

navigation

- [Main page](#)
- [Categories](#)
- [Getting Involved](#)
- [Wiki News](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

search

Go

Search

toolbox

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)

in other languages

- [Български](#)
- [Česky](#)
- [Dansk](#)
- [Deutsch](#)
- [English](#)
- [Español](#)
- [فارسی](#)
- [Français](#)
- [עברית](#)
- [Hrvatski](#)
- [Magyar](#)
- [Bahasa Indonesia](#)
- [Italiano](#)
- [日本語](#)
- [한국어](#)
- [Lietuvių](#)
- [Nederlands](#)
- [Polski](#)
- [Português](#)
- [Русский](#)
- [Slovenčina](#)
- [Српски / srpski](#)
- [Türkçe](#)
- [Українська](#)
- [中文（台灣）](#)

- page
- discussion
- view source
- history

Beginners' Guide (简体中文)

小贴士：本向导可以分页浏览，请从[这里](#)开始。

欢迎，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。 本文档将指导您使用**Arch 安装脚本**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。建议安装前先浏览一下**FAQ**。

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**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。

Contents [\[hide\]](#)

- 1 准备
- 1.1 刻录或写入最新的安装媒介

1.1.1 通过网络安装

1.1.2 在虚拟机中安装
- 1.2 引导 ArchLinux 安装媒介

1.2.1 检查是否启动到 UEFI 模式

1.2.2 启动问题解决
- 2 安装
- 2.1 修改语言

2.2 建立网络连接

2.2.1 有线连接

2.2.2 无线网络

2.2.2.1 不使用 wifi-menu

2.2.3 xDSL (PPPoE), 模拟调制解调器或者 ISDN

2.2.4 位于代理服务器后面
- 2.3 准备存储设备

2.3.1 选择分区表类型

2.3.2 Partitioning tool

2.3.3 Partition scheme

2.3.4 Considerations for dualbooting with windows

2.3.5 分区示例

2.3.5.1 Using cgdisk to create GPT partitions

2.3.5.2 Using fdisk to create MBR partitions

2.3.6 创建文件系统
- 2.4 挂载分区

2.5 选择安装镜像

2.6 安装基本系统

2.7 生成 fstab

2.8 Chroot 到新系统

2.9 配置系统

2.9.1 Locale

2.9.2 终端字体和键盘映射

2.9.3 时区

2.9.4 硬件时间

2.9.5 内核模块

2.9.6 Hostname
- 2.10 配置网络

2.10.1 有线网络

2.10.1.1 动态 IP

2.10.1.2 静态 IP

2.10.2 无线网络

2.10.2.1 添加无线网络

2.10.2.2 自动连接到已知网络
- 2.10.3 模拟信号猫, ISDN 或 PPoE DSL

2.11 创建初始 ramdisk 环境

2.12 设置 Root 密码

2.13 安装配置启动加载器

2.13.1 For BIOS motherboards

2.13.1.1 Syslinux

2.13.1.2 Grub

2.13.2 对于 UEFI 主板

2.13.2.1 Gummiboot

2.13.2.2 GRUB

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

- [Official Installation Guide \(简体中文\)](#)
- [Install from SSH \(简体中文\)](#)
- [Network Installation Guide](#)
- [General Recommendations \(简体中文\)](#)
- [General Troubleshooting](#)

2.14 卸载分区并重启系统

3 安装之后

3.1 用户管理

3.2 软件包管理

3.3 服务管理

3.4 声音

3.5 图形用户界面

3.5.1 安装 X

3.5.2 安装显卡驱动

3.5.3 安装输入设备驱动

3.5.4 配置 X

3.5.5 测试 X

3.5.5.1 疑难解答

3.5.6 安装字体

3.5.7 选择并安装图形界面

4 附录

准备

注意: 要从已经存在的 GNU/Linux 发行版安装 Arch, 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。

刻录或写入最新的安装媒介

[这里](#) 可以下载最新的官方 Arch Linux 安装媒介。ISO 大概每月更新一次, 建议每次安装前都下载使用最新版本。所提供的 ISO 映像文件同时支持32位和64位构架。

- 用你喜欢的软件把 ISO 映像文件刻录到光盘上。

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。如果遇到刻录异常的情况, 请试着将速度调至系统支持的最低速。

- 你也可以写入到U盘安装。更多信息, 请参阅 [USB 安装](#)。

通过网络安装

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, 但前提是你已经设置好一个服务器。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, 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](#)
- [Moving an existing install into \(or out of\) a virtual machine](#)

引导 ArchLinux 安装媒介

插入准备好的光盘或者闪存设备, 然后用它启动电脑。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`。

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。

小贴士: 基本安装所需内存为 64 MB

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](#)。

Live 环境所提供的shell为Zsh, 它能提供更好的Tab补全, 以及[grml config](#) 的其他特性。

检查是否启动到 UEFI 模式

如果使用的是**UEFI**主板并且UEFI启动模式已启用, 安装程序会自动启动内核 (EFISTUB via Gummiboot Boot Manager)。要测试系统是否已启动到 UEFI 模式, 检查 `/sys/firmware/efi/vars/` 是否存在文件:

```
# ls -l /sys/firmware/efi/vars/
```

注意: 内核模块 `efivars` 已经编译进内核, 无需手动加载。

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 `nomodeset` 然后按 `Enter`。另外也可以试一下 `video=SVIDEO-1:d`，这种方法如果可以使用的話不会禁用 **kernel mode setting**. 参见 **Intel** 获取更多信息。
- 如果屏幕不是空白，但是启动过程在内核装入时卡住，按 `Tab` 编辑内核行，在最后加入 `acpi=off` 然后按 `Enter`。

