界面的文件管理器

文件管理器有 `ranger` 和 `mc` 等，其他笔者没有试用。`ranger` 也并未安装成功。

`mc` 安装方法为：

```
$sudo yum install mc
```

该工具支持在远程终端中使用，直接在终端运行 `mc` 即可进入程序：

```
$mc
```

即可进入命令行的文件管理页面，根据提示可以操作和退出程序。

十八、纯字符界面的任务管理器

网上比较推荐 htop，这个工具是 top 的加强版。安装：

```
$sudo yum install htop* top*
```

该工具在远程终端中也可以使用：

```
$htop
```

即可进入命令行的任务管理页面，根据提示可以操作和退出程序。

十九、纯字符界面的视频编辑工具 FFmpeg（命令行工具）

FFmpeg 是一个非常强大的视频工具，可以在命令行快速处理视频，包括视频转换为图片、图片转换为视频、视频格式转换、视频剪切、视频添加水印、视频去水印、视频音频分离、视频音频合成、修改码率等等，功能十分强大。

简单的从视频分离出音频，从图片生成视频等功能 mplayer 工具也可以完成，参见[纯字符界面下的视频播放器 mplayer](#)。

1、安装

安装方法：

```
$sudo yum install ffmpeg* --skip-broken
```

使用示例，将视频转换为图片：

```
$ffmpeg -i ./input.mp4 ./test%d.jpg
```

视频格式转换：

```
$ffmpeg -i ./input.mp4 ./output.avi
```

2、视频剪切

在实际使用中，很多时候需要进行视频剪切，命令如下：

```
$ffmpeg -ss 0 -t 26.7 -accurate_seek -i ./input.mp4 -codec copy -avoid_negative_ts
```

```
1 ./output.mp4
```

在这条命令中，参数顺序不要改变，一定要保证参数“-ss 0 -t 26.7 -accurate_seek”在输入文件“-i ./input.mp4”的前面，因此最好不要改变顺序，否则会出现视频头部黑屏、时间定位不准等问题。这里，指定截取视频从 0 秒开始，长度为 26.7 秒。

3、视频拼接方法一

多段视频的拼接命令如下（以拼接两段视频为例）：

首先进行视频的转流（这步是必须的）：

```
$ffmpeg -i ./input_0.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts ./input_0.ts
```

```
$ffmpeg -i ./input_1.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts ./input_1.ts
```

然后将转流后的视频文件合并并同时输出为普通视频文件：

```
$ffmpeg -i "concat:input_0.ts|input_1.ts" -c copy -bsf:a aac_adtstoasc -movflags +faststart output.mp4
```

现在 output.mp4 文件就是最终的合并后的视频文件了，现在就可以删除刚刚创建的中间文件了：

```
$rm ./input_0.ts ./input_1.ts
```

关于视频剪切和合并的命令请查看文章：

https://blog.csdn.net/matrix_laboratory/article/details/53157383

<https://blog.csdn.net/east196/article/details/79821361>

4、视频拼接方法二

另外还有一种成功率较高的视频拼接方法（需要 ffmpeg 1.1 版本及以上）：

首先创建一个文本文件 filelist，把需要合并的文件按顺序列进去（文件内容如下）：

```
file 'input_0.mp4'
```

```
file 'input_1.mp4'
```

然后进行视频合并：

```
$ffmpeg -f concat -i filelist -c copy output.mp4
```

注意：使用这种方法（FFmpeg concat 分离器）时，如果文件名有奇怪的字符，要在 filelist.txt 中转义。

该方法参看文章：<https://www.jianshu.com/p/a9bccc12229b>

5、关于 ffmpeg 的其他使用教程

其他诸多功能，请查看命令手册：

<https://www.cnblogs.com/frost-yen/p/5848781.html>

<https://blog.csdn.net/xuyankuanrong/article/details/77527381>

<https://www.cnblogs.com/dwdxdy/p/3240167.html>

<http://linux.51yip.com/search/ffmpeg>

二十、纯字符界面下的网易云音乐客户端

网易云音乐开发了桌面客户端，[安装方法](#)见前文。有大神开发了网易云音乐的纯字符界面的命令行客户端（名为 NetEase-MusicBox），功能非常全，使用方法也很友好，支持账号和手机号码登录等，详细了解请移步该项目的 GitHub 主页：<https://github.com/darknessomi/musicbox> 其功能特性如下：

1. 320kbps 的高品质音乐
2. 歌曲，艺术家，专辑检索
3. 网易 22 个歌曲排行榜
4. 网易新碟推荐

5. 网易精选歌单
6. 网易主播电台
7. 私人歌单，每日推荐
8. 随心打碟
9. 本地收藏，随时加❤️
10. 播放进度及播放模式显示
11. 现在播放及桌面歌词显示
12. 歌曲评论显示
13. 一键进入歌曲专辑
14. 定时退出
15. Vimer 式快捷键让操作丝般顺滑
16. 可使用数字快捷键
17. 可使用自定义全局快捷键

使用该命令行的客户端，需要安装中文环境才能支持正常显示。[安装中文环境点这里](#)。

接下来进行安装工作，首先安装支持和依赖：

```
$sudo yum install aria* libnotify* dbus* qt* python-* --skip-broken
$sudo ln -s /usr/bin/aria2c /usr/bin/aria2
$sudo ln -s /usr/bin/aria2c /usr/bin/aria
$sudo pip3 install --upgrade pip
$sudo pip3 install sip
$wget
```

<https://www.riverbankcomputing.com/static/Downloads/sip/4.19.17/sip-4.19.17.tar.gz>

```
$tar -zxvf ./sip-4.19.17.tar.gz && cd ./sip-4.19.17
$python3 ./configure.py
$make -j4
$sudo make install
$wget
```

https://nchc.dl.sourceforge.net/project/pyqt/PyQt4/PyQt-4.12.3/PyQt4_gpl_x11-4.12.3.tar.gz

```
$tar -zxvf ./PyQt4_gpl_x11-4.12.3.tar.gz && cd ./PyQt4_gpl_x11-4.12.3
$echo "yes" | python3 ./configure.py -q /usr/lib64/qt4/bin/qmake
$make -j4
$sudo make install
```

