

Archlinux 安装指南

By: bstaint

Mail: bstaint@gmail.com

Blog: <http://www.bstaint.net>

Time: 12 年 12 月 23 日

谨以此文献给那些默默为 Archlinux 维护的和学习 Archlinux 的朋友。



目录

Archlinux 安装指南.....	1
目录.....	1
简介.....	2
一、准备安装.....	2
1.1 准备工具.....	2
1.2 VMware 配置.....	3
1.3 Live CD 启动盘制作.....	5
1.3.1 使用软件制作.....	5
1.3.2 grub4dos 相关.....	6
二、安装 Archlinux.....	7
2.1 进入 Live CD.....	7
2.2 分区格式化硬盘.....	8
2.2.1 分区.....	8
2.2.2 格式化.....	12
2.3 安装软件.....	13
2.3.1 挂载分区.....	13
2.3.2 网络配置.....	13
2.3.2.1 DHCP 联网.....	14
2.3.2.2 ADSL 设置.....	14
2.3.2.3 静态 IP 设置.....	15

2.3.2.4 其他网络环境.....	16
2.3.3 配置软件源.....	16
2.3.4 软件软件包.....	16
2.3.5 系统配置.....	17
2.3.5.1 生成 fstab.....	17
2.3.5.2 配置计算机名和时区.....	18
2.3.5.3 中文支持.....	18
2.3.5.4 建立 init 引导盘.....	19
2.3.5.5 安装 grub.....	19
2.3.5.6 设置 root 密码.....	19
2.3.5.7 其他的一些设置.....	20
三、Archlinux 设置.....	20
3.1 基础设置.....	20
3.1.1 分区是否挂载成功.....	20
3.1.2 常用软件包.....	20
3.2 安装 Xwindows 和字体.....	21
3.2.1 安装 Xwindows.....	21
3.2.2 安装常用字体.....	22
3.2.3 进入 Xwindows.....	22
四、附录.....	23
4.1 systemd 服务管理.....	23
4.2 本文所涉及命令.....	24
4.3 Archlinux 常用配置文件.....	25
4.4 Linux 常用软件.....	25

简介

该版为第五版，采用 archlinux-2012.12.01-dual.iso（目前最新）在第四版的基础上进行修正，主要讲解无 initscripts（`/etc/rc.conf`）如何配置静态 IP 和启用服务（`systemctl`），修正第四版本部分细节问题，并且加入新的内容，加入目录索引（其中配图除上方 Logo 以外均来自实际操作截图）。

Tip:

1. 文中所使用的命令我都以深灰色背景标注，其中命令中的#表示当前为 root 用户。
2. 文中出现括号部分为相关名词的解释。

一、准备安装

1.1 准备工具

由于设备限制，文中的截图均来自 VMware 等虚拟软件，实际操作中如果要安装可能会遇到各种问题，你可以善用搜索引擎或者发邮件给我一起来解决。

VMware（<http://www.vmware.com>）

说明：之所以采用这款虚拟软件乃是个人习惯所致，你可以采用其他同类产品，这里不过多介绍。

archlinux-2012.12.01-dual.iso（<http://www.archlinux.org/download/>）

说明：该版本为当前最新的 Live CD 镜像，其安装过程中需联网环境，如果无法联网请不必再尝试。

Archlinux Installation Guide (https://wiki.archlinux.org/index.php/Installation_Guide)

说明：时刻打开 Archlinux 的官方 wiki 可以获取到更多的帮助信息，另外也同样需要关注官方公告，由于滚动式发行版的软件均采用最新版本，所以难免会出现部分软件不兼容的问题，官方公告会及时发布相关修正内容，可以采用 RSS 订阅的方式来获取最新信息。

1.2 VMware 配置

首先使用 VMware 新建虚拟机，然后按照下图选择“我以后再安装操作系统”：

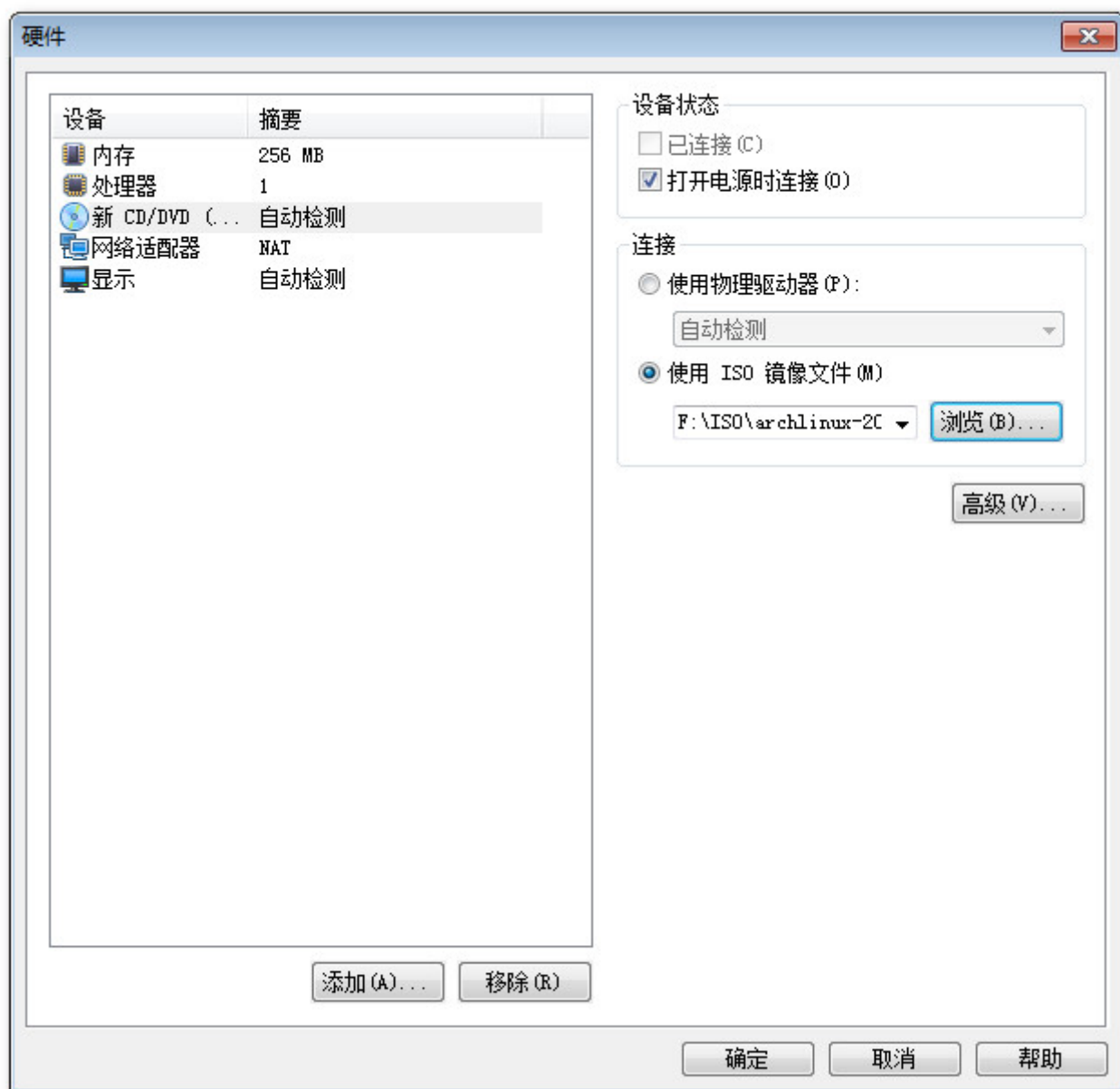


一直下一步来到选择操作系统设置这里，这里选择“Other Linux 2.6.x kernel”：



这里选择的操作系统跟我们实际安装操作系统并无任何关联，该功能为 VMware 的相关功能，主要用于自动安装系统使用，所以这里为了避免一些朋友认识上的误区，所以选择 **Other Linux**。

