配置你新的Slackware

我们假设你已经读过安装指南，并在电脑上装好完整的Slackware并准备好开始一场快乐的旅程。

此初学者索引将带你坚定的走上Slackware的道路。如果这是你第一次安装Slackware不要被命令行的启动登陆页面给吓到。让这篇文章指引你初始化配置你全新安装的Slackware系统吧。

在继续之前，一定要记住Slackware的包管理器不会做任何的依赖检查。因此如果你是刚接触Slackware安装全部(除了KDEI 系列)可以避免许多后续可能出现的麻烦。
Slackware的官方推荐[1]是“ *如果你有足够的硬盘空间，建议您使用完整安装以达到最佳的目标*“

快速安装预览

当Slackware完成首次安装与重启之后，你会注意到系统启动到了命令行（控制台）的登陆页面 – 不像其他发行版那样启动到一个图形登陆界面。不要被这一点所吓到。这就是学习Linux的第一课，接下来的短短几周会让你*更加*了解Linux系统。

安装过程并没有提供创建用户账户。此时只有“root”账户可用。你应该记得root的密码，是在安装过程的结尾时你自己设置的。用root（根）账户登陆 – 你会发现自己处于“ # ”的根权限命令行状态。

现在该干啥呢？ ”root ”用户并不会成为你日常使用的帐号Root意味着系统的维护与配置，软件的更新等等。
你要做的第一件事是创建一个没有root权限的新用户。之后，你需要安装“ 非开源显卡驱动 “ （如果你使用的是Nvidia或Ati的显卡），设置无线网络连接或者启动图形化桌面环境。你可以使用Slackware做很多事情！让我们从最基本的开始。

创建用户账户

你要做的第一件事是为自己创建一个非root的用户账户。有两种方法，但都是用的命令行。推荐使用Slackware的交互命令 `adduser`，比如：

```
# adduser
```

然后根据提示走。更多关于 `adduser` 的信息，请阅读 用户管理 页面。你也可以用非交互式的标准linux命令 `useradd` :

```
# useradd -m -g users -G
wheel,floppy,audio,video,cdrom,plugdev,power,netdev,lp,scanner -s /bin/bash
slacker
```

完成上述后你就可以用你的用户账户登陆了。
登出现在的root帐号（在root提示符后输入 `logout` ），然后使用你刚才创建的帐号登陆。下面开始真正的旅行吧！

使用自己的语言

Slackware的安装过程只提供英语，且电脑中的程序也是默认英语。如果你是非英语用户，想要使用汉语，可以查看这篇 ”本土化:调整Slackware到你自己的语言”

配置包管理器

现在你已经有一个在运行的Slackware你应当花点时间考虑电脑的安全问题。随着时间推移Slackware预装的一些软件可能被发现安全漏洞而 易遭攻击 。当被发现的漏洞可能危害您的电脑健康Slackware通常会发布一个软件包的补丁。这些补丁包可以在网上下载（在发行版的 /patchs 文件夹里），并在Slackware安全邮件列表中声明。

你有好几种方法保持你的Slackware安装软件最新。不建议你使用全自动的方法保持安全更新，虽然使用cron计划任务）可以做到这一点。

slackpkg

最好的选择是使用slackpkg，这是一个Slackware自己安装包工具中的包管理器。在你使用 slackpkg 之前你需要指定一个用来下载更新的网络镜像站。

网络镜像站的列表可以在这个文件中找到：

```
/etc/slackpkg/mirrors
```

使用文本编辑器如nano或者vi打开并取消注释 一条 镜像网址。确保你注释的镜像站包含你运行的Slackware版本！同时，选择接近你或者你认为比较快的镜像站。完成后，运行以下命令初始化包管理器的数据库：

```
# slackpkg update gpg
# slackpkg update
```