然后安装依赖 mpg123 用来播放歌曲：

```
$wget
```

http://mirror.centos.org/centos/7/os/x86_64/Packages/mpg123-1.25.6-1.el7.x86_64.rpm

```
$sudo yum install ./mpg123-1.25.6-1.el7.x86_64.rpm
```

还要修改一下 mpg123 的参数，否则播放音乐会出错：

```
$sudo sed -i '/ "mpg123_parameters": {/{n;s/ *"value": "\\[/,/"value":
["-b","144"],/g}' ~/.netease-musicbox/config.json
```

接下来就可以安装网易云音乐命令行客户端 NetEase-MusicBox 了：

```
$sudo pip3 install NetEase-MusicBox
```

查找客户端的安装目录并为该客户端建立软连接，方便打开：

```
$sudo find /usr/ -name musicbox
```

由此可以找到刚才安装的音乐目录在：

```
$sudo ln -s /usr/local/python3/bin/musicbox /usr/local/bin/musicbox
```

```
$sudo ln -s /usr/local/python3/bin/musicbox /usr/local/bin/music
```

至此网易云音乐命令行客户端就安装成功了，运行方法如下（在此之前首先运行中文环境）：

```
$musicbox
```

或者

```
$music
```

键盘快捷键

Key	Effect			Key	Effect	
j	Down	下移		a	Add	添加曲目到打碟
k	Up	上移		A	Enter album	进入专辑
h	Back	后退		g	To the first	跳至首项
l	Forward	前进		G	To the end	跳至尾项
u	Prev page	上一页		z	DJ list	打碟列表
d	Next page	下一页		s	Star	添加到收藏
f	Search	快速搜索		c	Collection	收藏列表
[Prev song	上一曲		r	Remove	删除当前条目
]	Next song	下一曲		J	Move Down	向下移动当前项目
=	Volume +	音量增加		K	Move Up	向上移动当前项目
-	Volume -	音量减少		C	Cache	缓存歌曲到本地
Space	Play/Pause	播放/暂停		,	Like	喜爱
?	Shuffle	手气不错		.	Trash FM	删除 FM
m	Menu	主菜单		/	Next FM	下一 FM
p	Present/History	当前/历史播放列表		q	Quit	退出
i	Music Info	当前音乐信息		t	Timing Exit	定时退出
P	Playing Mode	播放模式切换		w	Quit&Clear	退出并清除用户信息

关于网易云音乐命令行客户端的安装以及问题解决等也可以参看以下文章：

<https://github.com/darknessomi/musicbox>

<https://blog.csdn.net/zwkm6936367/article/details/29016179>

<https://my.oschina.net/u/2378180/blog/855632>

https://www.cnblogs.com/everfight/archive/2017/09/17/linux_soft_link.html

<https://github.com/darknessomi/musicbox/issues/374>

二十一、纯字符界面下的百度网盘命令行第三方客户端

纯字符界面下大神开发的命令行百度网盘客户端名叫 BaiduPCS-Go, [GUI 的官方客户端安装方法点这里](#)。

该命令行客户端很多文件操作命令都同 Linux 命令行是一样的, 包括 cp、mv、ls、cd 等命令。安装方法:

```
$wget
```

<https://github.com/iikira/BaiduPCS-Go/releases/download/v3.5.6/BaiduPCS-Go-v3.5.6-linux-amd64.zip>

```
$unzip ./BaiduPCS-Go-v3.5.6-linux-amd64.zip
```

```
&sudo mkdir -p /opt/Baidu/BaiduPCS-Go-v3.5.6-linux-amd64
```

```
$sudo cp -r ./BaiduPCS-Go-v3.5.6-linux-amd64/*  
/opt/Baidu/BaiduPCS-Go-v3.5.6-linux-amd64/
```

```
$sudo ln -s /opt/Baidu/BaiduPCS-Go-v3.5.6-linux-amd64/BaiduPCS-Go  
/usr/local/bin/baidupcs
```

到这里就安装好了, 运行方法为:

```
$baidupcs
```

然后可以在程序中输入 help 来显示帮助信息, 帮助信息包括该命令行客户端的详细使用说明。输入命令 exit 即可退出客户端。

关于该第三方客户端信息, 可参看网址:

<https://github.com/iikira/BaiduPCS-Go#linux--macos>

该客户端还有另外两个变体, 分别被不同层次地包装成了 GUI 软件, 地址如下:

<https://github.com/liuzhuoling2011/baidupcs-web>

<https://github.com/user1121114685/baidupcsweb>

二十二、纯字符界面下的 csv 文件处理器

科研工作中 csv 文件非常重要, 但是命令行不能使用 excel 之类的工具, 命令行的 csv 文件查看处理工具功能十分强大, 关于工具的介绍 (包括用法), 可以参看文章:

<https://zhuanlan.zhihu.com/p/62391322>

<https://codeday.me/bug/20170407/8750.html>

这里介绍强大的工具集 csvkit, 安装方法如下:

```
$sudo pip3 install update
```

```
$sudo pip3 install csvkit
```

安装完成之后找到该工具集的安装目录:

```
$sudo find /usr/ -name in2csv
```

然后为其建立软连接到系统变量所指的目录下 (查看到我系统中该 csv 工具箱被安装到

了/usr/local/python3/bin/目录下)：

```
$sudo ln -s /usr/local/python3/bin/in2csv /usr/local/bin/in2csv
$sudo ln -s /usr/local/python3/bin/csvformat /usr/local/bin/csvformat
$sudo ln -s /usr/local/python3/bin/csvcut /usr/local/bin/csvcut
$sudo ln -s /usr/local/python3/bin/csvgrep /usr/local/bin/csvgrep
$sudo ln -s /usr/local/python3/bin/csvsql /usr/local/bin/csvsql
$sudo ln -s /usr/local/python3/bin/csvclean /usr/local/bin/csvclean
$sudo ln -s /usr/local/python3/bin/csvjoin /usr/local/bin/csvjoin
$sudo ln -s /usr/local/python3/bin/csvstat /usr/local/bin/csvstat
$sudo ln -s /usr/local/python3/bin/csvpy /usr/local/bin/csvpy
$sudo ln -s /usr/local/python3/bin/csvjson /usr/local/bin/csvjson
$sudo ln -s /usr/local/python3/bin/csvsort /usr/local/bin/csvsort
$sudo ln -s /usr/local/python3/bin/csvstack /usr/local/bin/csvstack
$sudo ln -s /usr/local/python3/bin/csvlook /usr/local/bin/csvlook
$sudo ln -s /usr/local/python3/bin/sql2csv /usr/local/bin/sql2csv
```

