

- Download
- AUR
- Bugs
- Wiki
- Forums
- Home

ArchLinux 新手指南 (简体中文)

From ArchWiki

i18n

Dansk – Deutsch – English – Español – Français – Indonesia – Italiano – Lietuviškai – Magyar – Nederlands – Polski – Português – Română – Slovenský – Suomi – Svenska – Türkçe – Česky – Ελληνικά – Български – Русский – Српски – Українська – עברית – ไทย – 日本語 – 正體中文 – 简体中文 – 한국어

Contents

- 1 前言
 - 1.1 所有你一直想知道的关于安装Arch的事
 - 1.2 别怕 (DON'T PANIC) !
 - 1.3 授权协议
 - 1.4 Arch 之道
- 2 第一部分：安装基本系统
 - 2.1 一、获取最新的系统安装包
 - 2.1.1 CD 系统安装包
 - 2.1.2 USB安装方式
 - 2.2 二、启动Arch Linux系统安装程序
 - 2.2.1 登录并根据需要更换键盘映射表
 - 2.2.2 边安装边查看安装指南
 - 2.3 三、开始安装
 - 2.3.1 1、选择安装来源
 - 2.3.2 2、设置时间
 - 2.3.3 3、硬盘分区配置
 - 2.3.3.1 分区信息
 - 2.3.3.2 交换分区swap
 - 2.3.3.3 规划分区
 - 2.3.3.4 开始创建分区cfdisk
 - 2.3.3.5 IDE设备，为什么看起来它们和SCSI设备一样？
 - 2.3.3.6 设置文件系统挂载点
 - 2.3.4 4、选择软件包
 - 2.3.5 5、安装软件包

Summary

本文针对新手，提供了极其详尽的 Archlinux 的安装过程向导，和相关的简要说明。

相关文章

[Arch 官方安装指南 \(简体中文\)\(更通用的安装方式\)](#)

[ArchLinux新手指南附录 \(简体中文\)](#)

[ArchLinux安装完后的小技巧 \(简体中文\)](#)

[好用的应用程序 \(简体中文\)](#)

[Work and Play with Arch \(Chinese\)](#)

- 2.3.6 6、配置系统
 - 2.3.6.1 系统配置主文件/etc/rc.conf
 - 2.3.6.2 关于守护进程
 - 2.3.6.3 磁盘和分区配置文件 : /etc/fstab
 - 2.3.6.4 mkinitcpio.conf、modprobe.conf和 kernel26-fallback.conf
 - 2.3.6.5 (对于静态IP) 网络配置文件 : /etc/resolv.conf
 - 2.3.6.6 /etc/hosts
 - 2.3.6.7 /etc/hosts.deny和/etc/hosts.allow
 - 2.3.6.8 /etc/locale.gen
 - 2.3.6.9 Root密码
 - 2.3.6.10 Pacman镜像
- 2.3.7 7、安装启动加载器
- 2.3.8 8、系统重启
- 3 第二部分：完成基本系统并升级
 - 3.1 一、配置网络 (如果有需要)
 - 3.1.1 有线网络
 - 3.1.2 无线网络
 - 3.1.3 模拟信号调制解调器
 - 3.1.4 ISDN
 - 3.1.5 DSL (PPPoE)
 - 3.2 二、更新Pacman
 - 3.2.1 pacman是什么 ?
 - 3.2.2 1、配置pacman
 - 3.2.2.1 pacman总配置文件/etc/pacman.conf
 - 3.2.2.2 (如果有必要) 配置镜像列表 /etc/pacman.d/mirrorlist
 - 3.2.3 常用的命令
 - 3.2.4 软件包管理增强工具
 - 3.2.5 图形化管理工具
 - 3.2.6 Pacman的更多信息
 - 3.2.7 2、更新Pacman自己
 - 3.3 三、更新系统
 - 3.3.1 使用pacman
 - 3.3.2 滚动发布之美
 - 3.4 四、增加一个用户并设置用户组
 - 3.5 五、安装建立Sudo (可选)
- 4 第三部分：安装X并配置ALSA
 - 4.1 一、配置声卡：使用alsamixer
 - 4.1.1 测试声音
 - 4.1.2 保存声音配置
 - 4.2 二、安装X
 - 4.2.1 1、安装X
 - 4.2.2 2、安装显卡驱动
 - 4.2.3 nVIDIA图形显示卡
 - 4.2.4 ATI图形显示卡
 - 4.2.5 3、安装输入设备驱动软件包
 - 4.3 三、配置X：创建并修改/etc/X11/xorg.conf (如有需要)

- 4.3.1 配置文件xorg.conf介绍
- 4.3.2 1、创建 /etc/X11/xorg.conf
- 4.3.3 更多的创建/etc/X11/xorg.conf的办法
- 4.3.4 2、Input设备热插拔问题：请在测试X前阅读
- 4.3.5 3、测试 X
- 4.3.6 对xorg.conf除错
- 4.3.7 调整键盘布局
 - 4.3.7.1 如果开启了input热插拔
 - 4.3.7.2 如果关闭了input热插拔
- 4.4 简单的命令行界面X测试（如果需要）
 - 4.4.1 准备工作：配置~/.xinitrc
 - 4.4.2 开始执行测试
- 5 第四部分：安装并配置桌面环境
 - 5.1 一、安装字体
 - 5.2 二、选择默认启动的桌面环境 ~/.xinitrc
 - 5.3 三、选择安装一个桌面环境
 - 5.3.1 GNOME
 - 5.3.1.1 关于GNOME
 - 5.3.1.2 安装
 - 5.3.1.3 对GNOME有用的守护进程
 - 5.3.1.4 美化
 - 5.3.2 KDE
 - 5.3.2.1 关于KDE
 - 5.3.2.2 安装
 - 5.3.2.3 有用的KDE守护进程
 - 5.3.3 Xfce
 - 5.3.3.1 关于Xfce
 - 5.3.3.2 安装
 - 5.3.4 *box
 - 5.3.4.1 Fluxbox
 - 5.3.4.2 Openbox
 - 5.3.5 fvwm2
 - 5.4 四、优化和收尾
 - 5.4.1 HAL
 - 5.4.2 启动时让守护进程在后台执行
 - 5.4.3 美化液晶显示器(LCD)下的字体
 - 5.4.4 让所有的鼠标按键生效
 - 5.4.5 笔记本电脑可能遇到的问题
 - 5.4.6 设置CPU频率
- 6 第五部分：常用的应用程序
- 7 附录
 - 7.1 问题答疑
 - 7.2 术语
 - 7.3 接下来还有什么？

前言

所有你一直想知道的关于安装Arch的事

这篇文档将会指导你安装和配置Arch Linux——一个简洁、轻量级的GNU/Linux发行版，一个类UNIX的操作系统。本文是面向Arch新用户，因此尽力为所有人提供强大的参考资料和信息。本指南讲述该如何获得一个基本配置好的Arch Linux系统(拥有一个图形桌面环境)，但显然是不可能涵盖（甚至是预见）所有的信息。因此本指南会专注于某些关键性的步骤，想要进一步深入了解你可以查看Arch Linux Wiki或者Arch Linux论坛 (<http://bbs.archlinux.org/>)。同时有兴趣的话你还可以了解一下Arch之道，它概述了Arch Linux发行版背后的哲学思想。

Arch Linux的相关特性:

- 类Unix式的简古纯粹设计哲学 (**Simple UNIX-like design and philosophy**)
- 针对有一定经验的GNU/Linux用户开发的社区发行版 (**Community distro targeted at competent GNU/Linux users**)
- 所有的软件包都是为*i686/x86_64*所专门编译的
- 从底层开始就允许用户高度定制
- BSD风格的启动脚本和一个可以集中管理的配置文件 (**BSD-style init scripts, featuring one centralized configuration file**)
- **滚动升级**
- **Pacman**是一个用**C**写成的包管理器，既小巧又灵活，并且不会占用太多内存
- **ABS**: ABS是“**Arch Build System**”的缩写 (“Arch构建系统”)，这是一个类似于ports那样的包构建系统，能让您非常方便地从源代码开始构建您自己的可安装Arch包文件，便于您与AUR社区分享
- **AUR**: AUR是“**Arch User Repository**”的缩写 (Arch用户软件库)，它提供了数以千计的构建脚本来帮助安装Arch用户们所提供的软件包

别怕 (DON'T PANIC) !

要知道Arch Linux的安装方法和过程也许会和你所使用过的其它GNU/Linux发行版非常不同，特别是你是个新手的话将更能体会到这点。通常，GNU/Linux发行版会在一张或者一堆CD（甚至是DVD）上捆绑了大量的软件，包含了缺省桌面环境、缺省的应用程序和设置、硬件自动检测和图形化的安装程序，一切都是替你选择好了的。相反的，Arch Linux系统是让用户自己只是用一个bash shell和基本系统工具，从ncurses installer和基础系统上建造起来的。这就是Arch之道。不象其它大型发行版，它没有强加于用户的缺省环境或者配置。你可以在命令行使用pacman通过互联网连接从Arch软件仓库安装软件包，直到你的系统配置成你希望的样子。这给予了用户最大的弹性、选择性和对系统资源的控制。因为是你建造了它，所以你肯定能了解系统里具体细节，知道底层都有些什么。

Arch Linux系统是通过编辑文本文件进行配置的，没有提供图形界面的工具，所以需要对系统和linux相对熟悉。Arch Linux针对的是有能力的GNU/Linux用户以及希望花费时间学习系统知识的用户。

*Arch*是一个可以被用户塑造的工具。

授权协议

Arch Linux, pacman, documentation, and scripts are copyright ©2002-2007 by Judd Vinet,
©2007-2008 by Aaron Griffin and are licensed under the GNU General Public License Version 2.

Arch 之道

Arch的设计思想是“简古纯粹” (*keeping it simple*)。

这里的‘朴’不单指‘简单’或者‘用户友好’，而且还是‘没有不必要的增加、修改或者复杂性’。简而言之，即优雅、极简之法。

"Simple' is defined from a technical standpoint, not a usability standpoint. It is better to be technically elegant with a higher learning curve, than to be easy to use, and technically crap."
-Aaron Griffin

"The extraordinary part of [my method] lies in its simplicity..I have always believed that the simple way is the right way." - Bruce Lee “(我的方法中) 最超凡的部分是它的纯粹性，我一直相信纯粹的就是对的。”李小龙 (Bruce Lee)

Occam's razor: *Entia non sunt multiplicanda praeter necessitatem* or "Entities should not be multiplied unnecessarily." The term razor refers to the act of shaving away unnecessary assumptions and complications to get to the simplest explanation, method or theory.

Note: 仔细遵循本指南是成功安装一个配置正常的Arch Linux系统的基本要点，因此请认真仔细的阅读本文

- 你也可以将本指南打印出来，作为一个较为有用的Arch Linux用户手册。
- 如果你想在这个wiki里增加内容，请适当描述清楚“为什么会这样”以及“该怎么办”。最佳的文档应该能教会我们如何做和为什么这么做！
- Arch的wiki是项优秀的资源，可以第一时间 (http://wiki.archlinux.org/index.php/Main_Page) 用来解决日常问题；如果找不到答案也可以求助IRC和论坛 (<http://bbs.archlinux.org/>)。

欢迎使用Arch！现在，让我们开始吧。

本文按照安装过程分四大部分：

第一部分：安装基本系统

第二部分：完成基本系统并升级

第三部分：安装X并配置ALSA

第四部分：安装并配置桌面环境

第一部分：安装基本系统

一、获取最新的系统安装包

你可以从这里 (<http://archlinux.org/download/>) 获取Arch最新的官方系统安装包。最新版本是2009.08

- 不论是Core还是Netinstall系统安装包，都只是包含了安装**Arch Linux基本系统**所需的软件包。注意，基本系统是不包含GUI图形界面的。它主要包含了GNU的系列工具（编译器、汇编程序、链接、库、脚本，还有其它的一些有用的工具），Linux的内核，还有一些额外的库和模块。

- Core映像文件可以较方便的通过CD或者网络进行安装。
- Netinstall映像文件较小巧，本身没有提供软件包，整个系统都是通过internet来获取的。
- isolinux映像文件是给那些使用grub版本时发生问题的用户的。两者没有其它区别。
- Arch64问答([http://wiki.archlinux.org/index.php/Arch64_问答_\(简体中文\)](http://wiki.archlinux.org/index.php/Arch64_问答_(简体中文)))可以帮助你在32位版本和64位版本之间作出选择。

CD 系统安装包

把下载的.iso映像文件刻录到CD上。开机时用CD启动，出现界面，跳到 **二、启动Arch Linux系统安装程序**

Note: 通常为了刻录的稳定可靠（速率越低刻录质量越有保证），推荐刻录的最大速度不超过16倍速。有些用户推荐**速度为 4x 或者 2x**. 如果所刻录的CD有问题，可以尝试使用你系统所支持的最小速度方式刻录。