注意，包管理是由“root”用户完成的！

你需要时不时的更新slackpkg数据库，特别是知道新版本发布的时候。更新完数据库就可以更新软件了。关于此工具的更多信息你可以查看slackpkg

使用slackpkg安装更新

slackpkg update命令会连接到一个Slackware镜像站并更新本地已装软件的信息数据。这个命令不会*安装*软件的更新！
常用的更新软件的过程是这样的：

```
# slackpkg update
# slackpkg install-new
# slackpkg upgrade-all
# slackpkg clean-system
```

“install-new”“upgrade-all”和 “ clean-system”命令会在真正运行前给你列出将被操作的软件列表，这给你机会去选择/取消选择你不想更新的软件。
“clean-system”命令只是在你进行系统版本更新（如从14.1升级到14.2）时移除不是（或不再是Slackware的版本核心有关的包。

slackpkg clean-system命令会移除第三方提供的软件包！因此，必要时可以注意一下黑名单(/etc/slackpkg/blacklist)

留意软件更新

Slackawre要点一书中有保持软件的更新章节。如果你还没有读过，建议你读一读。

- 一种查询更新包（补丁）的方法是注册到Slackware安全的邮件订阅列表中，当你收到推送就可以开始更新了。
- 另一方法就是设置脚本去每天检查更新，有更新则使用邮件提醒。
  若想要这正常工作你应该配置好sendmail(通常这是开箱既用的)并知道如何设置cron任务。当然，你得有这样一个这样的脚本。
  一个例子脚本是rsync_slackware_patches.sh，它监视Slackware的ChangeLog.txt文件来检测更新。你下载这个脚本，编辑它以使其用你最喜欢的镜像站，并使它成为一个cron任务：

```
# wget http://www.slackware.com/~alien/tools/rsync_slackware_patches.sh
-O /usr/local/bin/rsync_slackware_patches.sh
# chmod +x /usr/local/bin/rsync_slackware_patches.sh
```

  为了更加适应你的环境，脚本中有一些你可能想修改的内容-比如脚本会把补丁下载到哪里。
  先运行一下这个脚本，看看它会报告什么东西：

```
# /usr/local/bin/rsync_slackware_patches.sh
[rsync_slackware_patches.sh:] Syncing patches for slackware version
'13.37'.
[rsync_slackware_patches.sh:] Target directory
/home/ftp/pub/Linux/Slackware/slackware-13.37/patches does not exist!
[rsync_slackware_patches.sh:] Please create it first, and then re-run
this script.
```

  你会发现你必须编辑这个脚本，定义一个此脚本要用的文件夹(*当然也得创建这个文件夹*)。完成以后，你需要运行一下这个脚本-首次下载补丁。
  然后你就可以以每天一次的频率使用cron运行这个脚本了。例如，想要这个脚本在每天的05:35运行一次，让它检查64位的Slackware-13.37的更新。通过以下命令打开crontab编辑器：

```
crontab -e
```

  添加以下内容到你的cron表中：

```
33 5 * * *      /usr/local/bin/rsync_slackware_patches.sh -q -r 13.37 -
a x86_64
```

  这个命令如果没有发现新的补丁，它将会安静的运行(意味着你不会收到邮件)。一旦脚本检查到了

更新，它就会把输出用邮件发出去。你会收到类似这样的一封邮件：

```
[rsync_slackware_patches.sh:] New patches have arrived for Slackware
13.37 (x86_64)!

.....................................................................

0a1,10
> Mon Sep 10 20:26:44 UTC 2012
> patches/packages/seamonkey-2.12.1-x86_64-1_slack13.37.txz:  Upgraded.
>   This is a bugfix release.
> patches/packages/seamonkey-solibs-2.12.1-x86_64-1_slack13.37.txz:
Upgraded.
>   This is a bugfix release.
> +--------------------------+
> Sun Sep  9 19:11:35 UTC 2012
> patches/packages/mozilla-thunderbird-15.0.1-x86_64-1_slack13.37.txz:
Upgraded.
>   This is a bugfix release.
> +--------------------------+
```