安装

您将看到一个命令行提示，自动以 `root` 登陆。

修改语言

提示: 对于大部分用户，这些是非必需操作。只在这些情况下有需要：打算用母语编辑配置文件，在Wi-Fi密码中使用了变音记号，或需要母语显示的系统信息（如错误信息）。修改仅影响安装过程。

键盘布局默认为 `us`（美式键盘映射），若正在使用非**美式**键盘布局，通过以下的命令选择相应的键盘映射表：

```
# loadkeys layout
```

把 *layout* 替换成你的键盘布局，如 `fr`，`uk`，`dvorak` 或 `be-latin1`。[[[这里](#)有国家与二位代码的映射表。

多数语言使用多于**26个英文字母**的字形，因此字体也需要进行相应的修改，否则这些语言的文字会显示成白方块。注意这些名称是区分大小写的：

```
# setfont Lat2-Terminus16
```

系统默认将语言设为美式英语。若想在安装过程中使用其他语言（以**德语**为例），从 `/etc/locale.gen` 中去掉相应**locale** 前的 `#`。请选择含 `UTF-8` 的行。

按下 `Ctrl+X` 退出，在保存提示出现时按下 `Y` 和 `Enter` 来使用相同的文件名。

```
# nano /etc/locale.gen

en_US.UTF-8 UTF-8
de_DE.UTF-8 UTF-8

# locale-gen
# export LANG=de_DE.UTF-8
```

记得按下 `LAlt+LShift` 启用新的映射

建立网络连接

Warning: 从systemd 197 版本开始，udev 不再按照**wlanX**，**ethX**的方式分配网络接口的名字。如果你是从其他发行版转来，请不要认为有线接口名字就是**eth0**，无线接口是**wlan0**，请用 `ip link` 命令查看接口的名字

安装程序会自动运行 `dhcpcd` 尝试配置有线连接。可以通过 `ping` 一个网站检查是否成功。

```
# ping -c 3 www.baidu.com

PING www.a.shifen.com (61.135.169.105) 56(84) bytes of data.
64 bytes from 61.135.169.105: icmp_seq=1 ttl=57 time=0.495 ms
64 bytes from 61.135.169.105: icmp_seq=2 ttl=57 time=0.603 ms
64 bytes from 61.135.169.105: icmp_seq=3 ttl=57 time=0.590 ms

--- www.a.shifen.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.495/0.562/0.603/0.055 ms
```

如果出现：`ping: unknown host` 错误，表示网络还不通，先检查网线接触是否良好，无线信号是否足够，如果它们没有问题，需要用下面方法手动配置。

如果可以连通的话，继续 **#准备硬盘**。

有线连接

如果使用静态**ip**配置连接到有线网络，请使用如下方式配置。

首先，禁用自动启动的 `dhcpcd` 服务：

```
# systemctl stop dhcpcd
```

用**ip**命令查看有线连接的接口名字。

```
# ip link

1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether 00:11:25:31:69:20 brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc mq state UP mode DORMANT qlen 1000
    link/ether 01:02:03:04:05:06 brd ff:ff:ff:ff:ff:ff
```

在这个例子中，有线连接的接口名字为 `enp2s0f0`。如果你不确定的话，有线连接一般以"e"开头，一般会以"lo"或"w"开头。你也可以用`iwconfig`命令查看哪些接口不是无线的来帮助你判断有线连接的接口名。

```
# iwconfig

-----

enp2s0f0  no wireless extensions.
wlp3s0    IEEE 802.11bgn  ESSID:"NETGEAR97"
          Mode:Managed  Frequency:2.427 GHz  Access Point: 2C:B0:5D:9C:72:BF
          Bit Rate=65 Mb/s   Tx-Power=16 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=61/70   Signal level=-49 dBm
          Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0
          Tx excessive retries:0   Invalid misc:430   Missed beacon:0
lo        no wireless extensions.
```

在这个例子中，`enp2s0f0`和`lo`都不是无线设备，所以`enp2s0f0`是有线接口。

你也应该了解以下信息：

- 静态IP地址，
- 子网掩码，
- 网关IP地址，
- 域名服务器(DNS)IP地址，
- 域名 (如果是本地局域网，可以任意取名).

激活接口，(例如 `enp2s0f0`):

```
# ip link set enp2s0f0 up
```

添加地址：

```
# ip addr add <ip 地址>/<掩码位数> dev <接口名>
```

例如：

```
# ip addr add 192.168.1.2/24 dev enp2s0f0
```

更多信息参看：`man ip`

用类似下面的命令添加网关，`ip 地址`替换为实际的网关地址：

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

例如：

```
# ip route add default via 192.168.1.1
```

编辑 `/etc/resolv.conf` 如下，替换你的DNS服务器IP地址和本地域名：

```
# nano /etc/resolv.conf

-----

nameserver 8.8.8.8
nameserver 114.114.114.114
search example.com
```

注意：目前，你最多只能包含三行 `nameserver` 字段. 要突破限制，请使用本地域名缓存，例如 [Dnsmasq](#)。

这时应该已经有一个可用的有线网络了，如果没有，请查阅详细的[配置网络](#)页面。

无线网络

如果需要用Wi-Fi连接网络，请采用以下步骤。

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。

首先确定网络接口名称：

```
# iw dev

-----

phy#0

    Interface wlp3s0
            ifindex 3
            wdev 0x1
            addr 00:21:6a:5e:52:bc
            type managed
```

