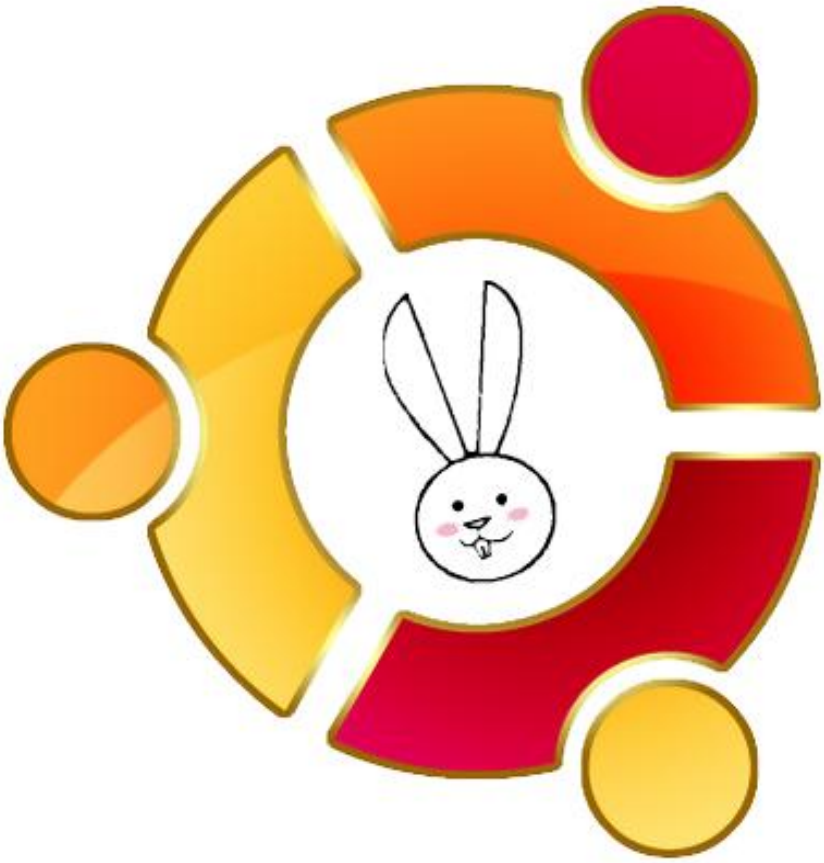
Ubuntu 爱好者们

:) 欢迎来到《笨兔兔的故事》！

作者将以趣味幽默的笔触，把 N 多乏味难读的技术文章，以第一人称形式生动表达，为学习 Ubuntu 提供便利条件。希望你能在《笨兔兔的故事》里寻找到自己需要的内容！



版权声明：

本文档（包括但不止 PDF、DOC 版），根据《署名-非商业性使用-相同方式共享 2.5 中国大陆 》发布，制作者：Yexiaoxing 详情参考

我叫 Ubuntu，主人喜欢叫我“笨兔”，但是我绝对不笨，与某种耳朵长尾巴短的哺乳动物也没有什么联系，其实，我是一个操作系统，我是一个 Linux，我是——Ubuntu。

在 2008 年的 4 月，我来到了这个世界，并由出生日期得到了自己的代号——8.04。当然，和我同一天出生的兄弟们还有很多，我只是其中之一。我们出生之后，每个人坐上一张被人们叫做“光盘”的碟子，奔赴世界各地，寻找自己的归宿。我们只有找到住处，才可以发挥我们的能力，而我们并不像人们一样，住在钢筋水泥的格子里面，我们住的地方，是一块叫做硬盘的空间。

住房的质量严重影响着我工作的效率，房子够不够大，屋里的过道够不够宽敞，都会对我的性能有影响。不过还好，我住的这个地方还不错，房子很宽敞，有 500G 大。不过不是我一个系统住，房子被隔成两个大屋子，每边 250G，我住在右边的一间，隔壁住的是一个 Windows XP，听说这家伙很厉害，不过我来的时候他正睡觉，于是也没有打招呼。（后来我知道，我们两个是不可能同时醒着的）250G 的屋子，对我来说实在是太宽敞了，我很欣慰，除了硬盘大之外，我发现这里其他的设施也是不错的。

硬盘只是我们操作系统休息和存放个人物品的地方，真正工作的时候，是不能在硬盘里的，那时候，我就需要从住的硬盘里，来到工作的地方，我们的工作间——内存。我的这个工作间也比较宽敞，有 4G 大，不过也不是我一个人用，也是要和隔壁的 Windows XP 共用的。鉴于他名字太长，我就管他叫“查皮”吧，不管他愿不愿意了，反正他也不知道。但内存的共用方式与硬盘不一样，不是一人一半，而是谁醒着谁就用，用完了，临睡觉前，一定要记得把内存里有用的东西，都搬回自己的那间硬盘里，以备下次使用。这倒不光是为了另外一个系统着想，主要是因为内存里很不保险，东西放在里面去睡觉的话，等你在醒来，准没，不管你隔壁有没有住着一个查皮。

其他的硬件，我也很和熟悉，这主要是因为我们在 Canonical 学校的时候就进行了充分的学习，所以这里的东西我基本上都会用的比较顺手。像 Realtek 的网卡啦，Intel 的南北桥，声卡以及 E8400 双核 CPU 啦，更是应用自如。Intel 是个大公司，他和我们的关系还是不错的，为我们提供了很多的教科书，说明书，基本都是讲如何使用他们的设备的，所以对于 Intel 的设备，我们都使用的比较好。这里唯一不能完全用起来的可能就是 GeForce 8800GT 的显卡了。不过基本的显示功能还是没有问题的，只是我不知道怎么使用它的 3D 加速功能。这个需要专门的手册，要去 nVidia 那里去要。对周围的一些熟悉以后，就等着我的第一次启动了。

平时，我和隔壁的查皮都是在睡觉。当主人有事情的时候，会让传达室的 GRUB 大叔来叫我们。G 大叔就住在传达室，传达室很小，只有 512Byte，门上贴着牌子——MBR，可能 G 大叔想说自己是个“明白人”吧。由于传达室地方实在太小，所以他会把一些有用的东西放在我的硬盘空间里，必要的时候来看看。他总是面无表情，每次主人来的时候，他就板着个脸说：你找谁呀？是 Ubuntu 还是 Windows XP？快说！就给你 10 秒阿。然后就倒数，如果主人没来得及说，他就会直接来叫我——因为我们熟，他是我带来的。如果是叫查皮，那我就不知道了，因为那时我肯定在睡觉，上面提到过，我们两个是不可能同时醒着的。而这一次，他径直来到我这里，拍拍我说：嘿，醒醒，开工啦！

一听说开工，我很麻利的蹦起来，以迅雷不及掩耳盗铃之势，嗖的一下就跃进工作室——内存里，用最快的速度进入工作状态。主人对此很满意，夸我说比那查皮麻利不少。然后，他下达了第一个命令：先去上网看看，找个快一点的软件源。

于是我赶快叫醒还在硬盘里睡觉的 Firefox——是的，看网页这个事我做不来，就得去找 Firefox，我喜欢叫她狐狸妹妹。狐狸妹妹轻移玉步，走进工作间——速度有点慢，不过还可以接受。然后开始工作，一下子找到某菜鸟入门新手指导帖之类的，找到一些著名的软件源列表，如 cn99 之流。然后，主人决定记录下这些地址，所以，我又叫醒了 gedit 小弟——对，编辑文件这件事我也做不来，得去找 gedit。gedit 个头很小，身体轻盈，一下子蹦进来，开始干活。然后，我又在主人听歌的要求下去叫醒 Rhythmbox——没错，你猜对了，放歌这事我还是做不来，后来又去叫 pidgin，去叫 Apt……等等，有人问为什么你总是叫别人干活，自己不干？我正在干，我要干的就是——叫人。

是的，我号称叫做操作系统，听起来好大的一个软件阿，好像操作系统就应该是啥都能干。不过，其实我们作为操作系统，并不能直接完成任何你需要的任务。我需要很多帮手，他们各自帮我完成各种不同的任务。可以说，我们在一起，才算的上一个系统，而我，是核心，是领导。没有我，他们不知道该做什么，而没有他们，我也不知道该怎么做。我们操作系统的最基本的职责就是管理，管理各种程序的执行，管理硬件资源的使用。比如 CPU，就是我们程序要用的重要设备，每个程序都要用，可是 CPU 很贵，不能发给每个程序一个（否则主人会破产）。狐狸妹妹来了，我会把 cpu 给她用，gedit 小弟也来了，他也要用，那么我就告诉他俩，一人用一会儿。但是这可不是每人 1／2 这么简单，狐狸妹妹要做的事情比较复杂，那么就让她多用一会，gedit 的工作很简单，就让他少用一会。主人关心