USB安装方式

Warning: 这样做会把USB盘里面原有的数据格式化。

UNIX 方式：

插入USB盘，找到其路径，用/bin/dd命令把.img映像文件写入USB (Linux下)：

```
dd if=archlinux-2009.08-{core|netinstall}-{i686|x86_64}.img of=/dev/sdx
```

if= 后面是.img镜像文件的路径，of= 是你的USB路径. 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)

检查镜像的完整性md5sum (可选):

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/sdx count=记录区数目 status=noxfer | md5sum
```

md5sum命令返回的值 (这是你检查U盘上文件得到的) 要与你下载的archlinux镜像文件的md5sum校验值相同;它们都应该和镜像站点的md5sum文件列表里相应的值相同。

Windows方式:

从<https://launchpad.net/win32-image-writer/+download> 下载Disk Imager。插入U盘，启动Disk Imager并选择映像文件，再选择U盘相应的磁盘驱动号，点击"write"即可。

二、启动Arch Linux系统安装程序

Usually, the first item is the preferred selection. However, choose Boot Archlive or Boot Archlive [legacy IDE] if you have trouble with libata/PATA, or have no SATA (Serial ATA) drives.

Also, to change boot options, press **e** for editing the boot lines. Many users may wish to change the resolution of the framebuffer, for more readable console output. Append: 插入刻录好的光盘CD或者USB盘，然后用它启动电脑。可能需要在BIOS里（通常是启动后在加电自检过程中按DEL、F1、F2、F11或者F12）改变一下启动顺序使得CD或者USB先被启动。

Note: 内存需求：

- Core : 160 MB RAM x86_64/i686 (选择所有软件包，开启交换分区)
- Netinstall : 160 MB RAM x86_64/i686 (选择所有软件包，开启交换分区)

启动了Arch Linux光盘后，用上下箭头高亮菜单，按下回车键进行选择。

通常，推荐选择第一项菜单。不过如果使用libata/PATA出现问题或者没有SATA (Serial ATA)驱动器，你也可以选择Boot Archlive或者Boot Archlive [legacy IDE]。

如果你需要改变启动参数，可以按‘e’键来编辑启动行内容。含多用户可能希望改变framebuffer分辨率以得到更好的终端显示。将

```
vga=773
```

加到kernel一行的后面，然后按<ENTER>，就可以得到1024x768的framebuffer。设置完毕后按**b**则使用相应的选择项进行启动。

系统将会启动并提示登录，这时可以用'root'进行登录（不要加上单引号）。

如果你的机器从live CD启动时发生错误或者一些其它**硬件hardware**错误，请参看Installation Troubleshooting。

登录并根据需要更换键盘映射表

如果你使用non-US键盘布局，可以用命令选择相应的键盘映射表。

```
# km
```

或者是是用loadkeys命令

```
# loadkeys layout
```

(把`layout`替换成你需要的，比如 "fr" 或者 "be-latin1")

边安装边查看安装指南

在刻录的livecd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vc/2 (alt + f3, alt+f4..... 分别可以切换到对应的控制台vc/3 , vc/4.....以此类推)，然后输入：

```
# less /arch/docs/official_installation_guide_en
```

按组合键`<ALT>+F1`回到控制台`vc/1`，继续安装系统。安装过程中可以随时切换回`vc/2`来查看安装指南。`less` 是个文本查看程序，位于`/usr/bin/less`

Tip: 请注意，官方指南仅涵盖基本系统的安装和配置过程。一旦安装完毕，强烈建议回到本wiki浏览更多安装后续已经相关问题。

三、开始安装

键入

```
/arch/setup
```

开始安装。

1、选择安装来源

接着会出现一个选择安装来源的提示。如果你用core ISO的话请选择CD，如果使用FTP ISO的话则选择FTP。

- 如果选择 CORE installer, 可以直接看下面的硬盘分区配置。
- FTP/HTTP only: 选择这个需要预先加载网络驱动以便联网安装. 使用Udev来加载这些模块非常有效.

按组合键`ALT + F3`切换到控制台`vc/3`，确认一下看其是否生效 invoking `ifconfig -a` . (Select OK to continue.) ——把后面的网络配置提前看

2、设置时间

有两个选择：

- UTC - 如果使用unix系统包括Linux, Freebsd等, 建议使用UTC
- local - 如果和Windows共享多引导系统, 建议使用local

3、硬盘分区配置

选择第一个菜单项"Prepare Hard Drive"。请注意，"Auto-Prepare" (自动分区) 并不是最佳的选择，因为它只分配了最基本的三个区 (`/`, `/home` , `swap`) 在这里我们用手工进行硬盘分区。选择"2. Partition Hard Drives"，接着选要分区的硬盘(比如：`/dev/sdx`)，然后按你需要编辑分区表。

Arch Linux 安装程序使用`cfdisk`公用程序来对硬盘进行分区。

接下来，熟悉GNU/Linux的用户可以跳过这段内容直接到下面[选择软件包](#)。

分区信息

对磁盘进行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 (格式化) 一个文件系统。分区的类型分为主分区 (Primary)、扩展分区 (Extended) 和逻辑分区 (Logical) 三类。

主分区可以被启动，一个物理磁盘最多能有4个主分区。要使用超过4个的话，只能使用一个包含了逻辑分区的扩展分区。（译者注：为避免误解，较易懂的解释应该是：一个磁盘不算逻辑分区最多只能有4个分区，这4个要么全是主分区，要么就是3个主分区+1个扩展分区，当然你将整个磁盘划为1个主分区也是没问题的，只要不超过4个分区即可）

扩展分区通常不会被直接使用，它是作为逻辑分区的“容器”存在。一个硬盘只能有一个扩展分区，可以在里面继续再划分逻辑分区。

当对一个磁盘进行分区时，是按这种数字顺序命名：主分区为sda1-3，跟着是扩展分区sda4（译者注：在/dev和fdisk -l里是看不到这个sda4的），在扩展分区中划分出来的逻辑分区分别是sda5、sda6，如此类推。

交换分区swap

交换分区Swap是虚拟内存驻留在硬盘上地方，内核可以用它来储存在物理内存中放不下的数据。

过去，常用的规则是将交换分区大小设置为物理内存的两倍。但是随着时间推移，现在的计算机已经有更大的内存容量，这条规则已经不适用了。通常，512MB内存的计算机，两倍 (1G) 的还是够用的。而1G内存的计算机，一倍的也够用了。如果你有更大的内存（比如超过1GB），也许可以完全把交换分区忘了，不推荐这样做！

- 如果你打算使用休眠到磁盘的电源方案，你必须要有不小于内存容量大小的交换分区，有些用户甚至会推荐大于物理内存容量15%的大小。

规划分区

磁盘分区的规划是十分个人的设置。每位用户的决策都会根据自己计算机的情况和需求而不同。至少需要有一个包含了根文件系统 (http://en.wikipedia.org/wiki/File_system) (/) 的主分区和一个交换分区。你还可以选择给/boot（主要用来存放内核文件）、/usr、/var和/home（用来存放用户个人数据）划出独立分区。一个较好的做法是给 / 和 /home 都划分独立分区。这样的话，当你因为某些原因重装Arch Linux（或者其它发行版），你还可以保留你的个人数据例如音乐、图片、桌面环境配置等。

在本案例中，我们划分一个分区给 /，一个给/home，另外还有一个交换分区。

开始创建分区cfdisk

让我们开始创建一个包含root(/)文件系统的主分区。

选择New -> Primary，然后输入你希望的分区大小（对一个完整的Linux系统，4到8G是个不错的选择）。分区要位于磁盘的开头部分。选择刚新建的分区，接着选择Bootable将它置为可启动。

同时选择分区文件系统类型Type，指定为'83 Linux'。

给你的/home目录创建另一个分区。选择另外一个主分区设成你想要的容量大小，这取决于你的用户会在他们的home目录里储存多少文件，其大小可能从几百M办公文档到上百G音频视频，所以这里我无法给出一个建议。如果你要使用磁盘所有的剩余空间，记得保给交换分区留一部分（例如1GB或1024MB）。

同样的，选择Type为'83 Linux'。

最后，我们给交换分区创建一个分区。选择一个介于512MB到1GB之间的大小并将Type改为82 (Linux swap / Solaris)。

现在的分区规划看起来应该象这样（基于你的选择大小也许会有所不同）：

Name	Flags	Part	Type	FS Type	[Label]	Size (MB)
sda1	Boot		Primary	Linux		(4096 – 8192)
sda2			Primary	Linux		(> 100)
sda3			Primary	Linux swap / Solaris		(512 – 1024)

选择Write并输入'yes'。要注意如果你删除了分区的话，这个操作会破坏你磁盘上的数据。选择Quit退出分区程序。选择Done离开此菜单，接着选择"Set Filesystem Mountpoints"。

IDE设备，为什么看起来它们和SCSI设备一样？

最新开发的Linux内核里包含了libata和PATA模块，所有的IDE、SATA和SCSI都采用sdx这种命名方式。这十分正常，无需担心。

设置文件系统挂载点

"文件系统格式 (filesystems) " 和 "文件系统结构 (file systems) " 的区别:

从技术层面确切的说，**文件系统格式(filesystems)**是一种数据格式，涉及信息吞吐，比如ext2；而"文件系统结构(file systems)"则是特定操作系统中文件和目录的布局，比如/home/username/games。因此，当你被问及是否创建**文件系统格式(filesystems)**时，是在问你是否要**格式化**该分区...但是当程序向你问及挂载点时，你应该给出的是该分区应该出现在你的Arch Linux "文件系统结构"中的位置。

文件系统类型：

强烈建议：创建一个交换分区 (SWAP)。选择一个合适的分区(本例中使用的是sda3). 安装程序就会询问你是否要创建swap 文件系统; 选择 yes. 接下来, 选择根分区的挂载点(本例中就是 sda1). 然后安装程序就会询问你需要哪一种文件系统。

Note: 采用何种文件系统的类型取决于实际应用需要，没有十全十美的文件系统，选择任何一个文件系统，都需要在数据安全和性能要求方面作出适当倾斜，下面只是简单的介绍各种Arch linux支持的文件系统

1. **ext2** - 古老、可靠的 GNU/Linux 文件系统。非常的稳定，不过 不支持日志记录功能。磁盘检测速度较慢，因而有些不便. ext2可以容易地转换成ext3。一般认为/boot分区采用此系统最合适，因为/boot里的东西就开机的时候读取一次，而且很少有变动，不需要日志文件系统，如果采用日志文件系统，日志的体积很可能远远大于内核的大小。

2. **ext3** - 基于ext2系统，并添加了日志记录功能。一般认为他比其他文件系统要 稍微的慢一点, 不过 非常的稳定并且使用最为广泛，由 GNU/Linux FS 负责维护和开发。

高性能文件系统:

3. ext4 - 最近加入支持的文件系统，是ext2和ext3的后续发展版本，Ext4增加了48位块地址，最大支持1EB文件系统，和单个16TB的文件，增加子目录限制到64,000。新增了很多功能，提供在线碎片处理，速度比ext3快，系统开销比ext3小。

Note: 由于此系统较新，目前仍存在一些问题，暂时建议/home分区不采用此文件系统，以保证个人数据安全。

4. ReiserFS - 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).

5. JFS - IBM提供的日志文件系统 (Journaled FileSystem首字母缩写) . 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 conjunction with the deadline scheduler. (参见 JFS.) 但是并不像ext或者ReiserFS那样应用广泛。

6. XFS - 非常高效的日志文件系统，特别适合处理大文件。在处理小文件的时候稍慢。在挂载的时候也很快。由于 /var 需要处理很多的小文件，所以可能不太适合作为/var 分区的文件系统。GNU/Linux 文件系统中，最先实现可用的在线碎片处理。应用不及ext和ReiserFS那么广泛。

- JFS 和 XFS 不能被磁盘工具压缩 (比如gparted 或者 parted magic)

未来文件系统

7、btrfs - 此文件系统尚未被Arch纳入正式支持。是Oracle提供的copy-on-write档案系统。目标在取代的ext3，提高ext3单一档案的大小，总档案系统大小或档案检查，加入目前ext3未支援的功能，比如：writable snapshots，snapshots of snapshots，内建磁盘阵列 (RAID) 支援，and subvolumes。

最主要的差别在于日志系统 (http://en.wikipedia.org/wiki/Journaling_file_system) (一种类似的在数据库环境中记录着日志变更的东西). 除ext2之外，其他文件系统都支持日志记录功能. ext3 完全兼容ext2,因此哪怕你可以用一个非常古老的CD来挂载他，也没问题。根分区目前最安全的选择就是使用ext3文件系统。ReiserFS,XFS,和JFS也可以使用，因为GRUB (启动引导管理器，稍候在安装部分将会介绍到) 都可以从这些文件系统中启动。你可以通过选择“yes”来创建文件系统 (并格式化分区)，接着你就会被提示添加附加分区。在我们的例子中, 只有sda2了。选择一个文件系统类型，并且把这个分区挂在到/home. 再次说明，创建文件系统并且选择Done。然后回到主菜单。

4、选择软件包

现在我们要选择安装哪些软件包到系统中。选择CD作为来源，如果你不止一个光驱的话还需要选择使用哪个的光驱。

软件包选择分为两个阶段。首先要选择软件包分类，然后出现所选分类中软件包的详细列表供你微调。使用空格键进行选择和取消。

***BASE**:包含最小的软件包集合，仅够构建一个可运行的系统骨架。 ***SUPPORT**:供网络和文件系统用的额外软件包，例如gpm、fuse、dnsutils、无线驱动、ndiswrapper、ntfs-3g诸如此类。 ***DEVEL**: 软件创建工具，如GCC、autoconf、automake和make等等。 ***LIB**:大量的库文件，如gmp、lebelf、libevent等等。

- 2008.06 版的软件包选择，只有一个: BASE ，选择之后进入其详细列表.

在这里，我们对>Select all packages by default'选择'yes'。

接下来屏幕将显示你所选的软件包。对于新手，较安全的做法是不修改它，而有经验的用户可能希望裁减掉一些不需要的软件包。（例如，不需要的文件系统类型公用程序，不需要的驱动等等。）

5. 安装软件包

接着，选择'Install Packages'。你会被询问是否希望保留pacman缓冲里的软件包。如果你选'yes'，日后你还可以选择对某些软件包进行降级，因此推荐保留（日后你还是可以清除这些缓存的）。安装脚本跟着就会安装所选软件包到你的系统中。这些软件包是二进制的、针对i686优化的，总共仅100MB，因此安装是相当便捷的。

6. 配置系统

系统接着会询问你是否要选择hwdetect来收集配置所需信息。新手应该选择是（'yes'）。

如果是相当熟悉自己的硬件、所需模块，有能力从零开始手动配置/etc/mkinitcpio和/etc/fstab的高级用户，可以选择否'no'。（当然，此选项是非常复杂的，已经超过本指南的范围，这里就不详细解释了。）

接下来，系统会询问你是否需要支持从USB设备、FireWire设备、PCMCIA设备、NFS共享、软RAID阵列、LVM2卷和加密卷上启动。如果需要则选'yes'，在本案例中以上都不选。

新手还应该选择让/etc/mkinitcpio/kernel26-fallback.conf使用hwdetect的结果，这将确保你的硬盘控制器相容次序。

现在系统会询问你使用哪个文本编辑器；如果你不熟悉vi/vim (http://en.wikipedia.org/wiki/Vim_%28text_editor%29) 的话可以选择nano (http://en.wikipedia.org/wiki/Nano_%28text_editor%29)。然后就会出现一个包含系统中大部分重要配置文件的菜单。如果你打算查找/etc/rc.conf中可用的选项，只需按Alt+F2切换到另外一个虚拟控制台进行查找，然后用Alt+F1切换回安装程序。

系统配置主文件/etc/rc.conf

Arch Linux继承FreeBSD的传统，使用`/etc/rc.conf`作为系统配置的主要文件。这个文件包含了广泛的配置信息，主要用于系统的启动。正如名字中所包含的意思，它还包含了供`/etc/rc*`文件使用的配置。`/etc/rc.conf`提供了一个Arch用户可以轻易访问到的，简洁而又范围广泛的精简系统资源配置方案。

■ LOCALIZATION部分

- **LOCALE=**: 系统场所 (locale) 设置，被所有国际化 (i18n-aware) 的应用程序使用。从命令行上运行`locale`可以列出可用的`locale`设置。此设置缺省是适合英文 (US English) 用户使用的`en_US.utf8`。（此处`locale`的设置必须与`/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。
- **HARDWARECLOCK=**: 指示开关机时进行同步的硬件时钟，是**UTC**时间（协调世界时 (Universal Time Coordinated)，过去曾用格林威治平均时(GMT)来表示）还是**localtime**（本地时间）。通常UTC较好，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windows进行双启动，就必须使用`localtime`（本地时间），它只把本地时间保存到硬件时钟里。
- **TIMEZONE=**: 时区 (TIMEZONE)。（所有可用的时区可在`/usr/share/zoneinfo/`里找到。）

- **KEYMAP=**: 可用的键盘映射表可在/usr/share/kbd/keymaps里找到。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。
- **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里找到可用的映射表。
- **USECOLOR=**: 如果你使用彩色显示器的话请选"yes"。