到这里该工具集就安装成功了。另外再安装一些辅助应用，方便处理 csv 文件：

```
$sudo yum install head tail more less header column body cols
```

Csvkit 工具集的使用示例（将 xlsx 文件转化为 csv 文件）：

```
$in2csv data/imdb-250.xlsx > data/imdb-250.csv
```

csvlook 可用于查看 csv 文件并格式化，--max-rows 设置行数，--max-columns 设置列数。

csvstat 统计列最大，最小，均值，unique 等统计值，参数如下：

- ① --max (maximum)
- ② --min (minimum)
- ③ --sum (sum)
- ④ --mean (mean)
- ⑤ --median (median)
- ⑥ --stdev (standard deviation)
- ⑦ --nulls (whether column contains nulls)
- ⑧ --unique (unique values)
- ⑨ --freq (frequent values)
- ⑩ --len (max value length)

关于 csv 文件的其他处理方式请参看文章：

<https://cloud.tencent.com/developer/ask/28421>

二十三、纯字符界面下的 ftp 工具

命令行的 ftp 工具主要有 ftp 和 lftp，安装方法如下：

```
$sudo yum install ftp* lftp*
```

匿名从 ftp 服务器上下载文件，则需要匿名登录，在提示输入用户名时，输入 anonymous，密码直接回车，就可以了。

ftp 常用命令:

查看目录: `dir`

下载: `get` 目标文件名 下载后的文件名

结束与服务器的 FTP 会话: `close`

结束与服务器的 FTP 会话并退出 FTP 环境: `quit`

例如, 匿名登陆上海交大的 CentOS 源服务器, 其网址为:

<http://ftp.sjtu.edu.cn/centos/>

登录方式:

`$ftp ftp.sjtu.edu.cn`

匿名登陆之后很多操作命令同 Linux 命令行是一样的。我尝试匿名登录如图 4-23-1-1:

```
[zhanghuoding@localhost ~]$ ftp ftp.sjtu.edu.cn/centos/
ftp: ftp.sjtu.edu.cn/centos/: 未知的名称或服务
ftp> ls
Not connected.
ftp> close
Not connected.
ftp> exit
[zhanghuoding@localhost ~]$ ftp ftp.sjtu.edu.cn
Trying 2001:da8:8000:6023::230...
Connected to ftp.sjtu.edu.cn (2001:da8:8000:6023::230).
220 (vsFTPd 3.0.2)
Name (ftp.sjtu.edu.cn:zhanghuoding): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||23305|).
150 Here comes the directory listing.
lrwxrwxrwx 1 1000 100 14 Dec 19 2005 Linux-HOWTO -> sites/tldp.org
lrwxrwxrwx 1 1000 100 23 Mar 05 2011 archlinux -> sites/ftp.archlinux.org
lrwxrwxrwx 1 1000 100 31 May 01 2007 centos -> sites/mirror.centos.org/centos/
lrwxrwxrwx 1 1000 100 19 Apr 03 2006 csw -> sites/blastwave.org
lrwxrwxrwx 1 1000 100 27 Nov 18 2005 debian -> sites/ftp.debian.org/debian
lrwxrwxrwx 1 1000 100 24 Sep 25 2011 debian-cd -> sites/cdimage.debian.org
lrwxrwxrwx 1 1000 100 25 Feb 25 02:20 debian-security -> sites/security.debian.org
lrwxrwxrwx 1 1000 100 30 Feb 25 2014 deepin -> sites/packages.linuxdeepin.com
lrwxrwxrwx 1 1000 100 29 Feb 25 2014 deepin-cd -> sites/cdimage.linuxdeepin.com
lrwxrwxrwx 1 1000 100 22 Jan 07 11:30 docker-ce -> pub/.mirror9/docker-ce
-rw-r--r-- 1 1000 100 4286 Mar 24 2015 favicon.ico
lrwxrwxrwx 1 1000 100 43 Dec 06 2011 fedora -> sites/download.fedora.redhat.com/pub/fedora
drwxr-xr-x 5 1000 100 4096 Jan 18 19:22 fedora-secondary
drwxr-xr-x 7 1000 100 4096 Mar 29 2016 html
-rw-r--r-- 1 1000 100 687 Dec 21 2012 index2.html
lrwxrwxrwx 1 1000 100 15 Dec 06 2011 linuxmint -> sites/linuxmint
lrwxrwxrwx 1 1000 100 18 Nov 26 2015 linuxmint-cd -> sites/linuxmint-cd
drwxr-xr-x 3 1000 100 26 Sep 08 2010 logs
lrwxrwxrwx 1 1000 100 24 May 24 2016 mageia -> sites/mirrors.mageia.org
lrwxrwxrwx 1 1000 100 27 May 25 2016 opensuse -> sites/download.opensuse.org
lrwxrwxrwx 1 1000 100 25 Dec 06 2011 openvz -> sites/download.openvz.org
drwxr-xr-x 12 1000 100 4096 Nov 19 2016 pub
lrwxrwxrwx 1 1000 100 23 Nov 04 2016 remi -> sites/rpms.remirepo.net
-rw-r--r-- 1 1000 100 72 Dec 06 2011 robots.txt
lrwxrwxrwx 1 1000 100 46 Sep 22 2011 scientific -> sites/ftp.scientificlinux.org/linux/scientific
drwxr-xr-x 5 1000 100 4096 Jan 08 12:15 sites
-rw-r--r-- 1 1000 100 333 Mar 03 2015 sjtu.edu.cn.html
-rw-r--r-- 1 1000 100 0 Nov 22 2016 test
lrwxrwxrwx 1 1000 100 24 Dec 09 2005 ubuntu -> sites/archive.ubuntu.com
lrwxrwxrwx 1 1000 100 25 Dec 09 2005 ubuntu-cd -> sites/releases.ubuntu.com
226 Directory send OK.
ftp>
```

