

Beginners' Guide (简体中文)

From ArchWiki

小贴士: 本向导可以分页浏览，请从[这里开始](#)。

Contents

- 1 前言
 - 1.1 介绍
 - 1.2 许可证
 - 1.3 Arch之道
 - 1.4 关于这份指南
- 2 准备
 - 2.1 获得最新的安装媒介
 - 2.1.1 检查下载文件的完整性
 - 2.1.2 光盘安装
 - 2.1.3 U盘安装方式
 - 2.1.3.1 Linux/UNIX 方式
 - 2.1.3.2 微软Windows方式
 - 2.1.4 网络安装方式
 - 2.1.5 在虚拟机中安装
 - 2.2 启动 ArchLinux 安装程序
 - 2.2.1 引导
 - 2.2.2 启动操作系统
 - 2.2.2.1 启动问题解决
 - 2.2.3 更换键盘映射表
 - 2.2.4 文档
- 3 安装
 - 3.1 选择安装源
 - 3.1.1 配置网络 (netinstall)
 - 3.1.1.1 Live 环境下设置 ADSL 桥接 (可选)
 - 3.1.1.2 Live 环境下设置无线网络(可选)
 - 3.2 设置编辑器
 - 3.3 设置时钟
 - 3.3.1 设置时区
 - 3.3.2 设置时间和日期
 - 3.3.2.1 与Windows共存时的时间设置
 - 3.4 准备硬盘
 - 3.4.1 磁盘分区：基本知识
 - 3.4.1.1 分区类型
 - 3.4.1.2 交换分区
 - 3.4.1.3 分区结构
 - 3.4.1.4 我的分区应该设多大？
 - 3.4.2 创建文件系统：基本知识
 - 3.4.2.1 文件系统类型
 - 3.4.2.2 关于日志
 - 3.4.3 选项1：自动调整

摘要

本文为新手提供了内容详实、解释清晰的 Arch linux 安装指导。

相关文章

Category:Accessibility (简体中文)

Official Installation Guide (简体中文)

Install from SSH (简体中文)

General Recommendations (简体中文)

General Troubleshooting

- 3.4.4 选项2：手动配置硬盘
- 3.4.5 选项3：手动配置块设备、文件系统和挂载点
- 3.5 选择软件包
 - 3.5.1 启动加载器
 - 3.5.2 软件包组
- 3.6 安装软件包
- 3.7 配置系统
 - 3.7.1 安装程序不能自动处理吗？
 - 3.7.2 /etc/rc.conf
 - 3.7.2.1 LOCALIZATION 部分
 - 3.7.2.2 HARDWARE 部分
 - 3.7.2.3 NETWORKING 部分
 - 3.7.2.4 DAEMONS 部分
 - 3.7.2.4.1 通用信息
 - 3.7.3 /etc/fstab
 - 3.7.4 /etc/mkinitcpio.conf
 - 3.7.5 /etc/modprobe.d/modprobe.conf
 - 3.7.6 /etc/resolv.conf
 - 3.7.7 /etc/hosts
 - 3.7.8 /etc/locale.gen
 - 3.7.9 /etc/pacman.conf
 - 3.7.10 /etc/pacman.d/mirrorlist
 - 3.7.11 Root密码
 - 3.7.12 完成
- 3.8 安装启动加载器
- 3.9 重启系统
- 4 配置
 - 4.1 更新
 - 4.1.1 配置网络 (若需要)
 - 4.1.1.1 有线局域网
 - 4.1.1.2 无线局域网
 - 4.1.1.3 代理服务器
 - 4.1.1.4 模拟调制解调器、ISDN 和 DSL (PPPoE)
 - 4.2 用 pacman 更新、同步、升级系统
 - 4.2.1 /etc/pacman.conf
 - 4.2.2 软件包源
 - 4.2.3 AUR
 - 4.2.4 /etc/pacman.d/mirrorlist
 - 4.2.5 rankmirrors
 - 4.2.5.1 Mirrorcheck 以检查最新软件包
 - 4.2.5.2 一个使用即时的源来更新你的的源列表的脚本
 - 4.2.6 熟悉 pacman
 - 4.2.7 更新系统
 - 4.2.7.1 忽略软件包
 - 4.2.7.2 Arch 滚动发布模式
 - 4.3 添加一个用户
 - 4.3.1 删除账户
- 5 其他
 - 5.1 Sudo
 - 5.2 声音

- 5.3 图形用户界面
 - 5.3.1 安装 X
 - 5.3.2 安装显卡驱动
 - 5.3.2.1 nVIDIA图形显示卡
 - 5.3.2.2 ATI图形显示卡
 - 5.3.3 安装输入设备驱动
 - 5.3.4 配置 X (可选)
 - 5.3.4.1 非 US 键盘
 - 5.3.5 测试 X
 - 5.3.5.1 消息总线
 - 5.3.5.2 启动 X
 - 5.3.5.3 若有问题
 - 5.3.5.4 需要帮助?
 - 5.3.6 安装字体
 - 5.3.7 选择并安装图形界面
 - 5.3.8 启动图形环境的方法
 - 5.3.8.1 手动
 - 5.3.8.2 自动
- 6 附录

前言

介绍

欢迎，本文档将指导您安装 Arch Linux：一个简单、轻量级、适合计算机水平较高用户使用的发行版。本向导写给 Arch 新用户，但是会尽量做到成为所有用户的参考和信息库。

安装前请先浏览一下FAQ。

ArchLinux 发行版的亮点：

- 简洁 的设计和哲学
- 所有软件包 (<http://www.archlinux.org/packages/?q=>) 针对i686 和x86_64架构编译
- 滚动发行 模式，一次安装、无缝升级到最新版本
- BSD风格的启动过程，配置集中于一个文件中
- mkinitcpio (简体中文): 一个简单的动态initramfs创建工具
- 使用Pacman作为软件包管理器 轻量且灵活，占用存储空间小
- Arch Build System: 类似 ports (BSD上面的包构建系统)的包构建系统。提供从源码创建 Arch 安装包的易用架构
- Arch User Repository: 成千上万的 Arch 构建脚本，这些脚本由 Arch 用户提供，您也可以在这里和别人分享构建脚本

许可证

Arch Linux, pacman、文档和脚本版权所有：© 2002-2007 by Judd Vinet, Copyright © 2007-2011 by Aaron Griffin

许可证协议：| GPLv2 (<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>) .

Arch之道

Arch背后的设计原则就是保持简洁。

'简洁'，在这里意味着'仅进行必要的扩展、修改和复杂化'。简单地说，它就是一个优雅简约的处事方式。

一些对简洁的思考，供参考：

- “ ‘简洁’ 的立场是技术，而不是可用性。学习曲线较高的优雅技术要高过易用的较差技术。 - Aaron Griffin ”
- “ Entia non sunt multiplicanda praeter necessitatem ” (拉丁语) 或者 “ 不应增加多余的东西。 ” -Occam's razor(奥卡姆剃刀原则) 术语 “剃刀” 指的是去除非必需的复杂化部分以得到简单的说明，方法和理论。
- “[我的方法中]最超凡的部分就是它的简洁性，大道至简。” - Bruce Lee(李小龙)

关于这份指南

社区维护的 Arch Wiki 是出色的资源，有问题先查 Arch Wiki。如果你在其他地方找不到问题的答案，IRC 频道(irc://irc.freenode.net/#archlinux) 和论坛 (<http://bbs.archlinux.org/>) 都是提问的好地方。同时，在对任何命令的使用有疑问的时候，都应该用 man 查询手册。

注意：严格按照指南操作是安装和配置完整 Arch Linux 系统的关键。所以，**请从头到尾地阅读它**。强烈建议你在进行安装配置之前完全阅读每一个章节

这份指南分为五个主要部分：

- 第一部分: 准备
- 第二部分: 安装
- 第三部分: 配置
- 第四部分: 其他
- 第五部分: 附录

准备

注意：要将 Arch 安装到已经存在的 GNU/Linux 发行版或者 LiveCD，请参阅 这篇文章 。这种方式特别适用于使用VNC或者SSH进行的远程安装。下面的内容假设使用传统的安装方式。

获得最新的安装媒介

你可以在这里 (<http://archlinux.org/download/>) 找到官方的 Arch Linux 安装媒介。最新的版本是 2011.08.19，本指南适合于最新版本。

- **Core** 和 **Netinstall** 映像文件都只提供创建 **Arch Linux 基本系统**必须的包。**要注意的是，此基本系统不包含图形用户界面。主要包含了 GNU 工具链(编译器，汇编器，连接器，库，shell和基本工具程序)，Linux内核，pacman (Arch 的软件包管理器)，一些附加库和模块。**
- **Core** 映像可以方便的从 CD 或者网络进行安装。
- **Netinstall** 映像则更小，自身不包含软件包；整个系统都是通过网络下载。
- Arch64 FAQ 中的信息可以帮你从32位和64位版本中选择。**双架构**镜像有两个架构的软件包，所以一个 CD 就能安装两个架构的系统。
- 记得下载 ISO 对应的校验码文件。

Pre-release 镜像可以通过这里 (<http://releng.archlinux.org/isos/>) 下载。**它们不是官方发布的，所以不提供官方支持。** 只有当官方安装镜像无法支持系统的硬件，而新的内核可能已经支持的时候才使用它

们。

检查下载文件的完整性

打开终端 cd 到下载文件所在的目录，然后执行 sha1sum 命令：

```
$ sha1sum --check name_of_checksum_file.txt
```

并在你所给出的 .iso 文件名后面显示 "OK" (请忽略其它条目)。如果没有显示 "OK"，请就重新下载文件。 md5sum 也是以同样方式检查。

光盘安装

用你喜欢的光盘刻录机和刻录软件把 .iso 映像文件刻录到光盘上，然后从启动 Arch Linux 安装程序继续。

注意: 不同刻录机和光盘的质量差别是非常大的。为了得到可靠的刻录质量，通常使用低速刻录，有些用户建议用4x甚至2x的速度。如果有过刻录异常的情况，请试着将速度调至系统支持的最低速。

U盘安装方式

更多信息，请参阅 从 USB flash 设备安装。

无论是使用读卡器还是使用 USB 接口，只要 BIOS 支持从中启动，都能用这种方式安装 ArchLinux。

注意，U 盘上的所有数据都会被覆盖！

Linux/UNIX 方式

警告: dd 将会无区别写入任何指定的位置，即使给出的位置是硬盘(将会导致数据丢失/文件系统崩溃)。

插入空白的或者无需要数据的闪存设备，确定路径，用 dd 程序把 .iso 写入闪存设备：

```
# dd if=archlinux-2011.08.19-"{core|netinstall}"-"{i686|x86_64|dual}".iso of=/dev/sd"x"
```

if= 后面是.iso镜像文件的路径， of= 是 USB 的路径. **注意**要使用 /dev/sdx 而不是 /dev/sdx1（就是要使用整个分区而不是第一分区）。usb设备需要有足够大的空间用于写入ISO镜像。