之后我们一直下一步直到按下“定制硬件”按钮（如果直接跳过新建虚拟机向导的朋友可以使用菜单栏上的“虚拟机” — “设置”）打开“虚拟机设置”对话框，我们可以删除不必要的硬件，例如软件驱动器、USB 控制器、声卡、虚拟打印机，然后设置新 CD/DVD（IDE）选择好我们的 Archlinux 系统镜像：



这样我们就简单的配置好一台 VMware 虚拟机，实际操作中仅仅需要我们将启动盘插入设备中等待开机。

1.3 Live CD 启动盘制作

这里简要的说下制作启动盘相关的内容，但并不具体说明，毕竟这个跟 Archlinux 并没有太大的关系，主要说使用 PowerISO 和使用 grub4dos 制作的一些区别跟一些注意事项。

1.3.1 使用软件制作

我们可以使用 PowerISO 制作 U 盘启动盘，菜单栏“工具”——“制作可启动 U 盘”，然后选择好我们的 U 盘和 ISO 文件，点击开始等待完成 U 盘启动盘制作：



当然，PowerISO 也具有刻录功能，只需要相关设备即可，由于我这里没有相关设备就不过多说明。

1.3.2 grub4dos 相关

grub4dos 相关的内容我这里不过多说明如何去安装，只是简单的说明下 menu.lst 文件的编写，我这里也给出我的 menu.lst 文件的相关内容：

```
title Install Arch Linux
find --set-root /grubhere.id
map --mem --heads=0 --sectors-per-track=0 (hd0,0)/archlinux-2011.08.19-core-i686.iso (0xff)
map --hook
chainloader (0xff)
boot
```

在引导的时候由于是将 ISO 文件读入内存中，所以比较慢，其中在启动中会出现错误，然后进入一个基本 shell 环境 (*sh-4.2#*)，需要执行下面的相关命令来进入 Live CD：

```
# mkdir /mnt
# mount -r -t vfat /dev/sdb1 /mnt
# modprobe loop
# losetup /dev/loop1 /mnt/archlinux-2012.12.01-dual.iso
# ln -s /dev/loop1 /dev/archiso
```

```
# exit
```

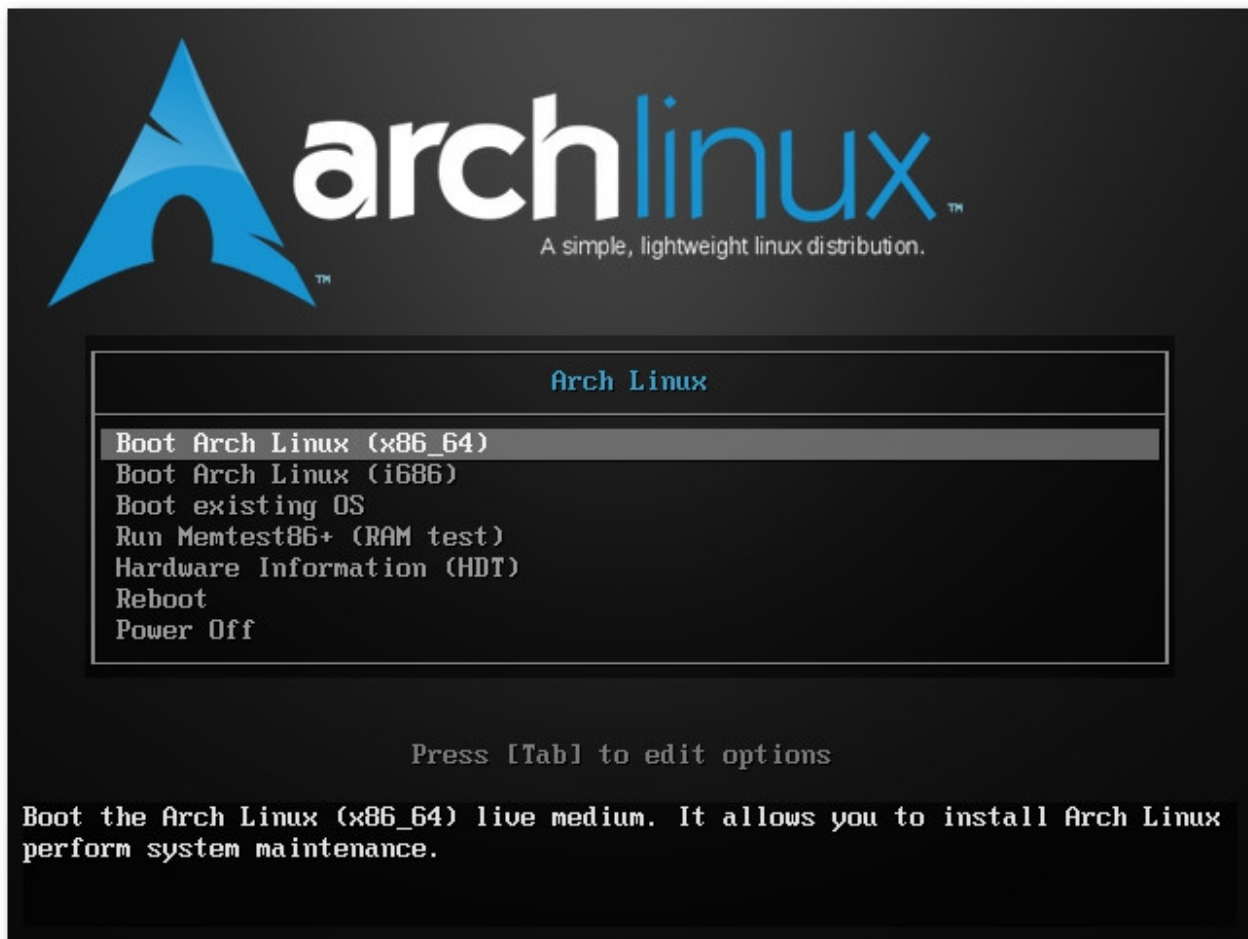
该命令主要是将 `sdb1` 块设备（U 盘）挂载到 `/mnt` 下，然后将 `/mnt/` 中的 `iso` 文件挂载到 `loop` 设备上。具体内容请自己查找相关文档。

二、安装 Archlinux

在第一章中我们已经简单的配置好安装 Archlinux 之前的环境了，接下来我们要做的就是打开电源进入 Live CD 然后进行相关操作，在这之前我们有必要学习下 Linux 的基本命令，当然本文并不会去说这些基础知识，需要各位提前准备了。

2.1 进入 Live CD

接下来我们开启电源等待进入 Archlinux 的 grub 引导界面：



上面就是 Archlinux 的 grub 引导界面，默认引导的是 64 位的，不过根据个人经验，我们优先选择 i686（原因是在 Archlinux 软件源中没有找到部分 64 位的软件），这个根据各位的 CPU 架构自行选择，按上下键选择然后回车确认，等待几秒钟进入 shell 环境（archlinux 引导速度是很快的）：

```
Arch Linux 3.6.8-1-ARCH (tty1)
archiso login: root (automatic login)
[ OK ] Started Initialize Pacman keyring.
[ OK ] Reached target Multi-User.
root@archiso ~ #
```

成功进入 Live CD 之后我们接下来就开始将 Archlinux 安装到硬盘中，在/root（进入 *shell* 默认路径）下有一篇安装相关说明（*install.txt*），我们可以查看一下：

```
# more install.txt
```

内容简单的说明了我们需要做什么，但并没有详细说明该怎样做（当然对于熟悉 *Linux* 的朋友并不是问题），所以才有了该文详细的说明我们该怎样做。

2.2 分区格式化硬盘

在接下来的操作中我们使用 *cfdisk* 来对硬盘进行分区，当然 LiveCD 中也带有 *fdisk*，两种分区软件皆可，当然前者属于半图形化，所以操作方便。