下面的示例中使用 `wlp3s0` 作为接口，`linksys` 作为 ESSID。请根据实际情况修改。

注意：输出和下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](#)。

- 启用接口：

```
# ip link set wlp3s0 up
```

除了驱动，一小部分无线网卡还需要固件。内核一般会自动探测并加载固件，如果启用网络时，缺少固件的接口会提示错误：`SIOCSIFFLAGS: No such file or directory`。若不确定，可以用 `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。

然后使用 `netctl` 提供的 `wifi-menu` 连接到网络：

```
# wifi-menu wlp3s0
```

到此，无线网络应该已经可用，如遇问题，请查看[无线设置](#)。

不使用 `wifi-menu`

手动配置方法，先扫描网路：

```
# iw dev wlp3s0 scan | grep SSID
```

连接网络：

```
# wpa_supplicant -B -i wlp3s0 -c <(wpa_passphrase "ssid" "psk")
```

将 `ssid` 替换为实际网络的名称，`psk` 替换为无线密码，请保留引号。最后，用 `dhcpcd` 获取 IP 地址：

```
# dhcpcd wlp3s0
```

如果不行，可以试试

```
# echo 'ctrl interface=DIR=/run/wpa_supplicant' > /etc/wpa_supplicant.conf
# wpa_passphrase <ssid> <passphrase> >> /etc/wpa_supplicant.conf
# ip link set <interface> up # May not be needed as dhcpcd should bring it up but may be needed for wpa_supplicant.
# wpa_supplicant -B -D nl80211 -c /foobar.conf -i <interface name>
# dhcpcd -A <interface name>
```

xDSL (PPPoE), 模拟调制解调器或者 ISDN

如果你使用拨号网络或者ISDN，请阅读[Direct Modem Connection](#)。

位于代理服务器后面

如果使用代理服务器，请 `export` `http_proxy` 和 `ftp_proxy` 变量。[查看这里](#) 获取更多信息。

准备存储设备

警告：磁盘分区有时会毁掉原分区内的数据。**强烈建议**先备份重要的数据。

选择分区表类型

可以选择使用 [GUID Partition Table](#) (GPT) 或者 [Master Boot Record](#) (MBR). GPT 更现代，建议新安装系统时使用。

- 如果要和 windows 双系统启动，需要仔细阅读 [Partitioning#Choosing_between_GPT_and_MBR](#) 作出选择。
- 推荐所有 UEFI 使用 GPT 分区格式，有些 UEFI 固件不支持 UEFI-MBR 启动。
- 有些 BIOS 系统不支持 GPT，参阅[and http://rodsbooks.com/gdisk/bios.html 这里](#)
- 如果要安装 Arch Linux 到U盘（USB闪存盘），请阅读[Installing Arch Linux on a USB key](#)。

Partitioning tool

新手建议使用图形化的分区工具。[GParted](#) 是个很不错的选择。可以从 "live" Linux 发行版 (如 [Parted Magic](#), [Ubuntu](#), [Linux Mint](#), 等等) 启动。硬盘在使用前需要先[分区](#)，然后将分区格式化为需要的[文件系统](#)。

小贴士：使用 `Gparted` 时，可以用选项设置成默认创建 "msdos" 分区，如果要创建 GPT 分区，可以选择 "Advanced" 并选择 "gpt"。

`gparted` 可能更容易使用，如果仅要创建几个分区，可以通过[fdisk 相关程序](#)快速完成，这些工具安装盘里面已经包含，参阅 [Partitioning](#)。

Partition scheme

You can decide into how many partitions the disk should be split, and for which directory each partition should be used in the system. The mapping from partitions to directories (frequently called 'mount points') is the [Partition scheme](#). The simplest, and not a bad choice, is to make just one huge `/` partition. Another popular choice is to have a `/` and a `/home` partition.

Additional required partitions:

- If you have a **UEFI** motherboard, you will need to create an extra **EFI System Partition** (about 1 GiB recommended).
- If you have a BIOS motherboard (or plan on booting in BIOS compatibility mode) and you want to setup GRUB on a GPT-partitioned drive, you will need to create an extra **BIOS Boot Partition** of size 1 or 2 MiB and **EF02** type code. Syslinux does not need one.
- If you have a requirement for a **Disk_encryption** of the system itself, this must be reflected in your partition scheme. It is unproblematic to add encrypted folders, containers or home directories after the system is installed.

如果你想现在设定交换分区，参见 **Swap** 获取更多信息，使用交换文件的话，修改文件的大小比修改分区大小更方便，但是目前交换文件不支持 Btrfs。

Considerations for dualbooting with windows

If you have an existing OS installation, please keep in mind that if you were to just write a completely new partition table to disk then all the data which was previously on disk would be lost.

The recommended way to setup a linux/windows dual booting system is to first install windows, only using part of the disk for its partitions. When you have finished the windows setup, boot into the linux install environment where you can create additional partitions for linux while leaving the existing windows partitions untouched.

此外，新的预装 Windows 8 电脑都默认使用安全启动。Arch Linux 当前还不支持安全启动，如果在 BIOS 中关闭安全启动则会出现一些预装的 Windows 8 现无法启动的情况。在某些情况下，同时在 BIOS 中关闭安全启动和 FASTBOOT 选项让 Windows 8 开机无安全启动是很有必要的。因此，这也许是一个更好的方案 to keep the Windows 8 install intact and have an independent hard drive for the Linux install - which can then be partitioned from scratch using a GPT partition table. Once that is done, creating several ext4/FAT32/swap partitions on the second drive may be a better way forward if the computer has two drives available. This is often not easy or possible on a small laptop. Currently, Secure Boot is still not in a fully stable state for reliable operation, even for Linux distributions that support it.