要检查镜像是否正确写入，记下读入和写出的记录区数目，然后执行以下命令进行检查：

```
# dd if=/dev/sd"x" count="number_of_records" status=noxfer | md5sum
```

md5sum 命令返回的值要与所下载 md5sum of the downloaded archlinux image file (2011.08.19) (<http://www.archlinux.org/iso/2011.08.19/md5sums.txt>) 相同，它们都应该和下载站中的值相同。一个典型的操作过程如下：

首先把 .iso 写到硬盘，

```
# dd if=archlinux-2011.08.19-core-i686.iso of=/dev/sdc
```

```
744973+0 records in
744973+0 records out
381426176 bytes (381 MB) copied, 106.611 s, 3.6 MB/s
```

其实是完整性检验，

```
# dd if=/dev/sdc count=744973 status=noxfer | md5sum

4850d533ddd343b80507543536258229 -
744973+0 records in
744973+0 records out
```

然后跳转到启动 Arch Linux 安装程序继续

微软Windows方式

从这里 (<https://launchpad.net/win32-image-writer/+download>) 下载 Disk Imager。插入闪存设备，启动 Disk Imager 并选择映像文件(Disk Imager 仅接受 *.img文件，所以应该在打开对话框中输入 "*.iso" 以选中 Arch 文件)，再选择U盘相应的磁盘驱动号，点击"write"即可。

也有其他方法来写入可引导 ISO 镜像到闪存。如果遇到无法连接闪存的问题，试试其他的 USB 接口和连接线。

继续 启动 Arch Linux 安装程序

网络安装方式

如果你选择网络安装，就没必要刻录光盘或者写闪存了，可以直接网络启动.iso映像文件，但前提是你已经有一个开启的服务器。细节请参考 这篇文章 ，然后继续 启动 Arch Linux 安装程序。

在虚拟机中安装

如果你对Arch Linux还不够熟悉，希望“无痛”安装使用，那么使用虚拟机是很好的方式。虚拟机还允许你一边开着浏览器查看手册、一边安装系统。经过虚拟机中的安装实践，实体安装对你不再是难题。

常见的虚拟机软件有 VirtualBox、VMware、QEMU、Xen、Varch、Parallels。

不同虚拟机软件的使用方法大同小异，一般有一下几步：

1. 创建虚拟硬盘
2. 配置虚拟机参数
3. 将下载的ISO镜像作为CD设备启动
4. 继续按照下面的教程安装

下面的文章对你也许有帮助：

- VirtualBox 中安装 Arch Linux
- Installing Arch Linux from VirtualBox
- VirtualBox Arch Linux Guest On Physical Drive
- Installing Arch Linux in VMware

启动 ArchLinux 安装程序

小贴士: 基本安装所需内存为 64 MB

小贴士: 安装过程可能被自动黑屏打扰，如果黑屏了，按 Alt 键就可以安全地唤醒正常显示画面

引导

插入准备好的光盘或者闪存设备，然后用它启动电脑。可能需要在 BIOS 里改变一下启动顺序，通常是在启动后的加电自检过程中按 **Delete**, **F1**, **F2**, **F8**, **F10**, **F11** 或 **F12**。

主菜单： 引导后应该可以看到主菜单，用上下箭头高亮选项，按下回车键进行选择，不同的 ISO 菜单略有不同。

启动操作系统

选择从主菜单选择 "Boot Arch Linux" 并按回车，系统将加载并给出登录提示，自动以 'root' 登录。

注意: 通过 SSH 连接远程安装的用户需要进行一些特殊的调整才能 ssh 到 live CD 环境，请参阅从 SSH 安装。

启动问题解决

如果使用英特尔显卡，在引导阶段可能黑屏，问题可能和 kernel mode setting 有关。一个可能的解决方法是重启电脑，并在 grub 菜单上按 Tab 键进入内核选项。在最后加上一个空格后加上：

```
i915.modeset=0
```

如果是ati显卡，可能在引导阶段出现竖线屏，解决方法也是修改grub引导内容，也是在 kernel 项最后加上一个空格后加上：

```
nomodeset
```

完成菜单项里边的设置后，只要按下回车键就会继续引导，更多信息，请参阅 Intel 文档。

如果屏幕没有黑，但是启动过程停在内核装入，按 Tab 编辑内核行，加入：

```
acpi=off
```

修改后只需回车就会继续启动过程。

更换键盘映射表

如果使用非 US 键盘布局，可以通过下面的命令选择相应的键盘映射表。

```
# km
```

或者是使用 loadkeys 命令

```
# loadkeys layout
```

(把layout替换成你需要的布局，比如 fr 或者 be-latin1

文档

live CD 里面已经包含了官方安装指南。在安装的时候可以边安装边查看，按组合键 **Alt+F2** 切换到控制台 tty2 (virtual console #2)，以 root 登录然后在 # 提示符后输入：

```
# less /usr/share/aif/docs/official_installation_guide_en
```

使用 less 是因为它的翻页功能使得查看文档更方便。

按组合键 **Alt+F1** 回到安装过程。（整个系统安装过程中, 你随时都可以切换 tty2 查看安装向导）

小贴士: 请注意官方指南仅涵盖基本系统的安装和配置，强烈建议安装后回到此 Wiki 进行安装后的其它相关操作。

安装

注意: 如果你使用的联网方式是通过HTTP和/或者FTP代理的方式并且想使用DHCP来配置你的网络,你需要在shell中执行 /arch/setup 之前设定环境变量 http_proxy 和/或者 ftp_proxy：

```
export http_proxy=http://<http_proxy_address>:<proxy_port>
export ftp_proxy=ftp://<ftp_proxy_address>:<proxy_port>
```

在 tty1 下以 root 身份执行安装脚本

```
# /arch/setup
```

然后你就会看到 ArchLinux 安装框架的屏幕。

选择安装源

出现欢迎界面之后，会提示选择安装来源。请选择一个合适的源（这里可以看到相关源的速度信息 源状态 (<https://www.archlinux.de/?page=MirrorStatus>) ）。如果使用 Netinstall 镜像，相当速度和更新状态可以在 [这里](https://www.archlinux.de/?page=MirrorStatus) (<https://www.archlinux.de/?page=MirrorStatus>) 查询。

- 如果选择 CORE installer，并且选择使用 CD 上的软件包，请选择来 CD-ROM 作为源。
- 如果使用 Netinstall 安装，选择 NET 并查看下面的 (配置网络)。

配置网络 (netinstall)

如果需要，安装程序会提示手动加载网卡驱动。Udev 可以非常有效的载入需要的模块，所以网卡模块应该已被载入。要验证这个过程，可以按下 <Alt> +F3 并执行 ip addr。完成后，按 <Alt> +F1 返回 tty1。

在下一个屏幕，选择 **设置网络**。如果正确列出接口和 HWaddr (**HardWare address**)，那么所需的模块就是已经被加载好了。如果没有列出接口，那么请通过安装程序侦测，或者从另一个虚拟控制台手动进行，选择接口并继续。

接下在的屏幕会提示你 Select the interface, Probe, 或 Cancel 。选择适当的界面然后继续。

随后安装程序会询问是否使用 DHCP。选择 Yes 后将执行 **dhcpcd** 查找可用的网关和 IP 地址。选择 No 会提示设置 static IP, netmask, broadcast, gateway DNS IP, HTTP proxy 和 FTP proxy。最后, 将返回 **网络安装菜单** 选定 **选择镜像** 并选择一个 FTP/HTTP 镜像, 完成后, 返回到主菜单。

小贴士: 为了实现最佳的下载速度,你最好选择你国家内的源,以及那些你可以信任的源(eg. universities 大学)

注意: archlinux.org 官方源是限速的50KB/s

Live 环境下设置 ADSL 桥接 (可选)

(如果你用调制解调器或者桥接模式的路由器连接 ISP)

切换到另一个虚拟终端 (**Alt+F2**), 以 root 登录, 然后调用

```
# pppoe-setup
```

如果配置正确, 用以下命令就可以连接到 ISP:

```
# pppoe-start
```

回到第一个虚拟终端(**Alt+F1**), 跳到 设置编辑器。

Live 环境下设置无线网络(可选)

(如果安装时需要无线连接)

现在在安装媒体中已经包含了无线网络驱动和工具。对无线网络硬件有一定的了解是完成无线网络设置的重要前提。注意下面的**在安装时执行**的快速设置过程中会初始化**live环境**中的无线网络硬件。安装好系统以后, 无线网络设置(或其他无线管理操作)**需要再次执行一遍**。

并且注意如果安装过程中并不需要无线网络, 不需执行下面的步骤。可以在系统安装好之后再设置无线网络功能。

注意: 下面的示例中使用 wlan0 作为接口, linksys 作为 ESSID。请根据实际情况修改。

基本过程为:

- 切换到虚拟控制台: **Alt+F3**
- root 登录
- (可选)确定网络接口:

```
# lspci | grep -i net
```

如果使用 USB 适配器:

```
# lsusb
```

- 用 `/usr/sbin/iwconfig` 确定 udev 已经载入驱动, 而且驱动程序创建了可用的无线内核接口:

```
# iwconfig

lo no wireless extensions.
eth0 no wireless extensions.
wlan0  unassociated  ESSID:""
        Mode:Managed  Channel=0  Access Point: Not-Associated
        Bit Rate:0 kb/s  Tx-Power=20 dBm  Sensitivity=8/0
        Retry limit:7  RTS thr:off  Fragment thr:off
        Power Management:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

本例中 wlan0 为可用接口。

注意: 输出和上面不相似表示驱动没有载入，需要自己加入，详情请查看 无线设置。

- 用 `/sbin/ifconfig <interface> up` 启用接口：

```
# ip link set wlan0 up
```

除了驱动，一小部分无线网卡还需要固件。启用网络时，缺少固件的接口会显示：

```
# ip link set wlan0 up

SIOCSIFFLAGS: No such file or directory
```

若不确定，可以用 `/usr/bin/dmesg` 查询内核日志，看看有没有来自无线网卡的固件请求 Intel 芯片组在启动时请求固件的例子：

```
$ dmesg | grep firmware

firmware: requesting iwlwifi-5000-1.ucode
```

如果没有输出，表示系统的无线芯片不需要固件。

注意: 网卡需要的无线接口固件包在 live 环境中预先安装到了 `/lib/firmware`，但是必须特别安装到实际的系统才能在重启后提供无线功能！向导的后面会介绍软件包的选择和安装。请确保在软件包选择的时候选中需要的无线模块和固件！如有任何疑问，请查看 无线设置，这是常见错误。

- 如果 ESSID 忘记了或未知，使用 `/sbin/iwlist <interface> scan` 可以搜索附近的网络

```
# iwlist wlan0 scan

Cell 01 - Address: 04:25:10:6B:7F:9D
        Channel:2
        Frequency:2.417 GHz (Channel 2)
        Quality=31/70  Signal level=-79 dBm
        Encryption key:off
        ESSID:"dlink"
        Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s
        Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s
        36 Mb/s; 48 Mb/s; 54 Mb/s
```

- 如果使用 WPA 加密：

使用 WPA 加密需要将密钥加密与 ESSID 一起保存在文件中，以供 wpa_supplicant 使用。因此需要几个额外的步骤：

作为简化和备份，重命名默认的 wpa_supplicant.conf 文件：