天台上，风很大，只有我们两个，面对面

-给我一个机会

-怎么给你机会

-我以前桌面不行，现在我想做一个好系统

-好啊，跟我CEO去说，看他让不让你做

-那就是要我闭源

-对不起，我是微软

-谁知道

十年后你还是不是

的程序，就得多用 cpu，主人不是很关心的，就可以少用一点 cpu。有的程序脾气不好，把着 cpu 就不放，我必须处理，有的程序确实工作量大，需要使用 cpu 相当长的时间，可是我也不能就真把 cpu 全都给他用，还是得让其他的每个程序隔一阵子都能用上一会，不至于一直闲着。而有的程序平时基本不需要做什么工作，可是又不能把他请回硬盘休息，那么我就要允许他在工作间睡觉，只是在必要的时候叫醒他，并把 cpu 给他用……怎么样？是不是有点乱？当个操作系统是很不容易的，当个好的操作系统，就更不容易了。当然，作为一个有理想的年轻人(或者说，年轻系统)，我是很信心做一个好操作系统的。

做个好系统，就要借鉴前人的经验，取长补短。听说隔壁的查皮就很牛，虽然不能和他直接对话，至少也侧面了解一些他的资料吧，也许有值得我学习的地方。于是借主人上网的时间，我让狐狸妹妹帮我找找相关的资料。狐狸妹妹虽然起床慢点，干活还是挺快的，特别是装备了 firefox 扩展以后。

查皮是个挺有名的操作系统，也算得上是名门之后。早在 1985 年，稍微有点软的公司就造出了查皮的老祖宗——Windows 1.0 名字比较土，不如查皮显得青春活力。而实际，这位老人家的表现也确实不咋样，大家都没怎么拿正眼看他。我还找到了他的一张照片，怪难看的。

两年后，1987 年 12 月 9 日，第二代 Windows 上市了，那时候的人都懒得起名字，于是就叫 Windows 2.0。还是那个有点软的公司，还是那张脸，跟他爹长的还真一样。

Windows 1.0 照片

并且，不单脸长得一样，遭遇跟他爹也差不多，基

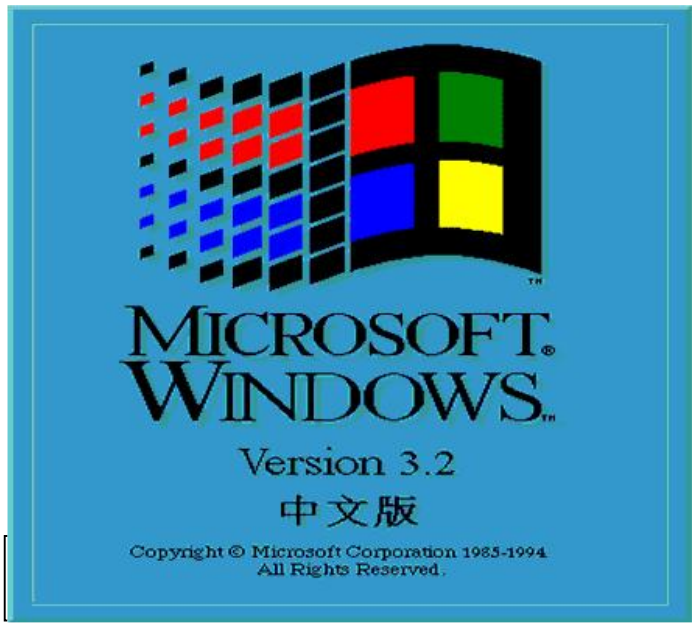
本上被打入冷宫，不被大家看好。不过 Windows 2.0 还是有一个地方比 Windows 1.0 强——2.0 有个好儿子，1.0 没有。

Windows 2.0 照片

Windows 3.0 终于让那个有点软的公司硬起腰板来了。相信很多老鸟也都是从 Windows 3.0 的一个重要分支版本 Windows 3.2 开始认识 Windows 的吧。1990 年 5 月 22 日，Windows 3.0 正是发布，第二年，1991 年发布了 Windows 3.0 的多国语言版本。而同年，一件更重大的事情发生了。

在 1991 年，芬兰，赫尔辛基大学的一名大学生的电脑上，我们的老祖先，Linux 的雏形正在一点一点的完善起来……代码一行一行的流入他的身体，那时候，Windows 3.0 已经广为人知，已经进入图形界面的时代，已经能够支持多种语言，而我们的老祖先还仅仅能够对世界说一句 Helloworld！可十几年后之后，就是另一番景象了。

再回来说 Windows 他们家吧。1992 年 Windows 3.1 出生，算是 3.0 的改进版，他增加了基本的多媒体支持和 TrueType 字体。TrueType 区别于点阵字体，可以放大缩小，看起来更好看。1994 年又发布了 Windows 3.2，相信他的样子大家都见过，什么？你没见过？好，贴张照片吧。



直到 1995 年，Windows 家的祖坟上终于冒青烟了，Windows 95 一下子把人们从 DOS 时代领进了窗口时代。出色的多媒体性能，人性化的操作，美观的界面（跟 dos，Win3.2 比）加上有点软公司强大的宣传攻势，

那时的 Windows 95 简直是家喻户晓，妇孺皆知，老少皆宜，人人必备，真乃居家旅行月黑风高杀人放火之必备良品。除了明显的，看得见的不同之处外，Windows 95 与他的前辈们的一个重要区别是，他是一个 16 位 / 32 位混合的操作系统，之前的那些都是 16 位的。也就是说，Windows 95 标志着个人电脑 32 位软件时代的开始。（顺便说一下，我们现在基本上还生活在 32 位软件的时代，隔壁的查皮就是 32 位的，而我嘛……是 64 位的^_^）总之，无论从哪方面说，Windows 95 都是成功的，那个“开始”按钮，留在了那个时代的历史中，并一直流传到了现在。

成功的东西要发扬广大，3 年后，Windows 98 问世了。这个系统是基于 Windows 95 编写的，他修正了 Windows 95 的近 3000 多个 bug（英语老师：你这里怎么不加 s！），添加了桌面主题等新的视觉特性，更重要的——他捆绑了 IE。有点软的公司终于意识到互联网的重要性，把 IE 作为基本的软件随系统一起安装。（其实 Windows 95 里面也有 IE，只是放在一个不起眼的阴暗角落里，生怕人知道似的）第二年，1999 年 6 月，Windows 98se 发布，也就是 98 第二版，他提供了 Internet Explorer 5、Windows Netmeeting 3、Internet Connection Sharing、对 DVD-ROM 和对 USB 的支持等，可以说，从 Windows 95 到 Windows 98se 这一脉，到这里达到了顶峰，也走到了尽头……

顺便说一下，Windows 95 一脉还有一个家伙——Windows ME。出生在 2000 年，不过，基本上可以忽略他的存在，他的一些激进性的改动没能获得广大用户的认同。重要的修改是系统去除了实模式 DOS，而由系统还原代替了。在概念上，这是一个大的改进：用户不再需要有神秘的 DOS 行命令的知识就可以维护和修复系统。但实际上，去除了实模式 DOS 功能对维护来说是一个障碍（现在的查皮都还有命令提示符，命令行才是王道阿），而系统还原功能也带来一些麻烦：性能显著的降低、硬盘空间的大量消耗，并且对一些通常的错误还原并不一定有效。（比查皮的系统还原差远了）所以，基本上可以把他算作 Windows 98 的一个不成功改版。

或许你会问，Windows 98se 之后，不是还有 Windows 2000，还有你隔壁那个查皮么，

怎么能说走到尽头呢？听我慢慢道来……

回到 1993 年，Windows 3.1 获得成功后，有点软公司不满足于个人用户的市场，开始进军服务器。于是，基于 OS/2 NT 的基础编制的 Windows NT 3.1 问世了，Windows NT 3.1 是个 32 位的系统，比桌面系统提前进入 32 位时代，由于是面向服务器领域，Windows NT 3.1 的稳定性要比桌面系统高很多（当然，跟 Linux 比……就算了）。不过这个版本似乎不如下一个版本名字更广为人知——Windows NT 4.0

1996 年 8 月，Windows NT 4.0 发布，增加了许多对应管理方面的特性，稳定性也相当不错，这个版本的 Windows 至今仍被不少公司使用着。Windows 的 NT 内核家族从此登上历史舞台，与以 Windows 95 代表的一脉并行发展。当千禧年的钟声敲响后，NT 家族的又一个精英——windows NT 5.0 问世了，为纪念千禧年，他还有另外一个名字——Windows 2000