如果你已经分过区，继续 **挂载分区**。否则，参考下面的例子。

分区示例

当前安装盘包含如下工具：`fdisk`，`gdisk`，`cfdisk`，`cgdisk`，`parted`。

Tip: 用 `lsblk` 命令显示连接到系统的硬盘和分区，这样可以确定要使用的分区。

此例使用 **cfdisk**。将 `cfdisk` 替换为 **cgdisk** 可以使用新的 **GPT** 分区。

示例硬盘将划分为15GB的根分区 `/`、剩下的作为 `/home`，选择 **MBR** 或 **GPT** 其中的一个。

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。详细分区介绍请参阅 **Partitioning**。

Using cgdisk to create GPT partitions

```
# cgdisk /dev/sda
```

Root:

- Choose New (or press `N`) – `Enter` for the first sector (2048) – type in "15360M" – `Enter` for the default hex code (8300) – `Enter` for a blank partition name.

Home:

- Press the down arrow a couple of times to move to the larger free space area.
- Choose New (or press `N`) – `Enter` for the first sector – `Enter` to use the rest of the drive (or you could type in the desired size; for example "30G") – `Enter` for the default hex code (8300) – `Enter` for a blank partition name.

完成之后，分区界面应该类似下面这样：

Part. #	Size	Partition Type	Partition Name
1	1007.0 KiB	free space	
2	15.0 GiB	Linux filesystem	
3	123.45 GiB	Linux filesystem	

请再次仔细检查分区大小和布局是否正确。

确认无误后，选择 Write (或者按 `Shift+W`) 完成分区，并输入 `yes` 将修改写入磁盘，然后退出。如果要重新开始，可以选择 **Quit** 退出分区程序。

Using fdisk to create MBR partitions

Note: There is also `cfdisk`, which is similar in UI to `cgdisk`, but it currently does not automatically align the first partition properly. That is why the classic `fdisk` tool is used here.

Launch `fdisk` with:

```
# fdisk /dev/sda
```

- Command (m for help): type `o` and press `Enter`

Then create the first partition:

1. Command (m for help): type `n` and press `Enter`

2. Partition type: Select (default p): **press** `Enter`
3. Partition number (1-4, default 1): **press** `Enter`
4. First sector (2048-209715199, default 2048): **press** `Enter`
5. Last sector, +sectors or +size{K,M,G} (2048-209715199....., default 209715199): **type** `+15G` and **press** `Enter`

Then create a second partition:

1. Command (m for help): **type** `n` and **press** `Enter`
2. Partition type: Select (default p): **press** `Enter`
3. Partition number (1-4, default 2): **press** `Enter`
4. First sector (31459328-209715199, default 31459328): **press** `Enter`
5. Last sector, +sectors or +size{K,M,G} (31459328-209715199....., default 209715199): **press** `Enter`

Now preview the new partition table:

- Command (m for help): **type** `p` and **press** `Enter`

```
Disk /dev/sda: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x5698d902

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            2048        31459327    15728640    83   Linux
/dev/sda2       31459328    209715199     89127936    83   Linux
```

Then write the changes to disk:

- Command (m for help): **type** `w` and **press** `Enter`

If everything went well *fdisk* will now quit with the following message:

```
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

In case this doesn't work because *fdisk* encountered an error, you can use the `q` command to exit.

创建文件系统

分区之后，还需要用 `mkfs` 将分区格式化为选定的[文件系统](#)。本例中 `root` 和 `home` 分区都使用 `ext4`:

```
# mkfs.ext4 /dev/sda1
# mkfs.ext4 /dev/sda2
```

如果你分了一个swap区，不要忘了格式化和启用它：

```
# mkswap /dev/sdaX
# swapon /dev/sdaX
```

UEFI 系统需要格式化 ESP 分区(通常是 `sda1`)，命令是：

```
# mkfs.vfat -F32 /dev/sda1
```

挂载分区

要检查当前磁盘的标识符和布局：

```
# lsblk /dev/sda
```

注意：不要在同一个目录挂载多个分区，而且按照顺序挂载很重要。

记下将要安装上 Arch 的磁盘/分区。每个分区都能通过一个数字后缀来辨认。如：`sda1` 即指某一磁盘的第一分区，而 `sda` 则表示整个磁盘。

注意要按照顺序挂载，先挂载根分区到 `/mnt` 。

```
# mount /dev/sda1 /mnt
```

然后挂载 `/home` 分区和其余单独分区(`/boot` , `/var` 等)。

然后在 `/mnt` 中创建 `home` 目录并挂载分区：

```
# mkdir /mnt/home
# mount /dev/sda2 /mnt/home
```

如果是 UEFI 主板，挂载 UEFI 分区：

```
# mkdir -p /mnt/boot
# mount /dev/sdaX /mnt/boot
```

选择安装镜像

安装前需要编辑 `/etc/pacman.d/mirrorlist`，将最想使用的镜像放到前面。`mirrorlist` 文件也会被 `pacstrap` 复制到新系统，所以最好现在就设置好。

注意: `ftp.archlinux.org` 限速 50KB/s.

ArchLinux 镜像状态 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** 可以自动根据更新状况生成离你比较近的镜像列表。

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` 会强制 `pacman` 刷新所有软件包列表，即使刚刚更新过。一般情况下，只要源有了更改，就应该使用 `pacman -Syy` 命令。

详情请参阅 **Mirrors**.

注意: 有些问题被报告到 **Arch Linux forums**，他们是一些由于网络导致的`pacman` 无法更新、同步的问题。(参考 **[1]** 和 **[2]**)。在实体机器中安装Arch linux 时，我们使用更换`pacman`的默认下载器的方法来应对这个问题(参考 **Improve Pacman Performance**)。当把Ach linux作为客户系统来安装在**VirtualBox**中时，这个问题可以通过在设置中使用宿主网卡而不是“NAT”来得到解决。

安装基本系统

注意: 安装前请设置系统时间，否则密钥验证会认为密钥过期，验证失败。设置方法位于 **Time**。

使用 **pacstrap** 脚本安装基本系统：

