

# **打造完美的 Linux 桌面 ArchLinux 2007.08-2**

文/Toy、Ning Bao

## 版权声明

这份文档发布在“[署名-非商业性使用-相同方式共享 2.5 中国大陆](#)”的创作共用协议之下。即，您可以自由：

- 复制、发行、展览、表演、放映、广播或通过信息网络传播本作品
- 创作演绎作品

惟须遵守下列条件：

- 署名. 您必须按照作者或者许可人指定的方式对作品进行署名。
- 非商业性使用. 您不得将本作品用于商业目的。
- 相同方式共享. 如果您改变、转换本作品或者以本作品为基础进行创作，您只能采用与本协议相同的许可协议发布基于本作品的演绎作品。

对任何再使用或者发行，您都必须向他人清楚地展示本作品使用的许可协议条款。

如果得到著作权人的许可，您可以不受任何这些条件的限制。

## 致谢

感谢 Ning Bao 兄参与这份文档的撰写工作。Archlinux 的灵魂这部分内容即是 Ning Bao 兄的杰作。Ning Bao 兄深入浅出的讲述方式，令人受益匪浅。

感谢下列朋友：

orionline、zlowly、latteye、rainy、lyzzzz、nacre、buwen、winsphinx、lh、matri、quark、skyround、reus、tom、zhuqin 等等。他们为这份文档提出了宝贵的意见和建议，并指出了明显的错误。没有他们，就没有今天这份略显完善的文档。

这份文档肯定还有其他错误，恳请方家指正。

# 目 录

版权声明.....	2
致谢.....	3
内容简介.....	6
为什么要选择 Archlinux.....	7
安装 Archlinux 基本系统.....	7
使用光盘引导.....	7
启动 Archlinux 安装程序.....	8
选择安装源.....	9
准备硬盘.....	10
1. 对硬盘分区.....	11
2. 挂载文件系统.....	14
选择包.....	15
安装包.....	16
配置系统.....	17
1. /etc/rc.conf.....	18
2. /etc/fstab.....	19
3. /etc/locale.gen.....	19
4. 设置 root 密码.....	20
5. 设置 Pacman 镜像.....	20
安装引导程序.....	20
退出安装.....	21
新建用户.....	21
更新系统.....	22
安装 X.Org.....	23
为 NVIDIA 显卡安装驱动.....	23
生成 xorg.conf 配置文件.....	23
编辑 xorg.conf 配置文件.....	24
1. Module 部分.....	24
2. Files 部分.....	24
3. InputDevice 部分.....	24
4. Monitor 部分.....	24
5. Device 部分.....	24
6. Screen 部分.....	24
安装桌面环境.....	24
安装登录管理器.....	24
安装 GNOME 桌面环境.....	24
安装终端.....	25
安装中文字体.....	25
安装中文输入法.....	25
安装 ALSA.....	27
安装常用软件.....	27
网络浏览.....	27
图像编辑.....	27
办公处理.....	27

即时通讯.....	28
音影播放.....	28
下载工具.....	28
新闻阅读.....	28
邮件收发.....	29
图像查看.....	29
文本编辑.....	29
FTP 软件.....	29
光盘刻录.....	29
文档查看.....	29
其他工具.....	30
安装主题.....	30
安装 Compiz Fusion.....	32
准备配置文件.....	32
安装 Compiz Fusion.....	33
自动启动 Compiz Fusion.....	33
安装 Avant Window Navigator (AWN).....	34
Archlinux 的包管理工具 —— Pacman.....	35
Pacman 简介.....	35
配置 Pacman.....	35
使用 Pacman.....	36
1、更新系统.....	36
2、安装包.....	36
3、删除包.....	36
4、搜索包.....	36
5、其他用法.....	37
Pacman 的 GUI 前端.....	37
Archlinux 的灵魂——PKGBUILD、AUR 和 ABS.....	37
PKGBUILD 和 makepkg.....	37
AUR 和 ABS.....	40
Yaourt.....	44
Yaourt 简介.....	45
安装 Yaourt.....	45
Yaourt 实战.....	45
参考资源.....	46

## 内容简介

这份文档详细描述 [Archlinux](#) 2007.08-2 的安装过程，包括基本系统、X Window System、桌面环境、中文支持、常用软件等等，并全面阐释 PKGBUILD、Makepkg、AUR、ABS、Pacman、以及 Yaourt 的用法。通过此文，你将最终获得一个轻快、灵活、最新的完美 Linux 桌面。

有关本文档的后续修订及更新情况，可通过 <http://linuxtoy.org/archives/archlinux-guide.html> 获悉。

## 为什么要选择 Archlinux

我不是一个 Archlinux 老手，使用 Archlinux 我才不过几个月时间，但我已经喜欢上了这个很有个性的 Linux 发行版。为什么要选择 Archlinux？我基于以下理由：

- 简单。Archlinux 信奉的哲学是 KISS，即保持简单。安装和配置 Archlinux 比我预想的要容易。Archlinux 的文件系统结构布局清晰，让人一目了然。
- 轻快。Archlinux 为 i686 进行优化，无论是系统的启动，还是运行程序，都感觉比较轻快。
- 灵活。不象其他的 Linux 发行版，为你默认安装一些不想要的东西。Archlinux 给你一个最基本的系统，是在此基础上搭积木，还是要魔方，随你怎么玩。这体现了 Archlinux 的灵活。
- 保持最新。几乎每个人都有喜新厌旧的心理。Archlinux 总能尽快满足你及时获取最新的软件。
- Pacman 和 ABS。Pacman 是 Archlinux 的包管理工具，与 Apt-get 类似，使用同样简单。ABS 作为 Archlinux 的编译系统，使程序从源代码编译是如此容易。Archlinux 还包括 AUR，让 Archlinux 用户彼此分享各种流行的包。

## 安装 Archlinux 基本系统

看到 Archlinux 的优点，是不是有一种跃跃欲试想要马上安装的感觉？不要着急，在安装 Archlinux 之前，还有一些准备工作需要作：

- 备份数据。首先，你应该备份现有系统中的重要数据，以做到有备无患。
- 收集信息。有些硬件的信息应当提前进行收集，在后面的系统配置中将会用到。
- 选择映像。Archlinux 当前最新版本是 [2007.08-2](#)，包括 CORE 和 FTP 安装 ISO 映像。你需要根据自己的实际情况进行选择，本文将以 CORE 为例，从光盘进行安装。Archlinux CORE ISO 映像文件大小为 158 MB，下载后刻盘备用。

## 使用光盘引导

当使用 Archlinux 安装光盘引导后，将出现以下画面：

```
Copyright 2002 - 2007 Judd Vinet <jvinet@zeroflux.org>
Distributed under the GNU General Public License (GPL)

ISOLINUX BOOT
Creation Tool: 'mkbootcd' written by Tobias Powalowski <tpowa@archlinux.org>

INSTALLATION SYSTEM
Arch Linux Don't Panic
Kernel: 2.6.22-ARCH
Architecture: i686
Creation Date: Wed Oct  3 13:40:13 UTC 2007

Available boot options (no input will boot the install/rescue system):
- 'arch <any_other_boot_option>' to boot the install/rescue system.
- 'arch root=/dev/??? <any_other_boot_option>' to boot into an existing system.
- 'lowmem <any_other_boot_option>' to boot the 96MB RAM install system.
- 'lowmem root=/dev/??? <any_other_boot_option>' to boot into an existing
  96MB RAM system.
- If you have trouble with IDE drives, use the "ide-legacy" boot option.
- If your system hangs during the boot process, any combinations of the
  boot options noapic acpi=off pci=routeirq nosmp may be useful.
- 'memtest' to start the memory test program memtest86+.

boot: _
```

Archlinux 提供几种不同的引导选项，你可根据自己需要选择。一般情况下，按回车即可。

## 启动 Archlinux 安装程序

稍等片刻，Archlinux 即引导进入这个画面：



```
Virtual consoles 1-4 are active.
-----
To change virtual console use ALT + F(1-5 or 12)

Logging:
- vc5 is used for setup logging.
- vc12 is used for kernel logging.
Documentation:
- Documentation can be read by executing:
  zcat /arch/archdoc.txt.gz | less
Keymap:
- To change to a non-US keymap, type 'km' at the console.
Normal Setup:
- When you are ready, please run '/arch/setup' to install Arch Linux.
For Experts:
- Use '/arch/quickinst' to install and bypass the setup routine.
Hit ENTER to enter the bash shell ...

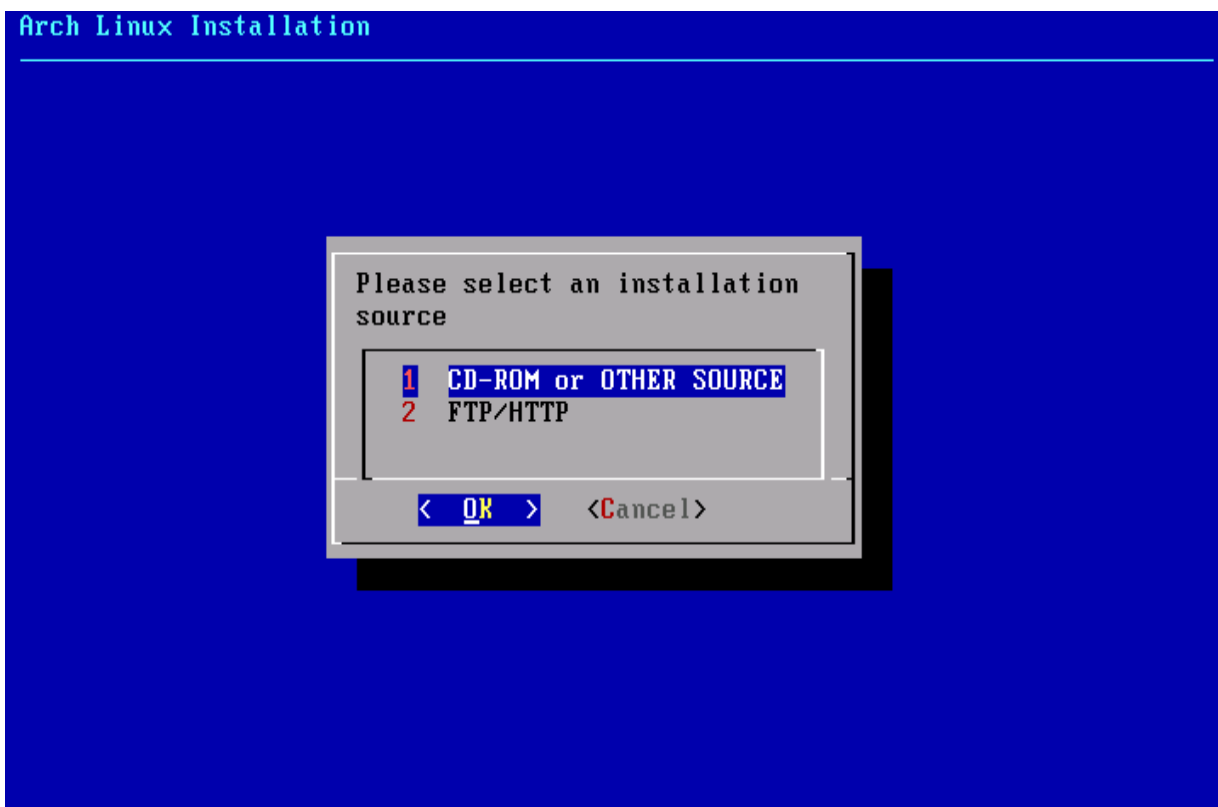
[Arch Linux: /]# /arch/setup_
```

该画面包括一些有用的安装信息，如安装日志的记录、文档的查看、键盘映射的更改等。我们直接在命令提示符后输入下列指令，以便启动 Archlinux 安装程序：

```
/arch/setup
```

## 选择安装源

在经过一段欢迎信息之后，我们将来到下一个画面：



因为我们是从光盘直接安装，所以保留默认选择即可。如果你是从 FTP/HTTP 安装，那么可以选择第二项。

## 准备硬盘

接着，我们会进入 Archlinux 的安装主菜单。首先是要求我们准备硬盘，即对硬盘分区和挂载文件系统。

## Arch Linux CDROM or OTHER SOURCE Installation

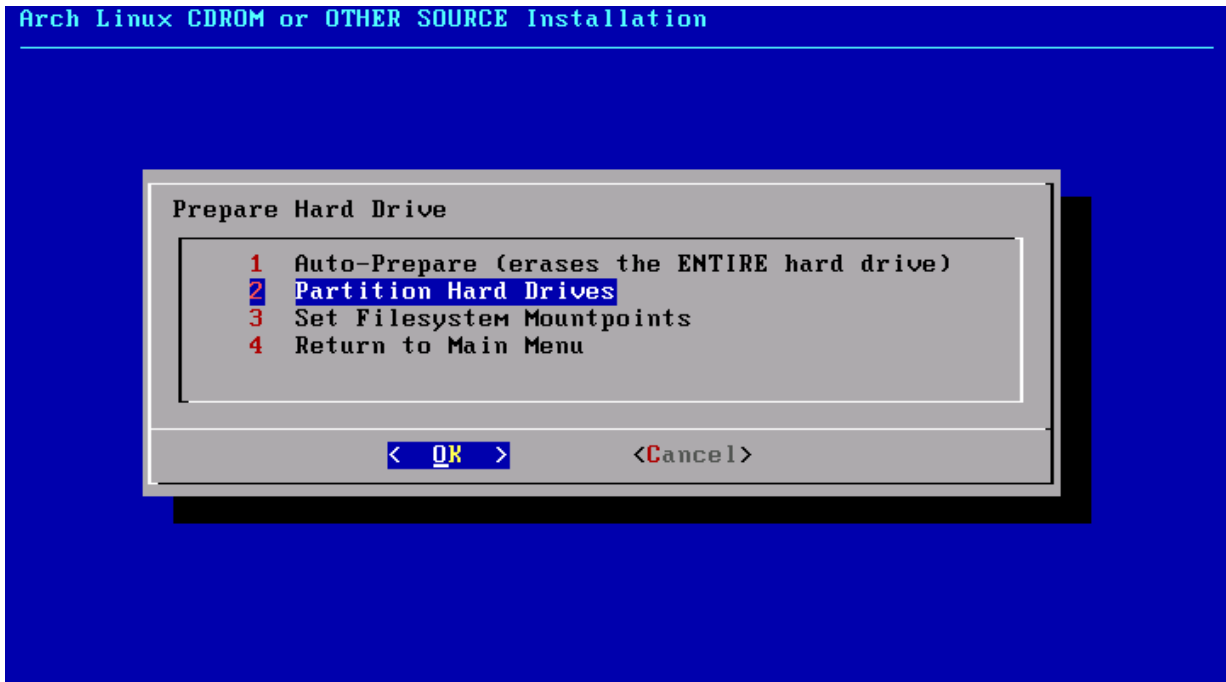


### 1. 对硬盘分区

如果你的硬盘上什么也没有，那么可以选择 Archlinux 安装程序的自动分区方案。不过，我们并不建议这样做。我们推荐在分区之前，根据自己的硬盘容量及实际需要来确定一个合理的分区方案。本文拟分四个区：