图表 4-23-1-1 匿名登陆上海交大 CentOS 源仓库

关于 ftp 的更多功能请参看文章:

<https://blog.csdn.net/longxibendi/article/details/5754646>

<https://blog.csdn.net/jubincn/article/details/6686822>

二十四、纯字符界面下的标记语言文件转换工具 Pandoc

该工具可以实现 Markdown 格式的文件到 LibreOffice Writer 文档格式的转换, 也可以实

现 LaTeX 文档到 EPUB 文档的转换，也可以实现 HTML 文件到幻灯片的转换，也可以实现讲 word 中的图片全部转存为文件到指定目录等等，还可以实现更多的转换等功能。

Pandoc 的相关介绍网页见：

<http://mobile.51cto.com/iphone-519160.htm>

其主页为：

<https://pandoc.org/>

安装方法很简单：

```
$sudo yum install pandoc*
```

英文版使用说明：<https://pandoc.org/MANUAL.html#options>

其他使用方式简介（中文）：

<https://www.jianshu.com/p/6ba04f669d0b>

<https://www.jianshu.com/p/1ebe64aa0442>

使用方式以及部分问题的排除和解决：

<https://www.jianshu.com/p/dcc2f95cc086>

使用示例，将 Markdown 文档渲染为 HTML 网页：

```
$pandoc input.md -o output.html
```

将一份 Markdown 文件转换为 docx 格式：

```
$pandoc input.md -o output.docx
```

文章 <https://www.jianshu.com/p/dcc2f95cc086> 中介绍了在将 LaTeX 文件转换为 PDF 文件过程中遇到的部分问题和解决方法（其实这个我觉得用 LaTeX 编译就生成 PDF 文件了，不用这种第三方的转换过程）。

二十五、纯字符界面下的 word 等文件格式转换工具

LibreOffice

该工具的主页：<http://libreoffice.org/>，该工具是一个 GUI 应用程序，但同时它也可以从命令行运行来快速转换一个或者多个文件。

安装：

```
$sudo yum install libreoffice*
```

然后就可以开始使用了，其使用模式一般是（soffice 或 libreoffice 效果相同）：

```
$soffice --headless --convert-to 目标格式（如 pdf） 源文件 --outdir 目标目录
```

例如：命令行转换 docx 文档到 Html 文件并进行查看

```
$libreoffice --invisible --convert-to html ./test.docx
```

命令行转换 docx 文档到 PDF 文件：

```
$soffice --headless --convert-to pdf ./test.docx
```

命令行转换图片到 PDF 文件：

```
$soffice --headless --convert-to pdf ./pic.jpg
```

为 PDF 文件添加水印，首先使用 libreoffice 将水印图片处理成 pdf 文件，再用 [pdftk](#) 的工具为文件打水印：

```
$ soffice --headless --convert-to pdf ./logo.jpg
```

```
$ pdftk source.pdf multistamp logo.pdf output result.pdf
```


关于该工具的更多使用教程请参看：

<https://blog.csdn.net/ljihe/article/details/77250206>

该工具的其他安装及部分使用方式参看：

<https://www.youcl.com/info/11465>

二十六、纯字符界面下的 PDF 处理工具

纯字符界面下的 PDF 查看器请参看 [PDF 查看器的安装](#)。另外还有优秀的 PDF 命令行处理工具 **pdftk** (the pdf toolkit)，它是一个功能强大的命令行的 PDF 文件编辑软件，可以合并/分割 PDF 文档、对 PDF 文件加密解密、给 PDF 文档加水印、从 PDF 文档中解出附件、将 PDF 文档变成一页等等。其官网为：<http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/>
该工具的安装方式：

```
$sudo yum install pdftk* pdf*
```

安装后运行，可能会提示缺少库文件，因此需要做如下安装：

```
$sudo yum install libstdc++-4.8.5-36.el7_6.2.i686
```

如果提示缺少共享库（例如缺少 `libstdc++.so.6` 库文件），可用如下命令查找包含该库文件的包，然后暗转该包即可解决：

```
$sudo yum provides libstdc++.so.6
```

```
$sudo yum install libstdc++-4.8.5-36.el7_6.2.i686
```

这里还缺少共享库 `libz.so.1`，一并安装：

```
$sudo yum install zlib-1.2.7-18.el7.i686
```

其使用方法如下：

将两个 pdf 文件内容页面进行追加合并（拼接）：

```
$pdftk 1.pdf 2.pdf output 4.pdf
```

将两个 pdf 文件内容页面进行交叉合并（一页 1 一页 2 一页 1 一页 2 穿插起来）：

```
$pdftk 1.pdf 2.pdf shuffle output 4.pdf
```

解密 pdf：

```
$pdftk secured.pdf input_pw foopass output unsecured.pdf
```

加密 pdf：

```
$pdftk 1.pdf 输出 1.128.pdf owner_pw foopass
```

将整个 PDF 文档旋转 180 度：

```
$pdftk in.pdf cat 1-endsouth output out.pdf
```

pdftk 可以随意删除或旋转页面。下面是一个把指定页码的页面移出 pdf 文档的示例：

```
$pdftk new.pdf cat 1-96 98-end output new1.pdf
```

为 PDF 文件添加水印，首先使用 [libreoffice](#) 将水印图片处理成 pdf 文件，再用 **pdftk** 的工具为文件打水印：

```
$ soffice --headless --convert-to pdf ./logo.jpg
```

```
$ pdftk source.pdf multistamp logo.pdf output result.pdf
```