Windows 2000 是主要面向商业的操作系统，他有 4 个版本：

Windows 2000 Professional，用于工作站及笔记本电脑，可以叫工作站版。

Windows 2000 Server 即服务器版，面向小型企业的服务器领域。

Windows 2000 Advanced Server 即高级服务器版，面向大中型企业的服务器领域。

Windows 2000 DataCenter Server 即数据中心服务器版，面向最高级别的可伸缩性，可用性与可靠性的大型企业或国家机构的服务器领域。外号：最牛版

好了，该我的好邻居出场了，我们就把时间定格在 2002 年。从 98 问世到 2002 年，桌面市场的 Windows 4 年没有什么变化了（Windows ME 是垃圾，Windows 98se 不过是 98 的升级版），而随着时代的发展，Windows 95 一脉的内核越来越不能适应现代软硬件需求的发展，于是，有点软的公司一咬牙一狠心一跺脚一哆嗦一没拿住——扔了！然后，拿过来一直表现良好的 Windows NT 系列的内核，开发出了新一代桌面版 Windows——Windows XP，也就是住我隔壁这位。Windows XP 有两个版本，Home edition 和 Professional。Home edition 面向家

庭用户，相当于 Windows 98。 Professional 面向工作站，相当于 Windows 2000 Professional。而对应 Windows 2000 其他用于服务器版本的系统，是 2003 年的 Windows 2003，也是一个很优秀的版本。好了，今天就到这了，主人要关机了，改天再来。

今天一起床就接到了一个任务，挺起还还听轻松，一般胡同里大妈大婶的，经常做这项工作，并且乐此不疲，这就是——串门。

事情是这样的，在我来之前，主人有许多照片存在了查皮那屋里，而现在可能他觉得这么珍贵的回忆全都堆在查皮那杂乱无章（在我看来）的屋里很不保险，所以，要我去复制一份，放在我这里。于是，我终于有机会去查皮那屋里看看他和他的同志们都长啥样。（前面说过，操作系统啥也干不了，他肯定也有一堆帮忙的应用软件，就像我的狐狸妹妹）不过，我串门可不像胡同大妈串门那么简单，认识门就行。我要想去查皮那屋，就得认识查皮在屋里规划的格式，或者说，认识查皮屋里的地里形势，或者说，认识查皮所用的文件系统。

基本上，查皮只会两种文件系统——换句话说，只会用两种方式规划整个屋子的空间，那就是 **FAT32** 和 **NTFS**。**FAT32** 是一种很老旧的格式了，连 **4G** 以上的文件都不支持，性能也不好，还不支持多用户的权限，所以基本不怎么用了。这个查皮也是，没有用 **FAT32**，而是用了另一个比较高级的格式——**NTFS**。那么，我就必须能够读懂 **NTFS** 格式的磁盘，我才能去查皮那里串门。要说以前，我们 **Linux** 是不太能读懂 **NTFS** 格式的磁盘的，毕竟是微软私有的格式，我的前辈们基本上只能勉强自从 **NTFS** 的磁盘上读取东西，往里写是不行的。不过自从 **Canonical** 学校为我们增加了一本 **ntfs-3g** 教材以后，读写 **NTFS** 就都不在话下了。不过虽然能够读懂，但是我自己是不会用这个文件系统的，我会用很多其他的文件格式，比如 **ext2**，**ext3**，**xfs**，**jfs**，**reiserfs**，**ufs**，**zfs** 等等，各有优势，我现在的屋里使用的是非常强大的 **xfs** 格式，至于怎么强大，以后慢慢细聊，现在，我要走了，去串门。

来到查皮的屋里才发现，我串门跟大妈串门的感觉实在不一样。大妈去串门得在人家醒着的情况下去。我来到这里，只能在他们睡着的时候。感觉我不像串门来的，反倒像小偷-_-b。要说这查皮还是真是不会收拾屋子阿，一地的磁盘碎片，多影响性能阿，我说他起床怎么那么慢呢。有人问:“啥碎片呀？我家有时候也有碎片，都是老婆和我吵架时候摔的……”对此，我可以用正宗的 C 语对你说 ： printf(“……-_-b\n”);

好吧，先不串门了，科普。

笨兔兔老师第一讲：什么是磁盘碎片

同学们都坐好啦，都把手机铃声关了，小灵通调成震动，BP 机直接扔了——台都没了你还留着它干嘛。好，上课了，首先说说什么叫磁盘碎片。磁盘，是我们程序居住的空间，我们用不同的方式对整个磁盘的空间进行管理。刚才说过了，包括各种方式，什么 ext3，xfs，查皮的 ntfs 等等。而磁盘里放的东西，就是一个一个的文件，同学们可以把磁盘想象成你家的屋子，文件就像一个个大大小小的箱子。每个箱子上面写着字，就是文件名。查皮喜欢把每个箱子都紧挨着放，一个挨一个，上下左右前前后后都紧贴着，这样，看上去很规整。可以让剩余的空闲空间比较完整。有同学说了，我家也这么收拾，这样很利索呀。不过，对于操作系统，这样做虽然也有好处，但是会有一些问题。

比如，一开始存了一个文件，也就是搬来了一个箱子，比如叫“日记”。查皮把它放在最靠墙的位置，然后又存了很多其他的文件，在“日记”文件的前前后后，左左右右，上上下下都放满了。忽然这一天，日记文件被修改了，加了点内容，就相当于往“日记”那个箱子里加了东西。可是箱子已经满了，再往里加，箱子就要增大(也就是文件大小变大，毕竟是比喻，不是真的箱子，大家不用费脑子想箱子怎么会伸缩。)，可是箱子周围堆满了其他的箱子，没地方了，怎么办呢？可以把边上的箱子挪开一点，原来的箱子就可以扩大了。可是边上的箱子要是少还好办，要是很多，还都装的铅块铸铁大理石阿什么的，那可就累死了。那怎么办呢，只好把新的内容放在另一

个小点的箱子里，放在别处。然后还得在原来的“日记”箱子上标注上：“日记（第一部分，第二部分在东墙根 ）”。然后在新的箱子上写：“日记（第二部分，结束 ）”。如果日子长了第二个箱子也被 n 多箱子挤在中间后，又要编辑日记文件，这个文件又变大了，就又要如发炮制出第三个箱子，乃至第四个，第五个…… 等到有一天，要读取这个日记文件的时候，查皮就开忙了——首先，到西墙角找到日记第一部分，翻腾出里面的内容，然后往箱子上一看“第二部分见东墙根”，然后查皮在跑到东墙根找第二个箱子，翻腾出里面的内容，然后再一看箱子“第三部分见大衣柜上头”，然后查皮搬梯子，上大衣柜一看“第四部分见厕所水箱后边”，再折腾到厕所“第五部分见屋子正中间从南墙数第两百四十八个箱子”……等到查皮把整个日记文件读完了，也累得半死了。这种情况，就是会影响性能的磁盘碎片。好，本节课到此结束，同学们自由活动吧，那位同学，快去捡你 BP 机去吧，说不定还能找着。

科普也科普完了，该干正事了。开始搬照片吧。

先拿出这屋的文件列表来看看——我当然知道文件列表在哪，因为我学过 NTFS 格式。好，上面写着，照片在窗台底下，好，我来到窗台底下，没看见照片，却发现了一个熟悉的面孔……

他带着个圆圆的眼镜，文质彬彬的样子，看上去像个学究，两道浓眉如同飞翔的海鸥。衣着并不华丽，倒也搭配的很是顺眼。我这里有他的照片：

OpenOffice.org 2.4 照片

人们喜欢叫他 OOo，可能是因为他的眼镜吧，而他的全名，叫做 OpenOffice.org——相信我，这确实是个软件的名字，当然，同时还是个网站的名字。之所以我认识这家伙，是因为在我屋里也躺着一个。

这并不奇怪，很多 Linux 下的软件都有相应的 Windows 版本，OO 老先生也是这样。基本上这个 OO 可以算是我屋里那个的兄弟吧，他们是相同的版本，相同的外表，相同的功能，只是一个跟着查皮混，另一个跟着我干。我绕过这位 OO 老先生，没有吵醒他的美梦（事实上我也叫不醒他 ）。终于自他身后的窗台下面发现了要复制的照片，不过别急，仔细看一下，果然，上面

写着“照片，第一部分，第二部分见里间屋写字台底下”哎～～我恨碎片……

来到里间屋，还没找到照片，先看见了床上躺着的查皮，这是我第一次看到这位可爱的邻居。他穿着红黄蓝绿四色的衣服，很是鲜艳。可是，不知道为什么，脸被涂黑了，上面还写着“使用正版，跟风黑屏”。看来主人是不希望自己的电脑里有盗版软件，所以才会在 Windows 下也用 OpenOffice。估计这个查皮是买电脑时候一起来的正版查皮。一边想着，一边来到写字台底下，找到了照片第二部分，往盒子上一看：“第三部分见……”Oh，God!