```
# mv /etc/wpa_supplicant.conf /etc/wpa_supplicant.conf.original
```

使用 wpa_passphrase，给出无线网络名称和 WPA 密钥并写入 /etc/wpa_supplicant.conf

如下命令为密码 'my_secret_passkey' 的无线网络 'linksys' 生成配置文件 (/etc/wpa_supplicant.conf)通过重定向输入文件：

```
# wpa_passphrase linksys "my_secret_passkey" > /etc/wpa_supplicant.conf
```

更多信息和问题处理请查看 WPA Supplicant。

注意： /etc/wpa_supplicant.conf 是用纯文本格式保存，在安装环境中没有问题，但是重启到新系统后，记得改变 /etc/wpa_supplicant.conf 的权限(比如:使用 `chmod 0600 /etc/wpa_supplicant.conf` 使得仅 root 可读)。

- 关联无线设备和访问点，加密方式(无, WEP 或者 WPA)不同，执行内容也有区别，先确定网络名 (ESSID)，然后：

加密方式	命令
无加密	<code>iwconfig wlan0 essid "linksys"</code>
WEP w/ Hex Key	<code>iwconfig wlan0 essid "linksys" key "0241baf34c"</code>
WEP w/ ASCII passphrase	<code>iwconfig wlan0 essid "linksys" key "s:pass1"</code>
WPA	<code>wpa_supplicant -B -Dwext -i wlan0 -c /etc/wpa_supplicant.conf</code>

注意： 可以通过 Arch 默认的网络守护进程、netcfg、wicd 或其它网络管理器自动连接网络。

- 使用上面的方法后，等待一段时间，确认已经成功连接到访问点再继续：

```
# iwconfig wlan0
```

输出应该显示网络接口已经连接访问点。

- 用 /sbin/dhccpd <interface> 获取 IP 地址：

```
# dhccpd wlan0
```

- 最后，用 /bin/ping 确认连接正常：

```
$ ping -c 3 www.google.com
```

```
PING www.l.google.com (74.125.224.146) 56(84) bytes of data.  
64 bytes from 74.125.224.146: icmp_req=1 ttl=49 time=87.7 ms  
64 bytes from 74.125.224.146: icmp_req=2 ttl=49 time=87.0 ms  
64 bytes from 74.125.224.146: icmp_req=3 ttl=49 time=94.6 ms  
  
--- www.l.google.com ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2002ms  
rtt min/avg/max/mdev = 87.052/89.812/94.634/3.430 ms
```

到此，无线网络应该已经可用，如遇问题，请查看无线设置。

Alt+F1 回到 tty1，继续 设置时钟

设置编辑器

现在你会被提示选择用哪个编辑器来更改配置文件。有两个选择：nano 或 vi。nano 通常认为对新手更容易，因为它的行为比较直观，比较像图形界面的字处理软件。例如键盘的方向键、退格键和删除键和期望的功能一样。最常用命令的菜单（例如，剪切是 **Ctrl-k**，粘贴是 **Ctrl-u**，退出是 **Ctrl-x**）显示在nano终端显示的下方。请参考 wiki 页面 Nano 和 Vi 获取更多说明。

设置时钟

设置时区

使用上下方向键选择时区，可以键入地区首字母跳转。然后回车选择。

设置时间和日期

设置硬件时钟模式。如果设置得与其它操作系统不同，将覆盖时间并导致时间错误。

- UTC (推荐)

注意: 为硬件时钟使用UTC并不意味着软件所显示时间必为UTC.

- **localtime** (不推荐) - 微软 Windows 系统默认使用，但还是可以设置为 UTC，如果设置为本地时间，linux 不会进行夏令时调整。

警告: 使用 localtime 将导致很多已知的不可修复的BUG.尽管如此,没有计划放弃对localtime的支持.

注意: 选择其它值将不会更改时钟（虚拟机的时候比较有用）。

与Windows共存时的时间设置

如果机器上同时装有Windows和Linux，有两种设置时间的方法：

- 推荐：Arch Linux 和 Windows 都使用UTC时间（需要设置注册表，参见该文 (https://help.ubuntu.com/community/UbuntuTime#Make_Windows_use_UTC)）。注意，Windows上不要启用网络时间同步，否则会重新启用localtime。如果要使用时间同步（NTP），可以在 Arch Linux 上安装ntpd。
- 不推荐：让 Arch Linux 使用localtime，之后（在配置系统部分），在/etc/rc.conf文件中，从DAEMONS去掉hwclock（使用Windows配置时间）。

准备硬盘

警告: 磁盘分区会毁掉原分区内的数据。强烈建议先备份重要的数据。

注意: 磁盘分区也可以在执行 Arch 安装程序之前通过 GParted (<http://gparted.sourceforge.net/download.php>) 或者其他工具进行。如果磁盘上已经按要求做好了分区, 那么跳到选项3 (Manually configure block devices, filesystems and mountpoints) 继续安装。

用 `/sbin/fdisk` 的 `-l` (小写 L) 参数检查当前磁盘的分区参数和布局。

切换到另一个虚拟控制台 (**Alt+F3**), 然后:

```
# fdisk -l
```

记下将要安装上 Arch 的磁盘/分区。

按 **Alt+F1** 切换回安装脚本。

选择第一个菜单项 **Prepare Hard Drive**。会出现四个选项:

- **选项1**: Auto-Prepare (自动调整)

此举会擦除**整个**硬盘, 并自动完成分区。部分参数可手动调整。

- **选项2**: Manually partition hard drives (手动配置硬盘)

推荐。使用此选项, 用户可以使用 `cfdisk` 工具自由地设置分区。分区完成后, 进入选项3。

- **选项3**: Manually configure block devices, filesystems and mountpoints (手动配置块设备、文件系统和挂载点)

如果之前已完成硬盘分区, 可以直接进入这一选项。选项3也是选项2的下一步。安装程序会检查所有可用分区和挂载点, 用户需要设置是否使用、如何使用这些分区。用户还需要选择分区识别方式, 即 `dev`、`label`、`uuid`。

- **选项4**: Rollback last filesystem changes (撤销最后更改)

撤销本次还未应用的操作, 恢复硬盘到更改前的状态。

前三个选项在下面有更详细的介绍。要进一步理解这些选项, 首先我们将对Linux分区和文件系统作简单的介绍。如果你已了解这方面知识, 可以跳过, 接着看下一节选择软件包。

Note: 如果要安装 Arch Linux 到U盘, 请阅读 [Installing Arch Linux on a USB key](#)。

磁盘分区: 基本知识

分区类型

磁盘分区是指在磁盘上定义若干区域(分区), 每个都象是一个独立的磁盘, 可以在其上创建一个文件系统(格式化)。

分区的类型有三种:

- 主分区
- 扩展分区
 - 逻辑分区

主分区可以被启动，一个物理磁盘或RAID卷最多能有**4个主分区**。如果需要三个以上分区，需要使用包含**逻辑分区**的**扩展分区**。扩展分区可以看作存放逻辑分区的容器。一个硬盘最多只能有一个扩展分区。而且，扩展分区也算主分区，所以有扩展分区的话，就只能有另外三个主分区。扩展分区可以划分为无限多个逻辑分区。如果使用多系统，Windows必须装在主分区。

进行磁盘分区时，可以观察到命名规则为：主分区为 sda1 到 sda3，跟着是扩展分区 sda4，在扩展分区中划分出来的逻辑分区分别是 sda5、sda6，以此类推。

交换分区

交换分区（swap）是在硬盘上划出的一段虚拟内存区域，当物理内存已经放不下数据时，用来临时应急充当内存。

以前，一般规则是交换分区为物理内存的两倍，在内存越来越大的今天，这个规则已经逐步过时。如果你的内存较小，只有512M，那么作为日常应用，需要 2 倍就足够了。如果系统有大量内存(超过 1024M)，则可以完全不用交换分区，因为可以在以后使用 交换文件 替代。本例中使用 1G 交换分区。

注意: 如果使用休眠到磁盘，必须要**等于**物理内存的交换分区。这种情况下，推荐使用超过物理内存容量10%-15%。

分区结构

如何对磁盘进行分区全靠个人喜好，下面提供一些指导。设计分区结构一般要考虑扩展性、速度、安全性以及磁盘空间。两种简单而常见的结构是：一，一个根分区，一个交换分区；二，只有一个根分区。请继续阅读，以做出最佳选择。如果同时安装了Windows，请阅读Windows和Arch双系统。

下面的挂载点可以放在单独的分区中：

/（root，根）

根文件系统是目录等级的最顶部，所有其它文件系统都通过它访问，即使位于不同的物理设备，所有目录和文件也都显示在根目录/下面。根文件系统必须能够启动、恢复、还原或者修复系统。所以/下面的一些目录不应该位于不同分区。（参见下面的警告）。

/boot

此目录包含内核、ramdisk 镜像和启动加载程序的配置文件和不同阶段使用的文件。/boot 还可以保存内核开始执行用户空间程序前使用的数据，包括保存的主启动扇区和扇区映射文件。/boot 是启动必须的，但是特别的地方是它可以位于单独分区。

警告: 除了 /boot，启动系统必须的分区为: /bin, /etc, /lib 和 /sbin。和 /boot 不同，这些目录必须位于/。

/home

为每个系统用户提供子目录，保存个人的数据和应用程序配置文件。

/tmp

存放诸如 .lck 文件之类的临时文件，这些文件可以阻止程序运行多个实例。程序结束后会删除 '.lck' 文件。程序无法保证 /tmp 在程序运行过程中不会更改，而且 /tmp 下的目录和文件会在启动时删除。

/var

存放可变数据，比如管理和登录数据、pacman缓存、ABS目录树等。该分区的存在使得挂载 /usr 为只读成为可能，以前在 /usr 读写的数据现在都改在 /var 下进行。

注意： /var 下存放着很多零散的小文件。如果使用独立分区，需要斟酌一下文件系统类型（见下）。

比起使用单独一个大分区，使用独立的分区有如下优点：

- 安全：在 /etc/fstab 中，每个文件系统可以单独配置为 'nosuid'、'nodev'、'noexec'、'readonly' 等。
- 稳定：只要有写权限，一个用户或恶意程序可能在文件系统里写满垃圾。位于其它文件系统上的重要程序将免受影响。
- 速度：经常写入的文件系统会有碎片（避免碎片的方式是不将文件系统写满），单独的文件系统不受其它分区的影响，而且可以单独进行碎片整理。
- 保险：就算一个文件系统坏掉了，别的不会受影响。
- 通用：使用独立文件系统，更有利于在多个系统间共享文件。每个文件系统都能够根据数据和使用状况单独选择合适的格式。

我的分区应该设多大？

这个问题最好根据个人需要回答，请参考下面这些原则：

- 根文件系统（ / ）包含 /usr 目录，这样如果安装很多程序，会变得很大。15-20GB 应该能够满足大部分人的要求。
- /var 文件系统除包含软件数据，还有 ABS 树和 pacman 缓存。缓存软件包快捷方便；可以按照要求降级软件包。/var 的大小会逐步增长；pacman 缓存在运行中逐步增大，但是可以很方便的清空。如果使用 SSD 应该将 /var 放到 HDD，将 / 和 /home 分区放到 SSD，以减少 SSD 的读写。8-12GB 对桌面系统来说应该足够了，服务器需要更大的 /var 文件系统。
- /home 文件系统一般存放用户数据、下载和多媒体文件。在桌面电脑中，/home 一般是远超过其它的最大的文件系统。记住如果重装 Arch。/home 中的数据不会改变（只要 /home 是独立分区）。
- /boot 一般只需要大概 100MB。
- 如果空间充足，可以为每个文件系统多留 25% 空间，以应对以后的变化，还可以避免碎片。

创建文件系统：基本知识

文件系统类型

不同分区可以使用不同文件系统，每种文件系统都有其优缺点和特性。下面对支持的文件系统作简要介绍，链接指向的维基百科页面有更详细的介绍。

1. **ext2 Second Extended Filesystem**，古老、可靠的 GNU/Linux 文件系统。非常稳定，一个缺点是不支持日志记录或隔离。不支持日志会导致在突然断电或当机时可能导致数据丢失。因为文件系统检查的时间很长，不适合根目录（ / ）和 /home 使用。ext2 可以容易地转换成 ext3。
2. **ext3：Third Extended Filesystem**，基于 ext2 系统，并添加了日志记录功能。ext3 向前兼容 ext2，非常成熟稳定。
3. **ext4：Fourth Extended Filesystem**，向前兼容 ext2 和 ext3，最大支持 1EB（也就是 1,048,576 TB）文件系统，支持单个 16TB 的文件，增加 ext3 中的子目录最大个数 32,000 到 64,000，可以在线碎片处理。
4. **ReiserFS(V3)**：Hans Reiser 主持编写的高性能的日志文件系统，使用一种非常独特有趣的数据存储检索方法。ReiserFS 效率非常高，特别在处理很多小文件的时候更是如此。ReiserFS 在挂载的时候相对比较慢。性能稳定。ReiserFS 现在的开发并不活跃（最新的版本是 Reiser4）。通常是 /var 目录的

好选择。

5. JFS : IBM提供的日志文件系统(**Journaled File System**) , 是第一个支持日志的文件系统。它在 IBM AIX® 操作系统中开发了多年, 之后移植到 GNU/Linux 中。JFS 效率非常高并且CPU资源占用率比其他任何一个文件系统都要低。并且在格式化、挂载和磁盘检测的时候都非常快, 在各方面的表现都非常突出, 尤其是与 deadline I/O 调度器结合。(参见 JFS.) 不如ext系列或者ReiserFS那样广泛支持, 但非常成熟稳定。

注意: JFS 文件系统无法用磁盘工具如 **gparted** 或 **parted magic** 缩小。

6. XFS : 另一个很早的日志文件系统, 原来由 Silicon Graphics 为 IRIX 操作系统开发, 之后移植到 GNU/Linux。特别适合处理大文件。在格式化和挂载的时候也很快。基准测试显示在处理小文件时稍慢。XFS 非常成熟, 同时支持在线碎片处理。

注意: XFS 文件系统无法用磁盘工具如 **gparted** 或 **parted magic** 缩小。

7. vfat : 即**Virtual File Allocation Table** (虚拟文件分配表) 。这种文件系统技术简单, 受各种系统广泛支持。这种格式常用于固态存储卡, 便于系统间文件交换。VFAT支持长文件名。
8. Btrfs : 也被叫做 "Better FS" , **Btrfs**是一个新文件系统, 它具有许多 Sun/Oracle 优秀的ZFS文件系统的功能。包括快照、多磁盘带状划分(striping)和镜像(没有 mdadm 的软 raid), 校验、增量备份和实时压缩(即可以节省空间, 还能提高性能)等等。在 2011 年 1 月仍处于 "不稳定" 状态。但是已经以实验状态进入内核主代码树。Btrfs 将成为 linux 未来的文件系统, 各主要发行版的安装程序都提供了将其作为根文件系统的选项。

警告: Btrfs 还没有磁盘检查(fsck)工具, 所以如果数据损坏将无法修复。

9. nilfs2 : **New Implementation of a Log-structured File System** , 由NTT开发。该文件系统将所有数据以连续的类似日志的结构储存, 新数据只添加不改写。这种设计减少了寻址时间, 相对传统的 Linux 文件系统能防止在崩溃发生后的数据丢失。

关于日志

除ext2, 上面的文件系统都支持日志记录 (http://en.wikipedia.org/wiki/Journaling_file_system) 。日志文件系统具有数据容错能力, 在数据写入文件系统前会记录日志, 若写入时发生崩溃, 可以避免元数据损坏。注意不同文件系统日志记录有区别。仅 ext3 和 ext4 支持 data-mode journaling, 同时记录数据和元数据的日志, 但默认不启用, 因为对性能影响很大。其它文件系统仅提供记录元数据日志的ordered-mode journaling。尽管都能在系统崩溃后将系统返回正常状态, data-mode journaling 为数据提供了更大的安全防护, 但性能有所降低, 因为数据会被写两次 (第一次到日志, 第二次到磁盘) 。可以根据数据的重要性选择文件系统。

选项1 : 自动调整

菜单选项 : **Auto-Prepare**

擦除硬盘所有数据, 自动准备将磁盘设置成如下结构 :

- ext2格式 /boot 分区, 默认大小100MB, 可调整。
- 交换分区, 默认大小256MB, 可调整。
- 独立的 / 和 /home 分区, 可设置大小。可选的文件系统包括ext2、ext3、ext4、reiserfs、xfs、jfs、nilfs2 (慎用) 、btrfs (慎用) 。要注意的是, 如果选择自动配置, / 和 /home 将使用同样类型的文件系统。

需要特别小心的是, Auto-Prepare 将完全地删除所选磁盘上的数据。请非常仔细地阅读安装程序给出的 **warning** , 确认分区的磁盘名是正确的。

如果不想使用自动配置，请使用下面的选项2或者GParted等软件自动配置。若使用多系统，则应手动配置。

选项2：手动配置硬盘

菜单选项：**Manually partition hard drives**

选择要分区的硬盘，安装程序会调出cfdisk让用户手动分区（如果使用固态硬盘，最好使用gdisk或GNU Parted (http://www.gnu.org/software/parted/manual/html_mono/parted.html) 等其他分区工具）。下面将以一个160GB的硬盘为例，讲解分区步骤。示例硬盘将划分为15GB的根分区/、10GB的/var、1GB的交换分区、剩下的作为/home。需要再次强调的是，真正的分区取决于个人的选择，此处的示例仅供参考。

选择 **New -> Primary** 并输入根分区（/）的大小（Size），将分区放到磁盘开始。选择**Type**，设置为 83 Linux，本例中新建的/分区显示为 sda1。

使用相同步骤创建第二个主分区 /var，大小10.256GB，**Type**（分区类型）设置为 83 Linux。新建的/var分区显示为 sda2。

然后是交换分区。大小1GB左右，**Type**为 82 (Linux swap / Solaris)。新分区的设备号为 sda3。

剩下的空间作为 /home 目录，步骤同上，**Type**设置为 83 Linux。新分区的设备号为 sda4。

完成之后，分区界面应该类似下面这样：

Name	Flags	Part Type	FS Type	[Label]	Size (MB)
sda1		Primary	Linux		15440 #root
sda2		Primary	Linux		10256 #/var
sda3		Primary	Linux swap / Solaris		1024 #swap
sda4		Primary	Linux		133000 #/home

选择 **Write** 并输入 yes。注意，该操作会破坏磁盘上的数据！选择 **Quit** 退出分区程序。

确保你已熟悉文件系统方面的知识，然后进行选项3。

注意: 因为最新的 Linux 内核加入了 libata 和 PATA 模块，所有的 IDE, SATA 和 SCSI 磁盘都会使用 sdx 命名，这很正常，不必担心。

选项3：手动配置块设备、文件系统和挂载点

菜单选项：**Manually configure block devices, filesystems and mountpoints**

进行该步前需要完成分区（选项2中完成），并且设备能通过dev目录、label或UUID识别。安装程序会列出所有可识别分区，每个分区都有一个类似 sda1 的标识，表示某硬盘上的第一个分区。而 sda 表示整块硬盘。

为每个分区选择并创建文件系统（格式化分区），并设置挂载点。还可以设置盘符（label）和mkfs格式化时的附加选项。

注意: 如果不需要 /boot 分区，可以忽略 /boot 分区不存在的警告。

接下来，返回到主菜单。

选择软件包

安装时的软件都来自 [core] 软件源，它们被分为 **base** (i686 (<http://www.archlinux.org/groups/i686/base/>) |x86_64 (http://www.archlinux.org/groups/x86_64/base/)) 和 **base-devel** (i686 (<http://www.archlinux.org/groups/i686/base-devel/>) |x86_64 (http://www.archlinux.org/groups/x86_64/base-devel/))。这里 (http://www.archlinux.org/packages/?repo=Core&arch=any&arch=i686&arch=x86_64&limit=all&sort=pkgname) 有软件包信息和简介。

启动加载器

首先选择 GRUB 或 syslinux 作为启动加载器。

软件包组

现在选择软件包类别：

注意: 为方便，默认选中了 **base** 中的所有软件包，请用空格键切换选择状态。

base

来自 [core] 软件源的最小基本系统，总是选中它并删除不用的软件包。

base-devel

来自 [core] 的附加工具例如 make 和 automake。大部分新手都应该安装它，后续会用到。

选中类别后，将给出软件包的完整列表以细化选择，用空格键切换选择状态。

注意: 如果需要连接无线网络，请选中 wireless_tools (https://www.archlinux.org/packages/?name=wireless_tools) 软件包。一些无线接口还需要 **ndiswrapper** 和特定 **firmware**。如果使用 WPA 加密，需要 **wpa_supplicant**。无线设置 页面将帮助您为无线设备选择正确的软件包。同时强烈推荐安装 netcfg，将帮助你在重启后设置网络连接。

选择好需要的软件包后，离开选择屏幕并继续到下一步，**安装软件包**。

安装软件包

安装选中的软件包到新系统，如果用 CD/USB 作为源，将安装 CD/USB 中的软件包。如果选择网络安装，将从互联网下载最新的软件包。

注意: 一些安装程序会询问是否希望保留 pacman 缓冲里的软件包。如果你选 'yes'，日后可以降级某些软件包，因此推荐保留(今后还可以清除这些缓存)。

下载软件包后，安装程序会做完整性检查，接着从下载的软件包中创建内核。

配置系统

小贴士: 理解并完全安装步骤设置是保证系统配置成功的关键。

本步骤将配置 Arch Linux 系统的主要配置文件。

你会看到一个菜单，包含了系统的主要配置文件。

注意: 此时，编辑，或者至少打开验证一下所有配置文件很重要。安装程序根据输入的信息创建这些文件。跳过这些关键配置步骤是个常见错误。

安装程序不能自动处理吗？

隐藏系统配置过程直接违背**Arch 之道**。确实，最近的内核和硬件检测工具可以提供绝好的硬件支持和自动配置，Arch 依然在安装过程中将配置文件展现给用户以维护**透明化和有利于系统资源控制**。在你完成这些配置文件的修改之后，就学会了简单的 Arch Linux 系统的配置，熟悉了基本结构，为更好的使用和维护新系统做好准备。

/etc/rc.conf

ArchLinux 使用 /etc/rc.conf 作为系统的主要配置文件。这个文件包含了广泛的配置信息，主要用于系统的启动。正如名字中所说，其中还包含了供 /etc/rc* 使用的配置信息。

LOCALIZATION 部分

LOCALE

系统本地化设置，所有国际化的应用程序都会使用这个设置。从命令行运行 `locale -a` 将显示可用的设置。此设置缺省是适合英文（US English）用户使用的 `en_US.utf8`。（此处本地化设置必须与 /etc/locale.gen 一致，**详见下文**。）中文一般是 `zh_CN.UTF-8`，如果使用之后乱码，先换回 `en_US.utf8`，等进图形界面后再换回来。

DAEMON_LOCALE

使用本地 daemon("yes" or "no").如果回答 "yes" 将把 \$LOCALE 使用环境变量, 否则使用 C locale (如果使用默认值 "no").

HARDWARECLOCK

设置开关机时进行时间同步的硬件时钟，可用选项为 **UTC** 时间 [通用协调时(Universal Time Coordinated)，过去曾用格林威治平均时(GMT)来表示] 还是 **localtime**（本地硬件时间）。通常 UTC 较为好，因为它大大简化了时区和日光节约时间（中国过去称为夏时制）的变更。而当你同时还有其它操作系统如视窗进行双启动，就必须使用本地硬件时间。

TIMEZONE

指定时区。（所有可用的时区可在 /usr/share/zoneinfo/ 中找到。）

KEYMAP

可用的键盘映射表位于 /usr/share/kbd/keymaps。请注意此设定仅在 TTY 中有效，在所有图形窗口管理器或者 X 中都不使用此设置。

CONSOLEFONT

如果你一定要修改此设置，可在 /usr/share/kbd/consolefonts/ 中找到可用的字体（比如终端中文显示，就需要设定）。

CONSOLEMAP

定义了启动时 setfont 程序加载的控制台映射表。有需要的话，可在 /usr/share/kbd/consoletrans 找到可用的映射表。

USECOLOR

如果你使用彩色显示器的话请选 "yes"。

LOCALIZATION 示例:

```
LOCALE="en_US.utf8"
DAEMON_LOCALE="no"
HARDWARECLOCK="UTC"
TIMEZONE="PRC"
KEYMAP="us"
```

```
CONSOLEFONT=
CONSOLEMAP=
USECOLOR="yes"
```

HARDWARE 部分

MODULES

如果知道缺少哪个重要的模块，可以在这里面具体指定。如果使用 loopback 文件系统，添加 "loop"。同时可以通过在模块前加感叹号(!)屏蔽模块，将强制 Udev 不载入屏蔽的模块。因为 Udev 可以检测到绝大部分设备，所以这里基本都是空的。

Example for HARDWARE:

```
# Scan hardware and load required modules at boot
MODULES=()
```

NETWORKING 部分

HOSTNAME

设置为你喜欢的主机名 (HOSTNAME)。这也是计算机的名字，设置后请在 /etc/hosts 设置相同的名字

例子:

```
HOSTNAME="arch"
```

interface

指定所有网络接口。多个接口请用空格隔开：(eth0 wlan0)

address

如果你想使用特定的IP,在这里设定. **如果使用DHCP就留空.**

netmask

可选, 默认255.255.255.0. 如果你想使用特定的子网掩码,就在这里设定. **如果使用DHCP就留空.**

broadcast

可选,如果你想使用特定的广播地址,在这里设定. **如果使用DHCP就留空.**

gateway

如果使用**静态IP**，将网关 (gateway) 地址设为路由器的IP。如果使用**DHCP**，可以忽略此选项，但有些用户报告说同样需要设置。

NETWORK_PERSIST

如果想阻止网络关机就"yes".如果你的启动设备在NFS上这个就是必须的.

NETWORKS

这个是可选的设定为可用,如果你使用netcfg并且使用dialog. 在启动的时候就启用netcfg的配置. 当你的机器运行在很多的网络环境中时,这个很有用 (比如笔记本(laptop)用户).

示例 静态 IP:

```
HOSTNAME=arch
interface=eth0
address=192.168.1.100
netmask=255.255.255.0
```

```
broadcast=192.168.1.255
gateway=192.168.1.1
#NETWORKS=(main)
```

示例 动态 IP (DHCP):

```
HOSTNAME=arch
interface=eth0
address=
netmask=
broadcast=
gateway=
#NETWORKS=(main)
```

使用静态 IP 时，请修改 `/etc/resolv.conf` 以指定 DNS，详情参见下面的部分。

注意: 自动连接到无线网络需要更多的工具(netcfg、wicd)和步骤。详情请查看 [无线设置页面](#)。

小贴士: 如果需要使用非标准 MTU 大小(即jumbo frames) 而且硬件支持，请查看Jumbo Frames 获得配置方法。

DAEMONS 部分

这个序列罗列了需要在开机启动脚本的名字，这些脚本位于 `/etc/rc.d/`。同时还支持异步启动以提高开机速度

```
DAEMONS=(network @syslog-ng netfs @crond)
```

- 如果脚本名字前有惊叹号(!)，表示它不被执行。
- 如果脚本名字前有"at"符号(@)，则表示它会在后台被执行，启动序列不等待脚本运行成功就会继续执行下一脚本。例如"mpd" 依赖 "network"，所以 network 后台启动将影响 mpd 的正常启动。
- 安装新服务后如果需要开启启动，请编辑此序列。

注意: 这种'BSD风格'的初始化正是Arch的行为方式，其它发行版则是在init.d目录里使用一大堆符号链接。

关于 DAEMONS

虽然此时不需修改daemons行，不过还是有必要解释一下什么是待命进程，因为在指南后面会需要它。

通用信息

daemon 表示那些运行于后台的程序，一般用于提供各种服务。一个明显的例子是网页服务器等待请求，服务器端的daemon程序将被请求的页面发送出去(例如:httpd) 或 SSH 服务器等待用户登录(例如:sshd)。有些daemon进程工作起来并不很显眼。例如写系统日志的待命进程（如syslog、metalog）和图形用户界面的程序(例如: gdm, kdm)。所有这些程序都可以被加入到daemons行中，随着系统启动而启动。本指南会对一些较有用的待命进程进行介绍。

小贴士: Arch 守护进程脚本位于 `/etc/rc.d/`

`/etc/fstab`

fstab(**file systems table**的缩写) 是系统配置的一部分，该文件列出所有可用的磁盘和分区，并指明它们初始化的方式或整合至全局文件系统的方式。`/etc/fstab` 文件一般被**mount**命令读取，**mount** 以设备上的文件系统为参数，将其添加到主目录结构。在启动过程执行到 3/4 左右的时候，`/etc/fstab` 会调用 **mount -a**，**mount** 读取 `/etc/fstab` 以确定挂载时使用的选项。如果 **noauto** 位于 `/etc/fstab` 中的文件系统之后，**mount -a** 不会在启动时挂载它。

这是一个 `/etc/fstab` 示例：

```
# <file system>          <dir>    <type> <options>          <dump> <pass>
tmpfs                    /tmp    tmpfs  nodev,nosuid 0    0
UUID=0ddfb25-9b00-4143-b458-bc0c45de47a0 /      ext4   defaults      0    1
UUID=da6e64c6-f524-4978-971e-a3f5bd3c2c7b /var    ext4   defaults      0    2
UUID=440b5c2d-9926-49ae-80fd-8d4b129f330b none    swap   defaults      0    0
UUID=95783956-c4c6-4fe7-9de6-1883a92c2cc8 /home   ext4   defaults      0    2
```

注意: `fstab` 提供了更多的信息和性能调整选项，例如 `'noatime'/'relatime'`。

`/etc/mkinitcpio.conf`

注意: 大部分用户都不需要在安装时修改此文件，但请阅读下面的解释信息。

此文件负责微调初始内存文件系统 `initramfs`。(曾被称为初始化内存磁盘或 `"initrd"`)。 `initramfs` 是启动时供内核读取的压缩镜像。目的是引导系统使其能够读取根文件系统。所以它需要装入 IDE、SCSI、SATA 或 USB/FW 等设备所需模块。`initramfs` 手动或通过 `udev` 装入所需的模块后，会将控制权交给内核，继续启动过程。因此 `initramfs` 仅需要包含读取根文件的必要文件。大部分内核模块都是在初始化过程中由 `udev` 装入。

mkinitcpio是下一代 **initramfs 生成器**。与以前的 **mkinitrd** 和 **mkinitramfs** 脚本相比具有许多优势。

- 使用 **glibc** 和 **busybox** 提供小巧轻量的早期用户运行环境。
- 可以利用 **udev** 在运行时进行自动侦测，可以避免装入大量不需要的模块。
- 基于钩子的初始脚本可以通过自定义钩子扩展，通过 `pacman` 软件包提供而不需要修改 `mkinitcpio` 本身。
- 已经支持 **lvm2**，**dm-crypt**，包括 `legacy` 和 `luks volumes`。 **raid**，**swsusp** 和 **TuxOnIce**从usb mass storage 设备唤醒和启动。
- 许多功能可以通过内核命令行配置，无需重新编译。
- **mkinitcpio** 脚本可以在内核中包含镜像，所以能够实现自我包含的内核。
- 十分灵活，许多时候都不需要重新编译内核。

如果根文件系统使用 RAID 或 LVM，必须配置正确的 HOOKS。详情请查看 RAID 和 `/etc/mkinitcpio`。如果使用非 US 键盘，请添加 `"keymap"` 钩子以载入本地键盘映射。如果使用 USB 键盘，请添加 `"usbinput"`。否则，如果出了什么问题无法启动，系统会要求输入密码以进行维护，但确实无法使用键盘。将 Arch 安装到外部硬盘、CF 卡或 SD 卡时，记得添加 `"usb"`，示例：

```
HOOKS="base udev autodetect pata scsi sata usb filesystems keymap usbinput"
```

如果需要通过如下设备启动，请按需配置 HOOKS：USB 设备、FireWire、PCMCIA、NFS、软 RAID、LVM2 卷、加密卷 和 DSDT 等。

`/etc/modprobe.d/modprobe.conf`

设置内核模块的装入。详细信息请访问内核模块。

/etc/resolv.conf

注意: 如果使用 DHCP，可以直接忽略此文件，dhcpcd 默认会动态创建和销毁此文件。可以通过配置改变默认行为，更多信息请查看Network 和 resolv.conf。

resolver 是 C 库中的一系列访问DNS(Domain Name System)的函数。DNS 的主要功能是将域名转化为 IP 地址。解析配置文件 /etc/resolv.conf 包含了解析器初始化时读取的信息。

如果使用静态 IP，需要在 /etc/resolv.conf 中设置域名服务器 (nameserver <ip-address>)，数量不限。例如，使用 OpenDNS:

