

第五章 其他非标准安装程序解析

透过前几章的介绍，您应该已经大致完成 Slackware Linux 的系统安装了。本章介绍的话题，主要是一些与系统安装有直接或间接关系的技术与观念，相信这些可以帮助您进一步了解 Linux。笔者也再次提醒，对于想把 Linux 摆在严肃场合的朋友来说，若没有把握系统安装的品质，不妨考虑重新安装，而透过更多的技术琢磨，重新安装后的系统必然可以更顺畅运转。

- 交大『云观 CD-ROM III』直接由 CD-ROM 执行之安装法
- bootdisk 磁片可以当作系统急救磁片
- 如何自制 bootdisk
- pkgtool 软体维护
- 手工建置档案系统
- 您需要虚拟记忆体吗?
- 手工建造 Swap Partition
- 手工建造 Swap File
- 手工设定 LILO
- 如何解除 LIL0 自动开机程序

□ 交大『云观 CD-ROM III』直接由 CD-ROM 执行之安装法

笔者很高兴看到交大一群朋友们的努力，出版了一份本土的 "Plug & Play" Linux CD-ROM，我想，凡是受益於这份 CD-ROM 的朋友，应该有必要知道，这 CD-ROM 主要是刘大川 ii 黄界木坚 ii 李欣?? ii 李建达以及张杰生等几位先生先进们的努力所成就。

由於本 CD-ROM 也附有标准 Slackware Linux 2.1.0，想正常安装 Linux 系统者（其实这才是笔者所推荐的，本节的 Plug & Play Linux 我感觉只适合用来作为 demo 用途，不过这算是笔者的偏见而已），请参考第四章的说明即可，不过，仍有一点须要注意的，就是 CD-ROM 上的路径，这部份您要回答 `//slackware/slakware`（注意不要拼错字）即可，也就是在 setup 画面选择 "slackware/slakware TransAmeritech CD" 选项。

这一节我们所要介绍的，是此次交大 云观三的一个特色之一 -- Plug & Play，这意思是说，好像电器用品一样，插上电源就可以玩了，虽然，这样说法或许有些夸张，其真实的情况是说，您「几乎」不需要额外的进行正常的安装程序，而只需要一些些硬碟空间，以及几步简单的程序，之後，就可以直接从 CD-ROM 上跑 Linux！

底下我们就来看看如何把 Linux 插上去就可以玩 :-)

一样的，我仍然建议读者先给 Linux 预备一个专有分割区（Partition）[注]，这个分割区不需要很大，根据笔者测试，大约只需要 13MB 到 15MB 空间即可进行安装，而安装后系统实际占用空间大约为 10MB 左右。

有了一个分割区，接下来的动作当然是建置档案系统，这就好像 DOS 的 FORMAT（格式化）动作。以上两个动作，仍然是标准的作业系统安装步骤，以下的步骤才是特殊之处。

前面几段话已经暗示了，我们其实没有办法把 Linux 直接在云观 III CD-ROM 上跑，真实的情况，是云观 III CD-ROM 设计了一个 /usr 目录，这目录收集了高达二百六十几 MB 的应用软体，这些都是可立刻执行的，也就是说可以把 CD-ROM 的 /cdrom/usr 目录 挂在 / 档案系统之下的 /usr。我们反过来说，我们之前之所以要设立一个 13MB 的分割区，是因为 Linux 系统的 / (root) 档案系统仍然必须设立好，且 / 档案系统下的 /bin ii /sbin ii /etc ii /boot ii /proc ii /tmp 等等目录架构是构成一部 Linux 系统所必须有的，这些虽然可以精简阳春，可是却不能没有，更重要的原因，Linux / 档案系统有一部份是必须可以读也可以写（Read & Write），而 CD-ROM 却只能唯读（Read Only）

)。

云观 III CD-ROM 所谓 Play & Play 的设计,就是把以上所提到的这些最基本档案系统,事先压缩成两个独立的档案,放在该 CD-ROM 内 (/cdrom/linux/root.tgz 及 /cdrom/linux/var/tgz),所以说,我们安装时,在观念上,就是直接把这两个档案直接解压缩在这一个 13MB 的分割区上,该分割区就立刻拥有一个完整的档案系统架构。这里,我们可以容易的看出,这样的安装法完全不同於标准 Slackware Linux 安装程序。