这样你就知道需要更新 slackpkg，安装最新的补丁。在自动检查更新（同时给你下载下来了）的同时让你控制更新（你来决定什么时候更新）。

配置网络

如果你选择安装了网络相关的包，在安装过程最后，你会被问几个问题，如：

- do you use DHCP;（你使用DHCP给你分配地址吗）
- or else, what IP address do you want to use;（或者，你想要使用哪个IP地址）
- what is your computer's hostname;（你的电脑的主机名是啥）
- do you have a (DNS) nameserver in the network?（你的网络中有DNS域名服务器吗？）

这些问题的回答会形成一些与网络有关的配置文件。

- `/etc/rc.d/rc.inet1.conf`
  这里是你的网络接口的详细情况（Slackware的`netconfig`工具只会配置你的`eth0`接口。如果你有额外的网络接口，你可以使用文本编辑器如`nano`或`vi`来编辑以详细配置你的网络接口。这里有一个关于它的手册(man（mannual的缩写)：

  ```
  man rc.inet1.conf
  ```

- `/etc/resolv.conf`
  这是你的域名服务器（DNS）及域名服务器表。如果你使用DHCP（那么你的DHCP客户端会为你更新这个文件。如果你想手动指定IP地址，你就得自己编辑这个文件。同样有个手册：

  ```
  man resolv.conf
  ```

- `/etc/HOSTNAME`
  这里定义你的主机名。

- `/etc/hosts`
  这里面有连接到主机名的回环接口(loopback)的定义。如果你不使用DNS服务器，或者你想指定服务器中没有或不合适的解析，你可以添加域名-IP地址表进去。有个手册：

```
man hosts
```

如该你想进一步了解如何配置网络，看看这个网页 Slackware网络配置综述

传统方式配置网络

有线网

想要配置 `eth0` 网络接口，（以root身份）运行

```
# netconfig
```

安装过程就是调用了这个命令。

`netconfig` 只处理 `eth0` 的有线网络。

假设你使用 `netconfig`配置你的有线网络，安装后基本上就是全自动连接了。

如果安装过程你没有配置网络，以root的身份运行 `netconfig`；然后运行

```
# /etc/rc.d/rc.inet1 eth0_start
```

你应该就可以连接到有线了。

无线网

如今Linux已经支持大部分无线硬件了。你可以去网上看看第三方硬件驱动中有没有你的硬件。如果你想看看你的电脑有没有识别到你的无线网卡，以root的身份运行

```
# iwconfig
```

如果它在所有接口都返回了“ *no wireless extensions* ”（没有无线扩展），那么内核中没有你的无线网卡的驱动，去网上找找看吧。

检测到无线接口后，配置文件一般在 `/etc/rc.d/rc.inet1.conf` 里。你可以在这里查看更多信息 无线配置指南。本地手册：

```
# man rc.inet1.conf
```

你还需要配置无线网络安全，如WEP或者WEP2不推荐不加密的无线网，不安全。注意WPA/WPA2加密不仅要配置 `/etc/rc.d/rc.inet1.conf`，也得在 `/etc/wpa_supplicant.conf` 里输入密码。

无线加密出问题的话，特别是WPA比较麻烦。一些常见问题可以看 关于网络的指南，以防你的电脑真的连不上热点。

图形界面配置网络

Slackware现在有其他配置和查看网络连接的方法。那就是安装守护进程（或者称之为后台服务）来让你很容易的在有线和无线间切换。特别适合可移动设备的用户。它有图形界面且不依赖传统的配置文件 - *事实上，如果配置文件真的配置好了会和这玩意儿冲突*

- 在版本表中的 *extra* 选项中有 wicd 项 (*extra* （额外）意味着不是核心部分，在完整安装选项中不会被安装)。
  安装完wicd包后，你必须给初始化脚本运行的权限，以便在开机时自动开启守护进程：

```
# chmod +x /etc/rc.d/rc.wicd
```