```
nameserver 208.67.222.222
nameserver 208.67.220.220
```

如果使用路由器，需要在路由器中指定 DNS 服务器，并在 /etc/resolv.conf 中将域名服务器指定成路由器的 IP(同时也是 /etc/rc.conf 中的网关)。例如：

```
nameserver 192.168.1.1
```

如果使用 **DHCP**，也可以在路由器中指定域名服务器，或者允许 ISP 自动指定。

/etc/hosts

此文件将 IP 地址与主机名和别名对应。 每行一个 IP 地址，每个主机应该用单行表示为：

```
<IP-address> <hostname> [aliases...]
```

将 /etc/rc.conf 指定的hostname设置为别名：

```
127.0.0.1 localhost.localdomain localhost yourhostname
```

警告: 为了程序的兼容性，必须严格遵照这个格式，**包括'localhost' 这项！**所以如果主机名为 "arch"，那么上面的内容应该变为：

```
127.0.0.1 localhost.localdomain localhost arch
```

错误的句法可能会导致网络性能低下或某些程序运行缓慢甚至无法运行。对新手来说这是十分普遍的错误。

注意: Arch Linux 安装程序的最近版本会在你修改 /etc/rc.conf 后自动将主机名加入。如果不是，请使用前面的指令更新文件信息。

如果你使用静态 IP，新加另外一行: <static-IP> <hostname.domainname.org> <hostname> 即:

```
192.168.1.100 yourhostname.domain.org yourhostname
```

小贴士: 为了方便可以用 `/etc/hosts` 给本地网络或互联网上的主机起别名, 例如::

```
64.233.169.103 www.google.com g
192.168.1.90 media
192.168.1.88 data
```

这样在浏览器中输入 'g' 就可以访问 google, 不用打 IP 地址也能访问多媒体和数据服务器。

`/etc/locale.gen`

`locale-gen` 命令从 `/etc/locale.gen` 读取系统的 locales 信息。这些设置稍后会被 **glibc** 和其他一些支持本地化的程序或者库用于自动翻译输出各种语言的"独特" 文本, 并且合适的显示地域、货币、时区以及日期格式、字符排列方式和其他一些本地化的特定标准。

默认情况下 `/etc/locale.gen` 是一个仅包含注释文档的空文件。编辑后, 这个文件将不会更改。每次 **glibc** 更新之后就会运行 `locale-gen` 一次, 重新生成 `/etc/locale.gen` 指定的本地化文件。

选定你需要的本地化类型(移除前面的 # 即可), 比如:

```
#en_US ISO-8859-1
#en_US.UTF-8 UTF-8
zh_CN.GB18030 GB18030
zh_CN.GBK GBK
zh_CN.UTF-8 UTF-8
zh_CN GB2312
```

现在, 安装程序将运行 `locale-gen` 脚本, 以生成你设定的 locales。你以后可以通过修改文件 `/etc/locale.gen` 变更 locale, 然后使用根用户 (root) 运行 '`locale-gen`' 脚本。

注意: 如果没有设置 locale, 将提示 "The current locale is invalid..." 错误。而这可能是 Arch 新手犯的最多的错误。

`/etc/pacman.conf`

`pacman` 会在每次启动时访问 `/etc/pacman.conf`。此配置文件分为许多段或源。每个源定义了一个 `pacman` 搜索软件包的源。此外还有 `options` 段定义了一些全局变量。

删除 '`Include =`' 和 '`[repository]`' 行前面的 # 将会启用对应的源。

注意:

- 默认设置应该能工作, 所以这时不需要修改, 但是建议验证一下。更多信息位于官方源。
- 选择源时, 头文件和括号行都要取消注释。否则会忽略选择的源。

`/etc/pacman.d/mirrorlist`

为 `pacman` 选择一个软件仓库镜像站。主站 `archlinux.org` 是有流量限制的, 下载速度限于 50KB/s。访问 `Mirrors` 获取更多关于 `pacman` 镜像的信息。注意此处选择的镜像将进入新安装的系统。

Root密码

最后，设置一个root密码并确定你已牢记它。返回主菜单然后接着安装启动加载器。

完成

选择 "Done"，之后，系统会重建镜像并返回到主菜单，这个步骤会花费一些时间。

安装启动加载器

示例中仅有一个操作系统，所以需要启动加载器。推荐使用 GRUB(**GR**and **U**nified **B**ootloader)，除此之外可以使用 LILO 或 Syslinux。如果使用其他启动加载器，请参考相关的 wiki 和文档。

提供的 **GRUB** 配置文件 (/boot/grub/menu.lst) 应该足够了，但请仔细检查文件内容以确保准确性(特别是要确定在第三行 UUID 指定的根分区)。如果要修改控制台的分辨率，可以用 vga= <number> 内核选项。(分辨率和数值对应表位于 menu.lst。)

示例：

```
title Arch Linux (Main)
root (hd0,0)
kernel /boot/vmlinuz-linux root=/dev/sda1 ro quiet
initrd /boot/initramfs-linux.img
```

从单独 /boot 分区启动：

```
title Arch Linux (Main)
root (hd0,0)
kernel /vmlinuz-linux root=/dev/sda3 ro quiet
initrd /initramfs-linux.img
```

将 **GRUB** 启动加载器安装到主启动记录 (本例中为 /dev/sda)。

如果计算机上同时还共存有其他操作系统请在此处添加启动项并修改参数。

警告: 请确定安装到 **/dev/sdX** 而 **不是 /dev/sdX#**! 这是常见错误。

小贴士: 详情参见 GRUB。

重启系统

好了，Arch Linux 基本系统已经安装配置完毕。退出安装并重启：

```
# reboot
```

小贴士: 确保移除安装媒介并修改 BIOS 中的启动选项；否则可能会启动后会回到安装界面！

配置

恭喜，欢迎来到 Arch Linux 系统!

此部分包含一些安装后的必须操作，包括更新新系统和添加正常非 root 用户。

更新

新的 Arch Linux 基本系统已经是可以工作的 GNU/Linux 环境了。以此为起点，你可以将这些优雅的工具集变成期望或要求的样子。

用 root 登录，我们将用 root 用户配置 pacman 并更新系统。

注意: 虚拟控制台 1-6 都是可用的，可以通过 <ALT> + F1...F6 切换。

配置网络 (若需要)

如果正确配置了系统，网络应该正常工作，通过 `ping example.com` 验证：

```
$ ping -c 3 example.com

PING example.com (192.0.43.10) 56(84) bytes of data:
64 bytes from 43-10.any.icann.org (192.0.43.10): icmp_req=1 ttl=248 time=25.6 ms
64 bytes from 43-10.any.icann.org (192.0.43.10): icmp_req=2 ttl=248 time=22.9 ms
64 bytes from 43-10.any.icann.org (192.0.43.10): icmp_req=3 ttl=248 time=23.6 ms

--- example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 22.912/24.062/25.632/1.156 ms
```

如果网络连接成功，继续 **用 pacman 更新、同步、升级系统**。

如果 `ping example.com` 收到 "unknown host" 错误，网络可能没设置好。请检查如下文件：

`/etc/rc.conf` - 检查 `HOSTNAME` 和 `NETWORKING` 部分中的错误。

`/etc/hosts` - 检查格式、拼写和错误。

`/etc/resolv.conf` - 如果使用静态 IP 请检查。如果使用 DHCP，默认会动态创建和销毁该文件。

小贴士: 网络配置的高级选项请查阅文章 [配置网络](#)。

有线局域网

用下面命令检查以太网：

```
$ ip addr
```

将显示所有接口，应该包含 `eth0` 或者 `eth1`。例子中将使用 `eth0`。

静态 IP

如果需要，可以设置新 IP:

```
# ip addr add <ip>/<netmask> dev <interface>
```

默认网关：

```
# ip route add default via <ip>
```

验证 `/etc/resolv.conf` 包含 DNS 服务器，如果没有请加上。再次用 `ping -c 3 www.google.com` 检查网络。如果一切正常，根据上面的静态 IP 调整 `/etc/rc.conf`。

DHCP

如果使用 DHCP 服务器/路由，请尝试：

```
# dhcpcd eth0
```

如果成功，按上面所说为动态 IP 调整 `/etc/rc.conf`。

无线局域网

无线网络连接请参见 Live 环境下无线快速设置。尽管已经不是在安装媒体上运行，只要在软件包选择时安装了无线连接需要的软件包，这些命令还是一样的。记住，无线设备可能需要固件(firmware) 才能运行。错误排查请访问 无线设置。

代理服务器

如果使用代理服务器，请在 `/etc/wgetrc` 中设置 `http_proxy` 和 `ftp_proxy`。

模拟调制解调器、ISDN 和 DSL (PPPoE)

参阅 Internet Access。

用 pacman 更新、同步、升级系统

下面将使用 pacman 升级系统。Pacman 是 Arch Linux 的软件包管理器(**package manager**)，负责管理 Arch linux 的整个软件包系统，处理安装、删除、软件包降级(通过 cache)、自定义编译软件包、自动解决依赖关系删除和本地查找等等。现在将用 Pacman 从远程源下载软件包并安装到系统。

注意: 如果通过网络安装，即使不是全部，大部分软件包已经是最新了。但是仍然建议完成整个更新过程。

`/etc/pacman.conf`

要修改 pacman 源或者选项，请编辑 `/etc/pacman.conf`。

软件包源

软件包源 是一个可以下载安装程序软件包的存储位置。Arch Linux 软件包维护者 (开发者和 Trusted Users) 维护了一系列官方软件源，包含了关键和流行的软件，可以通过 pacman 直接获取。本文档将列出官方支持的软件源，更多信息及每个源的目的请参阅 官方源。

大部分人会使用 `[core]`、`[extra]` 和 `[community]`。如果想在 Arch x86_64 上运行 32 位应用程序，请在 `/etc/pacman.conf` 中加入如下内容以启用 `[multilib]` 源：

```
[multilib]
Include = /etc/pacman.d/mirrorlist
```

AUR

Arch 用户仓库 包含 [community] 和 [unsupported] 分支。与其它分支不同，[unsupported] 不包含二进制文件，设计为无法通过 pacman 直接安装。此分支收集 Arch 用户通过 Arch 编译系统 编译软件的 PKGBUILD bash 脚本。[unsupported] 中的软件通常别的分支都没有，但 [unsupported] 软件包获得足够的投票后，一个 授信用户 会认领它，将其加入 [community] 二进制源。

注意: 一些 pacman 帮助程序 可以无缝访问 AUR.

/etc/pacman.d/mirrorlist

定义 pacman 源镜像和优先级。

注意: 如果安装媒介较老，mirrorlist 可能已经过期，升级时会产生问题(bug 报告在此 (<https://bugs.archlinux.org/task/22510>))。因此最好从 pacman 镜像列表生成页面 (<http://www.archlinux.org/mirrorlist/>) 获取最新镜像。将产生的列表复制到 /etc/pacman.d/mirrorlist 再继续。

用编辑器打开 /etc/pacman.d/mirrorlist 并取消离你近的镜像前的注释(删除前面的 '#')，然后刷新软件包列表：