费了半天劲，终于把照片都拷贝到了我的屋子里，把它们放在了专门放主人文件的分区下。有人忽然想问，查皮那里那么多碎片影响性能，那你怎么放这些文件呢？我更倾向于把文件都分散的放置，来了一个文件，我把它放在屋子正中。再有第二个文件，就放在第一个文件与墙的距离之间的正中位置，依此类推。这样，文件的码放也相对规律，文件与文件之间又留有一定的空隙，就不会产生碎片了。当然，这也不绝对，如果文件特别多，快把屋子占满，以至于我实在倒腾不过来的情况下，可能也会出现碎片。不过这种情况还是相对较少的。

刚刚休息了一下，主人又让我去叫醒一个家伙，他叫作 apt-get，什么？你不认识他？

人物志——超级牛力

这个家伙就像个公司里的人事部经理，来个软件走个软件的，都是他管。当别人夸奖他的时候，他总是自信的拍拍自己的胸脯说:“本 APT 有着超级牛力”。而他也确实很厉害，很敬业，也很专业，对于人才（对我来说也就是软件）的各种情况了如指掌。要招一个人来的时候，他会做好所有准备工作，这个人需要用什么样的库，或者需要什么其他的人才能一起协同工作，他都会事先做好准备。比如，主人想用 vim 来编辑文件，就叫 apt 去招 vim 来。apt 就会报告，说 vim 要来的话，首先需要准备好 libncurses 这个库，和 python 这种脚本语言的执行环境。征得同以后，他就会去网上找这些东西，并且运回家，把库放在该放的地方，相关的软件安排好住宿，然后再去找 vim 同志，请他过来帮忙干活，并且说明，环境都已经布置好了。每次新人来了之后都很感谢 apt 同志为自己做的这些准备工作，该有的东西，该来的助手都在，于是干活就事半功倍了。但把人才请来之后，apt 同志的工作还没有结束，他还要把现在的人事情况记录下来，以便主人哪天问起来的时候好如实汇报。哪天主人问一句:“我说超级牛力阿，咱这现在都有多少软件阿，都是谁阿？”apt 也能从容的回答。可以说，apt 这家伙对于我来说实在是非常重要的，没有他，笨兔就不是笨兔了。

不过 apt 也不是光为我们服务，以前，其实 apt 是 Debian 公司的人事部经理，人家 Debian 可是历史悠久的大公司，1993 年就成立了。apt 在这么大的公司里一直工作到现在，有大量的工作经验。当我们公司成立的时候，成功的请来他管理人事资源（当然，这并不影响他继续给 debian 打工），对他来说自然是轻车熟路，得心应手。21 世纪什么最宝贵？人才!绝对是人才！超级牛力这样的人才！

好，认识了吧。主人叫 apt 去找一个软件，名叫 preload。apt 同志翻开他那厚厚的记事本，查到了 preload 的资料，然后向主人报告：preload 工作需要的条件我们这里都已经满足了，可

以直接把他请来。本 APT 有超级牛力～～～～然后，在获得主人的同意后，apt 出发了……

“比海更广阔的是是天空，比天空更广阔的，是人的心灵。”

500g 的容量算不上海量，而屋外那个世界，却实在算的上比天空更广阔了，那就是网络。一个操作系统是孤单而无助的，只有接入了网络的操作系统，才真正能够发挥全部的能力。尤其是对于我来说，尤其是对于有 apt 做帮手的操作系统来说。apt 可以从网络上获得各种软件的资料并记录下来，当需要的时候，只要跟他说一声:“我要 xxx 软件”，apt 就直接去找去了，下载，安装，全都不用别人操心，他都给办了。如果没有网络，apt 到也不是全无用处，至少他可以用来管理安装光盘上的软件，可是就光盘那点容量，我自己都管理的过来，要什么软件自己搜索都不会慢多少，就体现不出 apt 同志的“超级牛力”了。除了 apt，很多其他的软件都是跟网络离不开的，比如狐狸妹妹，要是没网络，她就可以退休了。说起网络，实在是个很有意思的世界，有很多有意思的东西，不过，一个软件要想能够从网上取得信息，就需要懂得网络上的说话方式，懂得网络交流的语言，我们管它叫做——协议。懂得 http 协议的软件可以看网页，懂得 ftp 协议的软件可以传文件，不过这些都是上层的协议，底层，基本所有能上网的软件都要会的，算是 tcp/ip 协议了，apt 就懂得这门协议，所以，他可以去网上找想要的软件。

转眼间，apt 已经把 preload 请来了，并且做了一下安顿，完事后，看看没有什么其他的工作了，他就回硬盘去睡觉去了。要说 preload 这家伙我还真是没见过，不知道他是干什么的，有什么本事，不过既然主人要找他，总是有原因的吧。鉴于他名字念着不顺口，我们就叫他老 p 吧。

老 p 好像很勤快，一来了就跑到内存里准备干活，我正要看他到底会干些什么，哪知他什么也不做，就在那里看着别人忙活，时不时拿出个小本，记录着什么。一直到关机，大家都去睡觉了，他都没说话。第二天一开机，他又早早的跑进内存，看着别人忙忙碌碌，拿着小本记录，还是一言不发，还是不做多余的事情，直到再一次关机。第三天，第四天，涛声依旧……这家伙到底什么来头？

终于有一天，一如既往的起床，一如既往的看老 p 跑进内存，本以为他会一如既往的待在那里，一言不发，没想到他竟然说话了：FireFox 赶快起床，做好准备。我莫名其妙的看看老 p，心想：主人还没有发命令要用 FireFox 阿，怎么就把她叫醒了呢？再看看狐狸妹妹，只见她一如既往的伸个懒腰，一如既往的揉揉眼睛，一如……咦？忽然觉得不对，怎么今天叫床（-_-b）的声音跟以前不一样呢？扭头看看我问：头，你叫我？我愣愣的指指老 p：他……　老 p 却完全无视我俩的茫然，转脸又说：Audacious 起床，做好准备。Audacious 是一个多媒体软件，他会使用那个叫做声卡的硬件设备，唱出优美的歌声来。我问过我们这里学问最高的星际译王老先生，星爷告诉我 Audacious 这个名字是大胆，鲁莽的意思。大胆，唱歌，恩……我们就管这个会唱歌的家伙叫“想唱就唱”吧。想唱就唱也被老 p 叫了起来，跟狐狸妹妹一样迷惑的跑进内存，刚要问什么，这时主人发话了，要开网页。我马上明白了，看了一眼狐狸妹妹，她也很麻利，立刻进入工作状态。省去了平时狐狸妹妹起床的时间，反应快了不少，主人很满意。没过多会，主人果然又叫想唱就唱来唱歌了，一切都在老 p 的预料之中……

原来，老 p 这几天一直在记录分析主人的使用习惯，获得足够的数据之后，就可以知道哪些软件是常用的，哪些是不常用的，哪些软件哪些时候用，哪些软件哪些时候基本不用，正所谓金风未动蝉先觉，春江水暖鸭先知，主人用谁他先叫。有了他，整体系统的反应速度提高了，这就是他的能力，这就是他的本事，这就是他的价值。在这个世界里，没有一个程序是无用的，每个人都是人才——不同方面的人才。

自从那一次大家见识了老 p 的本领之后，都很乐意的听候他的调遣，整体的工作效率提高了一些。不知道查皮那里有没有类似的角色，于是就拜托狐狸妹妹去网上问问，结果发现在查皮发布的时候，有点软的公司就宣称，查皮有类似的功能，可以记录用户对软件的使用情况，使用的多的软件就能够较快的启动。而让人不解的是，5 年后，查皮的下一代，长得比他漂亮的 Vista(看到这个词，总让我想起 Visa，于是我总觉得这个系统很贵)系统发布时，有点软公司还在宣传，

Vista 系统增加了记录用户习惯的功能，用的多的程序将得到更快的启动速度。也不知道到底是加了没加，反正他们公司的系统，总是越用越慢倒是真的。到底为什么慢，我也说不清，因为他是一个闭源的系统。

我就知道你想问什么叫闭源。

笨兔兔老师第二讲：闭源和开源

什么是闭源呢？就是源代码不开放。我们知道，程序是程序员们一行一行的语句编出来的，c 语言也好，java 也好，这一行一行的语句，就是这个程序的源代码。有了源代码，就能够 100%的了解整个程序的构造，如何工作。而源代码是不能运行的，必须要把源代码变成可执行的二进制程序，这个过程叫做编译。源代码经过编译之后，才可以运行，但是编译之后的程序就不能够知道内部的构造了。我们平时在网上下载的各种程序，都是编译好的二进制程序，如果你想要它的源代码，对不起，不行！这是商业秘密，怎么能给你？给了你，我们的软件怎么卖钱？这种不开放源代码的程序，就叫闭源程序。打个比方，就好像肯德基。麦辣鸡翅谁都可以得到，只要花钱买就行，但是配方没人知道（虽然其实也没多好吃吧）。配方就相当于源代码，麦辣鸡翅就相当于编译好的二进制程序，制作过程就相当于编译过程。如果有了配方（源代码）你就可以自己作麦辣鸡翅（自己用源代码编译出二进制程序），甚至还可以根据口味对配方进行修改。（根据自己的需求修改源程序，为软件增加自己需要的功能）