```
# pacstrap /mnt base base-devel
```

如果 `pacman` 报告安装时遇到错误的签名(*error: failed to commit transaction (invalid or corrupted package)*)，请运行如下命令：

```
# pacman-key --init && pacman-key --populate archlinux
```

base

来自 **[core]** 软件源的最小基本系统。

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，后续扩展系统会用到它，安装**AUR**中软件包时，`base-devel`也是必须的。

如果不知道还需要什么软件，可以先跳过，后面可以用 **pacman** 安装。

注意: 如果需要连接无线网络，请选中**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。

生成 fstab

用下面命令生成 **fstab**。如果想使用 UUIDs，使用 `-U` 选项；如果想使用标签，用 `-L` 选项。

```
# genfstab -U -p /mnt >> /mnt/etc/fstab
# nano /mnt/etc/fstab
```

Note: 后面如果出现问题，请不要再次运行`genfstab`，如果需要，手动编辑`/etc/fstab`。

使用 `-U` 选项会用 UUID 指定分区，使用 `-L` 会用标签指定分区。

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。仅根分区需要在最后使用 "1"，其它可以使用 "2" 或 "0" (参阅**字段定义**)。

Warning: `/etc/fstab`文件在运行`genfstab`后应该被检查一下。如果之前你生成了一个EFI系统分区，那么 `genfstab` 给EFI分区添加了错误的选项，会导致无法启动。因此你需要移除EFI分区的所有选项，除了 `noatime`。对其他分区，替换 `"codepage=cp437"` 为 `"codepage=437"`，会挂载失败导致`systemd`进入恢复模式否则，在linux 3.8应该会被修复。

- 最后一列决定分区在启动时进行磁盘检查的顺序：只有非 `Btrfs` 格式的(`/`)根分区需要将最后一列设为 `1`，其它应该为 `2` 或 `0` (参阅**fstab#Field definitions**)。
- `btrfs`和 `swap` 分区设置成 `0`，不必进行磁盘检查。

Chroot 到新系统

Note: If you are planning to reboot Arch in EFI mode, read **For UEFI motherboards** as there are some things to take care of **before** entering the chroot. This is necessary to ensure the boot loader or manager can be correctly configured within the

chroot.

下面要 **chroot** 到新安装的系统：

```
# arch-chroot /mnt
```

注意：可以使用 `arch-chroot /mnt /bin/bash` 进入 `bash shell`.

到这一步之后，开始系统的主要配置，如果下面文件不存在，需要手动创建。

理解并完全安装步骤设置是保证系统配置成功的关键。

配置系统

Locale

glibc 和其他一些支持本地化的程序或者库使用 **Locales** 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。

需要编辑两个文件：`locale.gen` 和 `locale.conf`。

- 默认情况下 `/etc/locale.gen` 是一个仅包含注释文档的空文件。选定你需要的本地化类型(移除前面的 `#` 即可), 比如中文系统可以使用：

```
en_US.UTF-8 UTF-8
zh_CN.GB18030 GB18030
zh_CN.GBK GBK
zh_CN.UTF-8 UTF-8
zh_CN.GB2312
```

然后运行：

```
# locale-gen
```

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。

- `locale.conf` 文件默认不存在，一般设置 `LANG` 就行了，它是其它设置的默认值。

```
/etc/locale.conf
```

```
LANG=zh_CN.UTF-8
LC_TIME=en_GB.UTF-8
```

要使用其它 `LC_*` 变量，先运行 `locale` 查看可用选项。高级示例可以参阅[here](#). 不建议使用 `LC_ALL` 变量，因为它会覆盖全部设置。

终端字体和键盘映射

编辑 `/etc/vconsole.conf`。

KEYMAP

可用的键盘映射位于 `/usr/share/kbd/keymaps`。注意此设置仅对 `TTY` 起作用，不改变图形窗口或**X**的设置。

FONT

可用字体位于 `/usr/share/kbd/consolefonts/`，一般可用留空。

FONT_MAP

可选设置，定义要加载的映射，请参考 `man setfont`。可以删除或者留空。

Example:

```
KEYMAP=us
FONT=
```

Note: 详情参阅 [Console fonts](#) 和 `man vconsole.conf`。

时区

将 `/etc/localtime` 软链接到 `/usr/share/zoneinfo/Zone/SubZone`.其中 `Zone` 和 `Subzone` 替换为所在时区，例如

```
# ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

硬件时间

在 `/etc/adjtime` 中设置，默认、推荐的设置为**UTC**。详细信息请阅读[Time \(简体中文\)](#)。

请在同一个机器使用同一个硬件时钟模式，否则它们会覆盖时间导致时间错乱。

可以用下面命令自动生成 `/etc/adjtime`：

UTC(推荐使用):

注意：硬件时钟使用 **UTC** 不代表显示时间时使用 **UTC**.

```
# hwclock --systohc --utc
```

localtime, 不推荐, 但 Windows 默认使用此方式:

警告: 使用 *localtime* 可能导致一些无法修复的 bug。但目前还没有取消此设置选项的计划。

```
# hwclock --systohc --localtime
```

如果要和 Windows 进行双系统启动:

- Recommended: Set both Arch Linux and Windows to use UTC. A quick [registry fix](#) is needed. Also, be sure to prevent Windows from synchronizing the time on-line, because the hardware clock will default back to *localtime*. If you want such functionality (NTP sync), you should use [ntpd](#) on your Arch Linux installation instead.
- Not recommended: Set Arch Linux to *localtime* and disable any time-related services, like `ntpd.service`. This will let Windows take care of hardware clock corrections and you will need to remember to boot into Windows at least two times a year (in Spring and Autumn) when [DST](#) kicks in. So please don't ask on the forums why the clock is one hour behind or ahead if you usually go for days or weeks without booting into Windows.

内核模块

Tip: 一般情况下 udev 会自动加载需要的模块, 大部分用户都不需要手动修改。这里只需要加入真正需要的模块。

`/etc/modules-load.d/` 中保存内核启动时加入模块的配置文件。每个配置文件以 `/etc/modules-load.d/<program>.conf` 的格式命名。配置文件中包含需要装入的内核列表, 每个一行。空行和以 `#` 或 `;` 开头的行直接被忽略。示例:

```
/etc/modules-load.d/virtio-net.conf
```

```
# Load virtio-net.ko at boot
virtio-net
```

Hostname

按喜好设置**主机名**:

```
# echo myhostname > /etc/hostname
```

注意: 现在已经不需要再编辑 `/etc/hosts`。

配置网络

你需要再次配置网络, 但此次是为你所安装的系统而配置。该过程与**前面**的配置方法基本相同, 不同之处在于我们此刻将使其在计算机启动时自动运行。

注意: 了解更多网络配置相关信息, 请访问 [配置网络](#) 及 [Wireless 设置](#)。

Note: If you would like to use the old interface naming scheme (ie. eth* and wlan*) you can accomplish this by creating an empty file at `/etc/udev/rules.d/80-net-name-slot.rules` which will mask the file of the same name located under `/usr/lib/udev/rules.d` (alternatively, instead of an empty file, using a symlink to `/dev/null` is also an acceptable masking method).

有线网络

动态 IP

使用 dhcpcd

如果你只使用一个单一的固定有线网络连接, 你并不需要网络管理服务的话, 你可以简单的使用 `dhcpcd` 服务。在这里,

`interface_name` 是你的网卡名称:

```
# systemctl enable dhcpcd.service
```

如果不行, 试试:

```
# systemctl enable dhcpcd@interface_name.service
```

使用 netctl

复制一个简单的配置样本 `/etc/netctl/examples` 到 `/etc/netctl/`:

```
# cd /etc/netctl
# cp examples/ethernet-dhcp my-network
```

根据你的需要修改配置文件 (修改 `Interface`):

```
# nano my-network
```

启用这个 `my-network` 配置:

```
# netctl enable my-network
```

使用 netctl-ifplugd

警告: 不要将这种方式与 `netctl enable <profile>` 混用。

或者,你也可以使用 `netctl` 的 `netctl-ifplugd` 插件,方便地连接到网络:

安装 `ifplugd`, 这是被 `netctl-ifplugd` 所依赖的:

```
# pacman -S ifplugd
```

然后启用它.在这里, `interface_name` 是你的网卡名称:

```
# systemctl enable netctl-ifplugd@<interface>.service
```

Tip: `Netctl` 也提供了 `netctl-auto`, 可以与 `netctl-ifplugd` 一起管理有线配置。

静态 IP

使用 netctl

复制一个简单的配置样本 `/etc/netctl/examples` 到 `/etc/netctl/`:

```
# cd /etc/netctl
# cp examples/ethernet-static my-network
```

根据你的需要修改配置文件 (修改 `Interface`, `Address`, `Gateway` 和 `DNS`):

```
# nano my-network
```

`Address` 中使用 `/24` 这样 **CIDR 标记方式** 设置掩码为 `255.255.255.0`。

然后启用刚刚创建的配置文件:

```
# netctl enable my-network
```

无线网络

如果你的无线网卡需要固件 (如前所述 [#无线网络](#) 和 [这里](#)), 安装包包含你所需固件的软件包。大部分情况下 `linux-firmware` 会包括你需要的固件。对其他网卡来说, 他们的固件可能在另外的软件包中。比如:

```
# pacman -S zd1211-firmware
```

查看 [Wireless Setup](#) 以获取更多信息。

安装 `iw` 和 `wpa_supplicant`, 你需要他们来连接网络:

```
# pacman -S iw wpa_supplicant
```

添加无线网络

使用 wifi-menu

安装 `dialog`, `wifi-menu` 依赖于他:

```
# pacman -S dialog
```

在完成一下部分的安装并重新启动电脑后, 你可以使用连接到无线网络 `wifi-menu interface_name` 连接到无线网络。(

`interface_name` 是你的无线网卡名).

```
# wifi-menu interface_name
```

警告: 你一定要确保该命令是在电脑*重新启动*以后执行的。否则这个命令会和你在 `chroot` 之前的命令相冲突。此外, 你还可以使用一下命令提供的模版来配置一份网络设置文件, 这样你就不用担心 `wifi-menu` 命令的问题了。

使用人工配置 netctl 配置文件

从 `/etc/netctl/examples` 复制一份模版到 `/etc/netctl`:

```
# cd /etc/netctl
# cp examples/wireless-wpa my-network
```

按照需要修改模版 (modify `Interface`, `ESSID` and `Key`):

```
# nano my-network
```

启动时自动启用:

```
# netctl enable my-network
```

自动连接到已知网络

Warning: You cannot use this method in conjunction with explicitly enabling profiles, such as `netctl enable <profile>`.