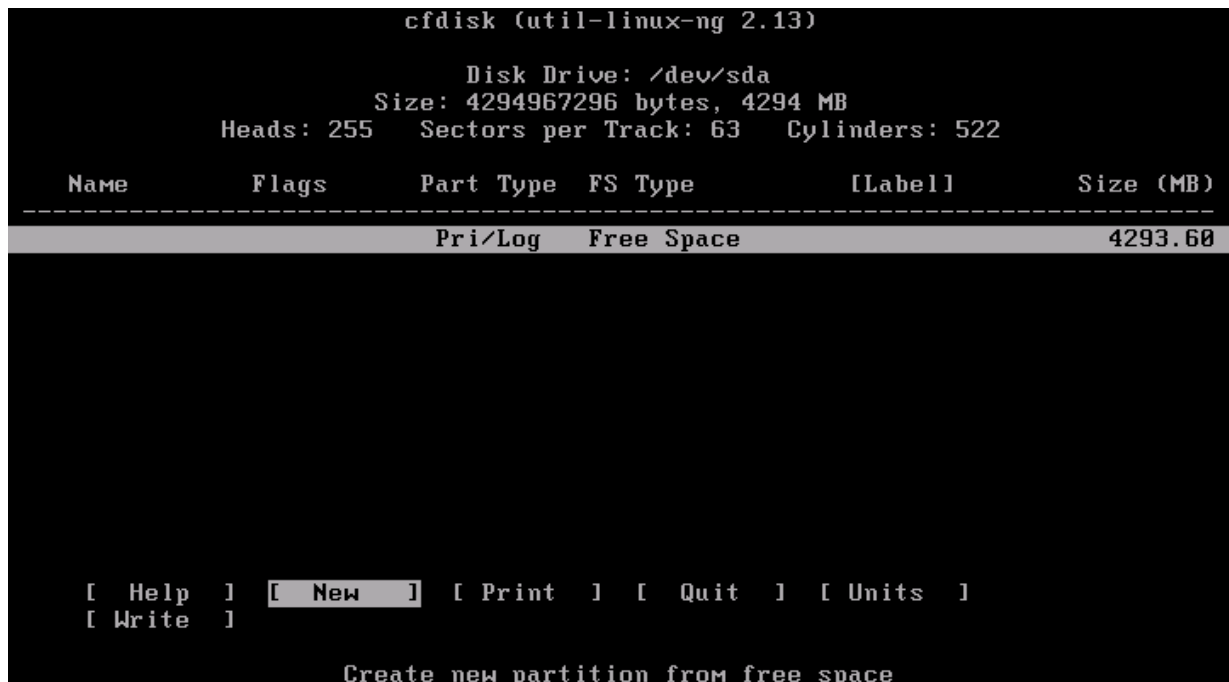
- /boot: 引导分区
- swap: 交换区
- /: 根分区
- /home: 用户目录区

现在，分区方案已经确定。因此，我们选择第二项进行手动分区。



Archlinux 安装程序会自动检测硬盘类型及容量，要求你选择分区的硬盘，并带你进入硬盘分区程序 cfdisk。

根据我们确定的分区方案，按 Tab 切换到 New 上，创建第一个分区 sda1。然后，依次建立余下的分区。



分区创立完成后，我们还需要做两件事：

1) 设置可引导标志。先选择 sda1，再切换到 Bootable 来完成。

```

cfdisk (util-linux-ng 2.13)

Disk Drive: /dev/sda
Size: 4294967296 bytes, 4294 MB
Heads: 255 Sectors per Track: 63 Cylinders: 522

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1      Boot      Primary   Linux        [Label]      32.91
sda2      Primary   Linux
sda3      Primary   Linux        3002.23
sda4      Primary   Linux        748.51

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition_

```

2) 保存创建的分区。通过切换到 Write 来写入分区。

```

cfdisk (util-linux-ng 2.13)

Disk Drive: /dev/sda
Size: 4294967296 bytes, 4294 MB
Heads: 255 Sectors per Track: 63 Cylinders: 522

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1      Boot      Primary   Linux        [Label]      32.91
sda2      Primary   Linux
sda3      Primary   Linux        3002.23
sda4      Primary   Linux        748.51

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Write partition table to disk (this might destroy data)_

```

最后，切换到 Quit 退出 cfdisk 分区程序。

```
cfdisk (util-linux-ng 2.13)

Disk Drive: /dev/sda
Size: 4294967296 bytes, 4294 MB
Heads: 255 Sectors per Track: 63 Cylinders: 522

-----
Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1      Boot       Primary   Linux         32.91
sda2      Primary   Linux     509.97
sda3      Primary   Linux     3002.23
sda4      Primary   Linux     748.51

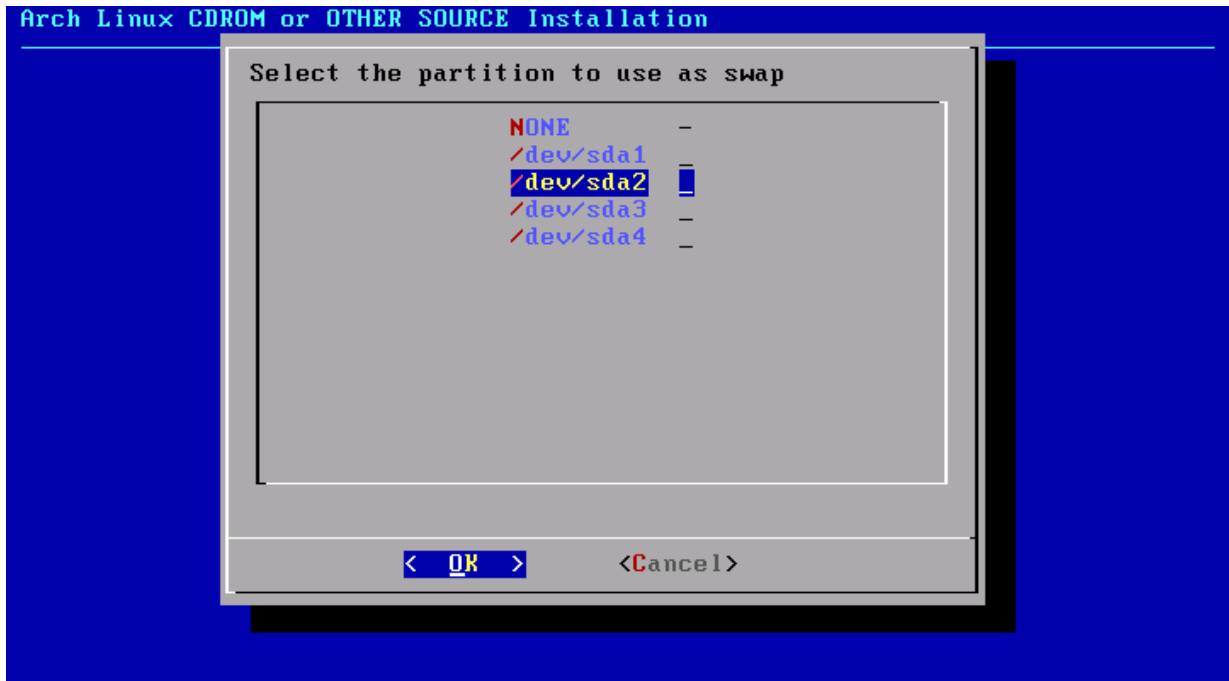
[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ]   [ Type ]  [ Units ] [ Write ]

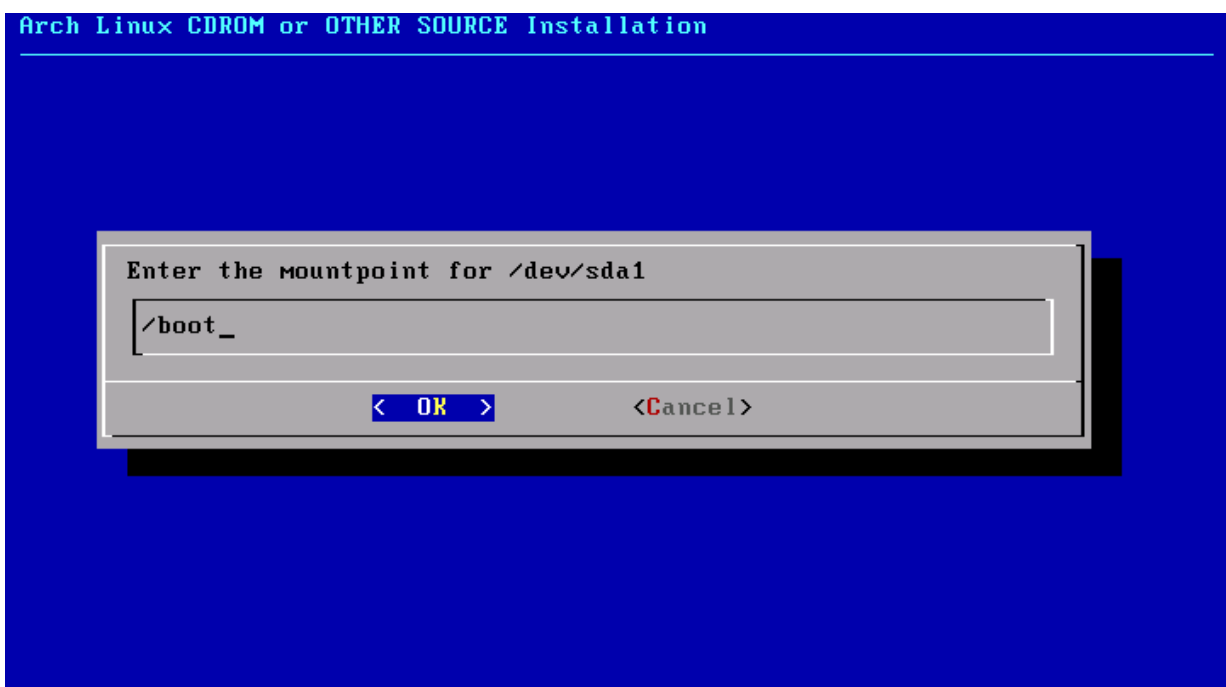
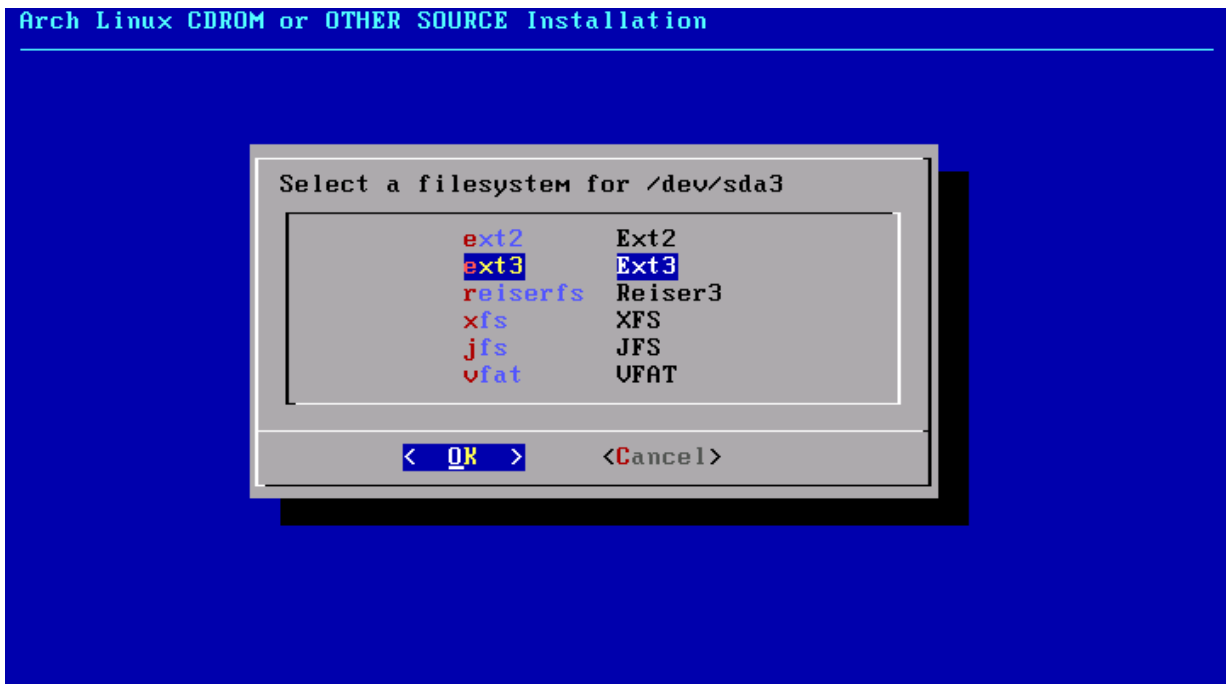
Quit program without writing partition table_
```

## 2. 挂载文件系统

在建立分区后，选择 DONE 条目返回准备硬盘画面。接着，我们选择 Set Filesystem Mountpoints 来将已建立的分区分挂载到文件系统上。

Archlinux 同样会自动检测硬盘容量，并首先要求你挂载 swap 区，我们选择 /dev/sda2。然后依次挂载 /、/boot 和 /home 区。与挂载 swap 区不同的是，挂载后几个分区会要求你选择一个文件系统类型。我们为 / 和 /home 选择 ext3，/boot 选择 ext2。你也可以选择其他的类型。另外，挂载 /boot 和 /home 时，需要自己输入挂载点，按原名输入即可。