```
LOCALE="zh_CN.UTF-8"
HARDWARECLOCK="localtime"
TIMEZONE="Asia/Shanghai"
KEYMAP="us"
CONSOLEFONT=
CONSOLEMAP=
USECOLOR="yes"
```

■ HARDWARE部分

- **MOD_AUTOLOAD=**: 如果你使用udev在启动时探测硬件并加载相应模块。请设置为"yes"，（适合缺省模块化的内核）。设置为"no"就得靠用户自己指定所需的有关信息。
- **MOD_BLACKLIST=**: 模块黑名单。在下面的例子中，IPv6模块和pcspeaker都被列入黑名单而不被加载。
- **MODULES=**: 指示了其它模块，如果你知道缺少哪个重要的模块可以加在里面。（hwdetect应该已经填写了最重要的模块）：

```
# Scan hardware and load required modules at bootup
MOD_AUTOLOAD="yes"
# Module Blacklist - modules in this list will never be loaded by udev
MOD_BLACKLIST=(net-pf-10 pcspkr)
#
MODULES=(e100 eepro100 miimii slhc snd-ac97-codec snd-intel8x0 soundcore)
```

■ NETWORKING部分

- **HOSTNAME=**: 设置你喜欢的主机名 (HOSTNAME)。
- **lo=**: 不修改此行。
- **eth0=**: 如果使用静态IP的话请调整网络接口的IP地址、掩码和广播地址。如果使用DHCP的话则设为eth0="dhcp"。
- **INTERFACES=**: 指定使用哪些网络接口。如果你不使用DHCP来配置设备，而是通过命令行用ifconfig手动配置设备的话，这里的变量值（名字必须等于打算进行配置的设备名一样）必须和ifconfig里加入的内容一样。
- **gateway=**: 如果你使用静态IP，将网关 (gateway) 地址设为路由器的IP。如果使用DHCP，则将此行用'#'注释掉。
- **ROUTES=**: 如果你使用静态IP，则将'gateway'前面的'!'去掉。如果使用DHCP的话则不需要修改此行。

例如，当使用DHCP：

```

HOSTNAME="arch"
lo="lo 127.0.0.1"
#eth0="eth0 192.168.0.2 netmask 255.255.255.0 broadcast 192.168.0.255"
eth0="dhcp"
INTERFACES=(lo eth0)
#gateway="default gw 192.168.0.1"
ROUTES=(!gateway)

```

■ DAEMONS部分

这个序列罗列了位于/etc/rc.d/里需要在启动过程中启动的脚本名字。

- 如果脚本名字前有惊叹号(!)，表示它不被执行。
- 如果脚本名字前有符号(@)，则表示它会在后台被执行，启动序列不会等待此脚本成功执行完毕，而是继续执行下一个守护进程脚本。
- daemon列表是有顺序的，系统根据先后执行相应的处理，所以不能随便放置，比如 networkmanager就不能放在hal的前面。

Note: 这种'BSD风格'的初始化就是Arch的处理办法，而其它发行版则是在一个init.d目录里使用一大堆符号链接（一个简单的文件 vs 一整个目录）

关于守护进程

虽然此时你不需修改daemons行，不过还是有必要解释一下什么是守护进程，因为过一会儿在指南里我们会需要它。

和Windows服务（service）类似，守护进程（daemon）是一个运行于后台的程序，等待事件发生并提供服务。一个较好的例子是webserver等待请求以发送页面，还有SSH服务等待用户登录。这些都是全功能的程序，还有些守护进程工作起来并不是那么明显的。例如向日志文件里写信息的守护进程（如syslog、metalog），当系统空闲时降低CPU频率的守护进程（如cpufreq），以及提供图形化登录的守护进程（如gdm、kdm）。所有这些程序都可以被加入到daemons行中，在系统启动时被执行。本指南会对一些较有用的守护进程进行介绍。按Ctrl+X退出nano编辑器。

磁盘和分区配置文件：/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，以决定某一特定设备被挂载时所采用的选项。

这是一个/etc/fstab的例子：

```

# /etc/fstab: static file system information
#
# <file system>      <dir>        <type>        <options>          <du
none           /dev/pts      devpts       defaults
none           /dev/shm      tmpfs        defaults
#/dev/cdrom    /mnt/cdrom   iso9660     ro,user,noauto,unhide
#/dev/dvd      /mnt/dvd     udf         ro,user,noauto,unhide
#/dev/fd0      /mnt/fd0     vfat        user,noauto
/dev/sda1      /            jfs         defaults,noatime,nodiratime
/dev/sda2      /home        jfs         defaults,noatime,nodiratime
/dev/sda3      swap         swap        defaults

```

Hwdetect应该已经有效地配置了一个可用的**fstab**，但最好还是简要地检查一下。

- 如果你打算用**hal**来自动挂载DVD之类的媒体，为了给即将安装的**hal**做准备，你可能要注释掉**cdrom**和**dvd**条目。

在笔记本电脑和桌面系统上，可以给 **/** 和 **/home**加上'**noatime**'和'**nodiratime**'选项来提高速度、性能，同时还能省电。无论你采用何种文件系统，这样做都应该是安全的。

更多信息请参照**Fstab**

mkinitcpio.conf、**modprobe.conf**和**kernel26-fallback.conf**

此时我们无须去修改这些配置文件。**mkinitcpio.conf**负责配置ramdisk（例如从RAID、加密卷启动），**modprobe.conf**可以用来设置内核模块的某些配置选项，而**kernel26-fallback.conf**则是initramfs后备配置。

(对于静态IP) 网络配置文件 : **/etc/resolv.conf**

- 如果你使用**DHCP**，完全可以忽略此文件，缺省情况下**dhcpcd**守护进程会自动创建和销毁此文件。

如果你使用静态IP，则可在**/etc/resolv.conf**里设置你的DNS服务器。每行的格式为**nameserver <ip地址>**，如果有多个DNS服务器则可以填入多行。

如果你使用路由器，也许会希望由你的路由器本身指定DNS服务器，而在**resolv.conf**里只是写入你的路由器的IP（同时也是**/etc/rc.conf**里的网关），例如：

```
nameserver 192.168.1.1
```

可选的，逐个加入你的DNS服务器，例如：

```
nameserver 4.2.2.1
nameserver 4.2.2.2
```

如果使用**DHCP**，你也可以在你的路由器里指定你的DNS服务器，或者如果你的ISP支持的话由ISP自动指定。

/etc/hosts

可以在此文件中加入你想要的*hostname*，必须和/etc/rc.conf里的设置一致，因此它应该象这样：

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

为了程序的兼容性，必须严格遵照这个格式，**包括'localhost'这项！**错误的句法可能会导致网络性能低下和/或某些程序运行缓慢甚至无法运行。这对新手来说是十分普遍的错误。

对于大部分用户，只要把*yourhostname*加入到行末就可以了，不过有些用户推荐以下句法：

```
127.0.0.1 yourhostname.domain.org localhost.localdomain localhost yourh
```

如果你使用静态IP，可以用这种句法加入新行：<static-ip> hostname.domainname.org hostname，例如：

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

/etc/hosts.deny和/etc/hosts.allow

如果你打算使用ssh的话，请根据自己的需要修改此配置。缺省设置是拒绝所有进入的连接，而不只是ssh连接。编辑/etc/hosts.allow 文件加入：

- 允许所有进入的ssh连接。

```
sshd:all
```

- 允许特定IP地址进入ssh连接。

```
sshd: 192.168.0.1
```

- 允许你指定的网段 (地址从 192.168.0.0 到 192.168.0.255)

```
sshd: 192.168.0.
```

- 允许某个IP地址范围

```
sshd: 10.0.0.0/255.255.255.0
```

如果你不打算使用ssh，为安全起见请保持此文件的缺省设置（为空）。

/etc/locale.gen

locale-gen 命令从 **/etc/locale.gen** 读取信息设定系统的locales。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殊标准。设置默认本地化的功能是类UNIX操作系统的一大特质

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get touched again。每次**glibc**更新之后**locale-gen**就会运行一次，以在**/etc/locale.gen**重新生成所有的本地化设置。

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
```

(你这里的设置必须和上文中提到的**/etc/rc.conf** 的设置一致哦。)

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**变更你的**locale**，然后使用根用户 (root) 运行‘**locale-gen**’脚本使之生效。

Note 如果你选择并设置了**locale**，他将提示你 “The current locale is invalid...” 错误。可能是因为大多数新Arch用户都会犯的普遍的错误，也是在论坛中问得最多的问题。

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。

Pacman镜像

为**pacman**选择一个软件仓库镜像站。记住主站 archlinux.org 是有流量限制的，下载速度限于 50KB/s.。

在列表中仔细查看，一般 xxx.cn 是中国镜像，xxx.au 是澳大利亚，等等...