经过了直接的解压缩安装,我们仍然需要非常简单的系统设定,而这部份又与 Slackware Linux 安装程序完全一样,於是我们可以完全引用 setup 程式来做,不过主要只做 setup 里面的 SOURCE 与 CONFIGURATION 两道步骤。当这到步骤也完成之後,我们重新开机,这时,我们可以看到系统一启动过程中,CD-ROM 灯号就立刻亮起来,而且断断续续,这是因为系统决大部分的东西都放在该 CD-ROM 内,系统必须依靠 CD-ROM 才可以完整启动机器,更重要的,我们所需的所有应用软体,也都摆在该 CD-ROM 内。

希望以上罗哩罗唆的说明,有助於观念上的补充,以下,我们就来看看实际上的操作细节。

步骤零,分割区规划:

首先您要根据第四章的说明,规划出至少 13MB 的硬碟空间,再次提醒您,您必须在 Linux 环境下用 fdisk 来执行规划,更具体的说,是用 bootdisk 及 rootdisk 两张磁片开机後 (有关 bootdisk 及 rootdisk 的选择与制作,请参考第三章),执行 fdisk。

步骤一,建置作档案档案系统:

假如想依照 Slackware Linux 的标准程序来做的话,可以参考第四章的说明,假如想迅速的手工操作的话,可以参考第五章「手工建置档案系统」乙节的说明。

以上两个步骤都可以用 Slackware Linux 的 setup 标准安装程序来完成。

接下来的步骤才是本节的精华,以下我们要做一些假设,我们假设所要安装的分割区为 /dev/hda2,使用之 CD-ROM 为 Panasonic 562B/563B,读者引用时,记得依照您的实际情况做修改。

步骤二. 安装前预备

以下步骤类似於 Slackware Linux setup 过程中,所指令的 Target 与 Source 等程序。

往下的步骤,您也同样选用恰当的 bootdisk 及 rootdisk 磁片来启动机器,之後以 root 帐号进入系统,接下来执行以下操作。

```
$ mount -t ext2 /dev/hda2 /mnt -o rw
```

这步骤与 Slackware Linux setup 程序的 Target 设定同样意思。

```
$ mount -t iso9660 /dev/sbpcd /var/adm/mount -o ro
```

这步骤与 Slackware Linux setup 程序的 Source 设定同样意思。请读者特别留意,假如您使用 Mitsumi 唯读光碟机,把 /dev/sbpcd 换成 /dev/mcd,是 Sony CDU-31A 的话换成 /dev/sonycd,是 Sony CDU-535 的话换成 /dev/cdu535,是 LMS/Philips 则换成 /dev/lmscd,其他 SCSI 唯读光碟机则换成 /dev/scd0。

步骤三. Linux Plug & Play 安装:

```
$ cd /cdrom/linux
$ install
```

正常的话,您会在萤幕上看到一连串档案拷贝的过程,这步骤可能维持一两分钟。

步骤四. 系统设定

这步骤您也可以引用 Slackware Linux 的 setup 标准安装程序来完成,不过大约有两点要留意,在 setup 程序中,只执行 SOURCE 以及 CONFIGURATION 两到程序即可, TARGET 不执行是因为刚刚步骤二时已经完成, SOURCE 之所以要重复做一次 (注意,步骤二也同样已经做了),我发现是因为 setup 会 umount 已经挂好的光碟机。

```
$ setup
```

这个指令就是您此刻唯一需要做的,请参考第四章的说明,细心的完成 CONFIGURATION 的操作,之后用 CONFIGURATION 过程中所制作的开机片重新启动机器。正常的话,您就可以使用这 Plug & Play Linux。

注. Linux 也可以在 UMSDOS 模式下,直接在 DOS Partition 下执行,但笔者不推荐这样做,尤其想把 Linux 摆在有生产力应用所在者,不要试。

☐ bootdisk 磁片可以当作系统急救磁片

这一章中,我们首先来看看 bootdisk 作为救急磁片的用途。

Slackware Linux 的 bootdisk 的第一个救急用途,就是用来启动硬碟内的 Linux 系统,什么时候需要用到呢?比方说您原本从硬碟开机突然失效了,更不巧您也没有做软碟开机片,这些情况您就可以使用 bootdisk 开机 (没有用到 rootdisk 磁片),之后在提示符号下使用 mount root=/dev/hda1 指令,详细情况如下面萤幕所示。

LILO

Welcome to the Slackware Linux 2.0.2 Bootkernel disk!