完成后，返回 Archlinux 安装程序主菜单。

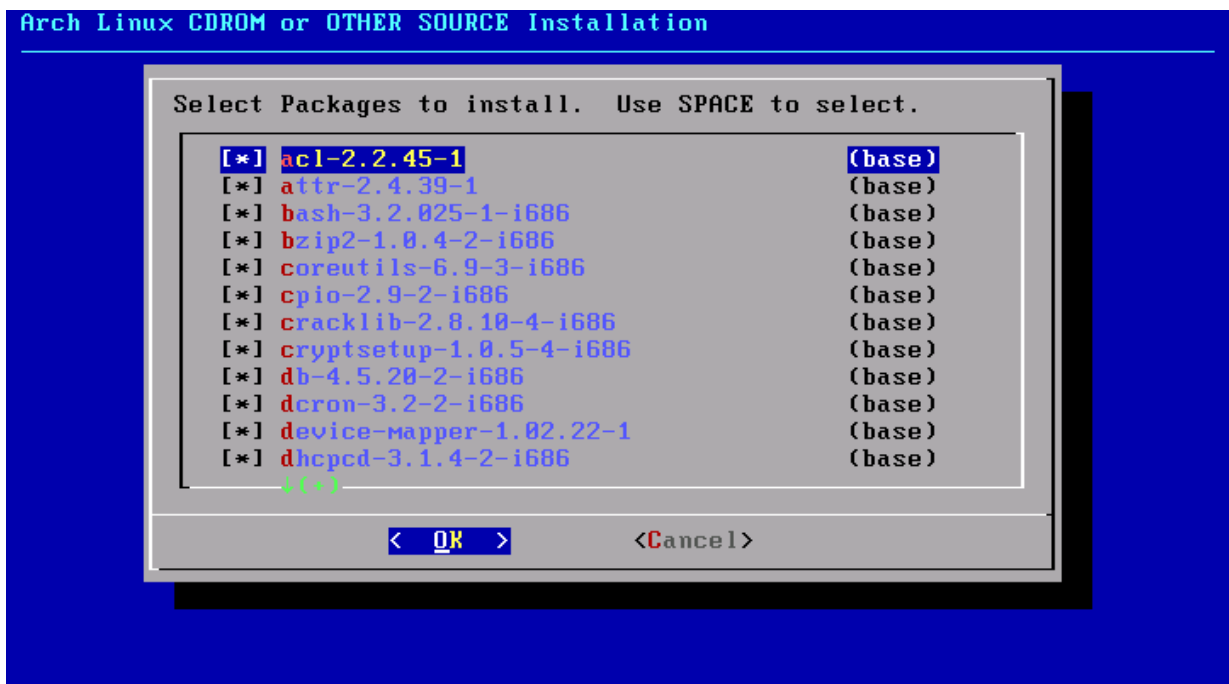
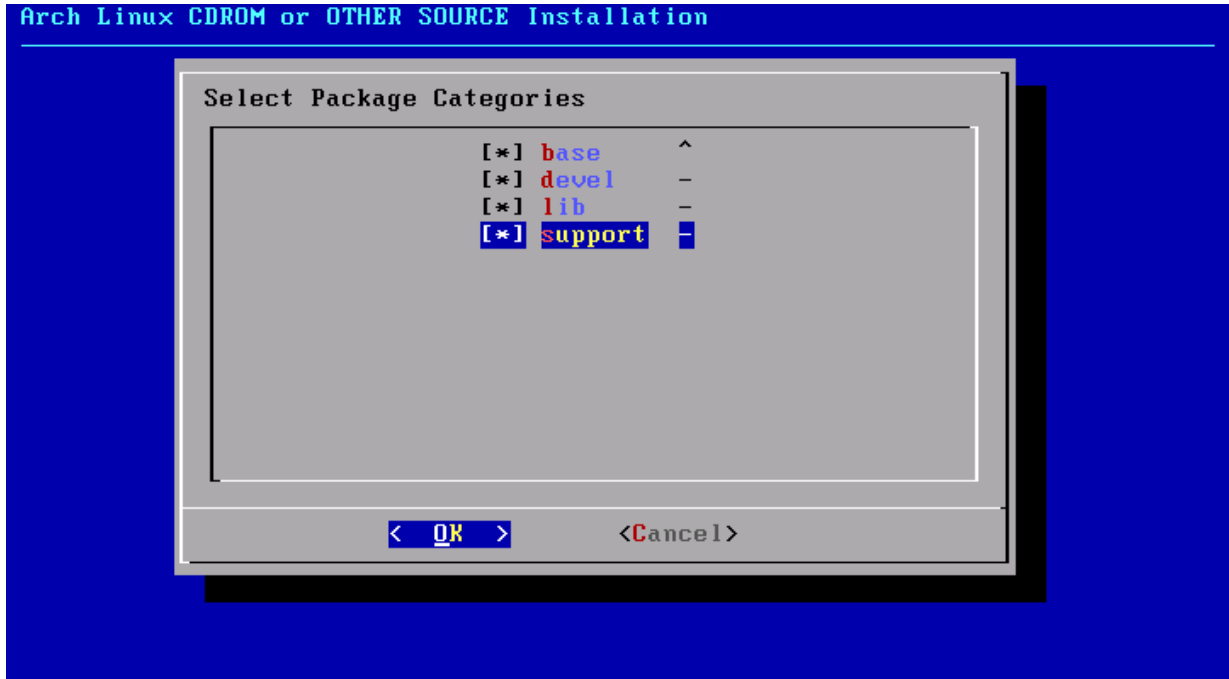
## 选择包

接下来，我们进行第五项任务：选择要安装的包。Archlinux 首先会要求选择安装介质，因为我们是从 CD-ROM 进行安装，所以保持默认。然后，我们选择 CD 驱动器，仍然默认。最后，选择包：

- base: Archlinux 中所包含的最基本的包。
- devel: 包含一些软件编译工具。
- lib: 包含应用程序所需的库文件。

→ support: 包含一些在网络和文件系统方面有用的包。

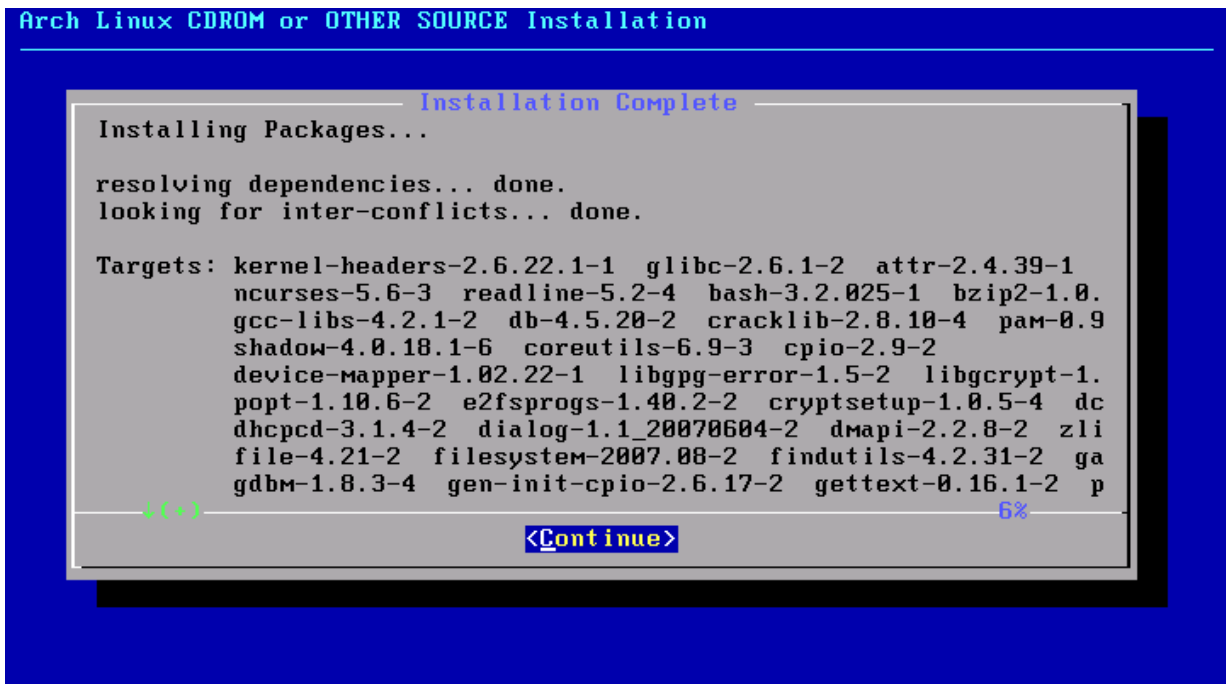
我们将选择以上四个分类的包，使用空格键可以完成选择过程。当 Archlinux 安装程序提示你是否默认选中所有的包时，按 Yes 后会进入具体的包选择画面。在此，你可以选择哪些包安装，哪些包不安装。完成后，按 OK 确认。



## 安装包

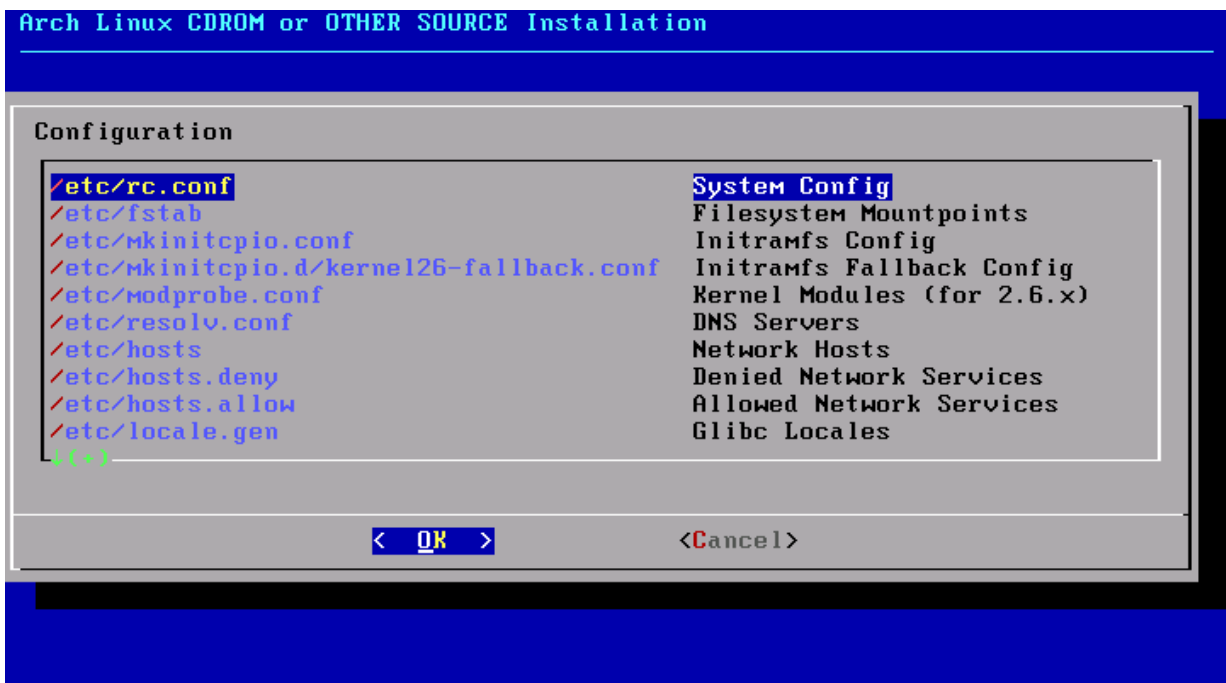
在安装包前，Archlinux 安装程序将提示你是否保存 Pacman 缓存，我们选择 No，即不保存 Pacman 缓存的包。Archlinux 将花一会儿时间来安装你所选择的包，你可以稍微休息一下。





## 配置系统

现在，我们将进入 Archlinux 安装过程中的一个重要环节，即配置系统文件。Archlinux 安装程序先会询问是否使用 hwdetect，按推荐选择 Yes，并回答是否需要支持从 usb、firewire、pcmcia 等设备引导。之后，我们需要选择所用的文本编辑器，可选 nano 和 vim，我们选择后者。然后，我们就到了如下的配置画面：



该画面包括需要配置的系统文件，主要有：

- /etc/rc.conf
- /etc/fstab
- /etc/mkinitcpio.conf

- /etc/mkinitcpio.d/kernel26-fallback.conf
- /etc/modprobe.conf
- /etc/resolv.conf
- /etc/hosts
- /etc/hosts.deny
- /etc/hosts.allow
- /etc/locale.gen

其中，有些配置文件可以暂时不管它，需要重点关注的是：

## 1. /etc/rc.conf

该文件中，你需要首先配置区域、时区、主机名、网络接口等内容。

### 1) LOCALE

LOCALE 定义系统的语言，默认为 `en_US.utf8`，即为英文。作为我们中文用户，可以将其设置为 `zh_CN.utf8`<sup>1</sup>。方法是，在进入 `vim` 编辑环境后，移动光标到该位置，按 `i` 进行编辑。

```
#
# /etc/rc.conf - Main Configuration for Arch Linux
#
#
# -----
# LOCALIZATION
# -----
#
# LOCALE: available languages can be listed with the 'locale -a' command
# HARDWARECLOCK: set to "UTC" or "localtime"
# TIMEZONE: timezones are found in /usr/share/zoneinfo
# KEYMAP: keymaps are found in /usr/share/kbd/keymaps
# CONSOLEFONT: found in /usr/share/kbd/consolefonts (only needed for non-US)
# CONSOLEMAP: found in /usr/share/kbd/consoletrans
# USECOLOR: use ANSI color sequences in startup messages
#
LOCALE="zh_CN.utf8"
HARDWARECLOCK="localtime"
TIMEZONE="Asia/Chongqing"
KEYMAP="us"
CONSOLEFONT=
CONSOLEMAP=
USECOLOR="yes"

20, 25                               Top
```

### 2) TIMEZONE

TIMEZONE 设置你所在的时区，由于我在中国西部，所以将其设置为 `Asia/Chongqing`。中文用户还可以设置成 `Asia/Shanghai`。由你所在的位置决定。另外，设置成 `PRC` 也可。

### 3) HOSTNAME

HOSTNAME 即设置主机名，我设置为 `linuxtoy`。

### 4) INTERFACES

INTERFACES 这部分设置你的网络接口参数。因为我是通过路由连接上网，所以我将 `eth0` 设置为 `dhcp`，即通过 DHCP 获得网络地址。你需要根据自己的实际情况来修改这些参数。

<sup>1</sup> 有些朋友建议设置成 `zh_CN.UTF-8`，以避免一些莫名其妙的问题。

在编辑完成后，按 `:wq` 保存并退出 vim 编辑环境。

## 2. /etc/fstab

该文件确定文件系统设置及挂载点，可以不用编辑，不过查看一下是否正确还是有必要的。

```
#
# /etc/fstab: static file system information
#
# <file system>      <dir>              <type>            <options>          <dump> <pass>
none                 /dev/pts            devpts             defaults            0       0
none                 /dev/shm            tmpfs              defaults            0       0

/dev/cdrom /mnt/cdrom  iso9660  ro,user,noauto,unhide  0       0
/dev/fd0 /mnt/fd0    vfat     user,noauto             0       0
/dev/sda1 /boot     ext2     defaults 0 1
/dev/sda2 swap      swap     defaults 0 0
/dev/sda3 /         ext3     defaults 0 1
/dev/sda4 /home    ext3     defaults 0 1
~
~
~
~
~
~
~
~
~
~
~/mnt/etc/fstab" 14L, 532C          1,1          All
```

## 3. /etc/locale.gen

这个文件包含系统所支持的区域及字符集。对我们中文用户来说，你需要去掉包括 `zh_CN` 这几行行首的注释符 `#`。

```
#ve_ZA UTF-8
#vi_UN.TCUN TCUN5712-1
#vi_UN UTF-8
#wa_BE ISO-8859-1
#wa_BE@euro ISO-8859-15
#wa_BE.UTF-8 UTF-8
#xh_ZA.UTF-8 UTF-8
#xh_ZA ISO-8859-1
#yi_US.UTF-8 UTF-8
#yi_US CP1255
zh_CN.GB18030 GB18030
zh_CN.GBK GBK
zh_CN.UTF-8 UTF-8
zh_CN GB2312
#zh_HK.UTF-8 UTF-8
#zh_HK BIG5-HKSCS
#zh_SG.UTF-8 UTF-8
#zh_SG.GBK GBK
#zh_SG GB2312
#zh_TW.EUC-TW EUC-TW
#zh_TW.UTF-8 UTF-8
#zh_TW BIG5
#zu_ZA.UTF-8 UTF-8
#zu_ZA ISO-8859-1

387,1 Bot
```