- 如果你不熟悉现有的镜像和它们所处的位置，那就随便选一个。稍后你可以编辑 “**/etc/pacman.d/mirrorlist**”，或运行本指南后面提到的**rankmirrors**脚本，它将自动配置最接近你的镜像。

返回主菜单。

7、安装启动加载器

因为本案例中没有第二个操作系统，因此我们需要安装一个启动加载器。推荐选择GNU GRUB (<http://www.gnu.org/software/grub/>) 作为启动加载器。另外一种选择是LILO (<http://lilo.go.dyndns.org/>)。缺省的**GRUB**配置 (`/boot/grub/menu.lst`) 应该足够了。唯一你可能想修改的是控制台的分辨率。将`vga=<number>`加入到第一个kernel行。（menu.lst中已列出一张分辨率和数字的对应表格可供查找。）

```
title Arch Linux (Main)
root (hd0,0)
kernel /boot/vmlinuz26 root=/dev/sda1 ro vga=773
initrd /boot/kernel26.img
```

参数“vga=773”表示1024x768分辨率、256色的framebuffer。

将**GRUB**启动加载器安装到主引导区（本例中为sda）。

8、系统重启

退出安装程序。

- 建议：为了方便接下来更进一步的安装配置，可以把光盘里的新手指南复制一份到电脑里，这样后面的安装就能查看相应的向导了

```
cp /arch/beginnersguide.txt /mnt/home/beginnersguide.txt
```

系统重启之后，进入控制台，可以使用下面命令查看这新手指南

```
less /home/beginnersguide.txt
```

所有安装完毕之后，可以删了这指南

```
rm /home/beginnersguide.txt
```

输入命令下面命令，重启：

```
reboot
```

一个新Arch Linux系统就会启动并进入登录提示符（你可能需要在**BIOS**将启动顺序改为优先从硬盘启动）。

祝贺你，一个崭新的Arch Linux基础系统闪亮登场了！

第二部分：完成基本系统并升级

你的新Arch Linux基础系统现在已是一个功能完整的GNU/Linux环境了，随时可以进行定制。从现在开始，你可以按自己的意图和需求改造它。由于arch是滚动发布的，所以安装包不一定是最新的系统，所以我们在安装完成之后，需要立即更新系统。

让我们开始吧。

登录入你的root账号。我们将以root身份配置pacman和更新系统，然后创建一个普通用户账号。

一、配置网络 (如果有需要)

- 如果安装脚本的自动配置不适合你，本段内容将帮助你配置大部分类型的网络。

如果一切正常的话，你的网络应该可以正常使用了。先ping www.google.com验证一下：

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

如果你已经成功建立网络连接，则跳到下面阅读使用pacman更新、同步、升级系统。

如果ping www.google.com后得到"unknown host"的错误，可以认为你的网络尚未正确配置。你需要重新检查以下内容的完整性和正确性：`/etc/rc.conf` # 特别留意HOSTNAME= 和 NETWORKING部分有无输入错误。

`/etc/hosts` # 复核一下格式。（看之前的内容。）

`/etc/resolv.conf` # 当你使用静态IP时需要检查。如果你使用DHCP，缺省这个文件会被自动创建和销毁，不过也可以改为你需要的内容。（参见Network。）

高级的网络配置方法可以在Network一文中找到。

有线网络

用以下命令检查网络

```
ifconfig
```

你应该看到eth0这一项。如果有需要，你可以这样设置一个新的静态IP：

```
ifconfig eth0 <ip address> netmask <netmask> up
```

并且这样设置缺省网关：

```
route add default gw <ip address of the gateway>
```

查看一下`/etc/resolv.conf`是否包含你的DNS服务器，不过缺失了则补填进去。再次用ping `www.google.cn`检查一下网络。如果一切正常，将`/etc/rc.conf`调整为刚才新设置的静态IP。如果你的网络使用DHCP服务器/路由器的话，输入：

```
dhcpcd eth0
```

如果这样可行的话，则将/etc/rc.conf按之前所述方法调整为动态IP。

无线网络

详见安装指南：无线设置

模拟信号调制解调器

如果你想使用外接电话线调制解调器，你至少要安装ppp包。根据需要修改配置文件 /etc/ppp/options 你可以参考 man pppd 的帮助。

你还需要定制一个 chat 协议脚本，在初始化握手连接建立以后，提供你的ISP用户名和密码。命令 pppd 和 chat 的帮助里有例子不管你是菜鸟还是高手都应该能搞定连接了。使用 udev 的话，串口设备一般是 /dev/tts/0 和 /dev/tts/1。

Tip: Read Dialup without a dialer HOWTO.

如果你不想为了 pppd 拔光自己的头发，你可能会选择安装 wvdial 或类似的配置工具来简化这个过程。如果你的猫是所谓的 WinModem，也就是内置 PCI modem 卡，你可以在这个网址找到大量信息：LinModem (<http://www.linmodems.org/>)

ISDN

通过三步，完成 ISDN 的安装和配置：

1. 安装配置固件Install and configure hardware
2. 安装配置ISDN 管理工具
3. 添加ISP信息

现在的 Arch stock kernels 已经包含了ISDN 所需要的modules，所以不需要重新去编译内核，除非 you're about to use rather odd ISDN hardware。把ISDN卡插入电脑或者插入USB ISDN-Box之后，you can try loading the modules with modprobe。绝大部分的ISDN PCI 卡都能被module hisax 处理，只需要两个参数：类型和协议。如果使用的是1TR6标准，则协议设定为 '1'，如果使用的是 EuroISDN (EDSS1)，协议设定为 '2'， '3' if you're hooked to a so-called leased-line without D-channel，如果使用的是US NI1，则设为'4'。

所有配置细节都可以在kernel documentation, more specifically in the isdn subdirectory, and available online中获知。类型的参数是根据电视卡来设定的，在README.HiSax kernel documentation中，有一个可能类型的列表。用相应的参数加载moduleChoose，比如这个：

```
modprobe hisax type=18 protocol=2
```

这个将加载适用于电视卡类型为ELSA Quickstep 1000PCI，德国使用的EDSS1协议的模块hisax。可以在日志文件/var/log/everything.log中查看是否模块被预加载。额外的USB ISDN Adapter需要加载些USB模块才能正常工作。

通过上述设定之后，如果电视卡能正常工作，那么把下面模块和参数加入/etc/modprobe.conf中：

```
alias ippp0 hisax
options hisax type=18 protocol=2
```

这么做的好处是，这些模块在不使用时不会被加载。另外的方法是可以在这只加参数，把hisax加入/etc/rc.conf里面的MODULES，这样做的坏处就是不管这些模块是不是在使用，都加载了。

完成上述步骤之后，需要安装管理工具

安装软件包isdn4k-utils, 阅读关于isdnctrl的说明手册。在说明手册里，将有详细的说明：如何创建一个配置文件that can be parsed by isdnctrl，安装实例。 Please note that you have to add your SPID to your MSN setting separated by a colon if you use US NI1.

使用isdnctrl utility配置ISDN卡之后，将可以拨号到你设定的PHONE_OUT参数的机器上，但提示用户名和密码错误。把用户名和密码加入到/etc/ppp/pap-secrets或者/etc/ppp/chap-secrets，这个主要根据的是你的ISP使用的协议授权方式，as if you were configuring a normal analogous PPP link。如果无法确定，可以两个文件都加。

完成上述设定之后，用root帐号建立一个拨号连接：

```
isdnctrl dial ippp0
```

有啥问题，请检查日志文件!

DSL (PPPoE)

仅当需要你的计算机去主动拨号连接ISP的情况下才需参考本段内容。如果你使用独立的路由器来负责这类工作，你只需要设置一个缺省网关即可。

在使用DSL连接前，你得先在计算机里安装一个连接到DSL-Modem的网卡。将你新装网卡加入到modules.conf/modprobe.conf或者MODULES列表之后，安装rp-pppoe软件包并运行pppoe-setup脚本来配置你的连接。当你输入完有关信息后，可以分别通过以下命令来进行连接和挂断：

```
/etc/rc.d/adsl start
```

和

```
/etc/rc.d/adsl stop
```

setup过程一般都是很容易、直观的，如果还有不懂可以阅读man手册。如果你想在启动时自动拨号，可以将adsl加入到DAEMONS列表中。

二、更新Pacman

pacman是什么？

Pacman是Arch Linux的软件包管理器 (**package manager**)。Pacman是用C语言编写的，速度快，使用简便，功能强大。它可以管理整个软件包系统，负责软件包的安装、删除、降级（通过缓存），也支持自编译的软件包，自动分析依赖关系，可以搜索远程和本地软件仓库，还有大量其它功能。这里我们将用pacman来从远程软件仓库下载软件包并安装到系统中。

对系统进行定制，pacman是Arch Linux最重要工具。

1. 配置pacman

pacman总配置文件/etc/pacman.conf

Arch通过pacman可以获得对下面 4软件仓库的管理:

[core]

[core] (“核心”) 仓库的简单原则是只提供一个基本的Arch Linux的基本工具：GNU系列工具，Linux内核，一个文字编辑器，一个命令行浏览器等（这里也有一些例外。比如说，vi和nano是同时提供的，允许用户选择一个或两个）。它包含所有使你的系统正常运转的“必须”安装的包，只有它们完美地工作，你的系统才能继续运行。这些是绝对“system-critical”（对系统来说至关重要）的包。由开发者负责维护。

- Core安装媒介 (*The Core installation media*) 简单地包含一个安装脚本，一个当前发布版本的核心仓库的快照 (*snapshot*)。

[extra]

[extra] (“额外的”) 仓库包含所有的对于一个基本的Arch系统不是必须，但是对于构建一个完整的系统环境有用的的Arch包。比如X, KDE和Apache，都可以在这里面找到。它也是由开发者维护的。

[testing]

[testing] (“测试”) 仓库包含的包是[core]和[extra]仓库的候选者。下面情况下，新的软件包将进入[testing]软件仓库：

- 可能会导致系统出问题的升级，需要测试。
- 需要重新打包其他软件包。这种情况下，所有需要重新打包的软件包将被放在[testing]仓库，直到所有重新打包完成，这些软件包才回归原有的软件仓库。

开发者维护。

[testing]是唯一的一个在名义上与其它官方仓库有冲突的仓库。（*that can have name collisions with any of the other official repositories.*）

*如果开启，[testing]必须放在pacman.conf的软件仓库的第一个。

警告: 开启[testing]软件仓库，容易导致系统崩溃，因此只建议有经验的用户开启。

[community]

[community] (“社区”) 仓库是由“信任用户” (*Trusted Users (TUs)*) 维护的，是Arch User Repository (AUR) (Arch用户仓库) 的一部分。它包含的是二进制的包，在用户仓库中有足够的“票数”，并由信任用户挑选出来的。像其它上面列出的仓库一样，[community] 可以用pacman获取。

AUR 也包含**unsupported** 分支，这不能直接通过pacman获取。*. [unsupported] 包含海量

PKGBUILD脚本，用来从源代码编译。这从其它仓库中是不能使用的。

* **AUR Helpers** 能帮助你无缝地使用AUR。

/etc/pacman.conf

每次调用pacman，它都会读取/etc/pacman.conf文件。这个配置文件分若干段，每一段定义了一个pacman用来搜索软件的仓库。唯一例外的是options段，它定义了全局选项。

```
nano /etc/pacman.conf
```