If you have any extra parameters to pass to the kernel, enter them at the prompt below. For instance, you might need something like this to detect the hard drive on PS/1 and ValuePoint models from IBM:

```
ramdisk hd=cyl,hds,secs (Where "cyl", "hds", and "secs" are the number of
                        cylinders, sectors, and heads on the drive. Most
                        machine won't need this.)
```

Also, in a pinch, you can boot your system with a command like:

```
mount root=/dev/hda1
```

On machines with low memory, you can mount root=/dev/fd1 or mount root=/dev/fd0 to install without a ramdisk. See LOWMEM.TXT for details.

If you would rather load the root/install disk from your second floppy drive: drive2 (or even this: ramdisk root=/dev/hd1)

DON'T SWITCH ANY DISKS YET! This prompt is just for entering extra paramters.

If you don't need to enter any paramters, hit ENTER to conitnue.

boot: mount root=/dev/hda1

把上面 /dev/hda1 换成您实际的情况就可以。

Slackware Linux bootdisk/rootdisk 组合也可以作为系统急救磁片,比方说您不小心修改 /etc/fstab 档案,但却由於有错误造成无法开机,类似这种例子,您就可以用 bootdisk 磁片开机,之後换插入 rootdisk磁片,继续执行开机作业,我们前面提很多次,这样子执行的就是一套阳春的 Linux作业系统,但虽然阳春,基本的系统工具程式都有,比方说 vi 文字编辑器。

由这软碟系统来修复硬碟内的 Linux档案系统的作法,是先把硬碟内的 Linux / (root) 档案系统挂上来,之後用 vi 修改造成错误的设定档案,或者用 e2fsck 或 fsck 来修复硬碟内的档案系统结构(就类似 DOS的 CHKDSK指令)。

```
$ mount -t ext2 /dev/hda2 /mnt
```

笔者发现 Slackware Linux 2.02 以後的 rootdisk 磁片里面已经没有 vi 及 e2fsck 等工具,假如您想使用 vi 的话,可以试看看 /mnt/usr/bin/vi。

☐ 如何自制 bootdisk

有些朋友会碰到 bootdisk 磁片开机後,无法认为电脑硬体设备,举个例子来说,比方您电脑配备有一个非常特别品牌的硬碟控制卡,而所以现有 Slackware Linux 的众多 bootdisk 都没有 driver,而您又确定知道新版的 Linux Kernel source 里面有 driver。您有这情况时,可以请别人帮忙,在别一部 Linux 机器上编译该份 Kernel 核心程式码,之後根据该份核心码制作一张您自己的 bootdisk。

自制 bootdisk 的操作,实际上是拿一片已经制作好的 bootdisk 磁片,把该磁片挂 (Mount) 一部运转中的 Linux 系统下,之後把我们事先编译好的系统核心拷贝到磁片上,这样会把磁片上旧核心盖过去,最後,再执行一些开机设定动作就可以了。细节如下:

0. 编译系统核心,把所需要的Driver编译进去
(请参考拙著『轻轻松松管理 Linux/Unix』)
1. 插入一张已经制作好的 bootdisk 磁片
2. mount /dev/fd0 /mnt # 挂上 bootdisk
3. cat zimage > /mnt/vmlinuz # 取代 bootdisk上旧 Kernel
4. rdev -R /mnt/vmlinuz 0 # 指定 bootdisk 开机後可读写
5. rdev /mnt/vmlinuz /dev/fd0H1440
指定 开机後 root (/) 档案系统所在,此处假设 A: 为三寸半磁碟
机。若是5 1/4寸 1.2MB磁碟,则改为 /dev/fd0h1200
6. rdev -r /mnt/vmlinuz 1440
指定 RAMDISK大小,若是 1.2MB磁片,可以改为 1200
7. cp -fa /boot/* /mnt/boot # 拷贝自己系统下的开机相关档案
8. lilo -r /mnt # 重新安装 LILO 软碟开机,萤
Added ramdisk # 幕上会看到 这三行讯息
Added drive2
Added mount
9. umount /mnt
- A. 完成,取出您的磁片,这就是一张自制的 bootdisk磁片