#### 4. 设置 root 密码

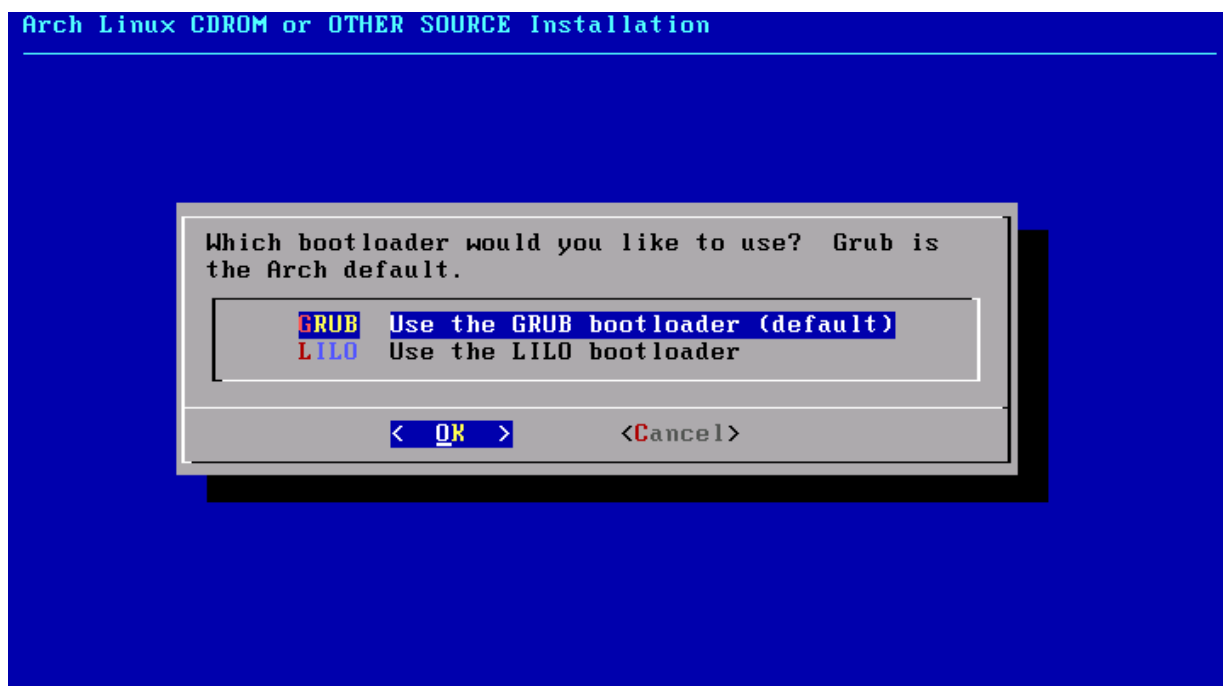
为 root 帐号设置一个密码，需要重复输入以便确认。

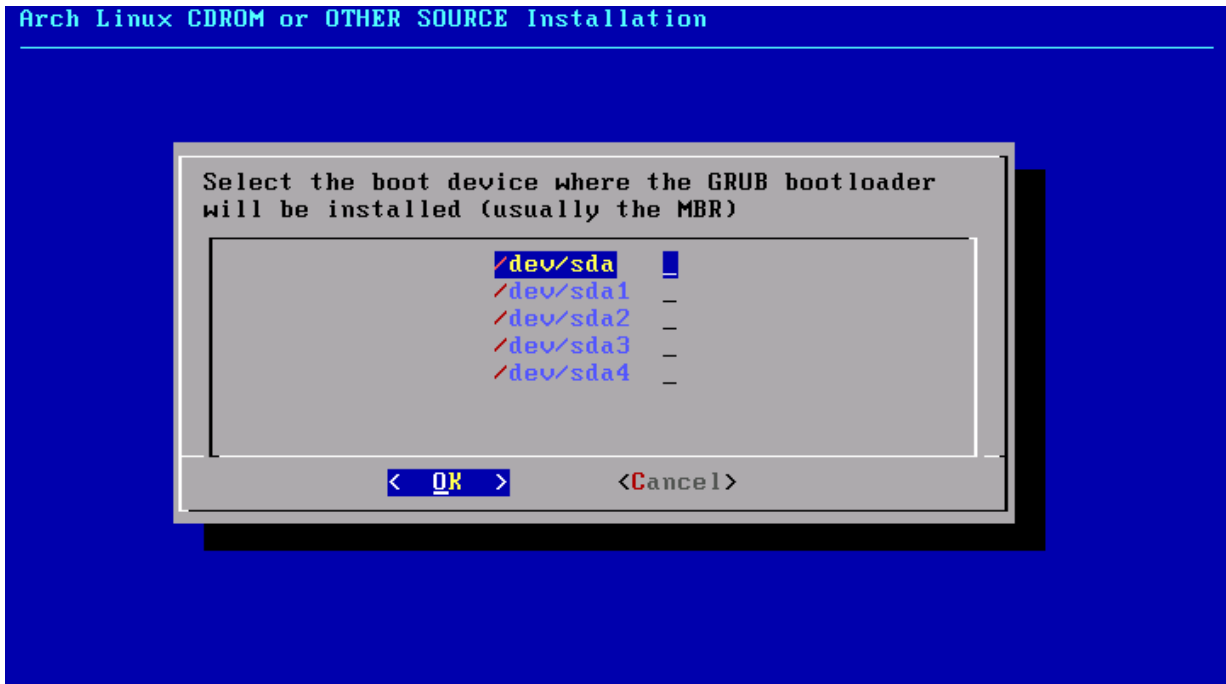
#### 5. 设置 Pacman 镜像

给 Pacman 包管理器设置镜像，选择较快的地址即可。

### 安装引导程序

系统配置完成后，回到主菜单。进入下一步，安装系统引导程序。我们选择 GRUB 条目。此时，Archlinux 安装程序让你查看 /boot/grub/menu.lst 的内容。接着，要求选择安装的位置，我们选 MBR，即主引导记录，第一项。

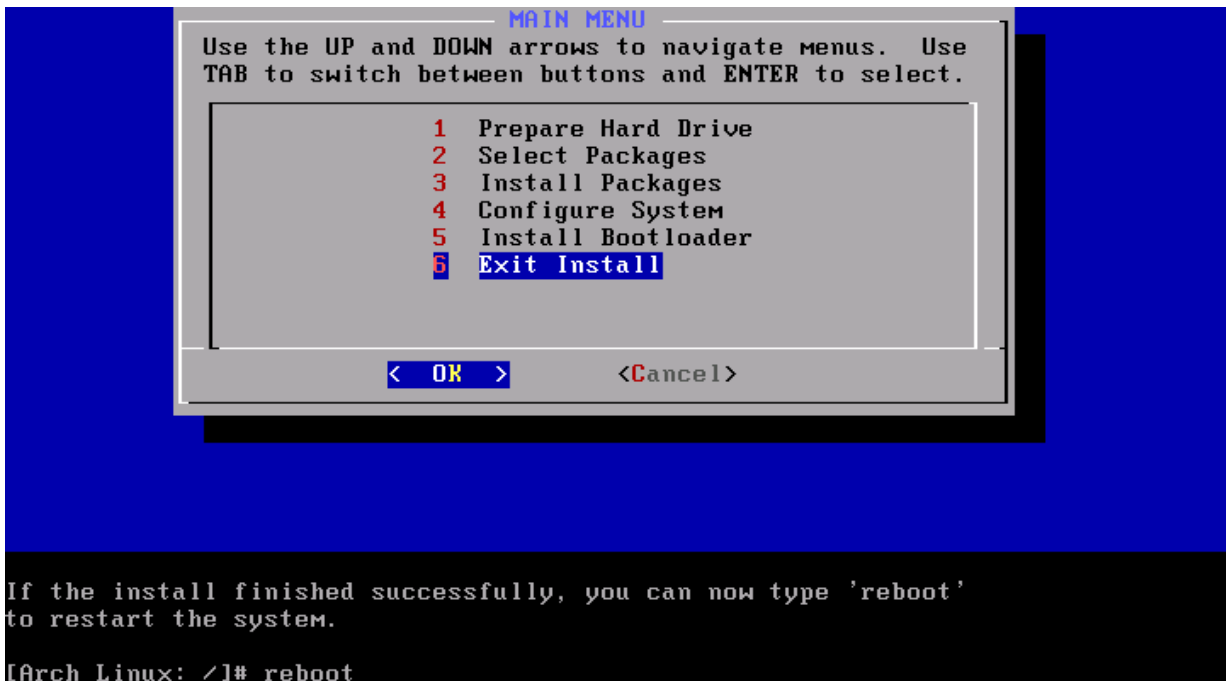




## 退出安装

还好，我们顺利完成了 Archlinux 安装程序的所有步骤。现在是时候退出了。按提示我们在命令行输入下列指令，以便重启系统：

`reboot`



## 新建用户

重启系统后，输入 root 帐号和密码登录系统。

```
GNU GRUB version 0.97 (638K lower / 260032K upper memory)

Arch Linux
Arch Linux Fallback

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.

The highlighted entry will be booted automatically in 2 seconds.
```

```
Arch Linux (Don't Panic) 2.6.22-ARCH (linuxtoy) (vc/1)
linuxtoy login: root
Password: _
```

首先要做的一件事情是，建立一个普通的帐号。可以通过以下命令完成：

```
useradd -m -s /bin/bash xiaodong
```

这将添加一个名为 xiaodong 的用户。接着，为该帐号设置密码：

```
passwd xiaodong
```

至此，Archlinux 的基本系统算是安装完成了。

## 更新系统

由于我采用宽带路由的上网方式，所以只需设置上面的 INTERFACES 参数为 dhcp 就可以了。如果你通过

DSL (PPPoE) 上网, 则需执行 `pppoe-setup` 脚本, 以便配置你的 ADSL 宽带连接。然后, 通过下列命令来启动连接或断开连接:

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

在我们进行后续的安装任务之前, 先把 Archlinux 更新到最新状态。你可以通过执行以下指令来完成 (需要 root 权限, 如果是普通用户, 可以先 su):

```
pacman -Syu
```

## 安装 X.Org

X.Org 是 X Window System 的开源实现。如果我们要在 Archlinux 中运行图形化的程序, 那么 X.Org 的安装是必不可少的。安装 X.Org 可以执行命令:

```
pacman -S xorg
```

该指令将为你安装 X.Org 所必需的包, 包括 X.Org 服务器、工具、字体、键盘驱动、鼠标驱动、显卡驱动等等。

值得注意的是, 默认的 X.Org 安装可能并没有包含你的显卡驱动程序。因此, 你需要单独为你的显卡安装驱动。你可以通过下列命令来进行搜索:

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

比如, 我在 VMWare 中测试时, 选择安装 `xf86-video-vmware`:

```
pacman -S xf86-video-vmware
```

## 为 NVIDIA 显卡安装驱动

作为 NVIDIA 和 ATI 显卡的用户, 为了获得更好的性能和 3D 加速功能, 可以安装它们的私有驱动。由于我本人是使用 NVIDIA 显卡, 所以本文就以安装 NVIDIA 的显卡驱动为例来说明。ATI 显卡用户可以参照 [Archlinux 维基的相关介绍](#)。

如果你象我一样使用较新的 NVIDIA 显卡, 那么可以执行以下指令来安装:

```
pacman -S nvidia
```

使用旧卡的朋友可以运行下列命令:

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

或者:

```
pacman -S nvidia-71xx
```

## 生成 xorg.conf 配置文件

在你使用 X.Org 前, 需要为其创建 `xorg.conf` 配置文件。手动创建比较费事, 我们选择相关工具自动生成, 然后再根据自己需要来编辑调整。

我们选择使用 `xorgconfig` 工具来生成 `xorg.conf` 文件。要启动该工具, 可以在命令提示符后执行:

```
xorgconfig
```

`xorgconfig` 是一个交互式的程序, 它会向你问一些有关鼠标、键盘、显示器、显卡等方面的问题。根据你的实际情况回答即可。

另外，使用 NVIDIA 显卡的朋友，也可以使用 `nvidia-xconfig`。

## 编辑 `xorg.conf` 配置文件

直接由工具所生成的 `xorg.conf` 配置文件并不能满足实际情况的需要。因此，我们还有必要对该文件进行适当的编辑。

### 1. Module 部分

Module 这部分定义默认载入的模块。比如，要载入 GLX 模块，可以去掉下列内容前面的注释符 #：

```
Load "glx"
```

### 2. Files 部分

在这部分中，你应当开启默认的字体搜索路径。

### 3. InputDevice 部分

InputDevice 部分可以让你配置鼠标、键盘等。

### 4. Monitor 部分

对显示器进行配置，注意输入正确的水平和垂直刷新率。必要时可以查阅显示器说明手册。

### 5. Device 部分

此部分配置显卡。NVIDIA 显卡用户可以加上以下选项，从而去掉 NVIDIA 烦人的标志。

```
Option "NoLogo" "true"
```

### 6. Screen 部分

Screen 部分可以设置显示器的色深和屏幕分辨率。

## 安装桌面环境

### 安装登录管理器

因为接下来我们将要安装 GNOME 桌面环境，所以我们选择 GDM。你也可以选择其他的登录管理器，比如 SLiM。要安装 GDM，可以执行命令：

```
pacman -S gdm
```

在 GDM 安装完成后，我们打开 `/etc/rc.conf` 配置文件，将 `gdm` 添加到 DAEMONS 中：

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

### 安装 GNOME 桌面环境

接着，我们执行下列指令来安装 GNOME 桌面环境：

```
pacman -S gnome
```



在安装 GNOME 时，你可以对其中所含的组件和程序进行有选择的安装。

## 安装终端

在登录到桌面环境之前，你应当安装一个合用的终端程序。我选择 GNOME Terminal。因此，在命令行执行：

```
pacman -S gnome-terminal
```

当然，你可以选择安装其他的终端程序，如 `rxvt-unicode`。

## 安装中文字体

对我们这些中文用户来说，由于要处理汉字，所以总是需要安装中文字体和中文输入法。在 Archlinux 中，你可以安装以下中文字体：

```
pacman -S ttf-arphic-uming ttf-arphic-ukai
```

同时，一些不错的英文字体也不妨安装上，比如：

```
pacman -S ttf-bitstream-vera
```

## 安装中文输入法

在 Archlinux 中包含 SCIM 和 Fcitx 中文输入法。我们选择安装后者：

```
pacman -S fcitx
```

当 Fcitx 安装完成后，将下列内容添加到你用户主目录中的 `.profile`<sup>2</sup> 中，以便让 Fcitx 自动启动：

```
export XMODIFIERS=@im=fcitx
```

```
export GTK_IM_MODULE=xim
```

```
export QT_IM_MODULE=xim
```

```
fcitx &
```