范例：

```

#
# /etc/pacman.conf
#
# See the pacman.conf(5) manpage for option and repository directives
#
# GENERAL OPTIONS
#
[options]
# The following paths are commented out with their default values listed.
# If you wish to use different paths, uncomment and update the paths.
#RootDir      =
#DBPath       = /var/lib/pacman/
#CacheDir    = /var/cache/pacman/pkg/
#LogFile     = /var/log/pacman.log
HoldPkg      = pacman glibc
# If upgrades are available for these packages they will be asked for first
SyncFirst    = pacman
#XferCommand = /usr/bin/wget --passive-ftp -c -O %o %u
#XferCommand = /usr/bin/curl %u > %o

# Pacman won't upgrade packages listed in IgnorePkg and members of IgnoreGr
#IgnorePkg    =
#IgnoreGroup   =

#NoUpgrade   =
#NoExtract   =

# Misc options (all disabled by default)
#NoPassiveFtp
#UseSyslog
#ShowSize
#UseDelta
#TotalDownload
#
# REPOSITORIES
#   - can be defined here or included from another file
#   - pacman will search repositories in the order defined here
#   - local/custom mirrors can be added here or in separate files
#   - repositories listed first will take precedence when packages
#     have identical names, regardless of version number
#   - URLs will have $repo replaced by the name of the current repo
#
# Repository entries are of the format:
#   [repo-name]
#   Server = ServerName
#   Include = IncludePath
#
# The header [repo-name] is crucial - it must be present and
# uncommented to enable the repo.
#
# Testing is disabled by default. To enable, uncomment the following
# two lines. You can add preferred servers immediately after the header,
# and they will be used before the default mirrors.
#[testing]
#TestingServer = /etc/pacman.d/mirrorlist

```

要开启一个软件仓库，只需要将'Include ='和'[repository]'行前的注释符号(#)去除。

- "Server ="行如果取消注释，将会强制此服务器被首先搜索，详细的仓库配置位于/etc/pacman.d/下。
- 注意：选择仓库时，请确认将仓库名和'Include ='行都一同取消注释。没有这样做的话会导致你选择的仓库被忽略！这是个很普遍的错误。

(如果有必要) 配置镜像列表 /etc/pacman.d/mirrorlist

更快的镜像能显著改善pacman的性能和你对Arch Linux的总体体验。

- 手动编辑/etc/pacman.d/mirrorlist：

```
nano /etc/pacman.d/mirrorlist
```

删除所有不在你所在洲或者太远的镜像。（在nano里你可以用CTRL-K删除每个不需要的行。）保存文件并退出。

- **rankmirrors**脚本可以根据镜像的响应延迟对其自动排名。**Rankmirrors**是python脚本，需要安装python：

```
pacman -S python
```

然后，运行**rankmirrors**脚本按响应延迟顺序对镜像进行排名：

```
rankmirrors -v /etc/pacman.d/mirrorlist
```

输出会列出所有镜像和它们的响应延迟（ping），例如：

```
# United States
# http://mirrors.easynews.com/linux/archlinux/$repo/os/i686 ... 0.96
# ftp://ftp.nethat.com/pub/linux/archlinux/$repo/os/i686 ... unreachable
# ftp://locke.suu.edu/linux/dist/archlinux/$repo/os/i686 ... 2.43
# ftp://mirrors_unixheads.org/archlinux/$repo/os/i686 ... 1.96
```

脚本将从低到高的响应延迟顺序显示：

```
Server = http://mirrors.easynews.com/linux/archlinux/$repo/os/i686
Server = ftp://mirrors_unixheads.org/archlinux/$repo/os/i686
Server = ftp://locke.suu.edu/linux/dist/archlinux/$repo/os/i686
Server = ftp://ftp.nethat.com/pub/linux/archlinux/$repo/os/i686
```

要注意的是这个结果对于镜像排列并非很精确，最接近你的镜像或者延迟最低的镜像并不一定是最佳选择。（反应迅速但是内容过时的镜像、或者低延迟但是窄带宽的镜像对你毫无用处）。

编辑/etc/pacman.d/mirrorlist，将最佳的镜像置于在列表顶端。为了体验多个镜像你可能要不断修改

此配置文件，请精明地选择。

常用的命令

与远程软件仓库同步和更新本地包数据库 (推荐在安装和更新包之前这样做):

```
pacman -Sy
```

升级系统中所有的包:

```
pacman -Su
```

一条同步、更新、**升级**所有系统中的包的命令:

```
pacman -Syu
```

安装或者升级单个软件包或者一串包(包括其依赖包):

```
pacman -S packageA packageB
```

移除软件包, 但保留依赖的包:

```
pacman -R package
```

移除单个包, 并且移除没有被别的软件依赖的依赖包 :

```
pacman -Rs package
```

移除包所有不需要的依赖包并删除其配置文件:

```
pacman -Rsn package
```

通过给定关键词 (列表) 搜索远程软件仓库数据库(repo):

```
pacman -Ss keyword
```

列出系统中所有的包

```
pacman -Q
```

在本地包数据库搜索 (查询)指定软件包 :

```
pacman -Q package
```

在本地包数据库搜索 (查询)指定软件包并列出相关信息 :

```
pacman -Qi package
```

To defragment pacman的数据库缓存和速度优化选项 :

```
pacman-optimize
```

统计当前系统中的包数量:

```
pacman -Q | wc -l
```

使用ABS和makepkg从源代码编译安装包 :

```
pacman -U packagename.pkg.tar.gz
```

Note: pacman有很多很多的选项和功能组合。你可以试试man pacman或者查阅It the pacman维基条目

软件包管理增强工具

- Yaourt 可查询aur的pacman增强工具
- Tupac 快速查询的yaourt增强工具
- Powerpill 多进程下载的pacman增强工具

Note: 强烈建议更新系统完毕后，安装yaourt和powerpill，以便更充分享受Arch Linux的方便和强大。

在/etc/pacman.conf文件里，最后添加两行(如果是i686的话，把x86_64换成i686即可):

```
[archlinuxfr]
Server = http://repo.archlinux.fr/x86_64
```

然后使用pacman安装yaourt

```
pacman -Sy yaourt
```

以后就可以使用yaourt命令来查找并安装包括AUR里的软件了。比如bcm无线网卡，需要安装b43-

firmware (此软件包在AUR仓库里，使用pacman无法查找到)，使用yaourt命令查找并安装：

```
yaourt b43-firmware
```

找到之后，根据相应提示，一直到安装完成。

图形化管理工具

- qt界面的shaman：比较成熟的pacman 图形化管理工具
- gtk界面的gtkpacman：目前BUG比较多，不稳定。

Pacman的更多信息

用此命令查看pacman手册：

```
man pacman
```

有空的话可以看看pacman (简体中文)维基条目。

2、更新Pacman自己

一开始，系统可能会提示你先更新pacman自身，这取决于Arch安装包的新旧程度：

```
pacman -Sy pacman
```

pacman开始查看是否需要更新，如果有提示需要更新，则按'y'，开始pacman更新。

有时候，还需要用户主动修改某些配置；请仔细阅读升级过程中的输出获取相关信息。

三、更新系统

使用pacman

现在我们要用pacman来更新系统。要更新、同步并**升级**你的系统，可以使用命令：

```
pacman -Syzu
```

pacman就会获取所有最新可用软件包信息并进行升级。（此时你可能会被提示先升级pacman自身。如果是这样，回答'yes'，然后在完毕后再运行一次pacman -Syzu命令。）

滚动发布之美

Arch是一个**rolling release**（滚动发布）的发行版。这意味着没有必要重新安装或执行完整的系统重建来升级到最新版本。只需要执行**pacman -Syzu**，就能保持你的系统更新到最新和bleeding edge（表示最新的、因而也并非完美的技术）。

四、增加一个用户并设置用户组

用root账号来处理日常工作是很危险的。Root是用来处理系统管理的任务。我们可以用以下命令增加一个普通用户账号：

```
adduser
```

大部分缺省选项都是没问题的，你可能还需要将storage、audio、video、optical和wheel加入到你的用户组中——特别是在你打算使用一个完整功能的桌面环境的情况下。

用户组和用户的关联定义在/etc/group文件中：

它包含了：

- **audio** - 关于声卡和相关软件的任务
- **wheel** - 给sudo使用
- **storage** - 负责存储设备
- **video** - 视频任务和3D加速
- **optical** - 和光驱有关的任务
- **floppy** - 软驱访问
- **lp** - 打印任务

请参看Groups一文了解你应该属于哪些用户组。你还可能需要将你的用户账号加入到以下用户组（需要以root身份运行此命令）：

```
usermod -aG audio,video,floppy,lp,optical,network,storage,wheel,dbus,hal US
```

详情可看usermod和gpasswd的man手册。

五、安装建立Sudo (可选)

安装前，请先移除会发生冲突的软件包：

```
# rm /usr/bin/{view,rview}
```

安装Sudo 和 vim (译者注：之所以要安装vim是为了配置sudo文件之用)：

```
# pacman -S sudo vim
```

以root身份，运行命令visudo，把用户加入sudo用户(a "sudoer")。如果不知道怎么使用vi，你可以在用visudo前设置VISUAL 或 EDITOR环境变量为你喜欢的编辑器。例如：

```
# export VISUAL=vim
```

或

```
# export EDITOR=nano
```

如果你对于用vi很适应，去掉EDITOR=nano这个变量就行：

```
# visudo
```

这命令将在一个特殊的vi会话中打开文件 /etc/sudoers。visudo复制文件到一个可编辑的临时文件中，用一个编辑器编辑它（vi是默认的），接下来运行一个安全检查（“sanity check”）。如果通过，这个临时文件就有正确的权限将重写的原始文件。t

注意:不要直接用编辑器编辑 /etc/sudoers；否则会出现这样的提示：Errors in syntax can cause annoyances (像root帐户不能使用的描述一样)。必须使用 visudo 命令来编辑 /etc/sudoers.

当用户输入以 "sudo"，作为开始的命令行时，临时授予该用户root权限。加入下面一行：

```
USER_NAME    ALL=(ALL)  ALL
```

USER_NAME 就是单个用户名。

更多信息，比如按<TAB> 自动命令补全，请见Sudo

第三部分：安装X并配置ALSA

一共有三步工作：

- 1、安装声卡驱动
- 2、安装x-window (包括显卡驱动)
- 3、配置X

Note: 在热插拔未配置正确的情况下，或者是/etc/rc.d/hal未启动的情况下，请不要启动开启X(包括重启使用gdm或者kdm登陆)，否则可能导致鼠标键盘失去响应！！！

一、配置声卡：使用alsamixer

高级Linux声音架构 (Advanced Linux Sound Architecture , 简称ALSA) 是一个Linux内核组件。它替换了原有的开放声音系统(OSS)，为声卡提供了设备驱动。除了声音设备驱动以外，ALSA还给应用程序开发人员提供了一个用户空间库，可使更高级的API无需直接与内核驱动打交道就能使用驱动功能。

Note: Alsa已经包含进最新的内核kernel，udev 将在系统启动时自动检测硬件，并自动根据声卡加载相应的内核模块。虽然声卡已经在工作，但电脑的声音默认是静音。

Note: OSS4.1已经放出，相对于老的版本，这是一个显著的提升的版本，如果使用ALSA有问题，或者是想尝试新的功能，可以使用OSS4.1。相关说明可见：OSS

Alsa-utils软件包里的alsamixer让我们可以从命令行来配置声音设备。（你也可以稍后在X环境中使用alsamixer。）默认情况下，内核加载模块snd_pcsp，这是一个alsa pc speaker模块。snd_pcsp通常先于“你实际的”声卡模块加载。大部分情况下，这模块最后加载更好，因为可以让alsamixer正确控制需要的声卡。

在/etc/modprobe.d/modprobe.conf中加入这行，可以使得snd_pcsp最后加载：

```
options snd_pcsp index=2
```

当然，如果你不想让snd_pcsp加载，可以把它放入黑名单，或者是在/etc/rc.conf的Modules里禁止它：

```
MODULES=(... !snd_pcsp)
```

Note: 想要生效，需要释放并重新加载所有声音模块，当然，系统重启就可以简单解决这问题。

- 安装alsa-utils软件包：

```
pacman -S alsa-utils
```

- 为了解决Alsa声卡独占问题，可以安装alsa-oss：

```
# pacman -S alsa-oss
```

- 把你的普通用户账号加入到audio用户组中：以root身份，运行/usr/bin/gpasswd程序：

```
gpasswd -a yourusername audio
```

- 以普通用户身份运行：

```
alsamixer
```

用左右光标键移动到Master和PCM声道上，按M取消掉静音。用向上键增加音量。（70-90的音量是较合适的范围）按ESC键离开alsamixer。有Speaker的，需要同时取消静音并调整之。

测试声音

以普通用户身份，使用 /usr/bin/aplay程序来测试你的声音配置：

```
aplay /usr/share/sounds/alsa/Front_Center.wav
```

你应该会听到一个女性声音说：“Front , center。”

保存声音配置

以root身份运行/usr/bin/alsactl :

```
alsactl store
```

这将创建'/etc/asound.state'文件，用来保存alsamixer的设置。

为了每次启动都可以自动恢复混音器设置，可将守护进程alsa加入到你的/etc/rc.conf中的DAEMONS列表。

```
nano /etc/rc.conf
DAEMONS=(syslog-ng network crond alsa)
```

注意，alsa守护进程在启动过程中仅仅是恢复音量设置。它和alsa音频库（和内核级API）是完全不同的。

更多详情请参看ALSA 安装设置。

二、安装X

X 窗口系统(一般叫**X11**或者更简单的叫**X**) 是一个网络互动和显示服务，提供位图显示器窗口. 提供标准化工具包 (toolkit) 和协议，用于**UNIX**类操作系统，构建一个图形界面(GUIs)

X 提供最基本、基础的框架，用于构建一个图形操作界面环境。包括：绘图、屏幕窗口移动，与鼠标或者键盘的交互。**X** 并不负责用户交互界面的部分：有独立的客户端程序负责处理这部分。比如 gnome、kde。

X 因为它先于**W**窗口系统，最初由斯坦福大学开发。

- 注意: 如果想使用一个**开源**显卡驱动，需要3d加速，强烈建议在安装Xorg之前安装库 libgl：

```
pacman -S libgl
```

(拥有专利的视频驱动提供它们自己的gl库安装 (*implementations*)).