2.2.1 分区

在分区之前我们要了解下我们的硬盘接口是 SATA 还是 IDE，两者的块设备是有区别的，相关内容请自行搜索，我这里简单列出：

IDE: /dev/hda

SATA: /dev/sda (*VMware7* 默认为 *SATA*)

我们使用下面的命令开始进入 *cfdisk* 界面：

```
# cfdisk /dev/sda
```

截图如下：


```

cfdisk (util-linux 2.22.1)

Disk Drive: /dev/sda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
Pri/Log   Free Space   8589.94 *

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ] [ Write ]

Create new partition from free space_

```

在 `cfdisk` 中使用上下左右键移动光标，使用回车确认，在分区完毕请选择 `Write` 将所操作的结果写入分区表中（不然先前操作均无效）

我们需要建立如下分区结构：

块设备	挂载点	大小	分区类型
/dev/sda1	/boot	200M	主分区
/dev/sda5	swap	512M	扩展分区
/dev/sda6	/	剩下部分	扩展分区

这里我给出一个划分 `sda1` 的例子，其他分区不再说明，首先选择 `New` 然后回车，会让你选择建立主分区还是逻辑分区，根据我们上面的分区表我们将该分区设置成主分区：

```

[Primary] [Logical] [Cancel]

Create a new primary partition_

```

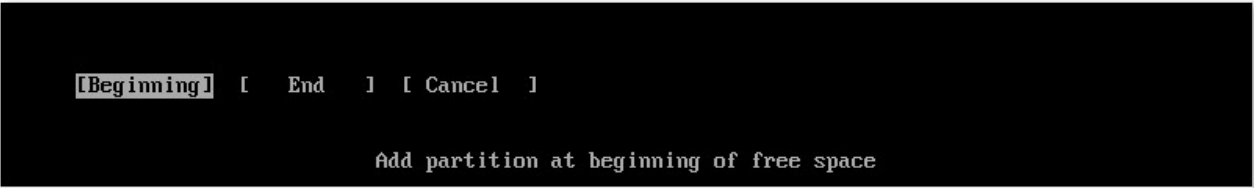
回车确认之后会让我们填写分区大小，是以 `M` 为单位的，我们填写 `200` 之后回车：

```

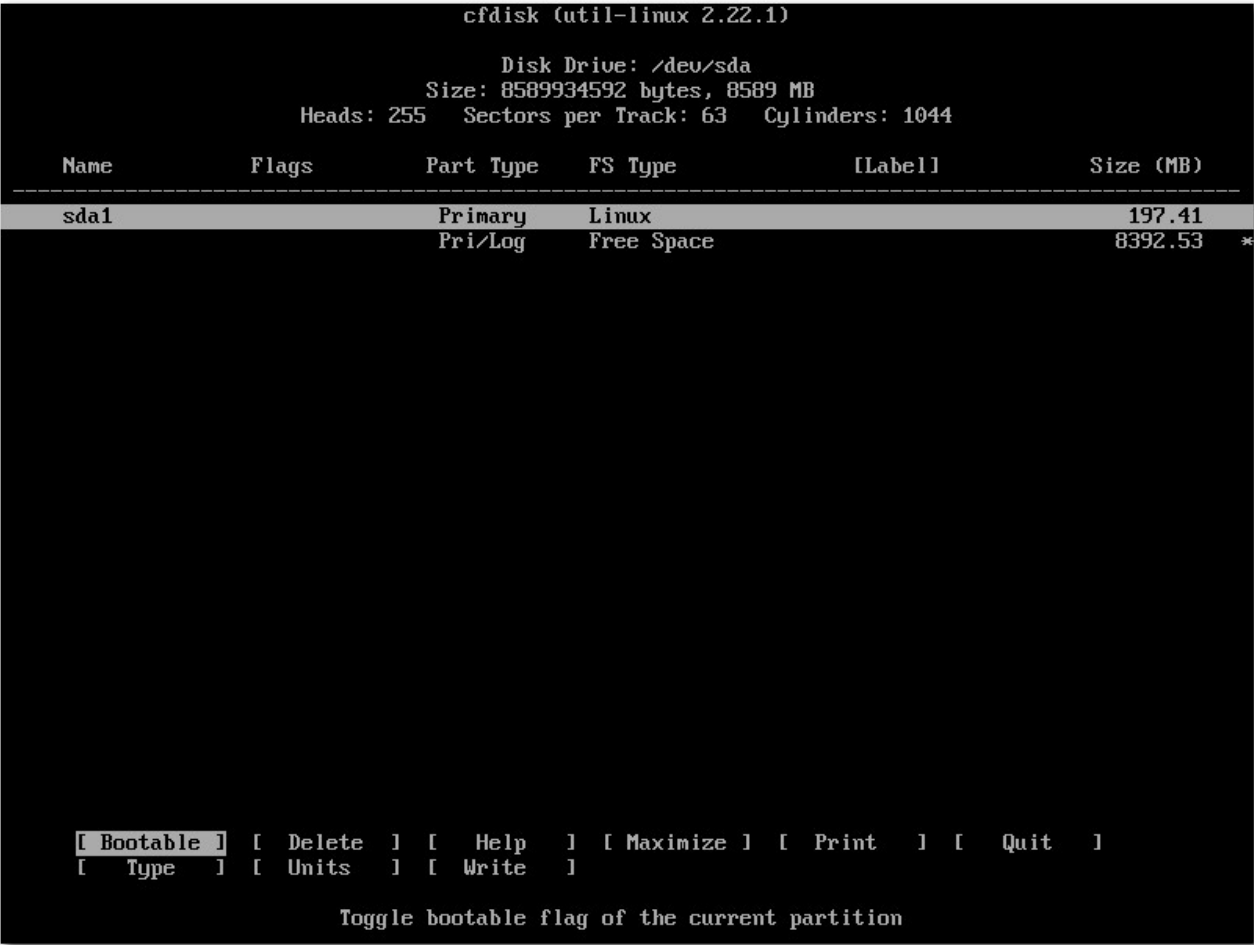
Size (in MB): 200

```

接下来会让我们选择在硬盘的空闲空间开头添加还是在结尾添加：



我们选择默认就可以了，回车之后就建立好一个分区，接下来我们可以将这个分区设置成启动分区，使用上下键选择分区，然后选择 **Bootable** 回车：



根据上面的操作我们将另外两个分区也划分出来，在划分完毕之后我们可以将 sda5 的 Type 设置成 82 (*SWAP*)：

```

cfdisk (util-linux 2.22.1)

Disk Drive: /dev/sda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

-----
Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1      Boot       Primary   Linux        197.41
sda5      Logical    Linux     509.97
sda6      Logical    Linux     7882.57 *
-----

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

Change the filesystem type (DOS, Linux, OS/2 and so on)