  你可以使用图形界面的 `wicd-client` 配置网络了，或者你如果运行的是Slackware 14你可以使用wicd的KDE微件（注widget百度百科称之为微件，类似桌面小插件）对于命令行爱好者，有 `wicd-curses` 提供X界面的配置方式。

- 自从Slackware 14之后，就有一个 网络管理器。在完整安装时会被安装，但是守护进程不是默认开启的，使用wicd你要给脚本运行权限：

```
# chmod +x /etc/rc.d/rc.networkmanager
```

  使网络管理器开机自启。你得使用X图形界面配置网络管理器。
  Slackware 14 包括了一个网络管理器的KDE微件。如果你使用其他桌面环境如XFCE你可以从 SlackBuilds.org 安装Gnome network-manager-applet

切换到通用内核

推荐你切换到Slackware的 *generic* 内核。这很简单，但也有以下这些步骤。

“ 通用”generic内核和作为默认内核的“ 巨大”huge内核之间的区别是啥呢？
“ 巨大 ” 内核是为能够实现成功安装而特制的，包含每一种驱动程序的内核。它考虑到了储存、（有线）网络、文件系统、加密驱动等等。这些内置驱动导致了内核很大（这也是它名字的来因）。当内核启动后会占据许多RAM内存（当然，相对来说，有1G的RAM的话少几M的内存问题也不大）
“ 通用 ” 内核不包含驱动，会按需加载驱动到内存，启动速度也会快一点。更小的占用意味着支持使用初始化内存盘initial RAM disk或者称 “initrd”初始化内存盘在一些情况下是必须的，比如软件RAID磁盘阵列），或者完全加密的硬盘。
总之，“ 巨大 ” 内核不支持初始化内存盘，“ 通用 ” 内核支持。为了最大的灵活性，我们使用“ 通用 ” 内核。

- 你必须创建初始化内存盘（简称“ 内存盘 ”？原文为英文缩写initrd内存盘作为内核启动阶段的临时的根文件系统，在启动阶段协助挂载真正的根文件系统。以root身份运行：

```
# /usr/share/mkinitrd/mkinitrd_command_generator.sh
```

这条命令实际上 *啥也没干*。它只是返回了如下一些信息 - 取决于内核版本，硬件状态，安装时的根文件系统选择等而变：

```
#                                                                Note
that package management is done as the "'''root'''" user!
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').□如果你需要内核启动时的初始化镜像，以防你使用未内置对你的储存或根文件系统支持的内核（如'通用'内核），这个脚本会为你推荐对应的命令）
# A suitable 'mkinitrd' command will be:

mkinitrd -c -k 3.2.29 -f ext4 -r /dev/sdb2 -m usb-storage:ehci-
hcd:usbhid:ohci-hcd:mbcache:jbd2:ext4 -u -o /boot/initrd.gz
```

运行脚本推荐的 `mkinitrd` 命令 (以root用户的身份) 来生成 `initrd.gz` 镜像。

- 如果你安装了 LILO (Slackware的默认启动器), 你需要修改配置文件 `/etc/lilo.conf` 来添加正确的启动项：

```
image = /boot/vmlinuz-generic-3.2.29
  initrd = /boot/initrd.gz # add this line so that lilo sees initrd.gz
  root = /dev/sda1
  label = Slackware
  read-only
```

实际上，如果你把通用内核的名字作为参数传递给“ `mkinitrd_command_generator.sh` ”脚本，它还会返回一个可以添加到 `/etc/lilo.conf` 的样式示例：

```
# /usr/share/mkinitrd/mkinitrd_command_generator.sh -l /boot/vmlinuz-
generic-3.2.29
```

推荐你 *添加新选项* 而不是在原来的选项上编辑。给你添的选项一个独特名字。重启后□LILO会给你两个选项：启动到新加的通用内核，或者启动到主打兼容的巨大内核（这东西依然是可用的）。