既然有闭源，那是不是还有开源呢？你答对了。Linux，就是一个开源的系统。

开源是什么？开源是一种精神，是乐于分享的理念。再举个例子，有一天你发现，蒸鸡蛋羹的时候往里面加点牛奶，可以让鸡蛋羹更滑嫩。知道了这个窍门，你很高兴的把它告诉你的朋友，让他们分享你的经验，于是大家很高兴的也学会了做这样的鸡蛋羹。这就是开源。你也可能不把它告诉别人，而是保留这个秘密，甚至申请个专利，然后开个店去卖京城独一份的奶香滑嫩鸡蛋羹。这就是闭源。当然，这之中没有谁对谁错，谁好谁坏，只是不同的理念而已。

以前讲过查皮他家的历史，现在就来说说我家的故事。话说 1991 年，那是一个夏天。有一位牛人在世界的互联网上画了好多圈——“Hello everybody out there using minix——I’m doing a (free) operating system”（英文圈多……）大家可能看不明白，我来逐一解释一下每个单词：第一个，Hello，这个是打招呼的意思，哦，你知道啦，那说第二个。everybody，每个人，跟我念，爱～唔～瑞～八～迪～，哎呀……呃，好了好了，不要着急，把西红柿鸡蛋都收起来吧，我直接说重点——minix。Minix 是一个操作系统，要想说清楚 Minix，得先从 Unix 说起，好咱一个一个来。

有你可死（Unix）——像爱情宣言？

UNIX 是一个历史悠久的系统。1965 年，鼎鼎大名的贝尔实验室加入了一项由通用电气(General Electric)和麻省理工学院(MIT)合作的计画——制作一套多使用者，多任务，多层次的 MULTICS 操作系统。贝尔实验室的大名大家都知道，晶体管、发光二极管、数字交换机、通信卫星、电子数字计算机、蜂窝移动通信、有声电影、立体声录音，以及通信网的许多重大发明都诞生自这里。麻省理工大学那更是历史悠久，技术雄厚。因为有这样的强强联合，所以，这个 MULTICS 操作系统的项目在 1965 年成立，到了 1969 年就……被取消了，主要原因是进度太慢。可见编操作系统不是一件容易的事儿。

真是世事难料阿，看似事情就这么结束了，然而，其实故事才刚刚开始，因为一位英雄的出现。

Ken Thompson 也在这个计划中，计划取消了，他很郁闷，因为他编了个星际旅行的游戏，没法玩了。这个程序之前运行在一台型号是 GE－635 的机器上，这个机器的系统大约就是他们

计划开发的 MULTICS 系统，但是反应比较慢，玩起来不爽。Ken Thompson 满怀希望的憧憬着项目完成的时候，系统能够优化到顺利的跑起来他的游戏，然而项目竟然取消了，怎么办呢？毛主席教导我们说：自己动手，丰衣足食。我估计 Ken Thompson 没有背过毛主席语录，但是他用自己的行动证明了其正确性。他在墙角淘换出一台 PDP－7 的机器，并且伙同 Dernis Ritchie 将星际旅行移植到了这台 PDP－7 上。这台幸运的 PDP－7 因此在历史上留下美名。就是这台：

当然，要想运行这游戏，得有个系统，系统从哪里来？不会从天上掉下来！还得自己动手。于是 Ken Thompson 和 Dernis Ritchie 又用汇编语言写出来个系统，这就是最初的，非常简陋的，UNIX 的前身。而这一切的努力，就是为了玩个游戏。-_-b

就这样，两位牛人完成了 UNIX 的最初雏形版。这个系统只支持两个使用者（估计做的时候没考虑别人，够他俩玩的就得）相对于那个 MULTICS 系统——MULTiplexed Information and Computing System，Brian Kernighan 开玩笑地戏称他们的系统其实是："UNiplexed Information and Computing System"，缩写为"UNICS"。后来大家取其谐音，就诞生了 UNIX 这个词。这一年，已经是 1970 年，史称 Unix 元年。直到现在，计算机中都是用 1970 年 1 月 1 日 0 点 0 分 0 秒为原点来记录时间。（计算机中的时间记录的是自 1970 年 1 月 1 日 0 点 0 分 0 秒开始，到现在经过的总秒数，再用这个秒数计算出年，月，日）后来， Brian Kernighan 觉得用汇编写的系统不好维护，于是……他发明了 C 语言（符合他一切自己动手的风格），然后用 C 语言又重写了一遍。从此，Unix 走上了发展的快车道，并且一直用到现在。许多世界级的大服务器，用的都是 Unix 系统。

没你可死（Minix）——到底什么时候死？

要说 Unix 确实是很牛的，很有技术含量的，是值得学习计算机科学和操作系统的同学们学习的，然而，Unix 也是天价的，广大穷苦的大学生们是买不起的。荷兰阿姆斯特丹的 Vrije 大

学的 Andrew S. Tanenbaum 教授深刻的体会到了这一点。他的学生们学习了计算机学习了操作系统原理，总得实践一下吧？总得找台机器用用吧？要用计算机就得有操作系统吧？买个 DOS 装上？虽然那时候 DOS 已经问世了，但是这么一个单用户单任务效率也不高的操作系统，实在不能指望它培养出什么软件人才。装个 Unix？学校还不想破产。于是 Andrew S. Tanenbaum 牛人拿起键盘——咱自个儿编一个吧！然后 Minix 就诞生了。Minix 取 Mini Unix 之意，自从 1987 年被编写出来，到 1991 年发展到 1.5 版，现在有两个版本，1.5 和 2.0。这个操作系统的初衷，是作为一个用来学习的模型。所以功能很简单，体积也很小。并且以后也没有进行进一步的开发和扩充，为的是能够让学生在一学期内能学完。那时候 Minix 在大学中用于教学是免费的，但是用于其他用途是需要给钱的。不过现在已经彻底免费了。它作为一个操作系统，其实并不算优秀，但它是一个源代码完全开放的操作系统，这使得有理想有志向有报复的黑客们，第一次能够完整的阅读到一个操作系统的全部代码。这其中，就包括芬兰赫尔辛基大学的学生，Linus Benedict Torvalds ……

离你可死（Linux）——靠，怎么还没死

那时候，Linus 是赫尔辛基大学计算机科学系的二年级学生。他的最大爱好，就是虐待计算机。测试计算机的能力和限制，整天研究怎么让计算机按照自己的想法去干活，怎么发挥计算机最大的性能，把可怜的机器累得精疲力尽呼哧带喘直到电容爆浆，吐血身亡。在学校，计算机上装的是教学用的 Minix 系统，虽然适合拿来学习，不过系统本身并不强大，渐渐的，这个教学用的操作系统已经不能满足 Linus 大侠的欲望，可是似乎又没有别的选择。上面说过了，Unix 奇贵无比，而且无论 Unix 还是 DOS，他们的代码都是不开放的，这系统只能拿来用，没法拿来折腾的。于是象其他牛人一样，Linux 自己动手了。

今天，我们知道，Linus 从那时起开始了一个事业，一个神话，但在当时，他并没有想那么多，只是为了学习 Intel386 体系结构保护模式运行方式下的编程技术。他并不知道自己即将创造的是一个在世界范围广泛使用的系统，而只觉得是自己一时的异想天开。因此，一开始他把自己写的这个操作系统命名为 FREAX。就此开始了这个“异想天开”操作系统的编写。大约 1991 年 4 月份的时候，就编写出了第一个可以运行的版本——0.00 版。这个版本可以启动，运行两个进程，分别在屏幕上打印出 AAA，和 BBB，然后……就没了。虽然连句整话都不会说，不过这是一个好的开始，至少能启动了。