```

使用上下键移动光标选择分区然后使用左右键选择 Type 然后回车：

```

cfdisk (util-linux 2.22.1)

Disk Drive: /dev/sda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

01 FAT12          27 Hidden NTFS WinRE  82 Linux swap / Solaris  C4 DRDOS/sec (FAT-16 <
02 XENIX root     39 Plan 9             83 Linux                  C6 DRDOS/sec (FAT-16)
03 XENIX usr      3C PartitionMagic recov 84 OS/2 hidden C: drive  C7 Syrinx
04 FAT16 <32M     40 Venix 80286         85 Linux extended        DA Non-FS data
05 Extended      41 PPC PReP Boot       86 NTFS volume set       DB CP/M / CTOS / ...
06 FAT16         42 SFS                87 NTFS volume set       DE Dell Utility
07 HPFS/NTFS/exFAT 4D QNX4.x            88 Linux plaintext       DF BootIt
08 AIX           4E QNX4.x 2nd part    8E Linux LVM             E1 DOS access
09 AIX bootable   4F QNX4.x 3rd part    93 Amoeba                E3 DOS R/O
0A OS/2 Boot Manager 50 OnTrack DM        94 Amoeba BBT            E4 SpeedStor
0B W95 FAT32      51 OnTrack DM6 Aux1   9F BSD/OS                EE BeOS fs
0C W95 FAT32 (LBA) 52 CP/M              A0 IBM Thinkpad hiberna EE GPT
0E W95 FAT16 (LBA) 53 OnTrack DM6 Aux3   A5 FreeBSD               EF EFI (FAT-12/16/32)
0F W95 Ext'd (LBA) 54 OnTrackDM6         A6 OpenBSD               F0 Linux/PA-RISC boot
10 OPUS          55 EZ-Drive          A7 NeXTSTEP              F1 SpeedStor
11 Hidden FAT12   56 Golden Bow        A8 Darwin UFS             F4 SpeedStor
12 Compaq diagnostics 5C Priam Edisk       A9 NetBSD                 F2 DOS secondary
14 Hidden FAT16 <32M 61 SpeedStor         AB Darwin boot           FB VMware VMFS
16 Hidden FAT16   63 GNU HURD or SysV  AF HFS / HFS+            FC VMware VMKCORE
17 Hidden HPFS/NTFS 64 Novell Netware 286 B7 BSDI fs                FD Linux raid autodetec
18 AST SmartSleep  65 Novell Netware 386 B8 BSDI swap              FE LANstep
1B Hidden W95 FAT32 70 DiskSecure Multi-Boo BB Boot Wizard hidden    FF BBT
1C Hidden W95 FAT32 (LB 75 PC/IX
1E Hidden W95 FAT16 (LB 80 Old Minix
24 NEC DOS        81 Minix / old Linux  C1 DRDOS/sec (FAT-12)

Enter filesystem type: 82

```

默认即为 82，只需要回车确认即可，之后选择 Write 写入分区表，中间会提示你是否将内容写入分区表，输入 yes 回车：

```

Are you sure you want to write the partition table to disk? (yes or no): yes

Please enter 'yes' or 'no'

```

最后选择 Quit 退出 cfdisk 界面。

2.2.2 格式化

在上一章我们详细说明了如何分区，接下来我们就可以将分区格式化从而进行以后的操作。首先我们格式化/dev/sda1（boot 分区），我们使用 ext4 来格式化该分区：

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

截图如下：

```
root@archiso ~ # mkfs.ext4 /dev/sda1
mke2fs 1.42.6 (21-Sep-2012)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
48192 inodes, 192748 blocks
9637 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
24 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

同样的我们将/dev/sda6 也格式化成 ext4，这里不再重复相关操作。

接着我们将/dev/sda5 格式化成 swap，命令如下：

```
# mkswap /dev/sda5
```

截图如下：

```
root@archiso ~ # mkswap /dev/sda5
mkswap: /dev/sda5: warning: wiping old swap signature.
Setting up swspace version 1, size = 497976 KiB
no label, UUID=0055214a-0833-448a-8cb0-39701c447420
```

这样我们就将整块硬盘进行了分区和格式化操作。

2.3 安装软件

在上一章中我们详细的说明了分区和格式化，从而让我们接下来将软件安装到分区中来，下面就开始详细说明相关操作。

2.3.1 挂载分区

我们使用下面的命令将我们的分区挂载到相关目录下：

```
# mount /dev/sda6 /mnt
# mkdir /mnt/boot
# mount /dev/sda1 /mnt/boot
```

结果截图如下：

```
root@archiso /mnt # mount | grep /mnt
/dev/sda6 on /mnt type ext4 (rw,relatime,data=ordered)
/dev/sda1 on /mnt/boot type ext4 (rw,relatime,data=ordered)
```

使用 mount 命令查看系统中所有挂载点，我们从上图中可以看到已经挂载上了。

2.3.2 网络配置

接下来我们就开始设置网络，确保能连接互联网安装软件包，其中又根据实际网络环境分为 DHCP 联网和静态 IP 还有 ADSL，由于目前我的网络环境适用于 DHCP，所以其他联网方法我只会将设置方法贴出，具体联网设置并不过多说明，如果有问题可以发邮件给我。

2.3.2.1 DHCP 联网

DHCP 联网很方便，我们只需要键入下面的命令即可，前提你的硬件环境都已经正确设置。

```
# DHCPcd eth0
```

截图如下：

```
root@archiso ~ # dhcpcd eth0
dhcpcd[1475]: sending commands to master dhcpcd process
```

我们可以使用下面的命令来查看是否已经获取到了 IP：

```
# ifconfig eth0
```

截图如下：

```
root@archiso ~ # ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.5.131 netmask 255.255.255.0 broadcast 192.168.5.255
    inet6 fe80::20c:29ff:fe18:40d9 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:18:40:d9 txqueuelen 1000 (Ethernet)
    RX packets 93 bytes 10927 (10.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17 bytes 2651 (2.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 18 base 0x2000
```

在上图中我们可以看到已经成功获取到了一个 IP，为了确保是否可以成功连接到互联网我们还可以使用 ping 命令来测试，这里我就不再过多说明了。

2.3.2.2 ADSL 设置

adsl 适用于宽带连接，我们需要使用下面的命令来配置拨号脚本：

```
# pppoe-setup
```

截图如下：

```
root@archiso ~ # pppoe-setup
Welcome to the Roaring Penguin PPPoE client setup. First, I will run
some checks on your system to make sure the PPPoE client is installed
properly...

Looks good! Now, please enter some information:

USER NAME
>>> Enter your PPPoE user name (default bxxxxnxx@sympatico.ca): 区域1

INTERFACE
>>> Enter the Ethernet interface connected to the DSL modem
For Solaris, this is likely to be something like /dev/hme0.
For Linux, it will be ethn, where 'n' is a number.
(default eth0): 区域2

Do you want the link to come up on demand, or stay up continuously?
If you want it to come up on demand, enter the idle time in seconds
after which the link should be dropped. If you want the link to
stay up permanently, enter 'no' (two letters, lower-case.)
NOTE: Demand-activated links do not interact well with dynamic IP
addresses. You may have some problems with demand-activated links.
>>> Enter the demand value (default no): 区域3

DNS
Please enter the IP address of your ISP's primary DNS server.
If your ISP claims that 'the server will provide DNS addresses',
enter 'server' (all lower-case) here.
If you just press enter, I will assume you know what you are
doing and not modify your DNS setup.
>>> Enter the DNS information here: 区域4
```

从图中我们可以看到四个区域，相关解释如下：

区域1：需要填写拨号帐号（例如电话号码）；

区域2：填写拨号的网卡；

区域3：确认是否将拨号程序是否始终执行；

区域4：设置DNS服务器；

下图接上：

```
PASSWORD

>>> Please enter your PPPoE password: 区域5
>>> Please re-enter your PPPoE password:

FIREWALLING

Please choose the firewall rules to use. Note that these rules are
very basic. You are strongly encouraged to use a more sophisticated
firewall setup; however, these will provide basic security. If you
are running any servers on your machine, you must choose 'NONE' and
set up firewalling yourself. Otherwise, the firewall rules will deny
access to all standard servers like Web, e-mail, ftp, etc. If you
are using SSH, the rules will block outgoing SSH connections which
allocate a privileged source port.