该工具的其他使用方式请参看文章：

<https://blog.csdn.net/fresherman/article/details/5979204>

<https://forum.ubuntu.org.cn/viewtopic.php?p=3208492>

二十七、纯字符界面下的 Markdown 文件展示工具（相当于播放 PPT）

该工具 GitHub 地址为： <https://github.com/visit1985/mdp>

另外，该工具支持远程终端显示，也就是说你可以将 md 展示文件放在远程主机上，然后通过 ssh 等方式远程登录，然后运行该程序展示文件将会在本机终端上完美显示。

安装方式：

```
$mkdir ./mdpInstall && cd ./mdpInstall
$git clone https://github.com/visit1985/mdp.git
$cd ./mdp && make
$sudo make install
```

到这里就安装成功了，可以测试一下仓库中自带的测试文件：

```
$ mdp sample.md
```

该测试文件会以彩色字体呈现在终端，按上/下/回车键可以翻页，按 q 即可退出展示，这个展示过程还是相当酷的。

运行命令即可使终端具有颜色淡出的功能：

```
$echo 'export TERM=xterm-256color' | sudo tee -ai /etc/bashrc
$echo 'export TERM=screen.linux' | sudo tee -ai /etc/bashrc
```

参照项目主页中的 sample.md 文件，即可编辑出属于自己的展示页面。

英文版的使用说明如下（也可以访问其主页 <https://github.com/visit1985/mdp> 查看最新版的使用说明）：

How to use it:

Horizontal rulers are used as slide separator.

Supports basic markdown formatting:

- line wide markup
 - headlines
 - code
 - quotes
 - unordered list
- in-line markup
 - bold text
 - underlined text
 - code

Supports headers prefixed by @ symbol.

- first two header lines are displayed as title and author in top and bottom bar

Review sample.md for more details.

Controls:

- h, j, k, l, Arrow keys, Space, Enter, Backspace, Page Up, Page Down – next/previous slide
- Home, g – go to first slide
- End, G – go to last slide
- 1-9 – go to slide n
- r – reload input file
- q – exit

第五部分 Linux 系统运维

Linux 系统运维知识非常重要，这里介绍一些方便工作开展的工具，后续会继续补充。

这部分介绍的内容相对来说比较简单，因为一般的系统运维的工具，系统都会自带，而且网上高质量的教程以及出排坑的论坛和博客数量非常多，因此这部分内容大多做一个工具的提示性介绍，不深入讲解，如果有需要，可以去网上找教程。

一、 文本编辑器 vim 和 vimx

文本文件的编辑功能非常基础和常用，笔者个人觉得 vim 及其增强版是最好用的，没有之一。该工具是系统自带的，但是其增强版 vimx 有更多易用的功能，安装方法见前文 [vimx —— 可以使用系统粘贴板的 vim](#) 来安装，使用方式可以在 [<https://www.cnblogs.com/softwaretesting/archive/2011/07/12/2104435.html>] 中查看，这个教程很清爽美观。

二、 远程登录和控制工具（一）——ssh 工具

ssh 是 SSH 远程控制协议族的一个实现。关于 ssh 有很多教程，且 ssh 工具是系统自带的，和它一起的有 scp 命令，ssh 用于远程登陆系统，scp 用于在本地计算机和远程服务器之间进行文件的相互传输。具体用法可以百度。

ssh 有一个增强版叫 openssh，OpenSSH 提供了服务端后台程序和客户端工具，用来加密远程控制和文件传输过程中的数据，并由此来代替原来的类似服务。安装方式：

```
$sudo yum install openssh*
```

ssh 和 scp 命令在登录时都需要人工手动输入密码，不能从文件、变量、参数等渠道获取，在首次登录远程服务器时，还需要手工输入“yes”来确认登录功能，这在大量服务器的管理维护工作中极为不方便。配置服务器五密码登录又觉得非常不安全。后文介绍的批处理脚本语言中的 tcl（[tcl 的安装点这里](#)）和 expect（[expect 的安装点这里](#)）都将完美解决这个问题。

三、 远程登录和控制工具（二）——sshpass

ssh 登陆不能在命令行中指定密码。sshpass 的出现，解决了这一问题。sshpass 用于非交互 SSH 的密码验证，一般用在 sh 脚本中，无须再次输入密码。

sshpass 允许你用 -p 参数指定明文密码，然后直接登录远程服务器，它支持密码从命令行、文件、环境变量中读取。系统默认没有安装，需要手动安装：

方式一：

```
$sudo yum install sshpass*
```

方式二（从文件下载编译安装）：

```
$wget
```

```
https://nchc.dl.sourceforge.net/project/sshpass/sshpass/1.06/sshpass-1.06.tar.gz
```

```
$tar -zxvf ./sshpass-1.06.tar.gz && cd ./sshpass-1.06
```

```
$./configure --prefix=/opt/sshpass && make && sudo make install && sudo
```

```
cp /opt/sshpass/bin/sshpass /usr/bin/
```

至此 sshpass 安装完成，从命令行方式传递密码（-p 指定密码）。远程登录：

```
$sshpass -p "123456" ssh user_name@192.168.1.134
```

远程文件传输：

```
$sshpass -p '123456' scp root@192.168.1.134:/home/test/t ~/test/
```

或者直接远程执行一条命令：

```
$sshpass -p '123456' ssh root@192.168.1.134 'ls -alh'
```

四、 远程登录和控制工具（三）——telnet

telnet 也能实现诸多远程登录和控制功能，但是不是一种安全的工具，因为其协议中的文件和用户名、密码等都是明文传输的。因此不推荐使用。

该工具一般都是系统自带的。

五、 批处理和交互脚本语言（一）——shell

shell 脚本是 Linux 下的一种非常基础且使用众多的 Linux 批处理的脚本语言，可以帮助我们自动实现很多功能。

六、 批处理和交互脚本语言（二）——Tcl

Tcl 是一种方便好用的脚本语言，除了出色的脚本语言的功能外，还可以在脚本中自动输入密码来 ssh 远程登陆服务器，并执行批处理操作。当我们需要同时登录多台服务器的时候，就可以编写一个 tcl 脚本文件，并在脚本文件中自动登录远程服务器并执行一些操作，