Install `wpa_actiond`, which is required for `netctl-auto`:

```
# pacman -S wpa_actiond
```

激活 `netctl-auto` 服务, 他会自动连接到已知的网络并且优雅的处理重新连接和漫游:

```
# systemctl enable netctl-auto@interface_name.service
```

提示: `Netctl` 同时提供 `netctl-ifplugd`, he 可以和 `netctl-auto` 同时使用, 用来处理自动有线连接。

模拟信号猫, ISDN 或 PPOe DSL

关于xDSL, 拨号和 ISDN 连接, 参见 [Direct Modem Connection](#).

创建初始 ramdisk 环境

注意: 大部分用户都不需要在安装时修改此文件。

如有需要, 配置 `/etc/mkinitcpio.conf` (参阅 [mkinitcpio](#)), 然后执行以下命令创建RAM盘:

```
# mkinitcpio -p linux
```

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系统的必要文件。大部分内核模块都是在初始化过程中由 `udev` 装入。

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`"。最好加入 `fsck` 以在系统启动时进行磁盘检查。示例:

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

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

你需要正确的配置 [hooks](#) 如果你的根分区位于USB硬盘, RAID, LVM, 或者 `/usr` 是一个单独的分区。

编辑 `/etc/mkinitcpio.conf` 并使用如下命令重新产生 `initramfs` 文件:

```
# mkinitcpio -p linux
```

提示: 在QEMU上安装Archlinux(比如使用 `virt-manager`) 或许需要在 `mkinitcpio.conf` 中添加 `virtio` 模块以成功启动。

```
# nano /etc/mkinitcpio.conf
-----
MODULES="virtio virtio_blk virtio_pci virtio_net"
```

设置 Root 密码

用 `passwd` 设置一个root密码:

```
# passwd
```

安装配置启动加载器

For BIOS motherboards

BIOS 系统有三个供选择 - Syslinux 和 GRUB。按个人喜好选择一个启动加载器。

- Syslinux 目前只能加载他所安装的硬盘上的文件。他的配置文件相对来说容易理解。模版如下 [这里](#)。
- GRUB 功能丰富且支持更复杂的情况。他的配置文件更像是一种脚本语言, 对新手来说可能写起来困难一些。我们推荐你使用自动产生的配置文件。

Syslinux

如果在之前选择使用 GUID 分区表, 需要安装[gptfdisk](#)。

然后安装 [syslinux](#), 使用 `syslinux-install_update` 命令自动安装文件 (`-i`), 设置分区为活动分区 (`-a`), 安装MBR (`-m`):

提示: 如果你的硬盘分区方式为GPT, 安装 [gptfdisk](#) 和 (`pacman -S gptfdisk`), 因为他们包含 `sgdisk`, he 可以用来设置GPT特有的启动标志。这时你需要使用 `-g` 而不是 `-m`

```
# pacman -S syslinux
# syslinux-install_update -iam ## 或者syslinux-install_update -iag
```

编辑 `/boot/syslinux/syslinux.cfg`，将 `/` 指向正确的根分区，这是必须的，否则 Arch 启动不起来。将 `"sda3"` 修改为实际的根分区。同样，修改 `LABEL archfallback`。

```
# nano /boot/syslinux/syslinux.cfg
-----
...
LABEL arch
...
APPEND root=/dev/sda3 rw
```

Syslinux 的详细使用和配置请阅读[Syslinux](#)。

Grub

提示：在后面命令中根据实际分区进行调整，不要使用类似 `/dev/sda1` 的内容。

BIOS 主板上的 GPT 分区需要 "BIOS Boot Partition"。参阅 [GPT 指令](#) 和 [安装到 GPT BIOS 启动分区](#)。

```
# pacman -S grub
# grub-install --target=i386-pc --recheck --debug /dev/sda
```

虽然手动配置 `grub.cfg` 完全可以工作，建议用户自动生成这个文件。

要搜索硬盘上安装的其它操作系统，请先用 `# pacman -S os-prober` 安装 [os-prober](#)。

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

详细配置信息请阅读[GRUB](#)。

对于 UEFI 主板

For UEFI systems, several options are available. A complete list of options is available at [UEFI Bootloaders](#). You may find that some options work while others do not. Otherwise, choose one as per your convenience. Here, we give some of the possibilities as examples:

- [gummiboot](#) is a minimal UEFI Boot Manager which basically provides a menu for [EFISTUB](#) kernels and other UEFI applications. This is recommended UEFI boot method.
- GRUB is a more complete bootloader, useful if you run into problems with Gummiboot.

For both the bootloaders, you need to have the `efivarfs` mounted at `/sys/firmware/efi/efivars` so they can be accessed from inside the chroot:

```
# mount -t efivarfs efivarfs /sys/firmware/efi/efivars
```

Note: For UEFI boot, the drive needs to be GPT-partitioned and an [EFI System Partition](#) (512 MiB or larger, gdisk type `EF00` , formatted with FAT32) must be present. In the following examples, this partition is assumed to be mounted at `/boot` . If you have followed this guide from the beginning, you have already done all of these.