The firewall choices are:
0 - NONE: This script will not set any firewall rules. You are responsible
        for ensuring the security of your machine. You are STRONGLY
        recommended to use some kind of firewall rules.
1 - STANDALONE: Appropriate for a basic stand-alone web-surfing workstation
2 - MASQUERADE: Appropriate for a machine acting as an Internet gateway
               for a LAN
>>> Choose a type of firewall (0-2): 0 区域6

** Summary of what you entered **

Ethernet Interface: eth0
User name: 
Activate-on-demand: No
DNS:             Do not adjust
Firewalling:     NONE

>>> Accept these settings and adjust configuration files (y/n)? _ 区域7
```

相关解释如下：

区域5：设置拨号密码；

区域6：设置防火墙类型；

区域7：询问是否确认这个配置输入y回车确认。

如果需要拨号使用命令：`# pppoe-connect`

这里我就不过多说明了，毕竟我这里没有实际环境，所以不便操作。

2.3.2.3 静态IP设置

接下来是讲解设置静态IP相关内容，根据wiki中的说明，需要建立network.target，我这里只是简单的演示就不再建立只是临时修改下IP：

```
# ip addr add 192.168.5.100/255.255.255.0 dev eth0
# ip route add default via 192.168.5.2
```

然后使用`# ifconfig`查看是否设置成功并且尝试Ping下局域网IP：

ifconfig 截图：


```

root@archiso ~ # ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.5.100 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 fe80::20c:29ff:fe18:40d9 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:18:40:d9 txqueuelen 1000 (Ethernet)
    RX packets 282 bytes 30125 (29.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 27 bytes 4110 (4.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 18 base 0x2000

```

Ping 截图:

```

root@archiso ~ # ping -c 3 192.168.5.1
PING 192.168.5.1 (192.168.5.1) 56(84) bytes of data.
64 bytes from 192.168.5.1: icmp_seq=1 ttl=128 time=3.44 ms
64 bytes from 192.168.5.1: icmp_seq=2 ttl=128 time=0.754 ms
64 bytes from 192.168.5.1: icmp_seq=3 ttl=128 time=0.670 ms

--- 192.168.5.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.670/1.621/3.441/1.287 ms

```

配置完 IP 之后仍然需要修改下 DNS 服务器, 这里需要编辑/etc/resolv.conf, 输入命令: `# nano /etc/resolv.conf`

Tip:

nano 编辑器相对于 vi 比较简单, 易操作, 保存: `Ctrl+X`, 查找 `Ctrl+W`。

添加如下内容:

nameserver 192.168.5.2

接下来尝试 Ping 域名是否解析成功。

2.3.2.4 其他网络环境

以上三种网络设置方法已经包含绝大多数网络环境, 但仍然有少部分网络环境需要硬件驱动的支持, 这里我不再过多说明, 毕竟我这并没有条件来一一列出, 一般该类问题请联系硬件厂家咨询。

2.3.3 配置软件源

根据上面的配置我们已经成功连接到互联网, 接下来我们需要修改软件源的配置文件以便加快安装速度, 毕竟在配置文件中的软件源镜像网站都是国外的, 我们可以将 163 提供的软件源添加到配置文件中。

使用编辑器打开/etc/pacman.d/mirrorlist, 输入命令: `# nano /etc/pacman.d/mirrorlist`

在文件开头部分添加上如下内容:

163

Server = [http://mirrors.163.com/archlinux/\\$repo/os/\\$arch](http://mirrors.163.com/archlinux/$repo/os/$arch)

截图如下:

```

GNU nano 2.2.6      File: /etc/pacman.d/mirrorlist      Modified
##
## Arch Linux repository mirrorlist
## Sorted by mirror score from mirror status page
## Generated on 2012-12-01
##
##163
Server = http://mirrors.163.com/archlinux/$repo/os/$arch_
## Score: 0.2, United States
Server = http://mirror.us.leaseweb.net/archlinux/$repo/os/$arch

```

可以根据实际情况添加其他镜像网站, 这里我就不再过多说明。

2.3.4 软件软件包

接下来我们就开始通过网络将软件包安装到硬盘上，具体安装时间根据实际情况，使用`# pacstrap /mnt base base-devel grub-bios`安装基础软件包，其中包含 grub，截图如下：

```
fuse-2.9.2-1 gdbm-1.10-1 glib2-2.34.3-1 gmp-5.1.0-1 gnupg-2.0.19-3
gpgme-1.3.1-4 groff-1.21-2 grub-common-2.00-1 hwids-20121203-1 iana-etc-2.30-3
iptables-1.4.16.2-1 isl-0.11.1-1 kbd-1.15.3-3 keyutils-1.5.5-3 kmod-12-1
krb5-1.11-1 libarchive-3.0.4-2 libassuan-2.0.3-1 libcap-2.22-3 libffi-3.0.11-1
libgcrypt-1.5.0-3 libgpg-error-1.10-2 libgssglue-0.4-1 libksba-1.2.0-1
libldap-2.4.33-3 libltdl-2.4.2-7 libmpc-1.0.1-1 libnl-3.2.11-1 libpcap-1.3.0-1
libpipeline-1.2.2-1 libsasl-2.1.23-9 libssh2-1.4.3-1 libtirpc-0.2.2-4
libusb-1.0.14-1 linux-api-headers-3.6.3-1 linux-firmware-20121118-1
mkinitcpio-0.12.0-2 mkinitcpio-busybox-1.20.2-1 mpfr-3.1.1.p2-1 ncurses-5.9-3
openssl-1.0.1.c-1 pacman-mirrorlist-20121128-1 pam-1.1.6-1 pambase-20120701-1
pcre-8.32-1 pinentry-0.8.2-1 popt-1.16-5 ppl-1.0-1 pth-2.0.7-4
readline-6.2.004-1 run-parts-4.3.4-1 systemd-196-2 sysvinit-tools-2.88-9
tzdata-2012j-1 xz-5.0.4-1 zlib-1.2.7-1 autoconf-2.69-1 automake-1.12.6-1
bash-4.2.039-1 binutils-2.23.1-1 bison-2.7-1 bzip2-1.0.6-4 coreutils-8.20-1
cronie-1.4.9-1 cryptsetup-1.5.1-1 device-mapper-2.02.98-1 dhcpcd-5.6.3-2
diffutils-3.2-1 e2fsprogs-1.42.6-1 fakeroot-1.18.4-1 file-5.11-1
filesystem-2012.12-1 findutils-4.4.2-4 flex-2.5.37-1 gawk-4.0.1-1 gcc-4.7.2-2
gcc-libs-4.7.2-2 gettext-0.18.1.1-4 glibc-2.16.0-5 grep-2.14-1 grub-bios-2.00-1
gzip-1.5-1 heirloom-mailx-12.5-3 inetutils-1.9.1-4 iproute2-3.6.0-2
iputils-20121126-1 jfsutils-1.1.15-3 less-451-1 libtool-2.4.2-7 licenses-2.9-1
linux-3.6.10-1 logrotate-3.8.2-2 lvm2-2.02.98-1 m4-1.4.16-2 make-3.82-5
man-db-2.6.3-1 man-pages-3.45-1 mdadm-3.2.6-1 nano-2.2.6-2 netcfg-3.0-1
nss-myhostname-0.3-3 pacman-4.0.3-5 patch-2.7.1-1 pciutils-3.1.10-1
pcmciautils-018-6 perl-5.16.2-2 pkg-config-0.27.1-1 ppp-2.4.5-5
procps-ng-3.3.5-1 psmisc-22.19-1 reiserfsprogs-3.6.21-4 sed-4.2.1-4
shadow-4.1.5.1-2 sysfsutils-2.1.0-8 systemd-sysvcompat-196-2 tar-1.26-2
texinfo-4.13a-7 usbutils-006-1 util-linux-2.22.2-1 vi-1:050325-3 which-2.20-5
xfsprogs-3.1.8-2

Total Download Size: 158.35 MiB
Total Installed Size: 568.25 MiB

Proceed with installation? [Y/n]
:: Retrieving packages from core...
linux-api-headers-3.6.3-1-i686 616.3 KiB 81.9K/s 00:08 [#####] 100%
tzdata-2012j-1-any 140.1 KiB 69.3K/s 00:02 [#####] 100%
glibc-2.16.0-5-i686 3.8 MiB 111K/s 00:37 [#####] 48%
```

从图中我们可以看到，需要下载 158.35M 的文件，安装之后是 568M，由于我这里网络环境所限，下载速度并不快，所以得等待一段时间。

Tip:

1. 如果终端睡眠黑屏，可以按下键盘上的任意键唤醒。
2. 中间如果出现“Operation too slow”错误会自动重新下载，如果多次出现会一直将软件包下载完毕但不会安装，请重新执行一遍命令。

下面是安装完毕的截图：