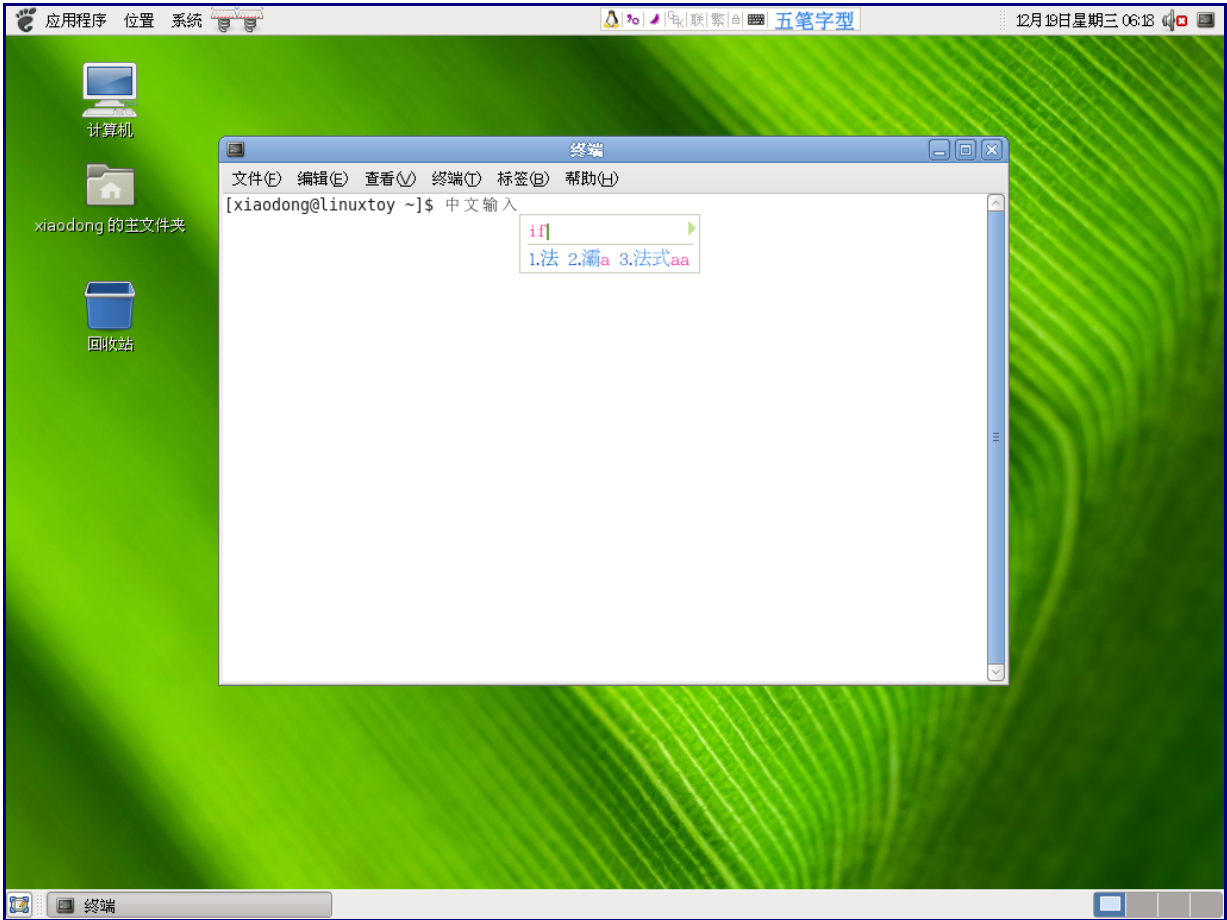
现在，重新启动系统。如果一切顺利的话，你将会看到如下登录画面：

---

2 有朋友建议加入 GNOME 会话中启动，以防止出现 SSH 登录时所产生的 2 类信息错误。



在输入你的用户名和密码后，即可进入 GNOME 桌面环境。



## 安装 ALSA

完美的 Linux 桌面怎么少得了美妙动听的音乐呢。现在，让我们为 Archlinux 安装 ALSA 吧。实际上，ALSA 已被包含到 2.6 版的内核中。因此，我们只需安装 ALSA 的相关工具即可。你可以在终端中输入下列指令：

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

接着，我们将当前用户添加到 audio 用户组，以便使用声卡设备 (注意把 xiaodong 换成你的用户名)：

```
gpasswd -a xiaodong audio
```

同时，将 alsa 添加到 /etc/rc.conf 配置文件的 DAEMONS 中：

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

值得注意的是，ALSA 默认是静音状态，你需要先打开音量。另外，也可点击“系统 → 首选项 → 音效”进行测试。

## 安装常用软件

### 网络浏览

我一直使用 Firefox 网络浏览器，所以执行以下指令来安装它：

```
pacman -S firefox firefox-i18n
```

该命令中的前者为 Firefox 主程序，后者为语言包。在安装完成后，可通过“应用程序 → Internet → Firefox”启动。

你也可以选择其他的网络浏览器，比如 Opera：

```
pacman -S opera
```

### 图像编辑

图像编辑软件首选 GIMP，要安装它可以执行命令：

```
pacman -S gimp
```

你可以通过“应用程序 → 图像 → GIMP”来运行 GIMP。

另外，矢量图形编辑软件可以使用 Inkscape：

```
pacman -S inkscape
```

命令行的可以装上 ImageMagick：

```
pacman -S imagemagick
```

用来捕获图像的 Scrot：

```
pacman -S scrot
```

也可以安装 GNOME Screenshot 这个图形化的工具：

```
pacman -S gnome-utils
```

### 办公处理

我们安装 OpenOffice.org 这套办公处理软件：

```
pacman -S openoffice-base openoffice-zh_cn
```

为了让 OpenOffice.org 运行于 GTK 2 模式，我们向 ~/.bashrc 添加如下内容：

```
export OOO_FORCE_DESKTOP=gnome
```

OpenOffice.org 安装完成后，可在“应用程序 → 办公”中找到相应的启动菜单条目。

## 即时通讯

要与朋友即时聊天，我们可以选用 Pidgin，它支持 Gtalk、MSN、QQ 等多种协议：

```
pacman -S pidgin
```

同时，Skype 也不可错过：

```
pacman -S skype
```

## 音影播放

音乐播放软件我选择 Quod Libet，你可以凭自己的喜好来安装：

```
pacman -S quodlibet
```

其他的音乐播放器包括 MPD、Audacious、Exaile、Amarok 等。

如果是看电影的话，MPlayer 不错，同时也加上浏览器插件和常用解码器：

```
pacman -S mplayer mplayer-plugin codecs gstreamer0.10-bad gstreamer0.10-ugly  
gstreamer0.10-ffmpeg gstreamer0.10-mad gstreamer0.10-mpeg2dec
```

如果需要 MPlayer 的前端，那么可以安装 SMPlayer：

```
pacman -S smplayer
```

当然，另一个选择 VLC 也挺好：

```
pacman -S vlc
```

## 下载工具

命令行下载工具，我们有 wget。此外，aria2 也不错，它支持断点续传和多线程下载：

```
pacman -S aria2
```

BitTorrent 下载工具，我们选用 Deluge：

```
pacman -S deluge
```

其他的包括 Azureus、rTorrent 等。

另外，我们把 aMule 也安装上：

```
pacman -S amule
```

为了能够让 aMule 直接从 Firefox 浏览器中处理 ed2k 链接，我们在 Firefox 的 about:config 中新建字符串 network.protocol-handler.app.ed2k，并将其设为 /usr/bin/ed2k。

## 新闻阅读

RSS 离线阅读软件，我们安装 Liferea：

```
pacman -S liferea
```

## 邮件收发

电子邮件客户端，可以选择 Thunderbird：

```
pacman -S thunderbird thunderbird-i18n
```

## 图像查看

我们安装一个轻快的图像查看软件 GQview：

```
pacman -S gqview
```

或者 GNOME 默认的 Eog：

```
pacman -S eog
```

## 文本编辑

如果要求简单的话，可以选用 Gedit：

```
pacman -S gedit
```

我们选择安装 Vim，要图形界面的话，可以加上 Gvim：

```
pacman -S vim
```

Emacs 迷们可以执行：

```
pacman -S emacs
```

Emacs CVS 版本可在 [community](#) 中找到。

## FTP 软件

我选择命令行的 Lftp：

```
pacman -S lftp
```

图形化的有 gFTP、FileZilla 等。

## 光盘刻录

我们选择 K3b，你可以通过以下指令安装：

```
pacman -S k3b
```

为了让当前用户能够使用光盘刻录设备，需要将其添加到 optical 用户组中 (请将 xiaodong 替换成你的用户名)：

```
gpasswd -a xiaodong optical
```

## 文档查看

查看 PDF 文档，可以安装 Evince：

```
pacman -S evince
```

CHM 文档，可以选用 KchmViewer。

## 其他工具

计算器:

```
pacman -S gcalctool
```

压缩/解压 rar、zip 等格式:

```
pacman -S unrar unzip
```

另外, 图形化的可以用 File-roller:

```
pacman -S file-roller
```

词典翻译, 我们安装 StarDict:

```
pacman -S stardict
```

同时, 词典文件需从 [StarDict 官方网站](#) 下载, 并释放到 /usr/share/stardict/dic/ 目录。

Java 支持:

```
pacman -S jre
```

Flash 插件:

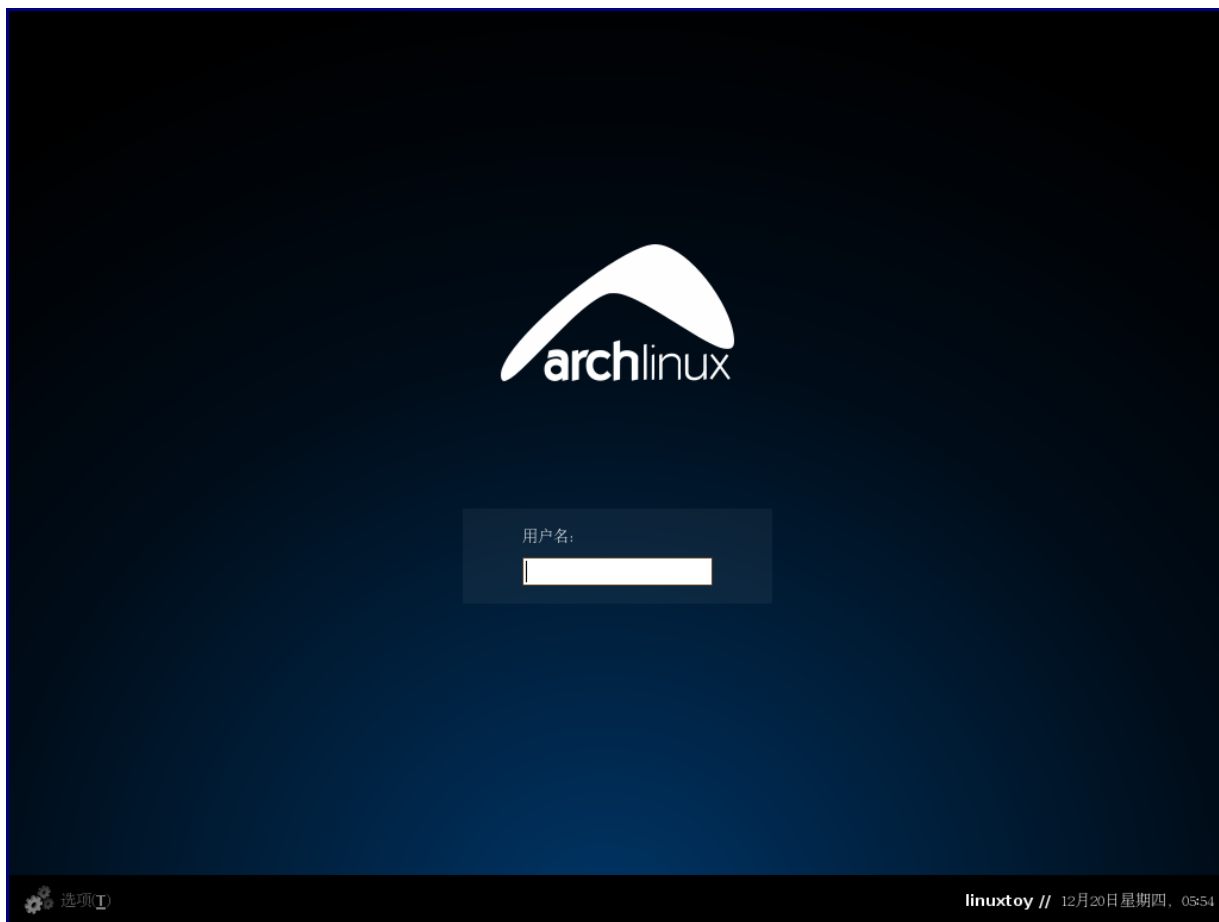
```
pacman -S flashplugin
```

## 安装主题

GDM 可以安装 [Arch Boomerang Underlight](#) 这套清爽的主题。下载后, 执行:

```
tar zxvf arch-boomerang-underlight-gdm.tar.gz  
mv arch-underlight-* /usr/share/gdm/themes/
```

然后在“系统 → 系统管理 → 登录窗口”中将样式更改为主题模式, 并选择已安装的主题即可。



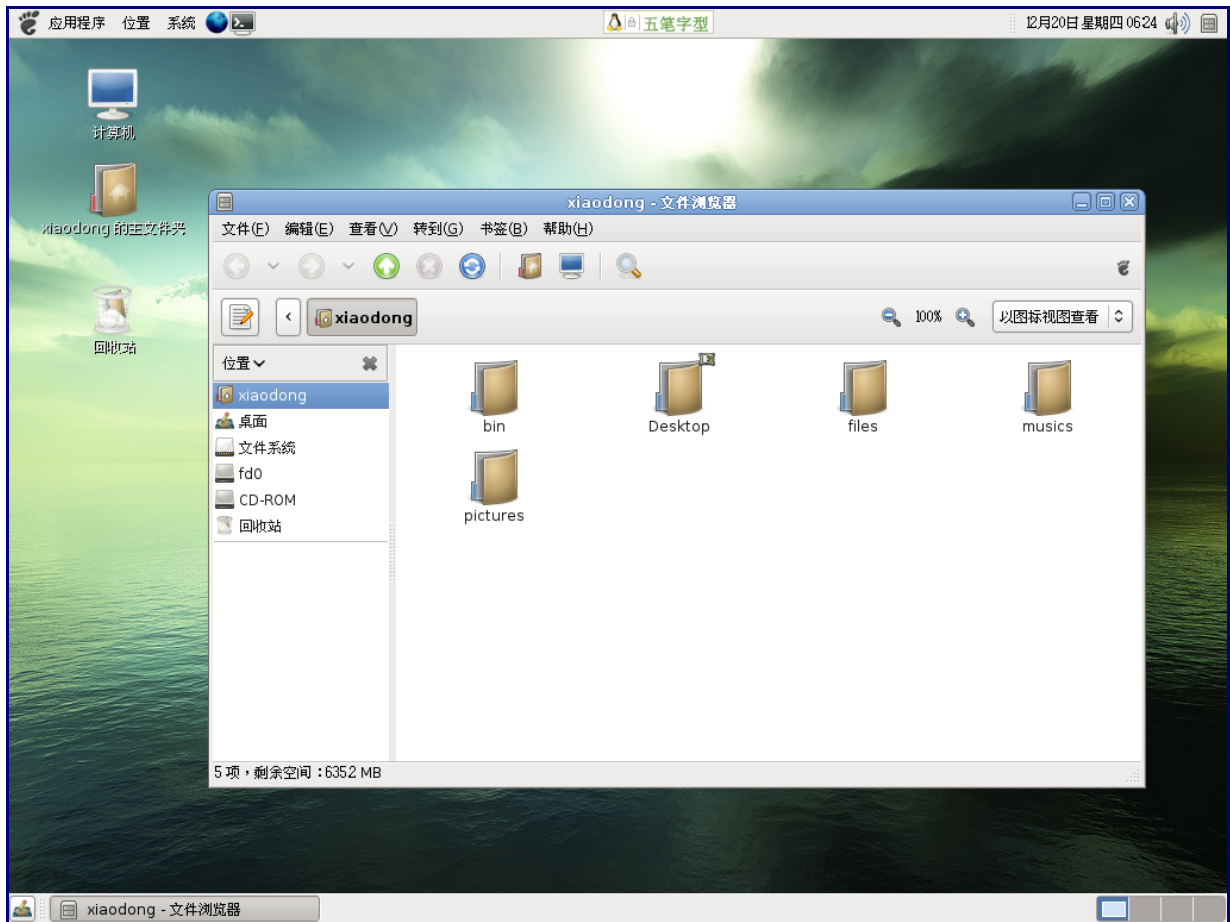
鼠标指针主题可以装上 [DMZ](#):

```
pacman -S xcursor-vanilla-dmz xcursor-vanilla-dmz-aa
```