☐ pkgtool 软体维护

有时候,您因为任何原因,需要增减系统内的应用软体,有这情况时您就必须借助 pkgtool 的帮忙。比方说,您想把某一个不常使用的软体拿掉,因为它占用非常大的硬碟空间,或者比方说,一开始时,您只安装了最基本的A 与 AP系列软体,您后来想要加装剩下一系列的软体,这时您就可以使用 pkgtool 程式来达成。

pkgtool 的使用场合主要是在安装完成後的系统之下,而不是 bootdisk/rootdisk 所执行的安装过程中。

pkgtool 的使用法非常简单,这部份笔者在这『轻轻松松管理 Linux/Unix』这本书也有提到。大体而言,您只需要下 pkgtool 这指令,就可以直接在萤幕画面上得知操作的细节,应该非常容易。

☐ 手工建置档案系统

在第四章中,我们提到,执行作业系统的安装过程中,必然要经过一道「建置档案系统」的程序 (DOS用 FORMAT, Linux用 mke2fs), 这道程序是用来把已经完成规划的分割区,进一步赋予其档案系统属性。

可惜前面的介绍中,由於 setup 自动的进行这道步骤,所以我们没有认识到如何收工来做这样工作,这一节就要补充这个不足。

作法上,通常我们必须先规划出一个新个分割区 (或者说已经规划好了),我们接着要想办法知道要所执行档案系统建置的分割区实际的大小 (Blocks数目),以下例子是用 fdisk 来取得数据。

```
# fdisk /dev/hdb
```

```
Command (m for help): p
```

```
Disk /dev/hdb: 16 heads, 63 sectors, 826 cylinders
Units = cylinders of 1008 * 512 bytes
```

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hdb1		1	1	100	50368+	83	Linux native
/dev/hdb2		101	101	405	153720	7	OS/2 HPFS
/dev/hdb3		536	536	736	101304	83	Linux native
/dev/hdb4	*	406	406	535	65520	a5	BSD/386

```
Command (m for help): q
```

假如我们的目的分割区为 /dev/hdb3,这时我们看其对应那一行,得到一个数据 101304 Blocks,根据这数据,我们执行以下指令:

```
$ mke2fs -c /dev/hdb3 101304
```

我们就简单介绍如此,想深入认识的朋友,可以用 man mke2fs 指令来查询线上使用手册。

☐ 您需要虚拟记忆体吗?

在前面第四章中,我们介绍到很多地方提起「虚拟记忆体」,您心理会不会问说,「我真的需要设计虚拟记忆体吗」?

这问题牵涉到一个实际的问题,就是您现有的记忆体够大吗? 大凡实际的记忆体不够时, Linux 便有可能开始用到虚拟记忆体,假如这时候连虚拟记忆体也用光了,系统便会有类似当机的反应 (多半不

会真正当机), 至少正执行中的应用软体因此原因当掉的可能非常大。

照笔者经验, 我通常会规划一个 Swap Partition 的虚拟记忆体, 而其大小不小於实际虚拟记忆体之大小, 以这样的情况观察系统的运转, 尤其是记忆体消耗的情况, 若发觉有需要, 可以再设立第二个 Swap Partition, 若没有办法增加 Swap Partition, 可以用 Swap File, 不过 Swap File 效率上比 Swap Partition 低。

最後提醒读者一个观念, 假如系统经常性的使用掉大量的虚拟记忆体, 这时您或许就该考虑扩充实际的记忆体了, 假如您希望系统运转的效率处在最佳状态的话。 對於把 Linux 使用在有严肃生产力场合的读者朋友来说, 尤其要注意这一点。

☐ 手工建造 Swap Partition

手工建造一个 Swap Partition 作为虚拟记忆体, 操作上非常简单, 我们同样是用 fdisk 规划出一个新的分割区 (假设硬碟内仍未满), 同时把该分割区的型别改别 Swap Partition, 之後我们用 mkswap 指令赋予其档案系统属性, 最後, 用 swapon 指令启动使用就可以。 一共三道步骤。