```
(134/134) installing grub-bios [#####] 100%
pacstrap /mnt base base-devel grub-bios 79.56s user 39.62s system 65% cpu 3:01.78 total
root@archiso ~ #
```

安装完毕会将安装所用的时间和系统负载都列出来。

2.3.5 系统配置

安装完所需要的软件包之后我们就开始进行系统配置，配置系统并不复杂，只要熟悉之后很容易完成。

2.3.5.1 生成 fstab

使用命令`# genfstab -p /mnt >> /mnt/etc/fstab`将生成 fstab，但是并没有 swap 分区（详见 2.2.1）相关内容，我们可以手工添加，使用命令：`# nano /mnt/etc/fstab`添加：
SWAP

/dev/sda1 swap swap default 0 0

截图如下：

GNU nano 2.2.6		File: etc/fstab		Modified
# UUID=d3eb96d4-8d92-40aa-9340-bc642aa24c1e				
/dev/sda6	/	ext4	rw,relatime,data=ordered	0 1
# UUID=35b59835-1dc2-45b8-8eec-2c333dd8c61				
/dev/sda1	/boot	ext4	rw,relatime,data=ordered	0 2
# SWAP				
/dev/sda5	swap	swap	default 0 0_	

这样我们就配置好 fstab 文件了。

2.3.5.2 配置计算机名和时区

首先我们需要切换 Linux 根目录，使用命令 `# arch-chroot /mnt`，截图如下：

```
root@archiso ~ # arch-chroot /mnt
sh-4.2# _
```

这样我就得到一个基本的 shell 环境，接下来我们就要在该环境下修改 ArchLinux 的一些配置。

- 1.修改计算机名使用 `# echo archlinux > /etc/hostname`
- 2.设置计算机时区 `# ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime`

结果截图如下：

```
sh-4.2# cat /etc/hostname & file /etc/localtime
[1] 6326
archlinux 计算机名
/etc/localtime: symbolic link to `/usr/share/zoneinfo/Asia/Shanghai' 软连接文件
[1]+  Done                  cat /etc/hostname
```

使用上图中的命令来确认下是否已经成功设置。

2.3.5.3 中文支持

默认是不支持中文的，需要手工开启，使用命令 `# nano /etc/locale.gen`，搜索 “en_US.UTF-8” 和 “zh_CN.UTF-8” 然后定位到相关位置将前面的 “#” 去掉，如下图：

```
#zh_CN.UTF-8 UTF-8
#zh_CN ISO-8859-1
#yi_US.UTF-8 UTF-8
#yi_US CP1255
#yo_NG UTF-8
#yue_HK UTF-8
zh_CN.GB18030 GB18030
zh_CN.GBK GBK
zh_CN.UTF-8 UTF-8
zh_CN GB2312
```

为了确保中文支持的比较全面，将 GBK 也同时开启，保存后使用 `# locale-gen` 更新字体缓存，截图如下：

```
sh-4.2# locale-gen
Generating locales...
en_US.UTF-8... done
zh_CN.GB18030... done
zh_CN.GBK... done
zh_CN.UTF-8... done
zh_CN.GB2312... done
zh_HK.UTF-8... done
Generation complete.
```

这样就配置好中文支持了。

2.3.5.4 建立 init 引导盘

为了确保 grub 能加载 Linux 内核，所以需要建立引导盘镜像文件，使用命令 `# mkinitcpio -p linux`，结果截图如下：

```
sh-4.2# mkinitcpio -p linux
==> Building image from preset: 'default'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux.img
==> Starting build: 3.6.10-1-ARCH
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [autodetect]
-> Running build hook: [modconf]
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [usbinput]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: 'fallback'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fallback.img -S autodetect
==> Starting build: 3.6.10-1-ARCH
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [modconf]
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [usbinput]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
```

成功会显示上图所示结果。

Tip:

以后如果更新内核之后都需要重新 `mkinitcpio`。

2.3.5.5 安装 grub

安装 grub，将 mbr 引导写入 `/dev/sda`，使用命令 `# grub-install /dev/sda`，截图如下：

```
sh-4.2# grub-install /dev/sda
Installation finished. No error reported.
```

之后我们生成 grub 配置文件，使用命令 `# grub-mkconfig -o /boot/grub/grub.cfg`，结果如下图：

```
sh-4.2# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
done
```

Tip:

以后如果要修改 `/etc/default/grub` 之后都需要重新 `grub-mkconfig`

2.3.5.6 设置 root 密码

接着我们重置下 root 密码，使用命令 `# passwd root`，截图如下：

```
sh-4.2# passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

设置好 root 密码在确认安全的同时不会忘记。

2.3.5.7 其他的一些设置

最后我们退出 Live CD 然后拿出光盘，尝试从硬盘启动，用下面得命令：

```
# exit
# umount /mnt/{boot,}
# halt
```

三、Archlinux 设置

通过上面的操作我们已经通过 Live CD 将 Archlinux 安装到了硬盘上，但是安装完毕之后只是一个最基本的操作系统，没有图形化界面和一些常用软件，我们同样需要联网安装来将 Archlinux 设置成一个易用的操作系统，根据个人爱好不同使用的软件搭配也不同。

3.1 基础设置

该章节我们会详细说明如何查看分区是否挂载成功，安装常用的几款软件。

3.1.1 分区是否挂载成功

经过安装重新开机使用 root 登录系统之后，我们得到一个 shell 环境，接下来我们的操作都会在该环境下操作完成。在第二章中我们自动生成了 fstab 并手工添加了 swap 分区，我们来查看下是否成功，使用 `df -h | grep /dev/sda` 命令查看分区，截图如下：

```
[root@archlinux ~]# df -h | grep /dev/sda
/dev/sda6      7.3G  902M  6.0G  13% /
/dev/sda1      183M   33M  141M  20% /boot
```

从图中我们可以看到我们的两个分区已经成功挂载，接下来看看 swap 分区，使用命令 `# swapon -s` 截图如下：

```
[root@archlinux ~]# swapon -s
Filename                                Type    Size    Used    Priority
/dev/sda5                               partition 497976    0        -1
```

三个分区都没有任何问题。

3.1.2 常用软件包

首先根据章节 2.3.2 以及 4.1 配置好网络然后查看下是否需要重新配置软件源，然后使用 `# pacman -Syu` 更新下软件源，截图如下：

```
[root@archlinux ~]# pacman -Syu
:: Synchronizing package databases...
core is up to date
extra is up to date
community is up to date
:: Starting full system upgrade...
there is nothing to do
```

上图内容表示软件源无需更新。

接下来安装几款常用软件，使用命令 `# pacman -S sudo net-tools`

```

[root@archlinux ~]# pacman -S sudo net-tools
resolving dependencies...
looking for inter-conflicts...

Targets (2): net-tools-1.60.20120804git-2  sudo-1.8.6.p3-1

Total Download Size:    0.66 MiB
Total Installed Size:   2.84 MiB

Proceed with installation? [Y/n] y
:: Retrieving packages from core...
  sudo-1.8.6.p3-1-i686           561.0 KiB   138K/s  00:04 [#####] 100%
 net-tools-1.60.20120804git-2-i686 113.4 KiB   142K/s  00:01 [#####] 100%
(2/2) checking package integrity [#####] 100%
(2/2) loading package files [#####] 100%
(2/2) checking for file conflicts [#####] 100%
(2/2) checking available disk space [#####] 100%
(1/2) installing sudo [#####] 100%
(2/2) installing net-tools [#####] 100%