安装完成后，可到“系统 → 首选项 → 外观 → 主题 → 自定义 → 指针”中选取。

一些不错的 GTK 主题引擎 Murrine、Rezlooks 等：

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



其他一些值得安装的软件包括：

- sudo：作为 root 帐号执行命令。
- libxft-lcd、cairo-lcd：使字体渲染更漂亮，主要针对 LCD 液晶显示器。
- bash-completion：对 Bash 自动完成功能的扩展。
- laptop-mode：对笔记本用户很有用。
- ntfs-3g：访问 NTFS 分区。

## 安装 Compiz Fusion

Compiz Fusion 合并自 Compiz 和 Beryl，它不仅将 Linux 桌面带入了 3D 环境，而且包含许多既丰富又绚丽的效果。

### 准备配置文件

要在 Archlinux 中安装 Compiz Fusion，首先确保 `/etc/X11/xorg.conf` 文件的正确配置。仍然以 NVIDIA 显卡为例：

1. Module 部分载入 GLX 模块：

```
Load "glx"
```

2. Device 部分添加下列选项：



```
Option "AddARGBGLXVisuals" "True"
```

以上是针对较新卡的配置，如果是使用旧卡的话，那么还应加上：

```
Option "RenderAccel" "true"
```

```
Option "AllowGLXWithComposite" "True"
```

3. 添加 Extensions 部分：

```
Section "Extensions"
```

```
Option "Composite" "Enable"
```

```
EndSection
```

使用其他显卡的朋友可以参考 [Archlinux 维基](#) 上面的相关文章。

## 安装 Compiz Fusion

现在，让我们来安装 Compiz Fusion，执行下列命令：

```
pacman -S compiz-fusion
```

这将安装 Compiz Core、Compiz Fusion 插件、Compiz Fusion 设置管理器、Emerald 及主题、Fusion Icon 等。

另外，GNOME 用户可以安装窗口装饰：

```
pacman -S compiz-fusion-gtk
```

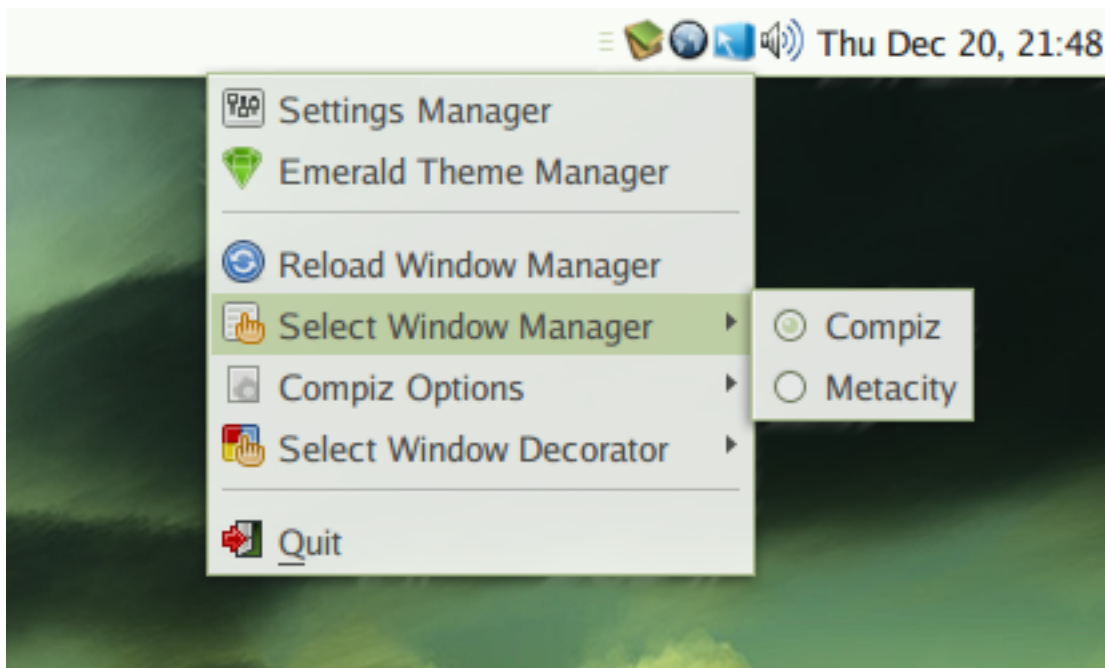
KDE 用户为：

```
pacman -S compiz-fusion-kde
```

## 自动启动 Compiz Fusion

要启动 Compiz Fusion，可以运行 Fusion Icon，它是一个系统托盘程序，通过它可以很方便的切换：

```
fusion-icon
```



从 Fusion Icon 中，我们可以将窗口管理器切换为 Compiz，窗口装饰切换为 Emerald。此外，该工具也可以调用 Compiz Fusion 设置管理器及 Emerald 主题管理器。具体的调整过程，你不妨亲自试试。

如果打算让 Compiz Fusion 自动启动，可以将 Fusion Icon 加入 GNOME 会话的启动程序组中。方法是，点击“系统 → 首选项 → 会话”，在启动程序标签中点击“添加”按钮，然后输入下列信息：

- 名称：填写 Compiz Fusion
- 命令：输入 fusion-icon
- 注释：填入 Compiz Fusion



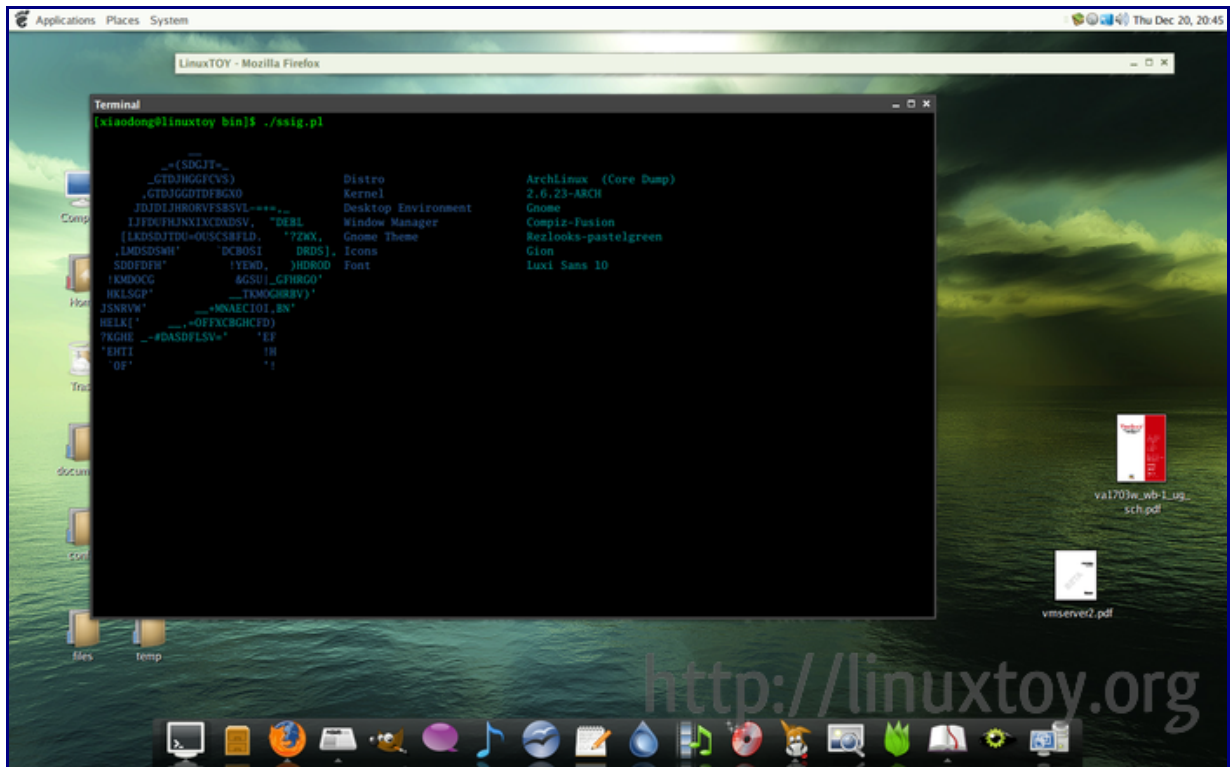
## 安装 Avant Window Navigator (AWN)

Avant Window Navigator 是一个漂亮的 Dock 程序，提供程序启动、窗口管理等，并包含许多插件。

你可以使用下列命令来安装 AWN：

```
pacman -S avant-window-navigator
```

AWN 可通过“应用程序 → 附件 → Avant Window Navigator”启动。自动启动的设置可以参考 Compiz Fusion 的做法。



## Archlinux 的包管理工具 —— Pacman

### Pacman 简介

[Pacman](#) 是 Archlinux 默认的包管理工具，该工具由 Archlinux 的创始人 Judd Vinet 所开发。截止写本文时为止，Pacman 的最新版本为 3.1.2，于 2008 年 2 月 20 日发布。使用 Pacman，你不仅可以更新 Archlinux 的整个系统，而且能够对包进行管理，包括安装、删除、升级等。同时，Pacman 也允许你搜索包和查看有关包的信息。此外，与 Apt 类似，Pacman 能够自动处理包的依赖。

### 配置 Pacman

Pacman 的配置文件为 `pacman.conf`，该文件位于 `/etc` 目录，使用文本编辑器就可对其进行编辑。以下为该文件的部分内容：

```
# /etc/pacman.conf
[options]
LogFile = /var/log/pacman.log
HoldPkg = pacman glibc
[core]
Include = /etc/pacman.d/mirrorlist
[extra]
Include = /etc/pacman.d/mirrorlist
[community]
```

```
Include = /etc/pacman.d/mirrorlist
[testing]
Include = /etc/pacman.d/mirrorlist
.....
[archlinuxfr]
Server = http://repo.archlinux.fr/i686
```

其中，行首具有 # 字符的为注释行。在 options 部分可以对 Pacman 进行设置。core、extra、community、testing 这些属于 Archlinux 的官方仓库，在其下的 mirrorlist 文件中可以定义服务器的镜像地址。如果你需要使用第三方的仓库，那么可以按如下的方式加入到 pacman.conf 文件中：

```
[仓库名称]: 如上面的 [archlinuxfr]
服务器地址: 如 Server = http://repo.archlinux.fr/i686
```

## 使用 Pacman

Pacman 是一个命令行工具，这意味着当你执行下面的命令时，必须在终端或控制台中进行。

### 1、更新系统

在 Archlinux 中，使用一条命令即可对整个系统进行更新：

```
pacman -Syu
```

如果你已经使用 pacman -Sy 将本地的包数据库与远程的仓库进行了同步，也可以只执行：

```
pacman -Su
```

### 2、安装包

→ pacman -S 包名：例如，执行 pacman -S firefox 将安装 Firefox。你也可以同时安装多个包，只需以空格分隔包名即可。

→ pacman -Sy 包名：与上面命令不同的是，该命令将在同步包数据库后再执行安装。

→ pacman -Sv 包名：在显示一些操作信息后执行安装。

→ pacman -U：安装本地包，其扩展名为 pkg.tar.gz。

### 3、删除包

→ pacman -R 包名：该命令将只删除包，不包含该包的依赖。

→ pacman -Rs 包名：在删除包的同时，也将删除其依赖。

→ pacman -Rd 包名：在删除包时不检查依赖。

### 4、搜索包

→ pacman -Ss 关键字：这将搜索含关键字的包。

→ pacman -Qi 包名：查看有关包的信息。

→ pacman -Ql 包名：列出该包的文件。

## 5、其他用法

- `pacman -Sw` 包名: 只下载包, 不安装。
- `pacman -Sc`: Pacman 下载的包文件位于 `/var/cache/pacman/pkg/` 目录。该命令将清理未安装的包文件。
- `pacman -Scc`: 清理所有的缓存文件。

关于 Pacman 更加详细的用法, 可以[阅读 Pacman 的手册页](#)。

## Pacman 的 GUI 前端

如果你对 Pacman 的图形化前端感兴趣, 可以参考 [Archlinux 的 Wiki 页面](#)。其中介绍了包括 Jacman、gtkPacman、Guzuta 等在内的 Pacman GUI 前端。

# Archlinux 的灵魂——PKGBUILD、AUR 和 ABS

## PKGBUILD 和 makepkg

其实, 我使用 Archlinux 的时间并不是很长。可是, 就是在这半年的时间里, 我感觉我学到了很多的东西, 对 GNU/Linux 有了更多的了解, 也在这个过程中深深地爱上了 Archlinux 这个发行版。

首先声明一下, 我不是一个程序员, 从来没有写过真正的 Code, 顶多是写一点 scripts, 或者做一些简单的网页什么的。和很多人一样, 我对计算机的接触是从 DOS/Windows 开始的。Windows 的图形界面很容易学。可是时间长了, Windows 就会运行的越来越慢, 我的硬盘上也就会有我根本不知道是什么的东西。而且, 在 Windows 上很多东西都是设定好的, 改变起来非常不容易。不要跟我提注册表, 那个东西实在是让我一头雾水。还有, 大部分 Windows 上的软件都不是自由软件。这意味着人们很难了解这些软件内部的情况, 你可能在毫无察觉的情况下就中了病毒或木马。

所以, 为了能够完全了解和控制我每天工作生活要用的计算机, 我开始学着使用 GNU/Linux。我一开始是用 Mandriva (那个时候还叫 Mandrake Linux)。后来是 Ubuntu (也包括其他一些 Debian 为基础的发行版)。Wow! 我一用上 Ubuntu 就有了完全不同的感觉。特别是非常好用的 apt-get, 加上庞大的自由软件库, 让我大开眼界。可是在使用 Ubuntu 一段时间以后, 我发现这个平台实在和 Windows 非常相似。Gnome 和 KDE 的界面都是在模仿 Windows。更糟糕的是, Ubuntu 会在一开始安装一些乱七八糟的东西, 大大影响了我的电脑的运行速度。我需要一点一点把我不用的东西去掉, 这个过程真的很不爽。我开始问自己为什么要放弃 Windows 呢?