```
# pacman -Syy
```

传入两个 --refresh 或 -y 标记强制 pacman 刷新所有软件包，不管是否被认为是最新的。**只要镜像有变化就执行 pacman -Syy** 是防止出现令人头疼问题的好习惯。

rankmirrors

另外,你可以使用 rankmirrors. rankmirrors 是一个在 /etc/pacman.d/mirrorlist 中取消注释一些与安装机器的软件版本最接近以及最快速的源的bash脚本.更快的源可以显著的提升pacman的性能,和arch的整体操作体验.这个脚本可以定期的执行,尤其是当你所选择的源和官方源有差异的时候.但是rankmirrors不能测试这些源的速度.可以使用 wget 或者 rsync 这2个工具可以在一个新生成的 /etc/pacman.d/mirrorlist }的有效的用来测试这些源的速度.

执行下面的命令可以完全的刷新pacman的数据库,并且安装 curl:

```
# pacman -Syyu curl
```

- 当你运行命令得到了错误,你可以使用 nano /etc/pacman.d/mirrorlist 来取消注释一个适合你的源地址

cd 到 /etc/pacman.d/ 目录:

```
# cd /etc/pacman.d
```

备份已经存在的 /etc/pacman.d/mirrorlist:

```
# cp mirrorlist mirrorlist.backup
```

编辑mirrorlist.backup注释掉一些在一个大陆或者相邻地域的源(比如china,japan)来测试rankmirrors.

```
# nano mirrorlist.backup
```

让rankmirrors带上参数-n对这个备份文件mirrorlist.backup执行操作,然后把输出重定向以方便生成一个新的/etc/pacman.d/mirrorlist源列表:

```
# rankmirrors -n 6 mirrorlist.backup > mirrorlist
```

注意: -n 6:将生成6个最接近的源

然后使用pacman使用新的源列表来刷新所有的包列表:

```
# pacman -Syy
```

Mirrorcheck 以检查最新软件包

因为rankmirrors不考虑镜像中软件包的更新状况,选择为最快的一两个源可能已经过期了。ArchLinux 镜像状态 (<http://www.archlinux.org/mirrors/status/>) 报告了镜像的各种状态,包括网络问题,数据收集问题、上次同步时间等等。如果需要最新的软件包,最好手动检查一下 /etc/pacman.d/mirrorlist,确保文件包含了最新的镜像。

此外 镜像生成器 (<http://www.archlinux.org/mirrorlist/>) 可以自动根据更新状况生成离你比较近的镜像列表。

一个使用即时的源来更新你的源列表的脚本

你可以根据上面的排名使用下面的shell脚本来更新你的源列表.如果你不在中国China,你可以改变下面的国家变量.

```
updatemirrors.sh
```

```
#!/bin/sh
```

```
[ "$UID" != 0 ] && su=sudo
```

```
country='China'
```

```
url="http://www.archlinux.org/mirrorlist/?country=$country&protocol=ftp&protocol=http&ip_version=4&use_mirror_status=on"
```

```
tmpfile=$(mktemp --suffix=-mirrorlist)
```

```
# Get latest mirror list and save to tmpfile
```

```
wget -qO- "$url" | sed 's/^#Server/Server/g' > "$tmpfile"
```

```
# Backup and replace current mirrorlist file
```

```
{ echo "Backing up the original mirrorlist..."
```

```
$su mv -i /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.orig; } &&
```

```
{ echo "Rotating the new list into place..."
```

```
$su mv -i "$tmpfile" /etc/pacman.d/mirrorlist; }
```

注意: 你需要拷贝上面的命令到一个新的文件,并且使用chmod +x 来让这个文件可以执行.如果你不是使用

的root来执行这个文件,这个脚本将让你输入命令来更新源列表文件

熟悉 pacman

pacman 是 Arch 用户最好的朋友。强烈建议通过 `man pacman(8)` 学习并使用它：

```
$man pacman
```

如果发现文字太长，屏幕显示不全，可以导出 `$MANWIDTH` 环境变量。在 root 的主目录创建 `.profile` 文件，加入：

```
# export MANWIDTH=80
```

更多信息请访问 `pacman` 或者查看 `pacman rosetta` 中与其他流行软件包管理器的对比。

更新系统

现在已经做好了更新整个系统的准备。在此之前，请阅读 新闻 (<http://www.archlinux.org/news/>) (或者 通告邮件列表 (<http://archlinux.org/pipermail/arch-announce/>))。开发者通常会针对已知问题提供需要配置和修改的重要信息。在升级前访问这些页面是个好习惯。

同步、刷新、升级整个系统：

```
#pacman -Syu
```

或者：

```
#pacman --sync --refresh --sysupgrade
```

pacman 会从服务器下载 `/etc/pacman.conf` 中定义的主软件包列表，进行所有可用的升级操作。此时可能会提示说 pacman 自己需要先进性升级，如果这样，请选择是并在完成后重新执行 `pacman -Syu`。

如果升级了内核，请重启系统。

注意: 时不时的，升级之后需要用户执行配置修改，详情参见 `Pacnew` and `Pacsave` Files。

pacman 输出会保存到 `/var/log/pacman.log`。

管理和更新软件包的常见问题位于 软件包管理 FAQs。

忽略软件包

执行 `pacman -Syu` 会升级整个系统。可以单独阻止某个软件包升级，比如某个升级会导致系统问题的时候。这时，有两个选择：使用 `pacman` 命令行选项 `--ignore` 忽略它，(详情参见 `pacman -S --help`) 或将软件包加入 `/etc/pacman.conf` 文件的 `IgnorePkg` 列表。更多信息请访问 `pacman`。

请注意有能力的用户应该用 `pacman -Syu` 保存系统最新而不是选择性的升级软件包。选择性的安装可能更和要求，但是事先提醒一下，这种方式更有可能产生问题。大部分的抱怨都是因为选择性升级、非正常编译或在软件安装。所以不建议使用 `/etc/pacman.conf` 中的 **IgnorePkg**，仅在极其个别的情况下使用。

Arch 滚动发布模式

请记住 Arch 是 **滚动发布** 的发行版。这意味着升级到新版本不需要重装或者重新建构。只需要定期执行 `pacman -Syu` 就可以将系统保持在最新前沿状态。升级后，系统完全是最新的。如果内核升级了，请记得**重新启动**。

警告: 进行系统升级时请小心，执行前请仔细阅读并理解 这个帖子 (<https://bbs.archlinux.org/viewtopic.php?id=57205>) 。

添加一个用户

Linux 是个多用户环境。请不要使用 `root` 登录进行日常工作。这不仅仅是坏习惯，还非常危险。Root 账户是用来做管理任务的。因此，使用 `adduser` 命令来添加一个普通的、非 `root` 帐号。

```
#adduser
```

程序将交互式运行，要求用户输入所需信息。下例中将创建用户archie：

```
Login name for new user []: archie
User ID ('UID') [ defaults to next available ]:
Initial group [ users ]:
Additional groups (comma separated) []: audio,lp,optical,storage,video,wheel,games,power,scanner
Home directory [ /home/archie ]:
Shell [ /bin/bash ]:
Expiry date (YYYY-MM-DD) []:
```

正如例子中所示，建议仅在Login name 和 Additional groups 输入内容，其它都留空。

Additional groups 中的列表是桌面系统的典型选择，特别推荐给新手：

- **audio** - 让任务可以调用声卡以及相关软件
- **lp** - 管理打印任务
- **optical** - 管理光驱相关任务
- **storage** - 管理存储设备
- **video** - 视频任务以及硬件加速
- **wheel** - 使用 `sudo`
- **games** - 得到那些属于游戏组的权限，比如手柄
- **power** - 笔记本用户需要这个
- **scanner** - 使用扫描仪

接着会给出用户信息预览，可以取消或者继续。

```
New account will be created as follows:
```

```
-----
Login name.....: archie
```



```
UID.....: [ Next available ]
Initial group....: users
Additional groups: audio,lp,optical,storage,video,wheel,games,power,scanner
Home directory...: /home/archie
Shell.....: /bin/bash
Expiry date.....: [ Never ]

This is it... if you want to bail out, hit Control-C. Otherwise press
ENTER to go ahead and make the account.
```

按下 **ENTER** 后将创建账户，并提升输入新用户的附加可选信息：

```
Creating new account...

Changing the user information for archie
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
```

最后，提示输入新帐号的密码：

```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Account setup complete.
```

新非 root 用户创建完成，同时还建立了用户主目录和登录密码。

删除账户

如果你弄错了账户设置，或者你想删除一个账户，或者你想要换个账户名，或者任何其他什么原因，使用 `/usr/sbin/userdel`：

```
# userdel -r [帐户名]
```

- `-r` 此选项将在删除账户的同时删除其主目录及其中的文件，用户的 mail spool 中的文件也会被删除。

更多信息位于用户和组，如果你想更换你的或者其他用户名，请参阅更改用户名。也可以查看 `usermod(8)` 和 `gpasswd(8)` 的 man page。

其他

现在应该有了一个完全可用的 Arch 系统，这是后续调整的基础。然而，大部分用户都希望有声音和图形的桌面系统。这个部分将提供添加这些附加功能的指导。

Sudo

安装 Sudo:


```
# pacman -S sudo
```

要将一个账户加到 sudo 账户中(a "sudoer")，必须得用 root 身份来执行 visudo 命令。

默认的，visudo 命令使用 vi 编辑器。如果你不知道如何使用 vi，那么你可以设置 EDITOR 环境变量来选择你想要的编辑器。例如，在这个例子中，使用 "nano" 编辑器：

```
# EDITOR=nano visudo
```

注意: 设置这个变量和启动 visudo 必须在同一行中完成。如果分成两行，这个变量将不起作用。

如果你擅长使用 vi，那就不带 EDITOR=nano 来执行 visudo 命令：

```
# visudo
```

这将在一个 vi 特别会话中打开 /etc/sudoers 文件。visudo 先将其复制到一个临时文件，然后使用编辑器编辑生成的临时文件 (默认是 vi)，编辑完成会执行完整性检查。如果检查通过，临时文件将会以正确的账户权限覆盖原先的文件。

警告: 不要用编辑器直接编辑 /etc/sudoers；sudo 配置文件中的句法错误将会引起麻烦 (例如使得 root 权限不可用)。请务必使用 visudo 命令编辑 /etc/sudoers。

在上一部分，你的账户已经加入了 "wheel" 组。要设定使用前缀 "sudo" 时给 wheel 组中用户完全的 root 权限，请编辑 sudo 配置文件：