1、安装X

构建一个GUI的第一步：用pacman来安装Xorg包.

```
# pacman -S xorg
```

对于新版的xorg，建议（可能大部分情况下是必须的）安装输入设备驱动evdev。它本来应该作为xorg-server的一个依赖被安装的，不过不知道为什么貌似缺失了：

```
# pacman -S xf86-input-evdev
```

3d的工具比如glxgears包含在包mesa里（如果需要，也安装）：

```
# pacman -S mesa
```

2、安装显卡驱动

到这一步，已经有了运行X Server的基本软件包，接下来需要把显卡的驱动安装到电脑里(e.g. xf86-video-<name>)。最简单的X.org配置方法就是安装正确的驱动包，然后使用自动脚本对 /etc/X11/xorg.conf 进行初始化，比如使用 Xorg -configure (arch团队提供hwd工具，但据以前用户的反映，这个现在还老出问题，新手可以使用，以后有能力再配置其他)。

如果不知道显卡类型，输入下面命令：

```
lspci | grep VGA
```

输入下面命令，查看所有开源驱动：

```
pacman -Ss xf86-video | less
```

下面是一个开源的显卡列表和相关说明：

- **xf86-video-apm** Alliance ProMotion video driver
- **xf86-video-ark** ark video driver
- **xf86-video-ati** ati video driver
- **xf86-video-chips** Chips and Technologies video driver
- **xf86-video-cirrus** Cirrus Logic video driver
- **xf86-video-dummy** dummy video driver
- **xf86-video-fbdev** framebuffer video driver
- **xf86-video-glint** GLINT/Permedia video driver
- **xf86-video-i128** Number 0 i128 video driver
- **xf86-video-i740** Intel i740 video driver
- **xf86-video-i810** Intel i810/i830/i9xx video drivers (deprecated - use -intel)
- **xf86-video-intel** Newer Version of Intel i810/i830/i9xx video drivers
- **xf86-video-imstt** Integrated Micro Solutions Twin Turbo video driver
- **xf86-video-mach64** mach64 video driver
- **xf86-video-mga** mga video driver (Matrox Graphics Adapter)
- **xf86-video-neomagic** neomagic video driver
- **xf86-video-nouveau** Open Source 3D acceleration driver for nVidia cards (experimental)
- **xf86-video-nv** nvidia nv video driver
- **xf86-video-openchrome** Openchrome drivers
- **xf86-video-r128** ati Rage128 video driver

- **xf86-video-radeonhd** Experimental Radeon HD video driver for r500 and r600 ATI cards
- **xf86-video-rendition** Rendition video driver
- **xf86-video-s3** S3 video driver
- **xf86-video-s3virge** S3 Virge video driver
- **xf86-video-savage** savage video driver
- **xf86-video-siliconmotion** siliconmotion video driver
- **xf86-video-sis** SiS video driver
- **xf86-video-sisusb** SiS USB video driver
- **xf86-video-tdfx** tdfx video driver
- **xf86-video-trident** Trident video driver
- **xf86-video-tseng** tseng video driver
- **xf86-video-unichrome** Unichrome video drivers
- **xf86-video-v4l** v4l video driver
- **xf86-video-vesa** vesa video driver
- **xf86-video-vga** VGA 16 color video driver
- **xf86-video-via** via video driver
- **xf86-video-vmware** vmware video driver
- **xf86-video-voodoo** voodoo video driver

- 说明：**vesa**驱动是最通用的软图形驱动，适用于大部分的显卡，但没有3D加速等功能。所以如果实在无法找到合适的显卡驱动，就只好选用vesa，理论上它应该能在任何显卡上工作。（这是原话，好像夸张了点！If you cannot find a suitable driver for your video chipset, vesa **should work.**）

用pacman来安装适合你显卡的驱动。例如：

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

(给Savage显卡的驱动。)

- 如果实在不确定需要安装那个显卡驱动，可以安装所有显卡驱动包，输入下面命令：

```
pacman -S xorg-video-drivers
```

- 如果显卡是nVIDIA 或者ATI的，并想安装nVIDIA或者ATI官方提供的显卡驱动，请看下面。

nVIDIA图形显示卡

nVIDIA显示卡的驱动程序被许多人认为是非常好的质量，而且带有高级3D性能。

在设置您的nVIDIA显卡之前您先要选择您要使用的驱动程序。Arch现在有3种不一样的nVIDIA驱动程序，各自支持不同型号的显卡：

1. **nvidia-96xx** 直到Ge Force 4的较新显卡
2. **nvidia-173xx** Geforce FX系列显卡
3. **nvidia** 在GF FX之后的最新GPU

Note: 一些特别老如TNT和TNT2之类的显卡所需的Nvidia-71xx系列闭源驱动，由于无法在Arch所用的新版Xorg下正常运行，并且nvidia也已不再继续支持它们，因此已被删除。你可以改用xf86-video-nv或者xf86-video-vesa驱动。

到nVIDIA主页，看哪个适合你电脑.区别只是安装，配置过程是一样的.

根据你的显卡，选择合适的nVIDIA驱动安装，比如：

```
pacman -S nvidia-96xx
```

nVIDIA软件包有个工具能自动升级你已有的nVIDIA驱动的配置文件/etc/X11/xorg.conf，输入下列命令即可实现：

```
nvidia-xconfig
```

它也有一些选项，用于更深入和具体地配置xorg.conf文件的内容和选项，例如：

```
nvidia-xconfig --composite --add-argb-glx-visuals
```

更详细的信息，请参考：nvidia-xconfig(1).

一些有用的妙法可用在“device”部分（你要清楚它们不一定都能对你的系统有用）：

```
Option      "RenderAccel" "true"
Option      "NoLogo"    "true"
Option      "AGPFastWrite" "true"
Option      "EnablePageFlip" "true"
```

将所有的DRI注释掉：

```
# Load      "dri"
```

多检查几次你的/etc/X11/xorg.conf以确定你的默认的depth（色深），horizontal sync(水平同步)，垂直刷新率，以及分辨率是可以接受的。

使用/sbin/depmod命令来更新内核模块依赖：

```
depmod -a
```

（也许需要重启。）

Tip: nvidia配置的高级的说明参考文章NVIDIA。

现在你可以跳到配置X来进一步熟悉，或者跳到测试 X继续安装过程。

ATI图形显示卡

ATI用户的驱动可以有几种选择。

- **xf86-video-ati**软件包提供的开源*radeon*驱动。
 - 原创的、完全支持Radeon芯片组到X1950（最新的R500芯片组）的逆向工程开源驱动。对9200系列以前的显卡完全支持，可以稳定的提供全2D和3D加速。从9500到X1950完全支持2D加速特性，良好的3D加速，不过缺乏某些闭源驱动的特性（例如能源节省特性仍处于测试阶段）。*xf86-video-ati*支持从HD2xxx (R6xx)到最新的显卡，不过目前仅提供2D支持。
- **xf86-video-radeonhd**软件包提供了开源的*radeonhd*驱动。
 - 此驱动支持ATI R500（Radeon X1000系列）以及更新的芯片组。该驱动由AMD提供文档，Novell为之开发。它支持RandR 1.2，目前的开发工作非常活跃。因此所支持的不同系列显卡的功能可以不一致。（有些用户反映它性能优异，而有些则说它很糟）。它还支持带音频的HDMI。
- AUR中的Catalyst软件包提供了闭源*fglrx*驱动。稍后将介绍闭源驱动。

开源驱动通常可以满足大部分需求而且较少出问题。

安装*radeon*ATI驱动：

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

安装*radeonhd*ATI驱动：

```
# pacman -S xf86-video-radeonhd libgl ati-dri
```

Arch的extra官方仓库曾经提供过闭源ATI驱动**Catalyst**的预编译包，不过在2009年3月以后，由于对闭源驱动的质量和开发进度不满意，官方取消了对它的支持。目前catalyst驱动可以在AUR (<http://aur.archlinux.org/packages.php?ID=22899>) 中找到。Catalyst驱动的安装信息可以在这里找到。

Warning: 闭源ATI驱动仅支持R600和更新的显卡（即HD2xxx及以上系列），不支持旧系列（X1xxx系列及以下）。

Tip: ATI配置的高级指引可以参看 ATI wiki 文章。

3、安装输入设备驱动软件包

最新的X需要你为包括键盘和鼠标在内的输入设备安装驱动。查看可用输入设备驱动的完整列表：

```
# pacman -Ss xf86-input | less
```

对大部分用户，*xf86-input-keyboard*和*xf86-input-mouse*应该可以满足基本配置。使用pacman来为你的输入设备安装需要的驱动，例如：

```
# pacman -S xf86-input-keyboard
```

三、配置X：创建并修改/etc/X11/xorg.conf (如有需要)

Note: 从X11R7.2版开始Xserver引入了自动配置特性。在大多情况下，即使没有xorg.conf文件它也能很好地运作。如果自动检测功能工作正常，而且你又不需要开启如 aiglx、compositing 等功能的话，你甚至不需要创建这个配置文件。

配置文件xorg.conf介绍

/etc/X11/xorg.conf 是你的 X-Window-System 的主要配置文件，你的 GUI (**Graphical User Interface**) 的基础，它是一个普通的文本文件，里面由“section”和“subsection”来安排。重要的 section 有 *Files*, *InputDevice*, *Module*, *Monitor*, *Modes*, *Screen*, *Device* 和 *ServerLayout*。各个 section 安排的前后次序没有关系，对同一种类型的设备也可以有多个section，比如你可能有多个显示器，或者你的本本既有鼠标又有触摸板。

1、创建 /etc/X11/xorg.conf

高级用户可以手动创建他们的xorg.conf文件。一般用户可以使用程序/usr/bin/Xorg附加参数 -configure 来生成一个基本的配置文件；以root身份输入下面命令：

```
# Xorg -configure
```

这里生成的文件是/root/xorg.conf.new

把这文件复制到 /etc/X11/：

```
# cp /root/xorg.conf.new /etc/X11/xorg.conf
```

更多的创建/etc/X11/xorg.conf的办法

- 说明：如果你的电脑已经有一个配置好的xorg.conf文件，不管这个文件是在什么别的发行版下生成的，只要Xorg的版本一致，就可以把该文件复制到你的Arch系统的/etc/X11/目录下。
- 可以使用程序/usr/bin/hwd生成xorg.conf。首先要安装这程序：

```
# pacman -S hwd
```

然后输入下面命令：

```
# hwd -xa
```

Note: 目前，hwd在生成的xorg.conf有一行失效。 使用完这程序后，需要手动删除或者是注释掉 Section "Files"下的RgbPath那一行。 via nano /etc/X11/xorg.conf or your equivalent, as

root/sudo.