Ubuntu 有一个很大很好的用户社区, 很多问题都可以在讨论区得到解决。然而, Ubuntu 的用户完全要依赖 Maintainer。我就有过这样的经历, 在发现一个 Bug 后得到很多其他人的确定, 可是 Maintainer 迟迟不作修改。还有, 也许有很多人都很想用一個比较新的软件, 但是大家都要等到有人能够而且愿意作 Maintainer 之后, 这个软件才会在 Repository 里出现。我或许能够在调试后自己从源代码编译, 可是我要如何和别人分享我的成果呢?

其实, 各种 Linux 发行版在本质上没有什么不一样。大家使用的软件都是要从源代码编译生成可以运行的二进制码。如果没有 rpm、apt-get 或者 pacman, 我们也是可以快乐生活的。只不过, 我们的生活会变得麻烦一些。如果要从源码安装一个软件, 我们通常是要做如下的步骤:

- `wget http://somewhere.org/source/package.tar.gz` (下载源代码)
- `tar xvzf package.tar.gz` (解压缩)
- `cd package` (进入源代码目录)
- `./configure` (设定)
- `make` (编译)

## make install (安装)

如果我们要像这样安装一个两个软件是没有什么问题的。但是如果我们要对付成百上千的软件/类库的话，这样的土办法是行不通的。于是出现了不同的 Linux 发行版，他们之间的区别只是在于如何管理成百上千的软件，特别是不同软件/类库之间互相依存的关系，也就是 `dependency` 的问题。

大多数 Linux 发行版都是以二进制包为基础的，这其中又分 Redhat（还有 SUSE、Fedora 等）、Debian（还有 Ubuntu、PCLinux 等）和 Slackware 阵营。为了解决管理大量软件包的问题，这些发行版采取了这样一个办法。他们找了一群大牛程序员来作 `Maintainer`，这些 `Maintainer` 负责把源代码编译成二进制码，加上一些控制信息（比如如何安装、`dependency` 等），然后一起打包放在服务器上。所以，最终用户根本不用接触源代码。如果你有兴趣的话，你可以抓一个 Debian 的 DEB 文件下来研究一下：

```
wget http://somewhere.org/package.deb
```

```
ar vx package.deb
```

你会发现你多了三个文件：

```
debian-binary
```

```
control.tar.gz
```

```
data.tar.gz
```

然后再用 `tar tzvf` 命令看一看 `control.tar.gz` 和 `data.tar.gz` 里面有什么东东，你就明白神奇的 `dpkg/apt-get` 是怎么一回事了。

二进制包固然是很方便，但是这种办法有一个很大的问题。那就是最终用户受到 `Maintainer` 很大的控制。比如说，我们并不知道 `Maintainer` 在编译的过程中是如何设定的（`./configure`）。如果我们要用不同的设定，就要自己从源代码从头开始。另外，如果某一个 `Maintainer` 心术不正，在二进制包里面加了木马程序，我们这些最终用户是很难察觉的。还有，设想一下如果某一个 `Maintainer` 外出休假了，那么你的软件也就不能及时更新了。

所以，也有一些发行版采取了完全不同的办法，这些发行版是以源代码为基础的。`Gentoo` 就是其中的代表。如果你用过 `Gentoo` 你就会知道 `ebuild` 文件。你如果有兴趣，可以从 <http://gentoo-package.com> 抓一个 `ebuild` 文件研究一下。你会明白 `Gentoo` 的用户其实从 `Gentoo` 得到的只有这些 `ebuild` 文件，在每一个 `ebuild` 文件里包含了安装使用一个软件需要的所有信息（从哪里下载源代码、如何编译、如何安装还有 `Dependency` 的问题等）。之后，`Gentoo` 的用户用 `emerge` 命令按照 `ebuild` 文件的指示编译、安装一个软件。这样做的好处是，`Gentoo` 的用户可以一目了然地了解每一个软件的编译、安装的过程。如果需要的话，`Gentoo` 的用户可以修改 `ebuild`，按照自己的需要编译一个软件。

我也用过 `Gentoo`。不过对于我这样的初学者，`Gentoo` 实在是太复杂了，有太多的参数要设定，`ebuild` 的编写也不是那么简单。还有，`Gentoo` 几乎不提供任何二进制包，所以绝大部分的软件都要从源代码编译，这是一个非常慢的过程。其实在大部分情况下，用户对一些软件的设定都是差不多的，没有必要让每一个 `Gentoo` 的用户都从头编译。所以，我需要找到一个发行版，既有 `Debian` 的易用性，又有 `Gentoo` 的灵活性。

我因此找到了 `Archlinux`。那么 `Archlinux` 又是如何解决从源代码到二进制码的问题呢？`Archlinux` 使用了 `makepkg` 这样一个工具。`makepkg` 会按照 `PKGBUILD` 文件生成一个二进制包。有些时候，`makepkg` 还需要 `install` 文件（主要用来显示提示信息、备份用户设置等）和其他的配置文件。那么 `PKGBUILD` 是什么呢？`PKGBUILD` 和 `Gentoo` 的 `ebuild` 一样，包含了安装使用一个软件需要的所有信息。下面是 `dwm`（一个非常非常简洁、高效的窗口管理器）的 `PKGBUILD` 文件：

```
# Contributor: Dag Odenhall <dag.odenhall@gmail.com>
```

```
# Contributor: Grigorios Bouzakis <grbzks@gmail.com> 是谁写了这个 PKGBUILD
```

```
pkgname=dwm 软件名称
```



```

pkgver=4.7
pkgrel=1  版本信息
pkgdesc="A dynamic window manager for X"
url="http://www.suckless.org/wiki/dwm"  软件说明和网站
arch=('i686' 'x86_64')  适用平台
license=('MIT')  版权
depends=('libx11')  Dependency
install=dwm.install  install  文件
source=(http://www.suckless.org/download/$pkgname-$pkgver.tar.gz
        config.h)  要下载的源文件
md5sums=('827b128514a3edb87e208e84fee0eb3f'
         '395e9a25f65605c4891e74c644b91530')  md5  验证码

build() {
    cd $startdir/src/$pkgname-$pkgver

    cp ../config.h .

    make X11INC=/usr/include/X11 X11LIB=/usr/lib/X11 || return 1
    make PREFIX=/usr DESTDIR=$startdir/pkg install || return 1

    install -m644 -D LICENSE $startdir/pkg/usr/share/licenses/$pkgname/LICENSE
&&
    install -m644 -D README $startdir/pkg/usr/share/doc/$pkgname/README
}  编译的过程

```

我们可以注意到在“编译的过程”这个部分，很多代码都和我们在 shell 里编译的命令一样。对 Archlinux 不要求用户学习太多新的东西，PKGBUILD 很容易理解，因为里面都是基本的 shell 命令。

好，我们把 PKGBUILD，dwm.install 和 config.h（dwm 比较特殊，config.h 包含所有的配置信息，所以在编译之前提供。其他的软件大多依靠外部的配置文件，像是 .bashrc 等）放在一个新的目录里之后。我们执行：

```
makepkg
```

之后，你会发现这个目录里出现了一些新的东西，包括：

```
dwm-4.7-1-x86_64.pkg.tar.gz
```

```
dwm-4.7.tar.gz
```

两个文件，还有两个目录

src

pkg

通过比较这些文件、目录里的内容和 PKGBUILD，你就会明白 makepkg 到底做了些什么：

1. 根据 source 里的内容下载了源代码文件 dwm-4.7.tar.gz；
2. 通过 md5 验证码确定下载的源代码文件和 PKGBUILD 的作者使用的是一致的；
3. 把源代码文件解压缩到 ./src/\$pkgname-\$pkgver（也就是 ./src/dwm-4.7）；
4. 按照 build() 里的内容编译源代码，并把编译好的内容放在 ./pkg 里；
5. 在 ./pkg 里加上其他的一些信息，包括 .PKGINFO 和 .INSTALL，也就是 dwm.install 的拷贝；
6. 把 ./pkg 里面的内容打包形成 dwm-4.7-1-x86\_64.pkg.tar.gz。

那么，我们有了这样一个 .pkg.tar.gz 这样一个二进制包之后，我们要如何安装呢？我们要使用这样一个命令：

```
pacman -U dwm-4.7-1-x86_64.pkg.tar.gz
```

这个命令又完成了那些事情呢？

1. 首先，二进制包被解压缩；
2. 按照 .INSTALL 的内容执行一定的命令；
3. 二进制包里面的内容会被拷贝到相应的目录（你注意到二进制包内的目录结构了吗？）；
4. 在 /var/lib/pacman/local 这个目录中建立 dwm-4.7-1 这样一个目录；
5. 这个目录里包含了四个文件 depends、desc、files 和 install；
6. depends 记录了 dependency，desc 是软件说明，files 记录了每一个安装到系统上的文件的路径，install 就是 .INSTALL 的拷贝。

从这以后，pacman 正是通过检查 /var/lib/pacman/local 里的内容来管理软件包的。比如说，在执行 pacman -R dwm 的过程中，pacman 首先在 /var/lib/pacman/local 找到了 dwm-4.7-1 这个目录，然后根据 files 的内容删除已安装的内容。Dependency 也是通过 depends 计算的。

OK！我已经解释了 PKGBUILD 的基本结构和 makepkg 的过程。基本上两步：从 PKGBUILD 到 .pkg.tar.gz 包，再从二进制包安装到系统。这样一种办法有很多好处。首先，PKGBUILD 非常方便用户交流。我的一个 PKGBUILD 如果编译成功了，就可以给别人用。PKGBUILD 的内容一目了然，不但有助于学习，也再不用担心木马的问题了。

另外，我通过一个小例子展现 Archlinux 的灵活性在哪里。比如，我要对 dwm 有自己的设置，也就是自己的 config.h，那我应该怎么做呢？我会做如下的事情：

1. 编辑 config.h，另存为 myconfig.h；
2. 编辑 PKGBUILD，把所有的 config.h 替换为 myconfig.h；
3. 把 pkgrel 变成 2。

之后通过 makepkg，我会得到一个文件 dwm-4.7-2-x86\_64.pkg.tar.gz，这个和原来的 dwm-4.7-1-x86\_64.pkg.tar.gz 可以区别开。我可以安装 dwm-4.7-2-x86\_64.pkg.tar.gz，如果有问题我还可以通过 pacman -U dwm-4.7-1-x86\_64.pkg.tar.gz 来安装原来的二进制包。我还可以用同样的办法生成一系列的 .pkg.tar.gz 包，这在软件的安装调试过程中非常有用。

好了，今天就讲到这里。有些人也许变得更疑惑了，因为在 Toy 的“[打造完美的 Linux 桌面 — Archlinux 2007.08-2](#)”系列中并没有提到 PKGBUILD 的问题，所有的软件都是通过 pacman -S 来安装的。

没关系，如果你理解了 makepkg 和 PKGBUILD，那么在我下一次谈到 AUR 和 ABS 之后，你就能完全明白了。

## AUR 和 ABS

我非常高兴看到我关于 PKGBUILD 和 Makepkg 的文章在 [LinuxTOY](#) 受到了欢迎。我想先针对一些读者的



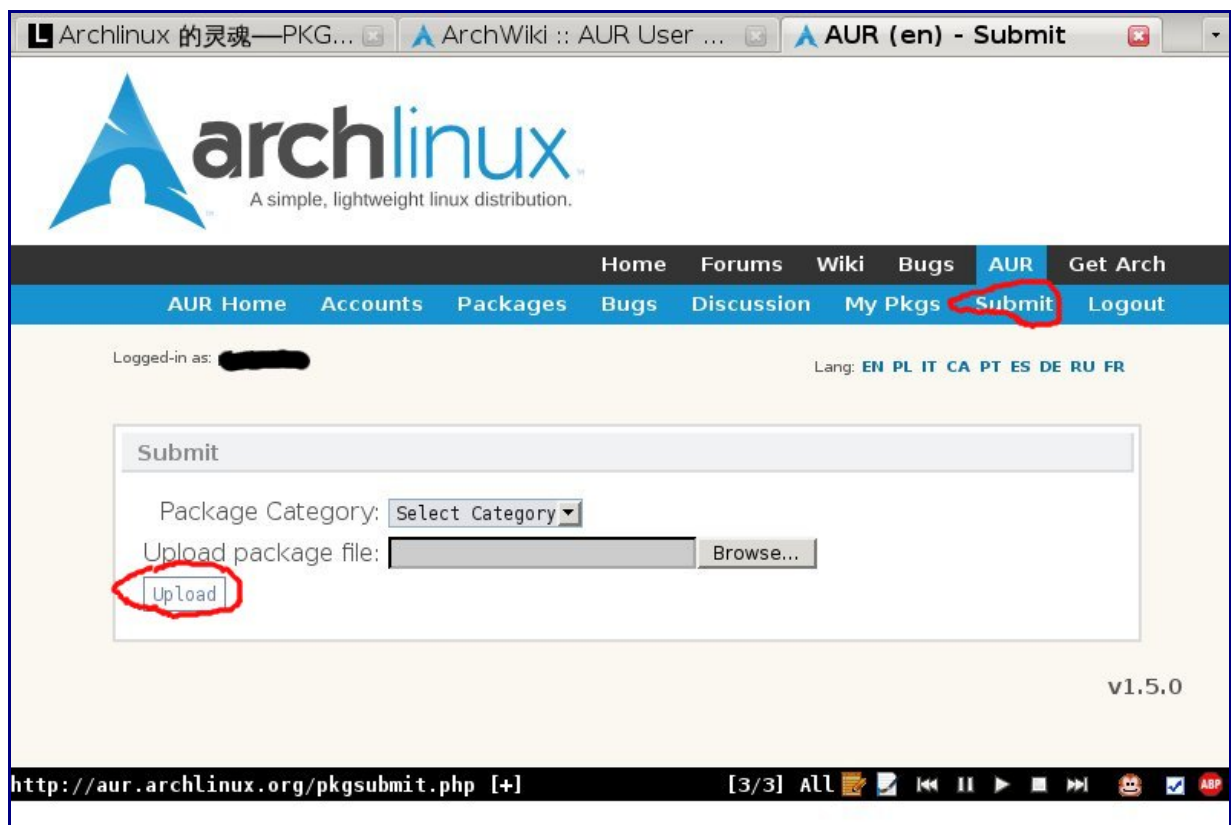
回复谈一点题外话。我先声明我一点也没有要诋毁 Debian 或 Gentoo 的意思，他们都是非常伟大的发行版，都有自己的特色。其实大多数的发行版都可以自己去定制，从而达到类似的目的。比如说，有的人提到 Gentoo 也有二进制包，比如像 Openoffice 这样的怪物。然而，我个人以为比较不同的发行版关键是要看它最核心的设计思想。比如说，Gentoo 根本就不是为了使用二进制包设计的。你要是都想用二进制包，就别费劲用 Gentoo 了。关于 Debian 阵营的发行版，我也想讲几句。正如一些朋友的回复所讲，DEB/apt-get 是非常好的管理工具，软件库也非常的大。我的笔记本现在还在用 elive，也是 Debian 的分支。我不喜欢 Debian 系列的发行版的原因不是它不能定制，而是他们非常依靠 Maintainer。我们可以自己做 DEB 包，然后呢？你的 DEB 包什么时候才能进入软件库呢？还有，只有你自己知道你的 DEB 是怎么做的，别人不能了解你编译打包的过程。Debian 本身打包的过程没有 Archlinux 的 PKGBUILD 来的简单明了。只要比较 Debian 的 Maintainer 手册和 Archlinux 的 Wiki 就可以看出这一点。