```

安装完毕我们就可以使用 `sudo` 和 `ifconfig` 命令了。

3.2 安装 Xwindows 和字体

为了我们以后能够使用图形界面，我们还需要安装 Xwindows 和显卡驱动和常用字体。

3.2.1 安装 Xwindows

安装 Xwindows 我们需要使用命令：`# pacman -S xorg-server xorg-xinit xorg-twm xorg-xclock xterm` 截图如下：

```

[root@archlinux ~]# pacman -S xorg-server xorg-xinit xorg-twm xorg-xclock xterm
resolving dependencies...
looking for inter-conflicts...

Targets (49): fontconfig-2.10.2-1  fontsproto-2.1.2-1  kbproto-1.0.6-1  libdrm-2.4.40-1
 libfontenc-1.1.1-1  libice-1.0.8-1  libpciaccess-0.13.1-1  libsm-1.2.1-1
 libutempter-1.1.6-1  libx11-1.5.0-1  libxau-1.0.7-1  libxaw-1.0.11-1  libxcb-1.9-1
 libxdmcp-1.1.1-1  libxext-1.3.1-1  libxfont-1.4.5-1  libxft-2.3.1-1
 libxkbfile-1.0.8-1  libxmu-1.1.1-1  libxpm-3.5.10-1  libxrender-0.9.7-1
 libxt-1.1.3-1  mtdev-1.1.3-1  pixman-0.28.2-1  renderproto-0.11.1-2  xbitmaps-1.1.1-2
 xcb-proto-1.8-1  xextproto-7.2.1-1  xf86-input-evdev-2.7.3-2  xkeyboard-config-2.7-1
 xorg-bdftopcf-1.0.3-2  xorg-font-util-1.3.0-1  xorg-font-utils-7.6-3
 xorg-fonts-alias-1.0.2-2  xorg-fonts-encodings-1.0.4-3  xorg-fonts-misc-1.0.1-2
 xorg-luit-1.1.1-1  xorg-mkfontdir-1.0.7-1  xorg-mkfontscale-1.1.0-1
 xorg-server-common-1.13.1-1  xorg-setxkbmap-1.3.0-1  xorg-xauth-1.0.7-1
 xorg-xkbcomp-1.2.4-1  xproto-7.0.23-1  xorg-server-1.13.1-1  xorg-twm-1.0.7-1
 xorg-xclock-1.0.6-1  xorg-xinit-1.3.2-3  xterm-287-2

Total Download Size:    14.16 MiB
Total Installed Size:   62.85 MiB

Proceed with installation? [Y/n] y

```

会显示需要安装哪些包然后输入 `y` 确认。

安装完 Xwindows 仍然无法正常使用，我们还需要安装显卡驱动，使用命令：`# pacman -Ss xf86-video | less` 查看有支持哪些硬件，截图如下：

```
[root@archlinux ~]# pacman -Ss xf86-video | grep -E "vmware|nouveau|intel|ati" & pacman -Ss nvidia | grep 310.19-2
[1] 1331
extra/xf86-video-ati 1:7.0.0-1 (xorg-drivers xorg)
  X.org ati video driver
extra/xf86-video-intel 2.20.16-1 (xorg-drivers xorg)
extra/xf86-video-nouveau 1.0.4-1 (xorg-drivers xorg)
  Open Source 2D acceleration driver for nVidia cards
  X.org ati Rage128 video driver
extra/xf86-video-vmware 12.0.2-2 (xorg-drivers xorg) [installed]
extra/nvidia 310.19-2
  X.org vmware video driver
[1]+  Done                  pacman -Ss xf86-video | grep -E "vmware|nouveau|intel|ati"
```

vmware 可以安装 xf86-video-vmware，使用命令：`# pacman -S xf86-video-vmware`，截图如下：

```
[root@archlinux ~]# pacman -S xf86-video-vmware
resolving dependencies...
looking for inter-conflicts...

Targets (11): damageproto-1.2.1-2 fixesproto-5.0-2 libgl-9.0.1-1 libglapi-9.0.1-1
               libxdamage-1.1.3-2 libxfixes-5.0-2 libxxf86vm-1.1.2-1 mesa-9.0.1-1
               suga-dri-9.0.1-1 xf86vidmodeproto-2.3.1-2 xf86-video-vmware-12.0.2-2

Total Download Size:   13.04 MiB
Total Installed Size:  50.01 MiB

Proceed with installation? [Y/n] y
```

同安装 Xwindows 一样输入 y 确认。

3.2.2 安装常用字体

为了省大家的时间，我已经将常用的字体都收集起来了，使用命令：`# pacman -S font-bh-ttf ttf-arphic-ukai ttf-arphic-uming ttf-dejavu ttf-bitstream-vera ttf-cheapskate ttf-fireflysung wqy-zenhei wqy-bitmapfont`，截图如下：

```
[root@archlinux ~]# pacman -S font-bh-ttf ttf-arphic-ukai ttf-arphic-uming ttf-dejavu ttf-bitstream-vera ttf-cheapskate ttf-fireflysung wqy-zenhei wqy-bitmapfont
resolving dependencies...
looking for inter-conflicts...

Targets (9): font-bh-ttf-1.0.3-1 opendesktop-fonts-1.4.2-1 ttf-arphic-ukai-0.2.20080216.1-3
              ttf-arphic-uming-0.2.20080216.1-3 ttf-bitstream-vera-1.10-8 ttf-cheapskate-2.0-8
              ttf-dejavu-2.33-3 wqy-bitmapfont-0.9.9-10 wqy-zenhei-0.9.45-3

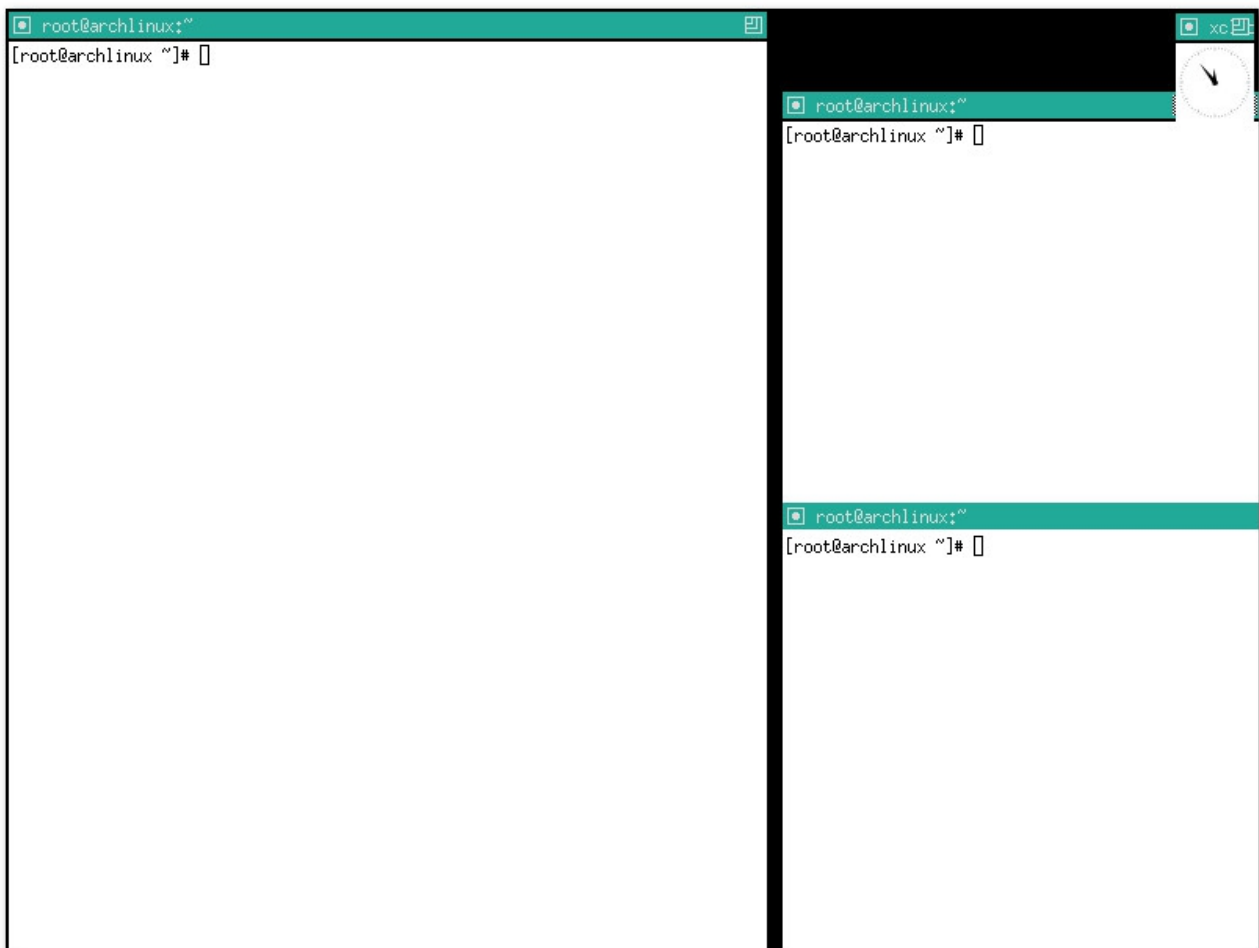
Total Download Size:   43.53 MiB
Total Installed Size:  126.00 MiB

Proceed with installation? [Y/n] y
```