- 使用/usr/bin/xorgcfg 脚本生成xorg.conf:

```
# xorgcfg
```

- 使用/usr/bin/xorgconfig生成xorg.conf，这个程序以互动方式来创建该文件，输入下面命令运行：

```
# xorgconfig
```

2、Input设备热插拔问题：请在测试X前阅读

xorg 1.5.x版本里已经开启热插拔Input设备。

Note: 当热插拔开启后，X将清空xorg.conf里所有使用the kbd and mouse 驱动建立的设备。这将导致X被冻结，使得鼠标和键盘失去响应。下面是解决办法。**方法只能二选一**

有两个办法可以修复这一问题：

- 1) 配置 input设备热插拔：安装驱动 xf86-input-evdev，同时配置HAL使其能使用kbd/mouse驱动。

安装xf86-input-evdev:

```
# pacman -S xf86-input-evdev
```

在X.org相关进程启动前，启动守护进程hal：

```
# /etc/rc.d/hal start
```

把hal加入/etc/rc.conf里的DAEMONS列表里，这样每次电脑启动，都会自动开启hal。详情请见Xorg input hotplugging。

说明: 开启hal必须首先有一个dbus用户和用户组，以root身份输入下面命令创建用户和用户组

```
usr/sbin/groupadd -g 81 dbus
usr/sbin/useradd -c 'System message bus' -u 81 -g dbus -d '/' -s /bin/false
```

- 2) 关闭input设备热插拔：在/etc/X11/xorg.conf的ServerFlags段，加入参数Option "AutoAddDevices" "False" 这将忽略hal的硬件检测，并使用你xorg.conf的keyboard/mouse配置。步骤如下

```
# nano /etc/X11/xorg.conf
```

加入下面三行：

```
Section "ServerFlags"
    Option "AutoAddDevices" "False"
EndSection
```

3、测试 X

在所生成的xorg文件路径`path/to/xorg.conf`前，加入参数`-config`，运行 X脚本，可以测试脚本配置是否正确。对于 Xorg 7.3 或者之前的版本，键入：

1. `Xorg -config xorg.conf.new`

X 将启动，在屏幕中央会有一个白色的 X，鼠标或是触摸板可以移动它，按 CTRL-Alt-Backspace 可以退出 X 从 Xorg 7.4 和更高的版本开始，这个测试将显示出一个黑色的屏幕，对于判断 X11 是否能正常工作会造成一些困扰。可以通过 retro 选项使用旧的模式：

1. `Xorg -config xorg.conf.new -retro`

如果看到黑灰的格子以及 X 型鼠标指针，就表示配置成功了。要退出测试，需要同时按下 `Ctrl+Alt+Fn` 来切换到用于启动 X 的虚拟控制台 (`F1` 表示第一个虚拟控制台) 之后按 `Ctrl+C`。

对xorg.conf除错

如果 X 出现错误不能运行，你需要手动检查你的配置文件：

```
nano /root/xorg.conf.new
```

- 确定 `Xorg -configure` 命令正确的识别了你的显卡，比如：

```
Section "Device"
    ...
    Driver   "savage"
    ...
EndSection
```

- 确定里面的 Section "Monitor" 设置了水平同步率和垂直刷新率，如果没有要把它加上：

```

Section "Monitor"
    Identifier      "Monitor0"
    VendorName     "Monitor Vendor"
    ModelName      "Monitor Model"
    HorizSync       30.0 - 130.0 # Safe for LCD's
    VertRefresh     50.0 - 100.0 # Safe for LCD's and most CRT's.
EndSection

```

如果你不知道这些刷新率到底是多少，你就得查一下你显示器的说明书。

- 在section "Screen"配置默认色深 (color depth) :

```

Section "Screen"
    Identifier "Screen0"
    Device     "Card0"
    Monitor    "Monitor0"
    DefaultDepth 24

```

(此处色深被设定为真彩24色。)

- 在subsection "Display" 加入你需要的Modes, at least under the Depth 24 header, e.g.:

```

SubSection "Display"
    Viewport   0 0
    Depth     24
    Modes    "1024x768" "800x600" "640x480"

```

- Add the following section, if you wish to enable eye candy which requires the composite extension:

```

Section "Extensions"
    Option "Composite" "Enable"
EndSection

```

- 上述调整完成后，再测试你的配置:

```
x -config /root/xorg.conf.new
```

- 在确认 X session开启，没有错误之后，把测试好的配置文件移到 /etc/X11下，作为系统的 xorg.conf配置文件:

```
mv /root/xorg.conf.new /etc/X11/xorg.conf
```

详细的介绍可见Xorg

接下来可以简单作测试 **Simple baseline X test**

调整键盘布局

如果开启了input热插拔

```
# cp /usr/share/hal/fdi/policy/10osvendor/10-keymap.fdi /etc/hal/fdi/policy
# nano /etc/hal/fdi/policy/10-keymap.fdi
```

Edit the "input.xkb.layout" key and possibly the "input.xkb.variant" key in this file. Restart hal:

```
# /etc/rc.d/hal restart
```

如果关闭了input热插拔

Add option lines in the "InputDevice" section of the /etc/X11/xorg.conf file specifying the keyboard layout and variant:

```
Option "XkbLayout" "be"
Option "XkbVariant" ""
```

详情请见 Xorg input hotplugging。

简单的命令行界面X测试（如果需要）

在安装了xorg、合适的显卡驱动、创建了一个/etc/X11/xorg.conf文件之后，如果想在安装完整桌面环境之前，快速测试下你的配置，使用**xterm**可以实现！Xterm是个简单的终端模拟程序，作为base xorg软件包的一部分会被自动安装，运行在**X Server**环境里。通过配置~/.xinitrc，Xterm可以让我们快速测试显卡驱动和配置文件/etc/X11/xorg.conf是否被正确配置。(另外，如果在没有/etc/X11/xorg.conf的情况下，你想测试下**X**自动检测是否满意，也可以使用这个)

准备工作：配置~/.xinitrc

One of the main functions of this file is to dictate what **X** Window client is invoked with the **/usr/bin/startx** and/or **/usr/bin/xinit** program *on a per-user basis*. (The **startx** script is merely a front end to the more versatile **xinit** command.) There are vast amounts of additional configurable specifications and commands that may also be added to **~/.xinitrc** as you further customize your system.

Note: **.xinitrc** is a so-called 'dot' (.) file. 在UNIX文件系统下，文件名以(.)开头的，是'隐藏文件'，使用普通的命令'ls'将不显示这些文件，使用加了参数命令`ls -a`可以显示。The 'rc' denotes *Run Commands* and simply indicates that it is a configuration file. Since it controls how a program runs, it is (although historically incorrect) also said to stand for "Run Control".

startx/xinit will start the **X** server and clients. To determine the client to run, **startx/xinit** will first look to parse a **.xinitrc** file in the user's home directory. In the absence of file **~/.xinitrc**, it defaults

to the global xinitrc in the xinit library directory; /etc/X11/xinit/xinitrc, which defaults to using the TWM window manager. (说明: 如果没有文件~/.xinitrc, /etc/X11/xinit/xinitrc 将被作为替代来使用, 这是个使用TWM and Xterm的默认配置。)

切换到普通, 非root用户:

```
# su - yourusername
```

- /etc/skel/ contains files and directories to provide sane defaults for newly created user accounts. The name **skel** is derived from the word **skeleton**, because the files it contains form the basic structure for users' home directories.
- 如果没有~/.xinitrc这个文件, 到该目录下创建一个, 或者从/etc/skel/复制一个同样的文件到你的home目录::

```
$ cp /etc/skel/.xinitrc ~/
```

- 编辑文件:

```
$ nano ~/.xinitrc
```

加入这行"exec xterm"像下面这样:

```
#!/bin/sh
#
# ~/.xinitrc
#
# Executed by startx (run your window manager from here)
#
exec xterm
# exec wmaker
# exec startkde
# exec icewm
# exec blackbox
# exec fluxbox
```

Note: 请确保~/.xinitrc里, 以exec开头的, 只有一行的注释被去掉。

开始执行测试

- 以普通用户身份输入下面命令, 开启X, 开始测试你的配置:

```
$ startx
```

或者是这命令

```
$ xinit
```

此时会打开一个**xterm** session，你可以在这里测试your keyboard and its layout

- 按Ctrl+Alt+Backspace，或者输入"exit"可以退出X Server。
- 如果开启X出现问题，在终端，查看日志文件/var/log/Xorg.0.log的错误信息。

更多的Xorg configuration介绍可见Xorg .

第四部分：安装并配置桌面环境

While The X Window System provides the basic framework for building a *graphical user interface* (GUI), a **Desktop Environment** (DE), works atop and in conjunction with X, to provide a completely functional and dynamic GUI. A DE typically provides icons, applets, windows, toolbars, folders, wallpapers, applications and abilities like drag and drop. The particular functionalities and designs of each DE will uniquely affect your overall environment and experience. Therefore, choosing a DE is a very subjective and personal decision. Choose the best environment for *your* needs.

- 如果你需要一个完整功能，类似于Windows和Mac OSX, 可以选择安装**KDE**
- If you want something slightly more minimalist, which follows the K.I.S.S. principle more closely, **GNOME** is a good choice
- **Xfce** is generally perceived as similar to GNOME, but lighter and less demanding on system resources, yet still visually pleasing and providing a very complete environment.

如果你的机器较老，或者是希望使用一个更轻量、更少GUI要求，可以选择只安装 **Window Manager**, 或者 WM. A WM controls the placement and appearance of application windows in conjunction with the X Window System but does NOT provide such additions as panels, applets, icons, etc.

- Lightweight floating WM's include: **Openbox**, **Fluxbox**, **fvwm2**, **Windowmaker**, **Pekwm**, and **TWM**.
- If you need something completely different, try **Awesome**, **ion**, **wmii**, **dwm**, or **xmonad**.

一、安装字体

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和bitstream-vera都是不错的字体集。而对于网页，你还可能需要Microsoft字体。如下安装：

```
pacman -S ttf-ms-fonts ttf-dejavu ttf-bitstream-vera
```

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
```

二、选择默认启动的桌面环境 `~/.xinitrc`

以普通用户身份编辑`/home/username/.xinitrc`，加入你要用的DE. This will allow you to use

startx/xinit from the shell, in the future, to open your DE/WM of choice :

```
nano ~/.xinitrc
```

反注释或者加入相应的桌面环境/窗口管理器的'exec ..'内容行。例如：使用Xfce4桌面环境：

```
exec startxfce4
```

记住在~/.xinitrc中只反注释掉一行**exec**。这条命令将一直运行到你注销桌面登录。一般的，shell将等到kde完成加载之后，在运行下一条命令。在命令前加 "exec"，是告诉the shell 这条命令最后执行，这样shell就不用等待其他命令了，直接载入桌面。接下来安装你所选的桌面环境/窗口管理器。

三、选择安装一个桌面环境

GNOME

关于GNOME

GNOME是GNU Network Object Model Environment的缩写。GNOME项目有两大部分：GNOME桌面环境 — 既直观又迷人的最终用户桌面，以及GNOME开发平台 — 一个用来创建可与桌面集成的应用程序的庞大的框架。

安装

安装基本的GNOME环境：

```
pacman -S gnome
```

安装额外的GNOME环境：

```
pacman -S gnome-extra
```

然后选择提示的所有软件包。

对GNOME有用的守护进程

如之前所述，守护进程(daemon)是运行在后台的程序，它等待事件发生并提供相应的服务。其中**Hal**守护进程可以自动挂载磁盘、光驱和U盘给GUI使用。**Fam**守护进程可以在GUI中实时反映文件的变动，如最近安装的程序、文件系统的变化等。**Hal**和**fam**使得GNOME的使用更为方便。**Hal**和**fam**软件包在安装GNOME时已一同安装，建议安装社区库的gamin来替代fam。

Note: 注意，fam是一个比较老的包，很容易造成较大的日志文件。

Note: 在当前的gnome环境中，你可以安装gamin来实现fam的功能。这时，你可以不用在DAEMONS中添加fam启动项了（不需要加入gamin，它会自动被启用）。

- 你需要启动**hal**守护进程。

启动**hal**：

```
/etc/rc.d/hal start
```

如果您想使用fam，同样需要启动：

```
/etc/rc.d/fam start
```

将**hal**加入到/etc/rc.conf的DAEMONS中以使得它们随系统启动而自动运行：

```
nano /etc/rc.conf
```

```
DAEMONS=(syslog-ng network crond alsal hal)
```

Note: 注意，如果使用fam，那么把fam加入到hal后面。

- 或许你还会希望安装一个图形界面的登录管理器。在GNOME下**gdm**守护进程是个不错的选择。

以root身份运行：

```
pacman -S gdm
```

让gdm启用有两种方式，一种是直接加入到DAEMONS的末尾，通过rc.conf来启动；另外一种方式是用过配置inittab来加载gdm。

- 通过inittab来加载gdm也很方便（推荐使用这种方式）：

```
nano /etc/inittab
```

修改默认启动方式：

将

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

改为

```
id:5:initdefault:
```

然后取消下面代码的注释：

```
x:5:respawn:/usr/sbin/gdm -nodaemon
```

Note: 实际上，可以看到gdm/xdm/kdm/slim等，使用哪个取决于你的喜好，启用喜欢的那个，将其他的注释掉。当然，本章介绍的是gnome，所以推荐gdm，下章中将推荐kdm：)

- 通过DAEMONS方式启动gdm也很方便：

```
nano /etc/rc.conf
```

```
DAEMONS=(syslog-ng network crond alsa hal gdm)
```

Note: 这是类似debian/ubuntu的做法，这种方式有一点不好：不管什么时候（非修复模式），系统总是会尝试进入gdm引导的X方式下。对于X已经配置得很好，并且不使用纯控制台的人可以使用这种方式。但是选择这种方式的话你将不能通过内核参数3或者5来区分进入纯字符控制台还是X11的gdm登录窗口。

- 开启Gnome

以普通用户身份启动X：

```
startx
```

或者这条命令也行：

```
$ xinit
```

如果你的 ~/.xinitrc 一直没配置，那你将可以一直使用下面这条命令来开启Gnome：

```
$ xinit /usr/bin/gnome-session
```

关于安装和配置GNOME的高级指引可以参见GNOME一文。接下来你可以浏览**优化和收尾**。或许你还会对Arch安装完后的小技巧一文感兴趣。

美化

如果不满足于缺省的GNOME主题和图标，那么可以安装一个不错的主题，比如murrine：

```
pacman -S gtk-engine-murrine
```

接着在System->Preferences->Theme里选择这个主题。

Arch Linux也做了些不错的主题，根据你自己的审美观，选择满意的安装吧：

```
# pacman -S gtk-engines gtk2-themes-collection gtk-aurora-engine gtk-candid
```

你还可以在Gnome Look (<http://www.gnome-look.org>) 里找到更多的主题、图标和壁纸。

KDE

关于KDE

KDE桌面环境(**K Desktop Environment**)的缩写。 KDE是一种著名的运行于GNU/Linux、 UNIX 工作站等的强大、自由图形工作环境。她拥有对系统内部各功能的强大整合能力、跨平台能力、易用性、优异的初始界面。

安装

Arch提供多个版本的kde: **kde**、**kdebase** 和 **KDEmod**. 选一个就行了，然后 **有用的KDE守护进程**:

1.) kde是完整的kde软件包,大概 ~300MB ,适合主要使用kde软件的系统.