选择什么样的发行版完全是要看个人需要。我选择 Archlinux，是因为它当初设计的时候就是要满足“KISS Rule”，也就是“Keep It Simple, Stupid”。或者说像爱因斯坦讲得：“Everything should be made as simple as possible, but no simpler”。Archlinux 的所有配置基本都是非常相似的脚本，加上简单灵活的 PKGBUILD 和 pacman，其实关于 Archlinux 本身真的没有太多新的东西要学习。大家有兴趣可以看看“Arch Compared To Other Distros”。Archlinux 实际上是强迫用户从零开始自己定制自己的系统，在这个过程中也就真正了解了 Linux 本身。

好了，现在言归正传谈一谈 AUR 和 ABS。

AUR 是指 Archlinux User-community Repository，也就是 Archlinux 用户社区的软件库。我们现在回忆一下在 Archlinux 中我们把源代码变成可以运行的二进制码需要哪些文件。我们需要：PKGBUILD，可能还有 .install 文件，加上一些补丁和必要的配置文件（像 dwm 的 config.h）。这样就足够了！当你成功使用 PKGBUILD 编译安装了一个新软件之后就可以通过 AUR 和其他的人分享你的成果了。具体的步骤是：

1. tar -zcvf package.tar.gz package-dir 把 package-dir 中所有所需的文件打包（包括 PKGBUILD, .install, patch 和其他的 config 等）
2. 前往 <http://aur.archlinux.org> 选择“Submit”（参照下图），并把你的 package.tar.gz 上传



AUR 会自动根据你的 PKGBUILD 内容把你的 Package 加到 AUR 里来。就是这么简单! 那么有人会问: “别的用户要如何使用 AUR 呢?”

这个就更简单了, 我们还是用一张截图来解释:

1. 首先下载 “Tarball” (红色的圈圈), 这个 Tarball 和你上传的内容是一样的, 无非是 PKGBUILD 什么的;
2. `tar xvzf package.tar.gz` 然后解压缩;
3. 然后的步骤你应该知道了, 那就是 `makepkg` 还有 `pacman -U`。

不过需要提醒的是, 为了对自己负责, 你应该在编译之前读一下 PKGBUILD 和 `.install` 的内容, 确定里面没有恶意的代码。另外, 我建议你一般不要以 `root` 的身份进行 `makepkg`。其实, `makepkg` 本身也做了这样的限制, 你不加 `-asroot` 的选项是不能 `makepkg` 的。这是因为, `makepkg` 会执行 PKGBUILD 里 `build()` 部分的代码。想一想, 如果有人在 `build()` 这部分加了这么一句 “`rm -r /home`”, 你就死定了!

如果你注意到, 在每个 AUR Package 的网页上都有投票 (绿色的圈圈) 和回复的功能 (蓝色的圈圈)。这些是帮助用户反馈意见的。

The screenshot shows the ArchLinux AUR (Arch User Repository) page for the package 'ccal'. The page is in English and shows the package details for 'ccal 2.4-1'. The maintainer is 'sssslang', and the package has 5 votes. The package is licensed under GPL and was first submitted on Tue, 06 Mar 2007 05:24:47 +0000. The package is available as a tarball and can be built using PKGBUILD. The dependencies are listed as 'gcc', and the source is 'http://thunder.cwru.edu/ccal'. The page also includes an 'Actions' section with 'Vote', 'Notify', and 'Flag Out-of-date' buttons, and a 'Comments' section with an 'Add Comment' button and a comment by 'sssslang' on 20070306 [05:24:47].

Archlinux 的灵魂—PK... ArchWiki :: AUR User ... AUR (en) - ccal

**archlinux**  
A simple, lightweight linux distribution.

Home Forums Wiki Bugs **AUR** Get Arch

AUR Home Accounts Packages Bugs Discussion My Pkgs Submit Logout

Logged-in as: [redacted] Lang: EN PL IT CA PT ES DE RU FR

**Package Details**

ccal 2.4-1  
<http://thunder.cwru.edu/ccal/ccal.htm>  
 The ccal utility is a simple-to-use command line program which writes a Gregorian calendar together with Chinese calendar to standard output.

unsupported :: i18n  
 Maintainer: [sssslang](#)  
 Votes: 5  
 License: GPL

Last Updated: (unknown)  
 First Submitted: Tue, 06 Mar 2007 05:24:47 +0000

**Tarball** :: Files :: PKGBUILD

Dependencies	Required by	Sources
<a href="#">gcc</a>	None	<a href="http://thunder.cwru.edu/ccal">http://thunder.cwru.edu/ccal</a>

**Actions**

**Comments**

Comment by: [sssslang](#) on 20070306 [05:24:47]  
 The ccal utility is a simple-to-use command line program which writes a Gregorian calendar together with Chinese calendar to standard output.

<http://aur.archlinux.org/packages.php?ID=9686> [3/3] Top

有些性急的朋友可能要问这个 AUR 和 pacman 取得的二进制包有什么联系？你应该记住，只要是 Archlinux，所有的东西一定是从 PKGBUILD 开始的。你通过 pacman 得到的二进制包也是从 PKGBUILD 编译而成的。在你的 /etc/pacman.conf 有很多 Repository 的设置，其中的 [core] 和 [extra] 是由 Archlinux 的核心成员维护的，这些软件库里的软件由于特别重要，每个人都要用，所以 Archlinux 的开发人员把二进制包提前做好，

你可以通过 `pacman` 取得了。然而，和 Debian 的 Maintainer 机制不一样，最终的用户可以很容易的了解这些软件的编译过程。如果需要的话，最终用户可以改变设置，重新编译这些软件。最典型的情形就是自己编译 kernel 的时候。这要通过 ABS 来解释清楚。ABS，也就是 Archlinux Build System。首先我们要安装 ABS：

1. `pacman -S abs`
2. `vi /etc/abs/abs.conf` 编辑 ABS 的配置文件；
3. 你会看到这样一行 `SUPFILES=(core extra !unstable community !testing)`，把你需要的 Repo 之前的 ! 去掉；
4. 然后以 root 身份执行 `abs`

之后又要如何使用 ABS 呢？ABS 所作的事情无非是把所有 Repo 里的软件的 PKGBUILD 下载到本地的硬盘中。这些 PKGBUILD 都放在了 `/var/abs` 中。你能通过 `pacman` 直接安装的二进制包其实也都是按照 ABS 的内容编译的。下面我还是用 `dwm` 的例子解释 ABS 的使用：

1. `su`
2. `cd /var/abs` 你可以看到这个目录里有 `core`，`extra`，`community` 三个子目录，正如 `abs.conf` 中的设定；
3. `mkdir local` 建立一个 `local` 目录，用来放你自己需要的软件的 PKGBUILD
4. `chown username:usergroup ./local makepkg` 要以非 root 身份进行
5. `exit` 退出 `su`
6. `cd local`
7. `cp -r ../community/x11/dwm ./` 从 ABS 中拷贝 `dwm` 的内容
8. `cd dwm`

下面不用我说了，你在这个目录里可以看到三个文件 `PKGBUILD`、`dwm.install`、`config.h`。你于是可以用 `makepkg` 和 `pacman -U` 来按照自己喜欢的方式安装 `dwm`。

OK，你实际已经清楚了解了 ABS 和 `pacman` 的关系，那么 AUR 又和 ABS 还有 `pacman` 有什么联系呢？说的直白一点，你上传到 AUR 的 PKGBUILD 要足够“有品”才能直接通过 `pacman` 使用。对于“有品”，我是这样定义的。你的 PKGBUILD 要有很多人用（很多人投票），没有恶意代码，没有太多的 Bug.....而判定你的 PKGBUILD 够不够“有品”的人是一些叫作 TU (Trusted User) 的人。这些人的工作是检查 AUR，关注那些特别受欢迎的 PKGBUILD。之后，他会仔细检查，确定这些 PKGBUILD 是不是安全。然后，他们会给这些 PKGBUILD 打上安全的标签，并且把这些 PKGBUILD 从 `unsupported`（我们上传的 PKGBUILD 一开始都是在 `unsupported` 中）移到 `community` 的 Repo 中。

在 `community repo` 里面的 PKGBUILD 会提前编译好，如果你在 `/etc/pacman.conf` 中开了 `community repo`，你就可以直接使用这些软件的二进制包了。也许有一天，你当初上传的 PKGBUILD 变得特别重要，这个软件可能被移到 `testing`，`extra` 或者 `core` 的 `repo` 中。补充一点，`testing repo` 里面一般是需要测试，又准备放到 `core` 或者 `extra` 中的软件。

Archlinux 就是这样，非常灵活。既有 `pacman` 这样好的二进制包管理工具，又有 ABS 和 AUR 这样方便的源代码服务。通过 ABS，你可以完全控制你自己的系统到底是如何建立的。如果在 `pacman -Ss` 的时候找不到一个软件，你可以到 AUR 去找，如果还是找不到，为什么不自己试着从源代码开始，写一个 PKGBUILD 然后放到 AUR 中和别人分享呢？

说到这里，我希望我已经把 Archlinux 最核心的东西讲明白了。有些人说我的文章写得比 wiki 里的文章清楚。其实，我写的东西只是在顺序上不一样。我是从 PKGBUILD 开始讲到 AUR 和 ABS，再到 `pacman`。这个顺序和 Archlinux 实际的开发过程是一致的，所以逻辑上容易理解很多。如果你从 `pacman` 入手反过来读，你可能就完全错过了理解 Archlinux 核心概念的机会。

Toy 会很快再写一篇关于 `yaourt` 的文章，`yaourt` 是一个把 `pacman` 和 AUR 结合起来的很好用的工具。有了 `yaourt`，你不需要去 AUR 网站也能在 shell 下直接取得 AUR 的 package，还方便了投票的过程。

## Yaourt

就一般情况而言，当 Archlinux 用户需要使用 AUR 中的包时，往往会执行到 [AUR 官方网站](#) 查找所需的包、

下载该包的 Tarball 文件、在命令行下对 Tarball 文件解压、通过 `makepkg` 编译打包、最后使用 `pacman` 安装这样一个过程。仔细打量这个过程，你是否觉得稍微有些繁琐呢？有解决的方案吗？回答是肯定的。这就是我们今天将要介绍的主角——[Yaourt](#)。

## Yaourt 简介

Yaourt 是一个由 Julien Mischkowitz 所编写的 Bash 脚本，它是将 Pacman 与 AUR 这两者相结合的绝佳工具。通过 Yaourt 安装 AUR 中的包十分方便，它不仅简化了上述繁琐的过程，而且把这一过程半自动化，使用者只需在它的交互模式中简单的回答几个问题即可。此外，Yaourt 支持将结果以鲜亮的颜色输出，非常抢眼。

## 安装 Yaourt

除了在 Archlinux 的 AUR 中可以找到 [Yaourt](#) 外，archlinuxfr 这个源中也包含 Yaourt。我们采用后者来安装 Yaourt。首先，将下列内容添加到 `/etc/pacman.conf` 文件：

```
[archlinuxfr]
Server = http://repo.archlinux.fr/i686
```

如果你的系统是 64 位，那么可以使用：

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

接着，我们可以执行下面的命令来安装 Yaourt：

```
pacman -Sy yaourt
```

另外，我们将 `aurvote` 和 `customizepkg` 这两个包也装上，前者用于对喜欢的包投票，而后者是定制 PKGBUILD 所需的：

```
pacman -S aurvote customizepkg
```

同时，你需要为 `aurvote` 建立一个配置文件 `.aurvote` (位于 `~/` 目录下)：

```
user=你的 AUR 帐号
```

```
pass=该帐号的密码
```

如果你没有 AUR 帐号，可到 <http://aur.archlinux.org/account.php> 注册一个。

## Yaourt 实战

为了说明 Yaourt 的使用，我们以一个实例来进行。譬如，我对 [Phatch](#) 这个批量图片处理程序非常喜欢，我希望在 Archlinux 中安装它。首先，我们来看一下，在 Archlinux 中是否存在 Phatch：

```
yaourt phatch
```

Yaourt 在搜索后返回如下结果：

```
1 aur/phatch 0.1.bzr435-1
```

```
Phatch is a simple to use cross-platform GUI Photo Batch Processor.
```

```
==> Enter n° (separated by blanks, or a range) of packages to be installed
```

```
Example: '1 6 7 8 9' or '1 6-9'
```

```
==> -----
```

从该结果我们可以断定，Phatch 在 Archlinux 的 AUR 中。现在，我们只需按 1 就可以安装它了。

在显示一些输出信息后，Yaourt 会让你作出第一个选择：是否编辑 PKGBUILD 文件。按 Y 回答并输入你喜欢的文本编辑器后，你可以针对 PKGBUILD 的内容进行修改。

```
==> Edit the PKGBUILD (recommended) ? [Y/n] ("A" to abort)
```

```
==> -----
```

然后，Yaourt 会询问是否继续编译。我们的回答当然是 Y。

```
==> Continue the building of 'phatch'? [Y/n]
```

```
==> -----
```

接着，Yaourt 询问是否安装已编译好的包，同样回答 Y 即可。

```
==> Continue installing 'phatch'? [Y/n]
```

```
==> [v]iew package contents [c]heck package with namcap
```

```
==> -----
```

最后，Yaourt 将检查投票情况，并问你是否要投票，按 Y 选择投票，按 n 表示不投票。

```
==> Checking for phatch's vote status
```

```
You have already voted for phatch inclusion/keeping in [community]
```

综观 Yaourt 的命令行选项，与 Pacman 非常相似。关于 Yaourt 的更加详细的用法，通过 `man yaourt` 可以获得参考。其实，除了从 AUR 安装包外，Yaourt 也可以从 Archlinux 的源安装包，此不赘述。

## 参考资源

Archlinux 官方网站: <http://www.archlinux.org>

Archlinux 官方论坛: <http://bbs.archlinux.org>

Archlinux 维基: <http://wiki.archlinux.org>

Archlinux 包查找: <http://www.archlinux.org/packages/>

AUR: <http://aur.archlinux.org>

Archlinux 中文网站: <http://www.archlinux.org.cn>

Archlinux 中文论坛: <http://www.linuxsir.org/bbs/forumdisplay.php?f=96>

Pacman: <http://www.archlinux.org/pacman/>

Yaourt: <http://archlinux.fr/yaourt-en/>

Archlinux tools: Pacman <http://polishlinux.org/linux/arch/archlinux-tools-pacman/>

Archlinux tools: AUR <http://polishlinux.org/linux/arch/archlinux-tools-aur/>