比较大，等待安装完成。

3.2.3 进入 Xwindows

安装完毕之后我们就可以正常进入 Xwindows 界面，输入 `# startx` 命令即可启动 Xwindows，截图如下：



最后我们就可以安装喜欢的桌面环境（以下是软件包名）：

- 1.xfce4
- 2.kde
- 3.gnome
- 4.openbox（需搭配其他软件共同使用）
- 5.其他

四、附录

下面内容主要包含本人收集的 Linux 软件和本文中所出现的命令以及 Archlinux 的常用配置文件。

4.1 systemd 服务管理

较之于第四版所用的版本相比，新版的 Archlinux 全面采用 systemd 来控制系统服务，我将举一个例子来说明怎样开启服务和关闭重启服务操作。

我们安装完毕系统之后开机并不是自动获取 IP 的，需要执行 DHCPd 才能获取 IP，我们可以将 DHCP 服务开启。

```
# systemctl start DHCP@eth0
```

```
# systemctl enable DHCP@eth0
```

截图如下：

```
[root@archlinux ~]# systemctl start dhcpcd@eth0
[root@archlinux ~]# systemctl enable dhcpcd@eth0
ln -s '/usr/lib/systemd/system/dhcpcd@.service' '/etc/systemd/system/multi-user.target.wants/dhcpcd@eth0.service'
```

下面是该命令的语法:

systemctl start|stop|disable|enable servicename

4.2 本文所涉及命令

1.3.2

```
# mkdir /mnt
# mount -r -t vfat /dev/sdb1 /mnt
# modprobe loop
# losetup /dev/loop1 /mnt/archlinux-2012.12.01-dual.iso
# ln -s /dev/loop1 /dev/archiso
# exit
```

2.1

```
# more install.txt
```

2.2.1

```
# cfdisk /dev/sda
```

2.2.2

```
# mkfs.ext4 /dev/sda1
# mkfs.ext4 /dev/sda6
# mkswap /dev/sda5
```

2.3.1

```
# mount /dev/sda6 /mnt
# mkdir /mnt/boot
# mount /dev/sda1 /mnt/boot
```

2.3.2.1

```
# DHCPd eth0
# ifconfig eth0
```

2.3.2.2

```
# pppoe-setup
# pppoe-connect
```

2.3.2.3

```
# ip addr add 192.168.5.100/255.255.255.0 dev eth0
# ip route add default via 192.168.5.2
# nano /etc/resolv.conf
```

2.3.3

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

2.3.4

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

2.3.5.1

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

2.3.5.2

```
# echo archlinux > /etc/hostname
# ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

2.3.5.3

```
# nano /etc/locale.gen
```

2.3.5.4

```
# mkinitcpio -p linux
```


2.3.5.5

```
# grub-install /dev/sda  
# grub-mkconfig -o /boot/grub/grub.cfg
```

2.3.5.6

```
# passwd root
```

2.3.5.7

```
# exit  
# umount /mnt/{boot,}  
# halt
```

3.1.1

```
# df -h | grep /dev/sda
```

```
# swapon -s
```

3.1.2

```
# pacman -Syu
```

```
# pacman -S sudo net-tools
```

3.2.1

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

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

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

3.2.2

```
# pacman -S font-bh-ttf ttf-arphic-ukai ttf-arphic-uming ttf-dejavu ttf-bitstream-vera ttf-cheapskate  
ttf-fireflysung wqy-zenhei wqy-bitmapfont
```

3.2.3

```
# startx
```

4.1

```
# systemctl start DHCP@eth0
```

```
# systemctl enable DHCP@eth0
```

4.3 Archlinux 常用配置文件

/etc/hosts

/etc/hostname（主机名配置文件）

/etc/localtime（时区配置文件）

/etc/default/grub（grub 配置文件）

/etc/conf.d/network（静态 IP 配置文件）

/etc/pacman.d/mirrorslist（软件源配置文件）

/etc/pacman.conf（pacman 配置文件）

4.4 Linux 常用软件

以下所列软件均为本人收集，部分软件来源于 aur 软件源，所以需安装 yaourt。

1.gvfs（说明：gvfs 是 gnome 桌面系统的虚拟文件系统。）

2.unrar unzip rar zip p7zip（说明：常用解压软件）

3.aliedit（说明：支付宝安全控件）

4.gcalctool（说明：计算器软件）

5.ibus ibus-pinyin（说明：中文输入法）

6.ntfs-3g（说明：挂载 ntfs 分区软件）

7.flashplugin（说明：Firefox Flash 插件）

8.gparted dosfstools（说明：分区管理软件）

- 9.keepassx (说明: 密码管理软件)
- 10.bleachbit (说明: 文件清理软件)
- 11.numlockx (说明: 控制 numlock 软件)
- 12.evince (说明: PDF 阅读软件)
- 13.gnochm (说明: CHM 阅读软件)
- 14.isomaster (说明: ISO 管理软件)
- 15.gvim (说明: 编辑器)
- 16.ghex (说明: HEX 编辑器)
- 17.poedit (说明: po 编辑器)
- 18.geany geany-plugins (说明: 小巧的代码编辑软件)
- 19.anjuta glib2-docs (说明: C/C++ IDE 以及 glib 说明文档)
- 20.dia (说明: 流程图绘制软件)
- 21.git (说明: 版本控制软件)
- 22.elinks (说明: 命令行下的浏览器)
- 23.firefox firefox-i18n-zh-cn (说明: Firefox 浏览器)
- 24.chromium (说明: Chrome 开源版浏览器)
- 25.transmission-gtk (说明: bittorrent 客户端软件)
- 26.amule (说明: ed2k 客户端软件)
- 27.filezilla (说明: FTP/SFTP 客户端软件)
- 28.Sylpheed (说明: 电子邮件客户端)
- 29.pidgin (说明: 多协议聊天软件)
- 30.macchanger (说明: MAC 地址修改软件)
- 31.whois (说明: WHOIS 软件)
- 32.dnsutils (说明: DNS 软件包)
- 33.freerdp (说明: 远程桌面客户端)
- 34.mplayer (说明: 视频播放器软件)
- 35.gecko-mediaplayer (说明: 视频播放器软件)
- 36.vlc (说明: 视频播放器软件)
- 37.deadbeef (说明: 音乐播放器软件)
- 38.mutagen (说明: 音乐播放器软件)
- 39.gtk-recordmydesktop (说明: 屏幕录制软件)
- 40.codecs (说明: 视频音频解码包)
- 41.virtualbox (说明: 虚拟机软件)
- 42.gimp (说明: 图片处理软件)
- 43.inkscape (说明: 矢量图绘制软件)
- 44.fceux (说明: FC 模拟软件)
- 45.goldendict (说明: 基于 WebKit 的词典软件)
- 46.libreoffice (说明: 文档处理软件)