```
%wheel    ALL=(ALL) ALL
```

现在只要简单地把账户加入到 wheel 组中，就能给与任何账户执行 sudo 命令的权利。

更多信息，例如 sudoer <TAB> 补全，参阅 Sudo

声音

如果需要声音，请访问ALSA页面，或者先完成下一部分，再设置声音。

Linux高级声音架构 (ALSA) (Advanced Linux Sound Architecture)是一个Linux内核组件，推荐先试着使用它。如果不能工作，可以试试OSS。OSSv4 已经以自由许可证发布，普遍认为要比 ALSA 替代的 OSSv3 好很多。设置请访问 OSS 文章。

如果对声音有更高的要求，Sound 包含了许多有用的文章。

图形用户界面

安装 X

X 窗口管理系统(**X11**或者**X**) 是基于网络的显示协议，提供了窗口功能，包含建立图形用户界面(GUI)的标准工具和协议。

现在，用 pacman 安装基础的 **Xorg** 组中的包。这是构建图形用户界面的第一步。

安装基础包:

```
# pacman -S xorg-server xorg-xinit xorg-utils xorg-server-utils
```

安装 mesa 以获得 3D 支持:

```
# pacman -S mesa
```

3D 工具 glxgears 和 glxinfo (<http://dri.freedesktop.org/wiki/glxinfo>) 都包含在 mesa-demos (<https://www.archlinux.org/packages/?name=mesa-demos>) 包里。如果需要这些工具，请安装这个包:

```
# pacman -S mesa-demos
```

注意: 如果是从 Virtualbox 虚拟机安装，需要用另一种方式安装 X，参见 此文，然后跳到下面的配置部分。

安装显卡驱动

接下来需要安装显卡驱动。

先要确定显卡类型，使用 /usr/sbin/lspci 程序可以获得显卡类型：

```
$ lspci
```

注意: vesa 驱动是最通用的驱动，大部分现代显卡都能使用。如果无法找到显卡芯片组的对应驱动，可以使用 vesa 但仅提供不带加速的 2D 性能。

输入下面命令，查看所有**开源**驱动:

```
$ pacman -Ss xf86-video
```

注意: NVIDIA 和 ATI 闭源驱动在后面介绍，如果有很多三维处理需求，例如游戏等，请考虑它们。

用 pacman 安装合适的显卡驱动，例如 Savage 驱动：

```
# pacman -S xf86-video-savage
```

小贴士: 一些 Intel 显卡需要进行配置才能有较好的 2D、3D 性能，参见 Intel。

nVIDIA 图形显示卡

NVIDIA 有三种驱动选择(外加 vesa 驱动):

- 开源 nouveau 驱动，提供了快速的 2d 加速和基本 3d 支持，基本混合已经够用。(注意: 还不能完全支持节能)。更能矩阵。(<http://nouveau.freedesktop.org/wiki/FeatureMatrix>)
- 开源但混杂过的 nv 驱动，很慢，只支持 2d。
- 闭源 nvidia 驱动，提供了很好的 3d 性能和节能支持。即使计划使用闭源驱动，还是建议先使用开源的 nouveau，在 X 能够正常工作之后再切换。Nouveau 通常能够直接使用，而 nvidia 需要配置和解决许多问题。参见 NVIDIA。

开源的 nouveau 适合大部分人，所以推荐使用：

```
# pacman -S xf86-video-nouveau
```

基本 3D 支持:

```
# pacman -S nouveau-dri
```

小贴士: 高级指令请访问 Nouveau。

ATI图形显示卡

ATI 用户有两种驱动可选(外加 vesa)

- **xf86-video-ati**软件包提供的开源**radeon**驱动，详情参见radeon 功能矩阵 (<http://wiki.x.org/wiki/RadeonFeature>)
- catalyst (https://aur.archlinux.org/packages.php?O=0&K=catalyst&do_Search=Go) 提供的闭源 **fglrx** 驱动，软件包位于AUR，只支持较新设备(HD2xxx 及之后)。以前位于 extra 软件源，但官方支持于2009年3月停止，因为其质量和开发速度令人失望。参见 ATI Catalyst。

推荐使用开源驱动，安装 **radeon** ATI 驱动：

```
# pacman -S xf86-video-ati
```

小贴士: ATI配置的高级指引可以参看 ATI wiki 文章。

安装输入设备驱动

Udev 应该能够正确检测设备；evdev (xf86-input-evdev) 是现代热插拔输入驱动，可以支持绝大部分设备。所以通常不需要安装输入驱动。此时，evdev 已经作为 Xorg 的依赖关系被安装。

如果 evdev 不支持您的设备，请从xorg-input-drivers组安装需要的驱动。

用 pacman 查找所有输入设备驱动：

```
# pacman -Ss xf86-input
```

注意: 如果准备禁用热插拔，请安装 xf86-input-keyboard 或 xf86-input-mouse 否则 evdev 将会成为输入驱动。

笔记本(或触摸屏)用户需要 synaptics 软件包以支持触摸板/触摸屏：

```
# pacman -S xf86-input-synaptics
```

小贴士: 触摸板优化或问题处理请访问Touchpad Synaptics。

配置 X (可选)

警告: 闭源驱动安装或配置后通常需要重新启动，参见 NVIDIA 或 ATI Catalyst。

X 服务器可以自动配置，即使没有 xorg.conf 文件也能很好地运作。如果仍然需要手动配置，请访问 Xorg。

非 US 键盘

如果不是使用标准 US 键盘，可以在 /etc/X11/xorg.conf.d/10-evdev.conf 中设置：

```
Section "InputClass"
    Identifier "evdev keyboard catchall"
    MatchIsKeyboard "on"
    MatchDevicePath "/dev/input/event*"
    Driver "evdev"
    Option "XkbLayout" "be"
EndSection
```

注意: XkbLayout 的值可能与 km 或 loadkeys 命令中使用的不同，例如 uk 布局对于 gb。

测试 X

此部分将解释如何安装基本的图形环境以测试X。使用的是简单的默认X窗口管理器 twm。

安装默认的测试环境：

```
# pacman -S xorg-twm xorg-xclock xterm
```

默认的 X 环境很简单，下面的部分 将安装桌面环境或窗口管理器以支持 X。

如果在创建普通用户前安装 Xorg 在 \$HOME 会有一个空 .xinitrc，请删除或编辑它再启动图形环境，若删除，X会运行默认环境(twm, xclock, xterm)。

```
$ rm ~/.xinitrc
```

消息总线

注意: 许多应用程序都需要 dbus，如果确定不需要它，请跳过这个部分。

安装 D-BUS:

```
# pacman -S dbus
```

启动 dbus 守护进程 并将 dbus 加入 DAEMONS 数组 使其自动启动。

启动 X

注意: 曾经用来杀掉 X 的 Ctrl-Alt-Backspace 快捷键已经过时，默认不能使用。要启用 Ctrl-Alt-Backspace 可以编辑 xorg.conf，参见 这里。

最后，启动 Xorg:

```
$ startx
```

或

```
$ xinit -- /usr/bin/X -nolisten tcp
```

这时应该能看到一些可以移动的窗口，鼠标也能够工作。完成验证之后，可以通过在终端输入 `exit` 退出X。

如果黑屏，可能需要切换到其他控制台(例如 CTRL-Alt-F2)，然后盲打 `root` 然后输入回车(<Enter>)接着是 `root` 密码和 <Enter>。

可以用 `/usr/bin/pkill` 命令杀死 **X** 服务器。(注意 **X** 大写)：

```
# pkill X
```

如果 `pkill` 还是不行，重启系统：

```
# reboot
```

若有问题

如出现问题，请从 `/var/log/Xorg.0.log` 中查看错误日志，注意所有以 (EE) 开头的行，EE 代表有错误。同时注意 (WW) 警告，可能预示着其他问题。

```
$ grep EE /var/log/Xorg.0.log
```

可以从 **X** 启动终端之外的虚拟控制台查看错误。

更多的 Xorg 配置和问题解决请查看Xorg。

需要帮助?

如果访问 Xorg 后还有问题，需要在 Arch 论坛提问，请安装和使用 `wgetpaste`:

```
#pacman -S wgetpaste
```

在论坛提问的帖子中，用 `wgetpaste` 提供如下文件的链接：

- `~/.xinitrc`
- `/etc/X11/xorg.conf`
- `/var/log/Xorg.0.log`
- `/var/log/Xorg.0.log.old`

`wgetpaste` 用法：

```
$wgetpaste </path/to/file>
```

在帖子中提供相应的链接，请同时提供硬件和驱动的信息。

警告: 解决 X 相关问题时，详细信息很关键。在 Arch 论坛提问时，请提供上面所说的内容的详细信息。

安装字体

(注：以下将会使用yaourt，关于yaourt的安装，请参考：Yaourt (简体中文))

在安装桌面环境/窗口管理器**之前**，也许你会先安装些美观的字体。Dejavu 是不错的字体集。英文字体优先选择dejavu字体

```
# pacman -S ttf-dejavu
```

对于中文字体，开源的文泉驿正黑矢量字体是不错的选择，它还内嵌了9pt-12pt的点阵宋体：

```
# pacman -S wqy-zenhei
```

当然现在流行的是安装1个字体:

```
# yaourt -S wqy-microhei
```

可能有人需要安装微软视窗下的字体，如下安装之：

```
# yaourt -S ttf-win7-fonts ttf-office-2007-fonts
```

请访问 [字体配置](#) 获取配置字体渲染的详细信息，Fonts 提供了字体选择建议和安装方法。

选择并安装图形界面

X 窗口管理系统为构建图形用户界面（GUI）提供了基础框架。

注意: 桌面环境的选择是主观的、因人而异。请根据实际需求选择最佳的环境。

窗口管理器(WM)

和 X 窗口系统协同工作，控制应用程序窗口的位置和外观。**详细信息请访问窗口管理器。**

桌面环境(DE)

在X之上并与其共同运作，提供完整的功能和动态图形界面。桌面环境通常提供图标、小程序（applets）、窗口、工具栏、文件夹、壁纸、应用程序和拖放等功能。**详情请访问 桌面环境。**

注意: 可以用 WM 和选定的应用程序建立自己的桌面环境。

安装图形界面之后，可以继续阅读一般建议页面获得后续动作的指导。

启动图形环境的方法

手动

有些人倾向使用命令手动启动图形界面。各桌面环境相关的命令请访问相关的 wiki。通用的X 命令请查看 Xorg 页面的相关部分。.

自动

如果要使用图形界面自动启动，请查看 显示管理器 获得启动管理器的使用方法。Start X at Boot 提供了两个不依赖显示管理器的轻量方法。

附录

常用应用程序 页面包含了常用软件的索引。

一般建议 页面包含了安装后的教程，包括设置 CPU 频率自动调节和字体渲染等。

Arch Linux 中文化 页面包含了关于系统、软件中文支持的内容。

Retrieved from "https://wiki.archlinux.org/index.php?

title=Beginners%27_Guide_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)&oldid=190310"

Categories: 简体中文 | Getting and installing Arch (简体中文) | About Arch (简体中文)

| Website Resources (简体中文)

-
- This page was last modified on 19 March 2012, at 16:55.
 - Content is available under GNU Free Documentation License 1.2.