Gummiboot

First install the [gummiboot](#) package and then run `gummiboot install` to install the bootloader to the EFI System Partition:

```
# pacman -S gummiboot
# gummiboot install
```

You will need to manually create a configuration file to add an entry for Arch Linux to the gummiboot manager. Create `/boot/loader/entries/arch.conf` and add the following contents, replacing `/dev/sdaX` with your root partition, usually `/dev/sda2` :

```
# nano /boot/loader/entries/arch.conf
-----
title Arch Linux
linux /vmlinuz-linux
initrd /initramfs-linux.img
options root=/dev/sdaX rw
```

For more information on configuring and using gummiboot, see [gummiboot](#).

GRUB

Install the [grub](#) and [efibootmgr](#) packages and then run `grub-install` to install the bootloader:

```
# pacman -S grub efibootmgr
# grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=arch_grub --recheck
```

Next, while using a manually created `grub.cfg` is absolutely fine, it is recommended that beginners automatically generate one:

Tip: To automatically search for other operating systems on your computer, install [os-prober](#) before running the next command. However `os-prober` is not know to properly detect UEFI OSes.

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

For more information on configuring and using GRUB, see [GRUB](#).

卸载分区并重启系统

如果还在 `chroot` 环境，先用 `exit` 命令退出系统：

```
# exit
```

卸载 `/mnt` 中挂载的系统：

```
# umount -R /mnt
```

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！

安装之后

现在应该有了一个完全可用的 Arch 系统，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。

用户管理

按照[用户管理页面](#)的介绍，建立想要的账号。不要用 `root` 进行日常工作，也不要服务器上给 `root` 开放[SSH](#)登录权限。`root` 账号应该仅用于系统管理工作。

软件包管理

Pacman 是 Arch 的软件包管理器。[pacman](#) 和 [FAQ](#) 页面提供了安装、升级和管理软件包的信息。

如果安装的是 Arch Linux x86_64，并计划使用 32 位程序，建议[启用 \[multilib\] 仓库](#)。

[官方软件仓库](#) 包含了各个仓库的详细介绍。

服务管理

Arch Linux 使用 [systemd \(简体中文\)](#) 管理系统服务。新用户有必要了解其基本使用方法。通常使用 `# systemctl` 命令进行系统管理，参见[此文](#)。

声音

[ALSA](#)，通常不需要什么设置就能直接工作，只需要解除静音。安装[alsa-utils](#)软件包，它包含了 `alsamixer` 工具，然后按照[此文](#)进行设置即可。

ALSA是一个Linux内核组件，推荐使用。如果不能工作，可以试试[OSS](#)。至于更高级的音频应用，请阅读[Sound](#)。

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。

安装基础的 [Xorg](#) 包：

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

安装 [mesa](#) 以获得 3D 支持：

```
# pacman -S mesa
```

安装显卡驱动

注意： 如果是从 Virtualbox 虚拟机安装，则不需要安装显卡驱动，参见 [此文](#)，然后跳到下面的配置部分。

Linux 内核包含了开源的视频驱动，支持硬件加速的 `framebuffers`。OpenGL 和 X11 的 2D 加速需要用户空间工具。

如果不知道显卡类型，请执行如下命令进行查询：

```
$ lspci | grep VGA
```

输入下面命令，查看所有[开源](#)驱动：

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

`vesa` 是一个支持大部分显卡的通用驱动，不提供任何 2D 和 3D 加速功能。如果无法找到显卡芯片组的对应驱动或载入失败，`Xorg` 会使用[vesa](#)：要安装：


```
# pacman -S xf86-video-vesa
```

要支持硬件加速，使用所有 GPU 支持的显示分辨率，需要安装正确的显示驱动，具体驱动请察看 [Xorg 页面中的列表](#)。

安装输入设备驱动

Udev 应该能够正确检测设备；evdev 驱动 [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-server](#) 的依赖关系被安装。

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
```

触摸板优化或问题处理请访问[Touchpad Synaptics](#)。

配置 X

警告：闭源驱动安装后通常需要重新启动，参见 [NVIDIA](#) 或 [AMD 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](#)。

如果使用Here you may set a [keyboard layout](#) if you do not use a standard [US](#) keyboard.

如果不是使用标准 [US](#) 键盘，可以 [设置键盘布局](#)

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`。

如果在测试中输入设备无法工作，请确保已经安装了[xorg-drivers](#)组中的必要驱动， 输入驱动的列表可以通过下面命令获得，按 `Q` 退出命令。

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

安装默认的测试环境：

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

如果要禁用 [hot-plugging](#)，需要安装[xf86-input-keyboard](#) 或 [xf86-input-mouse](#)，不禁用则使用 `evdev` (推荐)。}}

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
```

注意：X 必须在登录 TTY 启动，只有这样才能保持登录会话，默认的 `/etc/X11/xinit/xserverrc` 已经进行了处理。

最后，启动 X 会话：

```
$ startx
```

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，可以通过下面命令退出**X**：

```
$ exit
```

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`。

可以用下面命令杀死 **X** 服务器。(注意 **X** 大写)：

```
# pkill X
```

如果 `pkill` 还是不行，重启系统：

```
# reboot
```

疑难解答

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](#)。

如果还有问题，请到 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>
```

注意：解决 X 相关问题时，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：

```
# pacman -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。

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。

除了手动启动 X 的方法外，可以让图形界面自动启动，[显示管理器](#) 介绍了启动管理器的使用方法。 [Start X at Login](#) 提供了直接从终端启动的轻量方法。

附录

[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。

Categories: [简体中文](#) | [Getting and installing Arch \(简体中文\)](#) | [About Arch \(简体中文\)](#)