如果他就这么干下去，估计到今天，就不会有 Linux 这个东西了。一个人的力量是有限的，有道是人多力量大 ，众人拾柴火焰高，多个铃铛多个响，多根蜡烛多分光，一个篱笆三个桩，一个好汉三个帮，三个臭皮匠还顶个诸葛亮……铛！哎呦～ 好吧，就说这么多了。总之，Linus 让他的操作系统和互联网，亲密接触了。于是就有了前面说的这句“Hello everybody out there using minix——I’m doing a (free) operating system”（可算绕回来了）这是他当年在 comp.os.minix 上发布的消息，告诉大家，他正在写一个操作系统。并且，他还把他写的“异想天开”操作系统的代码上传到 ftp.funet.fi 的服务器上让大家下载，以便交流心得，共同学习。这就相当于你跑到网站上发帖子说：我研究出一种萝卜炖牛腩的方法，主料是啥啥啥，配料是啥啥啥，怎么怎么炖，大家都试试吧！（对不起，我又饿了）于是很多有兴趣的人就来尝 Linus 炖的牛腩，哦不对，是尝试 Linus 写的系统。不过当时那个服务器的管理员 Ari Lemke 看着这个异想天开的名字就不顺眼，想想，既然是 Linus 写的操作系统，又是类 Unix，或者说类 minix 的（minix 本身就是类 Unix 的，大家都是一家子），干脆，叫 Linux 吧。

Linux 被公布在网上之后，引来大家的围观，很多人觉得这个东西挺有意思。不过第一个对外发布的 0.01 版 Linux 还有很多的不完善（这简直是一定的）。这里先要说一个概念，Linux 是什么？确切的说，狭义的讲，Linux 只是一个操作系统的内核，他只是各位的 Ubuntu 系统里面

/boot/目录下的那个内核文件 vmlinuz-x.x.xx-xx-generic。就好比汽车，Linux 只是一个引擎，只是大家普遍的把装了 Linux 这种引擎的汽车叫做 Linux 汽车。那么既然 Linux 只是一个内核，要想工作就还需要很多周边的支持，比如文件系统，比如一个命令行程序，比如一些基本的软件。

由于当初 Linus 大侠是在 minix 系统上开发的，所以最一开始 Linux 用的文件系统是借用 minix 的文件系统。可老借别人的总不是个事，还是应该有自己的文件系统，就像查皮的 FAT 和 NTFS。前面说了，文件系统也就是自己管理自己这点硬盘空间的方式，自己的屋子用自己的方式管理，自然最顺手。这时候，来了个牛人叫 Theodore Ts'o。

Theodore Ts'o，1990 年毕业于美国 MIT 大学计算机科学专业。他爱好广泛，喜欢烹饪，骑车，还有折腾电脑（这都不挨着啊～），后来又玩上业余无线电报了，当然这都不是主要的。他看到 Linux 觉得很有意思，于是怀着极大的热情为 Linux 提供了邮件列表服务以便大家一起讨论问题，后来还提供了 ftp 站点来共享 Linux 的代码，并且一直用到现在。除此之外，技术上，他编写了 Linux0.10 内核中的虚拟磁盘驱动程序和内存分配程序。在感觉到 Linux 缺少一个自己的文件系统后，他提出并实现了 ext2 文件系统，并且 ext 系的文件系统一直都成为了 Linux 世界中事实上的标准，任何一个发行版都会默认支持。现在已经发展的遍地 ext3，期盼 ext4 了。Theodore Ts'o 算的上是 Linux 的顶级元老了。

另一位元老，一个英国人——Alan Cox。他工作于英国威尔士斯旺西大学，特别爱玩电脑游戏（又一个玩游戏的，可见玩游戏也不是坏事），尤其是网游（你看你看，还是网游），不过那时候的网游不像现在这样华丽，那时候是字符界面的，能想象嘛？字符界面的网游！那种叫做 MUD——Multi-User Dungeon or Dimension。玩 MUD 当然就得有计算机啊，就得有网啊，所以 Alan Cox 就开始逐渐的对计算机和网络产生了兴趣。为了提高电脑运行游戏的速度以及网络传输的速度，他开始接触各种操作系统，为自己选择一个满意的游戏平台，争取榨干电脑的每一个指令周期。经过自己考虑，他买了一台 386SX 电脑，并且装了 Linux0.11 版的系统。这主要

是因为预算比较紧张，即使 minix 他也买不起。（重复一下，minix 用于教学是免费的，但是其他用途要收费，包括个人用。）于是他开始使用 Linux，进而学习其源代码，并对 Linux 产生了兴趣，尤其是网络方面相关的代码。（整天琢磨怎么榨干他家那点带宽）在 Linux0.95 版之后，他开始为 Linux 系统编写补丁程序，以后逐渐加入 Linux 的开发队伍，并成为维护 Linux 内核源代码的主要人物之一。那个有点软的公司还曾经邀请他加盟，被他有点硬的拒绝了。

再有一位，Michael K. Johnson，他是著名的 Linux 文档计划的发起者之一，写了《内核骇客手册》一书，曾经在 Linux Journal 工作，现在在著名的商业发行版 Red Hat 的公司工作。

当然除了这些大牛，还有更多的大牛，中牛，小牛，牛犊，牛杂，牛尾，牛头肉，肥牛……（唉，又饿了）他们都为 Linux 的发展做出了自己的贡献。他们来自不同的国家，从事不同的职业，他们甚至从未见过面，但是他们为了一个共同的目标，通过网络，一起合作，利用自己的业余时间，义务的帮助 Linux 成长，才有今天这个可以合法免费使用的操作系统。这是什么精神？这就是软件国际共产主义的精神！（好吧，这个词是我造的）

话说这天又去查皮的屋里搬东西，看见有几个生面孔，长得怪猥琐的也不知道干什么的，查皮那还真是什么程序都有阿。刚把东西搬了回来，就见 apt 火急火燎的跑过来:“报告系统，本 APT 有超级牛力！我要用网络。”——是的，任何程序要使用任何硬件资源都要经过我的同意，因为我是操作系统。

我一边查看资源情况一边问:“这回去招什么软件呀？”

APT 说:“叫 AVAST。本 APT 有超级牛力！”

“这软件多大阿？”

“不大，本 APT 有超级牛力！”

“这软件干什么的呀？”

“本 APT 有超……”

“算了算了，我不问了，快去吧快去吧。”

话音未落，apt 就飞也似的跑出去了，从远处还传来他悠扬的声音:“……超级牛力~~！”

唉，就说现在是牛年了吧，也不至于这样啊。

过了一会,“超级牛力”回来了，带回来一个软件，看着软件个头不是很大，长得也比较难看，只有个很简陋的图形界面。我顺手拿过他的手册 man 了一下（Linux 下，你想知道一个软件是干啥的，怎么用，你就 man 他。当然，你得懂点英语。）才知道原来是个杀毒软件，还挺有名气的。可是……我又不中毒，主人装杀毒软件干什么？忽然想到了在查皮屋子里看到的几个萎缩的面孔——莫非是隔壁那哥们中毒了？

果然，在被“超级牛力”安顿好住处之后，AVAST 马上被主人叫进内存去工作。他先是去网上下载了最新的病毒库——就相当于一沓子通缉令，上面写着各种已知病毒的名字，相貌特征，作案手法等等信息，以便杀毒软件查对。下载完毕之后，就见他收拾好工具，整理一下装备，向着那黑洞洞的隔壁，出发了。

说起来，隔壁那个查皮还真是挺害怕病毒的，防不胜防啊。针对他的病毒多种多样，各有各的本领，虐待起查皮来真是八仙过海各显其能。他们有的会伪装成别的软件，比如查皮叫醒“记事本”去干活，却不知真正的记事本已经被病毒一棍子打死了，现在躺在那长得跟记事本一样的家伙已经是整了容的病毒。有的能够藏在正常程序的里面，正在工作的 IE 同志，很可能工作服的兜里就隐藏着病毒。并且病毒们现在基本都会随着查皮一起起床。当查皮被叫醒，伸个懒腰揉着眼睛走进内存的时候，他庞大的身躯后面可能正趴着 40 多只病毒。由于病毒是活的，要杀掉很困难，它们可能会有很多人共同作战，杀毒软件杀掉了内存里的强夫，内存里的大熊会把硬盘里强夫的复制版再叫起来。扭头杀毒软件去杀大熊，强夫会把杀掉的大熊抢救过来，结果谁也杀不了。有的病毒更暴力，自己先跑进内存，一看见有杀毒软件要进来，立马上去一铁锹拍死，然后藏起铁锹装着杀毒软件的声音说：“杀毒软件成功启动，没有发现病毒，噢耶~”然后斜着眼监视着 IE，一旦他要访问什么杀毒防毒相关的网站，二话不说，直接干掉！

但是 AVAST 去查毒就简单多了，因为这时候隔壁的查皮没起床，所有他那边的软件都在睡觉，病毒也一样，所以不会有任何反抗能力。AVAST 过去，只要根据通缉令一一对照即可，只听隔壁那边“阿~~”“厄~~~”“哎呦~”“我死得好惨哪~~~~~~~”等等叫声不绝于耳。过了很长一段时间，AVAST 回来报告：共发现病毒 7 种，共 214 只，全部歼灭。