在这个过程中不需要手工输入密码。

1. Tcl 的安装

Tcl 有的系统可以从软件仓库中安装：

```
$sudo yum install tcl tcl* tk tk*
```

如果软件仓库中找不到，则可以从源码安装。Tcl 可以从网站

<https://core.tcl-lang.org/index.html>

或者

<https://sourceforge.net/projects/tcl/files/>

下载到源码并进行安装：

```
$axel https://nchc.dl.sourceforge.net/project/tcl/Tcl/8.6.9/tcl8.6.9-src.tar.gz
```

```
$tar -zxvf ./tcl8.6.9-src.tar.gz
```

```
$cd ./tcl8.6.9/unix && ./configure && make
```

```
$sudo make install
```

即可安装成功。在终端测试是否安装成功：

```
$tclsh
```

命令提示符变成了百分号（%），说明已经安装成功了。输入 `exit` 退出 `tcl`。需要注意的是，运行 `tcl` 的 `bash` 需要用在终端输入命令 `tclsh`，而不是仅仅的 `tcl` 哦。用命令：

```
$sudo find /usr/ -iname "*tcl*" | grep "tclsh"
```

可以找到 `tcl` 的安装目录哦。

到这里 `tcl` 就安装成功了。使用或者语法教程请参看文章：

https://www.yiibai.com/tcl/tcl_basic_syntax.html

<https://blog.csdn.net/pfysw/article/details/79501846>

<https://www.cnblogs.com/rednodel/p/8329146.html>

使用 `tcl` 自动输入密码来进行 `ssh` 登录的代码见文章：

<https://blog.csdn.net/cheungjustin/article/details/6049015>

由于 `tcl` 的语法有些晦涩难记，本人觉得不是那么人性化，且正好有一个基于 `tcl` 的批处理脚本语言 `expect`，且该语言的语法形式简单，功能强大，尤其是交互功能等，因此笔者推荐使用 `expect` 脚本。安装方式见下节（[批处理和交互脚本语言（三）——expect](#)）。

2. Tk（Tcl 的扩展图形工具箱）的安装

用百度百科的话说：Tk 是 Tcl “图形工具箱”的扩展，它提供各种标准的 GUI 接口项，以利于迅速进行高级应用程序开发。从 1991 年发布至今，tk 有相当多的追随者，同时，有许多的编程语言就使用了 tk 作为底层库（虽然 tk 只是 tcl 语言的一个工具箱）。

在使用 python 进行图形界面开发的时候，就需要安装 tkinter 模块，而 tkinter 模块就是以 tcl/tk 为基础的，因此如果要使用 python 的 tkinter 模块，就需要安装 tcl/tk。

Tk 的安装在 tcl 的安装过程中有可能没有被安装，我们先安装 tcl，然后进行 tk 的安装，

从软件仓库即可安装：

```
$sudo yum install tk tk*
```

关于 tcl/tk 的教程网上也有很多，可以参看文章：

<http://ju.outofmemory.cn/entry/107455> （这篇文章排版很好）

<https://blog.csdn.net/chinalinuxzend/article/details/2430429>

<https://blog.csdn.net/ggf123456789/article/details/8882194>

七、 批处理和交互脚本语言（三）——expect

expect 脚本语言是基于 tcl 的，因此在安装 expect 之前，必须先安装 tcl！！

expect 的介绍前面已经铺垫过了，这里直接开始安装。可以通过软件仓库安装，也可以下载源码进行编译安装。

从软件仓库安装：

```
$sudo yum install expect expect* expectk
```

从源码编译安装：

expect 的源码可以从 <https://sourceforge.net/projects/expect/files/Expect/> 下载并安装：

```
$axel https://nchc.dl.sourceforge.net/project/expect/Expect/5.45.4/expect5.45.4.tar.gz
```

```
$tar -zxvf ./expect5.45.4.tar.gz
```

```
$cd ./expect5.45.4 && ./configure && make
```

```
$sudo make install
```

即可安装成功。在终端测试是否安装成功：

```
$expect -v
```

显示：

```
expect version 5.45
```

说明已经安装成功了。

关于该脚本语言的简单编写方式参见文章：

https://blog.csdn.net/f_zyj/article/details/85563926

<https://www.jianshu.com/p/b1e0f2242f89>

<https://www.linuxidc.com/Linux/2017-02/140362.htm>

但其实在使用这些简单的脚本之前，还是要先了解一下其语法的基本概要：

<https://www.cnblogs.com/lzrabbit/p/4298794.html>

<https://blog.csdn.net/chen3888015/article/details/80499874>

<https://www.cnblogs.com/operationhome/p/9154055.html>

最终，推荐使用详细、清爽、排版美观大方的教程：

http://xstarcd.github.io/wiki/shell/expect_handbook.html

或者

<https://www.jellythink.com/archives/373>

<http://www.linuxeye.com/Linux/expect.html>

<http://www.xuetimes.com/archives/781>

另外还有兄弟在网上提醒大家一点自己遇到的坑（Tcl/Expect 中利用 exec 调用管道"|"和 awk 的注意事项）：https://blog.csdn.net/iteye_15153/article/details/81899181

以及（ tcl 使用 exec 调用 shell 命令出错的问题解决 ）：
<https://blog.csdn.net/miao20091395/article/details/83308323>

第六部分 其他不常用的软件的安装

一、安装 kvm、xen、lxc 虚拟机和 docker：

1、安装 docker 的准备工作

在安装之前，如果是对于 centos，需要首先添加 docker 的源（如果不安装 docker，治安抓过其他的，则不用进行此步骤：

在/etc/yum.repos.d/目录中创建文件 docker.repo 并写入以下内容：

```
[dockerrepo]
```

```
name=Docker Repository
```

```
baseurl=https://yum.dockerproject.org/repo/main/centos/$releasever/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://yum.dockerproject.org/gpg
```