```
pacman -S kde
```

2.) kdebase是kde的基础软件包 (里面没有kdm) ,大概 ~80MB ,适合于只想把kde作为基础系统.

```
pacman -S kdebase
```

3.) KDEmod 是 Arch Linux独有的、社区驱动的项目，目标是把kde更加细致的模块化和达到其极致的运行效率。KDEmod项目的网站在<http://chakra-project.org/> 。

有用的KDE守护进程

KDE 需要**hal** (**H**ardware **A**bstraction **L**ayer) 作为守护进程 (daemons) . **kdm** (**K** **D**isplay **M**anager) , 提供一个 **图形化登录界面**.

Note: hal守护进程将自动开启dbus守护进程。kde4里已经不需要fam。kdm现在在kdebase-workstation这个包里，如果只安装kdebase，需要再安装此包才能使用kdm

hal 服务配合其他软件, 能自动挂接热插拔硬盘, 光驱, USB drives/thumbdrives for use in the GUI.

启动hal进程:

```
/etc/rc.d/hal start
```

编辑/etc/rc.conf的 DAEMONS部分:

```
nano /etc/rc.conf
```

把 **hal** 加入 DAEMONS 部分,以便他们在系统启动时加载. 如果想要kde的图形化登录界面 , 把 **kdm** 也加进去 (或者不加 , 在/etc/inittab里面设置) :

```
DAEMONS=(syslog-ng network crond alsal hal kdm)
```

Note: 如果你安装的是KDEmod3 , 则使用kdm3替换kdm

- 这种方式能启动运行等级3, (/etc/inittab 默认多用户模式), 然后启动 KDM 进程来登录.
- 有些用户喜欢使用 /etc/inittab 方式来启动系统到运行等级5. 细节请参考 Display Manager.
- 如果你喜欢启动时进入命令行, 即运行等级3, 然后手动启动 X, 则不加入kdm, 或者是加个前缀 (!)禁止它.
- 现在可以试试效果了 , 用普通用户身份开启X Server:

```
startx
```

或者

```
$ xinit
```

更多关于安装和配置KDE的文章 , 请看KDE.

到此步 , 可以看 #优化和收尾这部分了. 你可能想参考 Post Installation Tips wiki文章.

Xfce

关于Xfce

和GNOME、KDE一样 , Xfce是一个桌面环境 , 其目标是快捷、轻量 , 同时又美观而易用。它还提供了一整套的应用程序如根窗口应用 (root window app) 、 窗口管理器、文件管理器、面板等等。Xfce是用GTK2 toolkit编写的 (类似GNOME) , 和其它大型桌面环境一样 , 它也有自己的开发环境 (库文件、守护进程等)。但不象GNOME或者KDE , Xfce是轻量化的 , 其设计更接近于CDE而不是Windows或者Mac。它的开发周期比较慢 , 但是十分稳定、速度很快。在老硬件上Xfce可以运行得很好 , 而在新机器上更是运行如飞。

安装

安装完整的xfce环境以及主题和extras：

```
pacman -S xfce4 xfce4-goodies gtk2-themes-collection
```

如果你使用KDM或者GDM的话会显示一个xfce会话。另一种启动方式是：

```
startxfce4
```

关于安装和配置xfce的高级指引可以参见Xfce一文。

恭喜！欢迎来到Arch Linux系统的Xfce桌面环境！接下来你可以浏览[优化和收尾](#)，或者下面的其它内容。或许你还会对Arch安装完后的小技巧一文感兴趣。

*box

Fluxbox

Fluxbox © 是另一个流行的快速轻量级的 X 下的窗口管理器. 它是在 Blackbox 0.61.1 的代码基础上开发的. Fluxbox 看上去很像 blackbox, 它的界面方案, 颜色, 窗口按钮等都跟blackbox相似 (界面方案100% 兼容).

安装fluxbox:

```
pacman -S fluxbox fluxconf
```

如果你使用 gdm/kdm 登录管理器, fluxbox 会被添加到session选项列表. 或者你可以修改 .xinitrc 在最后一行加上:

```
exec startfluxbox
```

更多信息可参考 Fluxbox.

Openbox

Openbox是一个标准的，顺手的，快速的，轻量级的，可扩展的窗口管理器。

Openbox与你的应用程序一起工作工作，并使你的的桌面便于管理。这是因为它的开发方式和通常的窗口管理器的方式是相悖的。 Openbox was written first to comply with standards and to work properly. Only when that was in place did the team turn to the visual interface.

Openbox is fully functional as a stand-alone working environment, or can be used as a drop-in replacement for the default window manager in the GNOME or KDE desktop environments.

安装openbox

```
pacman -S openbox obconf obmenu
```

一旦openbox安装好，你将得到一个消息to move menu.xml & rc.xml to ~/.config/openbox/ in your home directory:

```
mkdir -p ~/.config/openbox/
cp /etc/xdg/openbox/rc.xml ~/.config/openbox/
cp /etc/xdg/openbox/menu.xml ~/.config/openbox/
```

In the file "rc.xml" you can change various settings for Openbox (or you can use OBconf). In "menu.xml" you can change your right-click menu.

To be able to log into openbox you can either go via graphical login using KDM/GDM or startx, in which case you will need to edit your ~/.xinitrc (as user) and add the following:

```
exec openbox
```

For KDM there is nothing left to do; openbox is listed in the sessions menu in KDM.

Useful programs for openbox are:

- PyPanel or LXpanel if you want a panel
- feh if you want to set the background
- ROX if you want a simple file manager and desktop icons

More information is available in the Openbox article.

fvwm2

FVWM是一个非常强大的面向X桌面系统的”ICCCM-compliant的多虚拟桌面的窗口管理器“ (ICCCM-compliant multiple virtual desktop window manager)。开发活动很活跃，支持也是极好的。

输入下面命令安装 fvwm2

```
pacman -S fvwm
```

fvwm会自动添加进 kdm/gdm会话列表. 如果没有的话，添加

```
exec fvwm
```

到你的用户文件 .xinitrc.

注意：这个稳定版已经有些年头了，如果需要最新的fvwm版本，可以安装unstable repo里的fvwm-devel 包。

四、优化和收尾

HAL

如果你已经安装了一个桌面环境，但是又觉得用起来不是很方便，那么现在就是个安装HAL的好时机了。HAL使得你的手机、iPod、移动硬盘等等可以即插即用。它会挂载设备并在你的桌面或'和'My Computer'目录中创建相应的图标，使得你可以在插入设备后就能访问到它，无需每次都对每个新设备来手动配置/etc/fstab文件或者udev规则。

KDE、GNOME和XFCE都使用到HAL。

安装过程描述可以在HAL一文中找到。其它一些信息可以在Wikipedia ([http://en.wikipedia.org/wiki/HAL_\(software\)](http://en.wikipedia.org/wiki/HAL_(software))) 上找到。

启动时让守护进程在后台执行

要加速系统的启动过程，可以在/etc/rc.conf中给守护进程前加上'@'符号使之在后台执行，例如：

```
DAEMONS=(@syslog-nd @network crond @alsa @hal @fam @kdm)
```

这使得守护进程在后台被加载，无需等待它们加载结束。

在守护进程前加上(!)可以禁止系统启动时加载它，例如：

```
DAEMONS=(@syslog-nd @network !crond @alsa @hal @fam @kdm)
```

另一种方式是直接将不需要的守护进程删除。

美化液晶显示器(LCD)下的字体

液晶显示器和彩显或者电子纸等的不同成像原理，可能会导致字体显示效果也不太一样，所以有时需要对字体显示效果进行配置 See Fonts

让所有的鼠标按键生效

看这个 Get All Mouse Buttons Working

笔记本电脑可能遇到的问题

- (1) 触摸板问题 参看http://wiki.archlinux.org/index.php/Touchpad_Synaptics
- (2) 摄像头问题 参看http://wiki.archlinux.org/index.php/Webcam_Setup
- (3) 无线上网问题 参看<http://wiki.archlinux.org/index.php/Wireless>
- (4) 快捷键问题 参看http://wiki.archlinux.org/index.php/Extra_Keyboard_Keys

设置CPU频率

现在的处理器都能够降低频率和电压，以降温和降低电源消耗。降温能使系统更稳定，笔记本用户肯定是想这样做的，但是即使是台式机也会对系统有利。

安装CPU频率工具用：

```
pacman -S cpufrequtils
```

把CPU频率在/etc/rc.conf里加入到启动进程(daemons)中。编辑配置文件/etc/conf.d/cpufreq改

```
governor="conservative"
```

这使得在需要的情况下可以动态地增减CPU频率（这是对于台式机来说也是一个安全的选择）。改变最小频率(min_freq)和最大频率(max_freq)以匹配你的CPU的情况。如果你不知道频率，在载入频率测量模块后运行“cpufreq-info”。你也可以将min_freq和max_freq注释掉或删除，这样就能自动进行。加入频率测量模块到/etc/rc.conf的模块行。几乎所有的笔记本和台式机都可以简单地使用acpi-cpufreq驱动，但是也有其它的选择，如：p4-clockmod, powernow-k6, powernow-k7, powernow-k8, and speedstep-centrino。

通过下面这个命令载入这个模块：

```
modprobe <modulname>
```

通过下面命令行开启cpufreq：

```
/etc/rc.d/cpufreq start
```

更多细节，请看 Cpufrequtils

第五部分：常用的应用程序

更多信息，请看好用的应用程序 (简体中文).

附录

新手指南附录ArchLinux新手指南附录 (简体中文)

想得到更多的信息和支持，你可以去homepage (<http://www.archlinux.org/>) , search the wiki , forums (<http://bbs.archlinux.org/>) , IRC channel , 以及mailing lists (<http://www.archlinux.org/mailman/listinfo/>) 。

问题答疑

See Arch FAQs for newbies 还有这里Arch forum thread (<http://bbs.archlinux.org/viewtopic.php?t=29055>) 也是面向初学者的。

术语

更多信息关于Arch术语，查阅[this article](#).

接下来还有什么？

你还可以阅读：

[Arch 官方安装指南](#)

[ArchLinux社区用户软件仓库 \(AUR\)](#)

[Get All Mouse Buttons Working](#)

[改善Pacman性能](#)

[Kernel Compilation](#)

[Pm-utils 电源管理](#)

[CPU 频率调节](#)

美化

Retrieved from "[http://wiki.archlinux.org/index.php/ArchLinux_%E6%96%B0%E6%89%8B%E6%8C%87%E5%8D%97_\(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87\)](http://wiki.archlinux.org/index.php/ArchLinux_%E6%96%B0%E6%89%8B%E6%8C%87%E5%8D%97_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87))"

Categories: [简体中文](#) | [获取并安装Arch \(简体中文\)](#) | [关于Arch \(简体中文\)](#) | [HOWTOs \(简体中文\)](#)

- This page was last modified on 15 April 2010, at 06:12.
- Content is available under [GNU Free Documentation License 1.2](#).