说起来 214 个还真不算多，经常听说有人一查查出好几千个呢，也可怜了查皮了。有人问，查皮那里的病毒那么可怕，你这里怎么没有病毒呢？好 ~

笨兔兔老师第三讲：为什么 Linux 不中毒

首先我们来了解一下病毒，病毒是什么？其实说简单了，病毒只是一个程序，一个坏坏的程序。既然是程序，就跟其他的正常程序一样，依赖于不同的平台。啥意思？就是说，给查皮打工的，没法给我干活，给我干活的，也不理查皮那一套。我要是拎过一个查皮那边的程序跟他说，快起床干活。他压根也听不懂，闭上眼睛继续睡，语言不通啊。所以，病毒也一样，针对查皮的

病毒传染不了我，针对我们 Linux 的病毒也不可能传染查皮。

那有没有针对 Linux 的病毒呢？答案是有的。第一个 Linux 病毒诞生于 1996 年，澳大利亚的一个叫 VLAD 的组织用汇编语言编写了 Linux 系统下的第一个病毒：Staog，不过这个病毒只是个试验品，只是证明一下 Linux 也会感染病毒。这个病毒会感染二进制文件，获取 root 权限，然后说：你看，我获取了 root 权限。炫耀完了也就算了，并不做任何破坏性的事情。后来也有了一些有破坏性的病毒，但是数量很少，经过科学家计算，一个不装任何杀毒软件或防火墙的桌面用 Linux 在互联网上中毒的几率大约比一个人花两块钱买彩票中五百万后立刻被雷劈中的概率大那么一点点。（这是哪门子科学家算的）病毒少，这是 Linux 不容易中毒的一个原因。可为什么病毒少呢？

话说有一个邪恶的人，出于某种邪恶的目的，他想编个 windows 病毒。他买书学习 Windows 的知识，找熟悉 windows 的高人前辈们学习。经过种种努力，编出了一个病毒，然后把这个病毒放在自己的网站上，只要是使用 windows 系统和 ie 浏览器上网的人一登陆这个网站，就必定中毒。放上去之后，他等着，看着有 1000 人来到了他的网站，看着其中 900 多个纯洁的查皮系统感染了病毒（总有不用 ie 的，防护比较到位的 windows 吧），他很有满足感，他觉得自己成为大牛了。

话说有另一个邪恶的人，出于某种邪恶的目的，他想编个 Linux 病毒。他买书学习 Linux 的知识——不过好像不太好找，好不容易找到基本也都是基础知识。找找高深的吧，还都是英文。好吧，英文的也看，对着字典慢慢研究。哦，对，还可以找找高人指导，不过……也不好找哈，找了半天找到一个高人，拜他为师吧。经过师父指点和自己的努力，他学到了很多 Linux 的知识。然后费劲心机编了一个 Linux 的病毒，然后把这个病毒放在自己的网站上，只要使用 Linux 系统，firefox 浏览器上网的人一登录这个网站，就必定中毒。放上去之后，他等着，看着有 1000 人来到了他的网站——998 个人都是 windows 系统……好吧，好歹还有俩用 Linux 吧，可

其中一个不用 firefox，而是用 Opera。邪恶的家伙咬咬牙，忍！看最后一个——哈哈，这家伙是 Linux+firefox，只要登录准中毒。可是只见着人来了转转又走了，一点事没有，临走还顺手改了自己的主页，上面写着：小子，跟我玩你还嫩点。——师父留……

通过对比我们得出结论——写 Linux 病毒，没前途！

除了以上所说的原因意外，Linux 以及周边软件的开源本质，也导致了病毒较少。比如我吧，主人要装什么软件，都是叫 apt 去找，apt 可不是四处瞎找，而是去 Ubuntu 官方的软件源里去找——因为这些软件是开源的，所以可以随意的拿来，放在一起，做成软件源，供 Ubuntu 们统一下载。官方的东西，自然没有病毒了，哪个娘也不能害自个孩子不是？Windows 就不一样了，它上面的软件基本都是闭源的，要装，得自己上网搜，在某个网站搜到了，下载下来装。可这"某个网站"，就不知道他靠不靠谱了。谁知道上面的软件有没有病毒呢？那么，那个有点软的公司不能也开个官方的软件源，让大家都去他那下软件么？当然不能了，都是闭源的软件，你拿来用都要给人家钱的（当然，也有免费的），拿来分发可能压根就是不允许的。另一方面，Linux 的开源导致了大家都可以对其进行完善，一旦发现漏洞，随便谁都可以去修复这个漏洞，只要他有能力。可 windows 呢？发现了漏洞，也只能漏着，等着有点软公司去修。人家要是不修（比如食堂伙食不好导致工人罢工），谁也没辙。

好了，这节课先上到这里，下次见。

小知识：世界上第一个真正意义上的电脑病毒诞生在 1987 年，名字叫做 C-BRAIN。这个病毒程序是由一对巴基斯坦兄弟所写的，他们在当地经营一家电脑商店，由于当地盗拷软件的风气非常盛行（其实……我们现在……），因此他们的目的主要是为了防止他们的软件被任意盗拷。只要有人盗拷他们的软件，C-BRAIN 就会发作，将盗拷者的硬盘剩余空间给吃掉。（说起来目的还是可以理解的）要说这么个病毒破坏力也还不大，不过毕竟开了先河，自此以后，各种各样的

病毒层出不穷。开始，编写病毒的人多是为了好玩，或者为了证明自己能力。比如 DOS 时代很有名的雨点病毒，发作时键盘暂时锁定，屏幕上的字符（DOS 时代，当然满屏幕都是字）像下雨一样刷刷的往下掉，电脑前那哥们眼泪也刷刷的往下掉，不过别着急，等所有的字符都掉到屏幕底下了，也就完了，您接着干您的。不过，过一会又得刷刷一遍，但是并不破坏什么。相比之下，有些病毒就是恶意的了，比如著名的黑色星期五，算是老牌的了。还有当时名噪一时的 CIH，号称首个可以破坏硬件的病毒——其实就是改写了 BIOS 程序，使主板无法工作。自那以后，现在主板的 BIOS 基本都是可插拔的了。

AVAST 给查皮杀光了病毒以后，据说查皮工作起来顺畅了不少。不过他似乎并不知道发生了什么，仍然很自以为是的摆出一副傲慢的表情，该怎么干活还怎么干活，也不说小心点别再染上病毒。（当然，这一切不是我自己看到的，是听人说的。主人叫戴眼镜的 OO 老先生记录下了给查皮杀毒的前前后后，还拜托狐狸妹妹把这些记录的问题放到了他的 BLOG 上，他们两个聊天的时候告诉我的）我很看不惯他这样的表现，也不喜欢他对权力很强的占有欲。

查皮这个系统在安装结束的时候，会有一个创建用户的步骤，输入用户名，以后就用这个名字登陆了。这个用户是有管理员权限的。当然也可以不输入，无论输入不输入，系统里都会有一个很重要的用户——Administrator

我，也就是 Ubuntu 这个系统在安装结束的时候，也会有一个创建用户的步骤，也输入用户名还有密码，以后也就用这个名字登陆了。这个用户也是有管理员权限的。当然也可以不输入，无论输入不输入，系统里也都会有一个很重要的用户——root

有的同学举手了，说：我知道了，root 就相当于 windows 里的 administrator，有着最高的权限。很好，领悟的很快，但是——并不准确。

Windows 下权利最高的是谁？是 Administrator 吗？很遗憾，不是，是 SYSTEM！也就是系统自己，查皮他自己。任何管理员的权利都不能大于查皮自己 的权利。你可以试试去把 C:\WINDOWS 下的 regedit.exe 删了。能么？“哇！我删了耶，没报错。”别着急，刷新几下看看，是不是又出来了？ 查皮会保护自己，不叫人类破坏。这个初衷看似还是好的，但是当查皮自己中毒的时候，就不一样了。当他中毒时，就像被外星生命寄生了的人类（异型看过吧？），就不再是正常的人了，不正常的查皮仍然会努力保护自己，不让人动他身上的任何部分——包括已经中毒变坏的部分。

那 Ubuntu 下（其他 Linux 也是一样）权利最高的是谁？毫无疑问的是 root！是这个用来给人类登陆的用户。root 在系统中拥有真正的，至高无上的权力，他真的无所不 能，他可以运

行 rm * -rf （危险动作，切勿尝试，后果自负）删除系统中的所有文件，或许我会语重心长的警告他，这么干很危险滴，这么干就都删光光了，这么干我这个系统就嗝屁了，不存在了！但是，当他确认的告诉我，他现在很清醒很冷静，知道自己在干什么之后，我会义无反顾地留着两行热泪按照他的命令去做！哪怕他要格掉整个硬盘，我也照办。这真是，君叫臣死，臣不得不死。他叫我格，我不能不格。（windows 下是不可能在系统运行的时候格掉系统盘的）