2、安装虚拟机等

具体关于各个虚拟机的介绍可以自行查阅百度。现提供各个软件安装方法汇总如下：

```
install kvm kvm* *kvm *kvm* docker docker-engine docker* *docker *docker* lxc lxc* *lxc *lxc*  
--skip-broken --nogpgcheck
```

```
install kvm docker docker-engine lxc lxc* --skip-broken --nogpgcheck
```

```
install kvm libvirt python-virtinst qemu-kvm virt-viewer bridge-utils --skip-broken --nogpgcheck
```

```
groupinstall Virtualization 'Virtualization Client' 'Virtualization Platform' 'Virtualization Tools'  
--skip-broken --nogpgcheck
```

```
install kvm kmod-kvm qemu kvm-qemu-img virt-viewervirt-manager --skip-broken --nogpgcheck
```

```
install kvm kvm* *kvm *kvm* --skip-broken --nogpgcheck
```

二、安装 flash plugin:

1、64 位系统:

wget http://linuxdownload.adobe.com/adobe-release/adobe-release-x86_64-1.0-1.noarch.rpm

```
rpm -ivh adobe-release-x86_64-1.0-1.noarch.rpm  
yum install flash-plugin
```

三、给 LibreOffice 安装中文字体

```
install libreoffice-langpack-zh-Hans
```

四、解决 libstdc++.so.6 版本过旧问题

```
install libstdc++-4.4.7-11.el6.i686
```

五、安装 chrome

1、64 位系统:

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm  
rpm -ivh google-chrome-stable_current_x86_64.rpm
```

解决依赖问题:

```
install lsb  
install libXScrnSaver
```

六、配置 kde 桌面语言中文显示（十分不常用）

我装的是 kde 桌面，刚装好时桌面是英文显示的，要显示中文，方法如下：

```
listkde*chinese
```

```
install kde-l10n-Chinese.noarch
```

第七部分 一些常用功能的简化脚本

Linux 下功能十分丰富，但是在使用过程中都需要带比较长的参数，而且每次使用的参数都是相同的，或者是需要做相同逻辑的批量处理的情景比较常见，因此将这些功能写成脚本放在系统路径下，方便调用。

一、递归修改目录中的子目录和文件名（全角——>半角）

在有些时候（例如系统没有中文输入功能或者不能输入中文符号的时候），对含有中文符号作为文件名的文件或者目录的处理就无从下手。因此，该脚本提供一个手段，将指定目录中的文件或者目录名递归修改，修改其中的全角符号为半角符号，并且将所有的空格修改为下划线。

命令为：

```
$sudo tee -i /usr/bin/changeFileOrFolderNameFromFullToHalfCorner.sh <<-“EOF”
#!/bin/bash

echo "#####"
echo "# Preparing to Recursively modify files in the folder"
echo "#####"

fileList='./fileList'
readonly fileList
globalName=""

changeName()
{
    oldName="$*"
    oldName=${oldName#*changeName }
    if [ "$oldName" == "s" ]
    then
        echo "There is function changeName"
        return
    fi
#    echo "$oldName"
    newName=$oldName
    newName=`echo "$newName" | sed 's/ 《/[/g`
    newName=`echo "$newName" | sed 's/》/]/g`
    newName=`echo "$newName" | sed 's/<[/g`
    newName=`echo "$newName" | sed 's/>]/g`
    newName=`echo "$newName" | sed 's/—/--/g`
    newName=`echo "$newName" | sed 's/ 【[/g`
```

```

newName=`echo "$newName" | sed 's/【 /]/g`
newName=`echo "$newName" | sed 's/（ /(/g`
newName=`echo "$newName" | sed 's/） /)/g`
newName=`echo "$newName" | sed 's/· /_/g`
newName=`echo "$newName" | sed 's/、 /+/g`
newName=`echo "$newName" | sed 's/"/"/g`
newName=`echo "$newName" | sed 's/"/"/g`
newName=`echo "$newName" | sed 's/: /:/g`
newName=`echo "$newName" | sed 's/; /;/g`
newName=`echo "$newName" | sed 's/! /!/g`
newName=`echo "$newName" | sed 's/¥ /$/g`
newName=`echo "$newName" | sed 's/~ /~/g`
newName=`echo "$newName" | sed 's/, /,/g`
newName=`echo "$newName" | sed 's/。 /./g`
newName=`echo "$newName" | sed 's/? /?/g`
newName=`echo "$newName" | sed 's/ /_/g`

newName=`echo "$newName" | sed s/'\'/'/g`
newName=`echo "$newName" | sed s/'\'/'/g`
newName=`echo "$newName" | sed 's/{}/{/g`
newName=`echo "$newName" | sed 's/}/}/g`

globalName="$newName"

if [ "$oldName" != "$newName" ]
then
    echo "Change file name from \"$oldName\" to \"$newName\"."
    mv "$oldName" "$newName"
else
    echo "File \"$newName\" need not to be moved."
fi
}
travFolder()
{
    currentDir="$*"
    currentDir=${currentDir#*travFolder }
    fList=`ls "$currentDir"`
    cd "$currentDir"
    echo "=====current path $currentDir=====
    echo "$fList" > $fileList
    cat $fileList
    while read f
    do
        if [ "$f" == "$@" ]

```

```

        then
            continue
        fi
        if test -d "$f"
        then
            changeName "$f"
            travFolder "$globalName"
        fi

        if [ "$s$f" != "sfileList" ]
        then
            changeName "$f"
        fi
    done < $fileList
    rm $fileList
    cd ..
}
travFolder "$1"
EOF

```

然后赋予该脚本可执行权限：

```
$sudo chmod +x /usr/bin/changeFileOrFolderNameFromFullToHalfCorner.sh
```

然后为该文件建立软连接：

```
$sudo ln -s /usr/bin/changeFileOrFolderNameFromFullToHalfCorner.sh
/usr/bin/file_folderNameChangeFromFullToHalfCorner
```

使用方法：

```
$file_folderNameChangeFromFullToHalfCorner ./
$file_folderNameChangeFromFullToHalfCorner ~/first/second/
```

二、简化使用 mplayer 视频播放器

该工具的安装请参见 [mplayer 视频播放器的安装](#)。

由于在纯字符界面或者命令行的时候，mplayer 每次播放的时候都需要带一大堆参数，难记、而且还浪费时间，因此有必要写一个简短点的脚本方便平时使用。笔者将该脚本命名为 shell_my_MPlayer.sh 并放置在 /usr/bin 目录中，为其添加可执行权限，并建立软链接 myMPlayer，之后，播放单个视频的命令格式为：\$myMplayer filename
播放视频列表的命令格式为：\$myMPlayer -playlist list

以上命令中，fileName 和 list 都包含文件路径+文件名。

执行以下三条命令：