- 修改完 `/etc/lilo.conf` 后保存，然后运行

```

# lilo -v
```

来使改变生效。然后，重启。

- 看看 `mkinitrd` 的手册页面 (`man mkinitrd`) 来获得更多信息。
- 如果你使用grub或者其他的启动器，使用对应的修改方法。
- 如果你使用通用内核却没有制作 `initrd.gz`，启动时会内核报错。

启动图形界面

有需要的话配置X

Slackware使用的是 X.Org 的X-Window框架□X服务会自动检测显卡并加载合适的驱动。如果自动检测没工作□X在启动时崩溃了），你就得编辑 `/etc/X11/xorg.conf` 来指定对应的显卡选项和分辨率。你可以使用

```
# X -configure
```

来在你当前目录下生成基础的 `xorg.conf` 配置文件。你可以继续修改，然后放到 `/etc/X11/` 文件夹。详细配置方法可查看X的手册，注意是 `xorg.conf` 的手册 (`man xorg.conf`)□

非自由的显卡驱动

许多人使用英伟达□Nvidia□或超威□AMD□的显卡（图形处理器□GPU□□这些高性能显卡的供应商一般只提供非自由的驱动（可能只提供编译好的二进制文件）。这些驱动可能只在特定的 OpenGL 驱动的软件中加速图像。如果你用的这样的卡，可以看看“ 有版权的显卡驱动”.

选择桌面环境/窗口管理器

想选择 窗口管理器 或 桌面环境，运行 `xwmconfig`：

```
$ xwmconfig
```

选择其中的一个选项。注意你如果使用的是根用户运行 `xwmconfig` 将为所有用户设置。使用你自己的账户只为你自己设置。
选完后就可以运行

```
$ startx
```

你想要的桌面环境或者窗口管理器就启动了。

图形界面登录

想在启动后不进入命令行的登录，而是直接进入图形界面来登录，把默认运行级别调到4.编辑文件 `/etc/inittab` 找到这样一行

```
   id:3:initdefault:
```

改成

```
   id:4:initdefault:
```

注意，这与别的发行版本不同；其他很多是第5级为图形界面。而在Slackware中，5级和3级是等价的（也就是启动到命令行）。

在图形级别下，你会首先收到来自显示管理器的问候（要求你登录□□Slackware默认会寻找可用的GDM□Gnome显示管理器□□KDM□KDE显示管理器）以及XDM□X显示管理器） - 按顺序找。你可以安装第三方登录管理器如 SliM 但你必须编辑 `/etc/rc.d/rc.4` 来让它首先被搜寻。

进一步探索

命令行

在安装图形桌面前，你可以探索学习命令行，学习一些命令，以及一些软件的非图形界面状态□Slackware比较好的一点是它有许多的覆盖范围大的命令行软件。
例如，浏览网页可以用lynx或者links□是基于命令行的浏览器。你可以听歌（甚至网络音乐流媒体），播放器比如有moc□mpg123□ogg123□

32位和64位混用

如果你安装了64位版本的Slackware (经常被称为 *slackware64* 或 *Slackware for x86_64*) 你会发现它拒绝运行一些像 Wine的32位软件。这种情况你可以阅读 使用混合运行库 □

Slackware文档

Slackware用户也能从好的文档中受益良多（否则你也没耐心读到这了）我们建议你看这个Wiki来学习额外技巧与用法。同时别忘了看看Slackware的DVD或CD的根目录！你会发现Slackware的自己的手册。每个文档都值得一看。

升级系统

如果你在用Slackware发现新版本可用，我们有一个不错的 指南 可以看看： 把你的Slackware升级到新版本

当追寻 当前版本，你一定要先看看 ChangeLog.txt □看看升级前后是否有什么准备步骤。升级到稳定版，看看CD/DVD或官方镜像里的, `UPGRADE.TXT` 和 `CHANGES_AND_HINTS.TXT` 是个好主意。

slackware, beginners, guide

1)
见DVD或CD1的根目录中的Slackware-HOWTO

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/zh:slackware:beginners_guide

Last update: **2020/12/23 08:57 (UTC)**