以下我们就来看一下实际操作系范例。

步骤一.

```
# fdisk /dev/hda          ← 在Shell提示符号後键入fdisk指令

Command (m for help): p    ← 我先来看看目前的设定情况

Disk /dev/hda: 15 heads, 17 sectors, 1001 cylinders
Units = cylinders of 255 * 512 bytes

   Device Boot   Begin    Start      End  Blocks   Id  System
/dev/hda1  *        1         1      161    20519    4   DOS 16-bit <32M
/dev/hda2             302       302     1001    89250   81   Linux/MINIX
/dev/hda3             260       260       301     5355   81   Linux/MINIX

Command (m for help): n    ← 增加一个Partition
Command action
  e   extended
  p   primary partition (1-4)
p          ← 指定Primary
Partition number (1-4): 4   ← 指定第四个分割区 (前三个已经满了)
First cylinder (162-1001): 162 ← 回答162 (162为系统自动侦测得知)

Last cylinder or +size or +sizeM or +sizeK (162-259): 259 ←
259也是系统自动侦测, 这两个数据, 您依照自己情况调整
Command (m for help): t    ← t 变更分割区的 ID

Partition number (1-4): 4   ← 第4个分割区 (您依照情况调整)

Hex code (type L to list codes): 82 ← 82代表Swap Partition
Changed system type of partition 4 to 82 (Linux swap)

Command (m for help): p    ← 设定完毕, 我们先瞧一瞧

Disk /dev/hda: 15 heads, 17 sectors, 1001 cylinders
Units = cylinders of 255 * 512 bytes

   Device Boot   Begin    Start      End  Blocks   Id  System
/dev/hda1  *        1         1      161    20519    4   DOS 16-bit <32M
/dev/hda2             302       302     1001    89250   81   Linux/MINIX
```

/dev/hda3	260	260	301	5355	81	Linux/MINIX
/dev/hda4	162	162	259	12495	82	Linux swap

果然新造一个/dev/hda4 partition, 且指定为Linux Swap

Command (m for help): w ← 无误後, 将以上设定存起来
The partition table has been altered.
Please reboot before doing anything else.

步骤二.

执行步骤一後, 重新开机, 接着执行步骤二及步骤三。

```
$ mkswap -c /dev/hda4 12495 ← 这里 /dev/hda4为刚设定的Swap
                             Partition, 12495为该分割区的大小
Setting up swapspace, size = 12787712 bytes
```

步骤三.

```
$ /sbin/swapon /dev/hda4
```

假如您是临时需要扩充虚拟记忆体, 上面这三道步骤已经足够了。但假如您想让系统在每次开机後都自动取用这些虚拟记忆体, 这时您就要有额外的设定, 否则每次您都要重复执行步骤三, 系统才知道如何取用该分割区。

要设定这分割区为固定系统的虚拟记忆体很简单, 方发之一, 就是把 /sbin/swapon /dev/hda4 指令加入 /etc/rc.d/rc.local 档案内, 或者另外的方法, 在 /etc/fstab档案内加入

```
/dev/hda4 none swap
```

一行就可以。同样的, 假如您的分割区不是/dev/hda4的话, 请更改这数据以符合您的情况。

您可以用vi 来编辑/dev/fstab档案以便加入这一行, 下图就是笔者/dev/fstab档案的内容。

/dev/hda2	/	ext2	defaults	
/dev/hda4	swap	swap	defaults	← 加入这一行
none	/proc	proc	defaults	

或者您可以使用下列指令(这适合不懂vi的读者们)。

```
$ cd /etc
$ cp fstab fstab.backup ← 先备份, 以免万一失手
$ echo "/dev/hda4 swap swap defaults" >> fstab
```

```
$ /sbin/swapon -a ← 使生效, 而无须重新开机
```

经过这一额外的动作後, 我以後每次的开机, 系统都会自动把这一个12MB大的Swap Partition 扩充加入原来的记忆体。为确定这设定是否生效, 您重新开机, 并观察在开机的时候, 是否有下面这一段讯息,

```
adding Swap: 12488k swap-space
```

有的话, 事情就妥当了。

☐ 手工建造 Swap File

至於设置Swap File与设置Swap Partition道理也很类似，我们来看下面步骤。

1. 执行dd 以建置一个Swap File (此例为/SWAPFILE, 名字可以自取)
2. 执行"mkswap swap-file block-size"
3. 执行"swapon swap-file"

在步骤一，我设定8 MB(8196) 的Swap File, 该Swap File取名为SWAPFILE, 该档案位於第二个Partition (/dev/hda2)的"/"目录下。这档名您可自取，所在位置也随您喜好。

步骤一

```
$ dd if=/dev/hda2 of=/SWAPFILE bs=1024 count=8192
8192+0 records in
8192+0 records out
```

步骤二

```
$ mkswap /SWAPFILE 8192
Setting up swspace, size = 8384512 bytes
```

步骤三

```
$ /sbin/swapon /SWAPFILE
Adding Swap: 8188k swap-space
```

经过这三道步骤後，这一个Swap File已经暂时成为您系统记忆体的一部份了，不过一旦您关机，这一个Swap File就又失效了。於是我們便又想到，如何永久取用该Swap File?