```
$sudo tee -i /usr/bin/shell_my_MPlayer.sh <<-'EOF'
#!/bin/bash

#####
#You can use this script to play a video like                                #

```

```

#   $shell_my_MPlayer.sh ~/my/testVideo.avi           #
#   $shell_my_MPlayer.sh -playlist ~/my/testVideo.avi #
#You can also use this script to play many videos by a list like      #
#   $shell_my_MPlayer.sh -playlist ~/my/list           #
#   $shell_my_MPlayer.sh ~/my/list                     #
#You can play a directory like                                         #
#   $shell_my_MPlayer.sh ./                                  #
#   $shell_my_MPlayer.sh /home/user/video                #
#   $shell_my_MPlayer.sh /home/user/video/              #
#   $shell_my_MPlayer.sh -playlist ./                   #
#   $shell_my_MPlayer.sh -playlist /home/user/video      #
#   $shell_my_MPlayer.sh -playlist /home/user/video/    #
#####

```

```

#this script need two parameters at least
if test $# -lt 1
then
    echo "Sorry! You have to set one parameters for excute at least."
    echo "You can set one or two parameters and try again."
    exit
fi

```

```

para0=$0
para1=$1
para2=$2
paraAll=$*
playList="playlist"
readonly playList
playFile=""

```

```

fun_PlayList()
{
    subLen=$(( ${#para2} - 1 ))
    lastChar=${para2:$subLen:1}

    if [[ "s${lastChar}" == "s*" ]]
    then
        para2="${para2:0:$subLen}"
        subLen=$(( ${#para2} - 1 ))
        lastChar=${para2:$subLen:1}
    fi
}

```

```

if test ! -e "$para2"
then
    echo "No such file or directory!"
    exit
fi

if test -d "$para2"
then
    currentPath=$( pwd )
    if [[ "s${para2:0:1}" == "s~" ]]
    then
        cd ~
        para2="$( pwd )${para2:1}"
        cd "${currentPath}"
    else
        if [[ "s${para2:0:1}" == "s." ]]
        then
            if [[ "s${para2:1:1}" == "s." ]]
            then
                temp=$( pwd )
                temp=${temp%/*}
                para2="${temp}${para2:2}"
            else
                para2="$( pwd )${para2:1}"
            fi
        fi
    fi
    cd "${para2}"
    para2="$( pwd )/"
    cd "${currentPath}"

    if test ! -r "${para2}"
    then
        echo "This directory dose not have read or write permissions!"
        exit
    fi
    if test ! -w "${para2}"
    then
        echo "This directory dose not have read or write permissions!"
        exit
    fi

    rm -f "${para2}${playList}"
    ls "${para2}"* > "${para2}${playList}"

```



```

        playFile="${para2}${playList}"
    else
        if test -f "${para2}"
        then
            if test ! -r "${para2}"
            then
                echo "This directory dose not have read or write permissions!"
                exit
            fi
        fi

        if is_TextFile
        then
            playFile="${para2}"
        else
            fun_PlayVideo
        fi
    fi

    mplayer -vo fbdev2 -geometry 4000:0 -zoom -vf scale -x 400 -y 225 -ao alsa -loop 0 -playlist "${playFile}"
    exit
}

fun_PlayVideo()
{
    playFile="${para2}"
    mplayer -vo fbdev2 -geometry 4000:0 -zoom -vf scale -x 400 -y 225 -ao alsa -loop 0 "${playFile}"
    exit
}

is_TextFile()
{
    #For a shell function,if return 0,means function execute successfully,and $? equals true
    #if return 1,means function execute failed,and $? equals false
    while :
    do
        temp1=$( file "${para2}" )
        temp2="${temp1}"
        temp1="${temp1#*symbolic link to *}"
        if [[ ${#temp2} -gt ${#temp1} ]]
        then
            temp1="${temp1:1}"
            temp1="${temp1%\'}"
            para2="${temp1}"
        fi
    done
}

```

```

        else
            unset temp1
            unset temp2
            break
        fi
        unset temp1
        unset temp2
    done

    temp1=$( file "${para2}" )
    temp2="${temp1}"
    temp1="${temp1#*${para2}: }"
    temp3="${temp1}"
    temp1="${temp1% text*}"
    if [[ ${#temp3} -gt ${#temp1} ]]
    then
        unset temp1
        unset temp2
        unset temp3
    #If this file is a text file,return 0.
        return 0
    else
        unset temp1
        unset temp2
        unset temp3
    #If this file is not a text file,return 1.
        return 1
    fi
}

##### start #####
if [[ $spara1 == s"-playlist" ]]
then
    para2=${paraAll#*${para1}}
    para2=`echo $para2 | awk '{gsub(/^\s+|\s+$/, "");print}'`
#    para2=${para2// /\ }

    fun_PlayList
else
    para2=${paraAll#*${para0}}
    para2=`echo $para2 | awk '{gsub(/^\s+|\s+$/, "");print}'`
#    para2=${para2// /\ }

    if test ! -e "$para2"

```

```

then
    echo "No such file or directory!"
    exit
fi

if test -d "$para2"
then
    fun_PlayList
fi

if test -f "$para2"
then
    if test ! -r "${para2}"
    then
        echo "This directory dose not have read or write permissions!"
        exit
    fi
    if is_TextFile
    then
        fun_PlayList
    else
        fun_PlayVideo
    fi
fi
fi
EOF

```

```

$sudo chmod +x /usr/bin/shell_my_MPlayer.sh
$sudo ln -s /usr/bin/shell_my_MPlayer.sh /usr/bin/myMPlayer

```

然后就可以享受\$myMplayer filename 和\$myMPlayer -playlist list 的简洁快感啦！

注：本脚本中设置的视频播放窗口大小宽高分别为 400 像素*225 像素，即 16:9 的视频尺寸比例，且窗口位置在电脑屏幕右上角。