很简单，您可以用"vi"编辑/etc/rc档案，加入"/sbin/swapon /SWAPFILE"一行。或者，底下用更简单的示范，让不懂vi的朋友参考。下面指令中，/SWAPFILE是您之前取的档名，/sbin/swapon是启动指令。

```
$ cp /etc/rc.d/rc.local /etc/rc.d/rc.local.old ← 备份，以免万一失手
$ echo "/sbin/swapon /SWAPFILE" >> /etc/rc.d/rc.local
```

或者假如您不喜欢上面步骤三的方法，也可以这样做：

```
$ cd /etc
$ cp fstab fstab.backup ← 先备份，以免万一失手
$ echo "/SWAPFILE swap swap defaults" >> fstab

$ /sbin/swapon -a ← 使生效，而无须重新开机
```

☐ 手工设定 LILO

LILO 是 Slackware Linux 内建开机选择程式，它可以用来启动Linux，也可以用来启动硬碟内的任何一个作业系统。在笔者看来，LILO 拥有是简单好用且强悍且安全的特色，非常值得推荐，最重要的，您安装好 Slackware Linux後，它就在那里。

比较详细的 LILO 介绍，请参考『轻轻松松管理 Linux/Unix』，这里，笔者只稍微介绍而已，补充一些 setup 自动安装程式所没有介绍的。

```
boot = /dev/fd0          # 这行用在软碟开机 启动场合
# boot = /dev/hda        # 这行用在硬碟 MBR 启动 启动场合
# boot = /dev/hda2       # 这行用在硬碟 superblock 启动场合
install = /boot/boot.b
delay = 50
#
image = /vmlinuz          # 启动 Linux
    label = linux
    root = /dev/hda1
    vga=normal
other = /dev/hda3         # 启动 DOS
    label = dos
    table = /dev/hda
```

当 /etc/lilo.conf 如上示范设定完成後，摆一张格式话过的磁片在第一个软碟机内，执行 lilo 指令即可。这张词篇就可以用来开机。

□ 如何解除LILO自动开机程序

本书作者其实最推荐把 LILO 安装在软碟上面，也就是说尽量由软碟开机，而不去动到硬碟的 MBR，这样子，假若以後不跑 Linux了，MBR 里面也不会残留着 LILO，您也就不需要阅读本节介绍。

假如您已经把 LILO 安装在硬碟 MBR，因为任何理由您想把 MBR 清理乾淨，在以後用硬碟开机时不想再看到 LILO 字眼，一般的作法很简单，拿一片 DOS 开机片，先确定该磁片内附有 FDISK.EXE 工具程式。用该 DOS 磁片启动电脑，之後执行 A:/> FDISK /mbr 指令即可。

```
*      作者：曾瑞源      著作权声明      *
```

```
*      yuan@UUserV.Net.tw      *
```

```
*      *
```

```
*      任何个人都可未经授权自行列印阅读，这里「个人」指任何人，
```

```
*      「自行列印」指的是非删改作者原作情况下，自行由电脑印表机
```

```
*      列印。本着版权规范不限制您列印数量，但凡商业性出版使用 ii
```

```
*      转排版印刷都不被允许。      *
```

```
*      *
```

```
*      关于本着作物(电子书)的转发行规定，您被鼓励将之摆放在任何
```

```
*      Internet FTP档案库上 Gopher/WWW站 ii 或者任何联盟的PC BBS
```

```
*      站台，作者欢迎并感激您愿意如此做，因为这样做受益的朋友必
```

```
*      然会更多。最後，本版权声明是本着作的一部份，任何将本声明
```

```
*      与本着作分离的动作已经侵权并违法。其它未定事宜，或您认为
```

```
*      本版权声明有不合时宜之处，请联络作者。我再次感谢您读我，
```

```
*      再见。      *
```

```
*      *
```