会有这样区别的原因，还是我们两个的理念不同。查皮认为，人类是会犯错误的，很可能一不小心就把系统搞坏了，所以必须加以限制，有些事情让做，有些事情无论如何不能让他们做。而我总觉得，人类是聪明的，他们知道自己在干什么——尤其是用 root 登录进来的人，我认为他是了解我，了解整个电脑才会用 root 登录进来做事情的。所以他的命令不会受到任何的阻挠。而一般的用户，会用普通帐号登录，既然用普通帐号登录，就说明他们承认自己只是个使用者，可能会做错事。那么我就会稍微进行限制，让他们不会破坏我，也不会破坏其他用户的东西。所以，当你用 root 账户登录进来的时候，一定不要辜负我对你的信任。

骑白马的不一定是王子，也可能是唐僧。

烧香的不一定是和尚，也可能是熊猫。

蓝脸的不一定是戏台上的窦尔墩，也可能是我隔壁的查皮。

今天听说查皮蓝屏了。他就是这样，好像比较禁受不住刺激，对工作间的要求比较高，一旦哪个程序带进来只小虫子（bug），查皮经常吓的脸色变蓝，念叨一堆英文字母然后就开始倒数，数完了，就把整个机器重启了。查皮的这种毛病让好多人郁闷不已，那他到底为什么蓝脸？蓝脸以后又是在干什么呢？

前面我说过，操作系统的本职工作就是管理——管理硬件资源，管理各种程序。就好像老师管理一个班的学生，老板管理一个部门的职员。不过，无论学生还是职员，都有可能不听话，程序也是如此。查皮整天坐在工作间（内存）里吆喝:“QQ 快起床，IE 呀，你看看这内存里就这么大地方，你一个浏览器要占多少啊。QQ 怎么还没起床啊?快点快点。我说瑞星啊，你能不能别让你那狮子到处乱跑啊，它净用 CPU 了，快把它赶开。QQ 呢？QQ 呢？，怎么还不起啊，再不起老大该怒啦，有 40 多 MM 等着他去聊天呢。啥？你说迅雷占着网络你起来也没用？唉，迅雷你也是，就那么点带宽，就说你下的这个什么 ubuntu dvd 挺大的吧，就不能留个 5k，10k 的给 qq 用用？你瞧瞧人家 IE，也能下载，人家那……咦？IE，你怎么站那不动了？IE!IE!靠，又没响应了，还得拍晕了从来……”每天在这样的高强度压力下工作，查皮有些心力交瘁。怪不得查皮连续不断电的干上几天就不行了，而我可以连续干上几个月都没问题。查皮的神经就这样每天紧绷着，程序来个假死什么的还算好解决，可要是哪个程序忽然抽风，再内存里追跑打闹，上窜下跳，查皮一时手足无措，就容易蓝屏了。蓝屏之后，他会向老大（我管他叫主人）报告，自己为什么蓝了，问题发生在内存的哪个区域，发生了什么，并且把当时内存里的情况如实的记录下来，写成一份《工作间突发事件记录》一边记录一边报告记录的百分比——这就是他在倒数。记下来这个干什么呢？牛人们可以拿着这份记录，分析到底是哪里出了问题。不过好像一般人都不是牛人，谁也没看过查皮的记录。

对于工作间的使用，查皮和我还有一点不同。查皮总是喜欢尽量留出空间来，好给新起床的程序用。可是我总觉得，查皮怎么能知道还会有什么程序要运行呢？要是没有程序要来了，工作

间里还空那么大地方，不让正在工作的程序用，那不是浪费么？我还是习惯尽可能的把东西都搬进工作间里。除了程序们申请多少内存就尽可能给多少之外，剩下的部分，我就把一些可能会用到的库啊，命令啊啥的统统都搬进来，能占多少占多少。那有人问，要是你把这里边都沾满了，待会有程序要进来咋办？很简单啊，我再搬出去呗！程序要进来，也不是一下子都进来，他也得把他的东西一点点搬进来，他往内存里搬的时候，我就往外搬，不耽误。所以，当有程序要启动，跟我说：我要 10 平米的地方放东西。我就先答应他说，好，放吧，有地。然后在他往里搬的时候我再给他腾地方。也可能他要 10 平，但是只用了 2 平，那我就先腾出 2 平来，等他再要我再腾。他们管我这个方法叫 Copy-on-write。查皮就不同了，可能是因为他比较胖的缘故吧，他比较懒，不愿意搬来搬去这么折腾。基本上他只是在必须用啥东西的时候才把那东西搬到内存里，让内存留出尽可能多的空间。这样，当有程序管他申请内存的时候，他就可以用手一指：那块地，归你。然后就不用管了。实在内存不够用的时候就找个比较闲的程序，命令他：你，去硬盘里先忍会。(顺便说说，这个 32 位的查皮，并不能够完全利用起这 4G 大的内存空间，而是只能用到 2.5G，浪费啊。)

所以，经常跟查皮打交道的人，总觉得内存里空着的地方越大越好。当他们看到我把内存占的那么满的时候，总觉得很不爽，唉 ~ 我冤枉啊。

听说有一部电影，叫做《Big Buck Bunny》，这部电影长达 9 分 56 秒——还没电视节目中间的广告长，但是他有一个特点，一个和我一样的特点——他是开源的。怎么个开源？他是在开源的平 台上用开源的软件制作，并且免费下载观看还可以获得他的原始制作文件，（blender 的文件）如果你愿意，还可以进行修改，编个续集什么的。有人问，你说这些干嘛？这个电影跟你这个操作系统有什么关系？他本来跟我没什么关系，但是随着一件事情的发生，他就跟我有关系了——主人想看看。

任务下达下来，马上开始准备解决方案。首先是狐狸妹妹如春风摆柳般走了过来，顺便带来了一阵叮叮当当稀里哗啦的响声——一身的插件和扩展啊～ 来到工作间，狐狸妹妹掏出一个插件在手中一晃：我有 Flash，直接去个什么土豆啊，地瓜啊，西红柿什么的网站去看就得了，最省事。这时候，墙角有一位慢 条斯理的说话了："要说这看片啊～还得我来～你那个不专业。那电影才 10 分钟，剧情肯定没什么可看的，人家看得是个效果，要的是清晰度。我看那，还是下载下来，我去放吧。"我扭头一看，说话的是 MPlayer，要说这家伙在多媒体部门里可算是个元老了，而且能力相当强，什么片都能放，什么 rmvb，flv，avi，wmv 全都不在话下。就算您没图形界面，人家跟字符界面照样给你放电影。哪怕您显示器都不带色（念 shai 三生）儿，人家能给您拿字符拼出电影看。现在时代发展了，都高清了，人家也不甘落后，照样能支持，什么硬解码软解码的，通吃。既然人家这么专业，我看八成就得用他了，不过这事情我不做主，还是得等主人的吩咐。果然，主人也觉得应该下载下来看，于是，我们就忙活起来了。

要完成看片的大业，需要群策群力，精诚合作！

首先，虽然狐狸妹妹的建议没有被采纳，但她并不沮丧，收起 Flash 插件掏出另外一个扩展——Downthemall！听这名字就知道是干什么的了。狐狸妹妹先出门去找狗狗大哥（学名叫 google），打听到了《Big Buck Bunny》的下载地址，然后掏出 downthemall，把地址一填，就开始下载。要说这一身的扩展实在没白装，哪个都有独到的功能，不一会，一部电影就下载回

来了。然后就该 MPlayer 上场了，他先拿过片子看看格式，是 ogg 的，然后掏出了相应的解码器。解码器是干什么的？要知道，片子的格式很多，就好象现实中看电影，有数字电影，就要用数字放映机。胶带的，就得拿传统的放映机。在家里看光盘的，就得拿 DVD 机，看录象带的，就得拿录像机。 MPlayer 就像个电影放映员，解码器就是放映机。放映员在怎么牛，也得有各种放映机做支持，没放映机他啥也放不了。Mplayer 掏出解码器，开始放起影片来。这就完了么？还差得远呢，要想让主人看上片子，还少不了图形部门的支持。

图形部门主要负责给主人显示漂亮的图形界面。他们那的老大叫 Xorg，他会跟硬件打交道，会用显卡（当然，用显卡也得经过我），能在显示器上画东西，想画什么画什么，谁要想显示点东西给主人看，都得经过他。 要想跟 Xorg 打交道，在显示器上显示出图形来，得懂他们图形部门的黑话——学名叫协议。他们说话使用一种叫做 X 的协议，这个 X 不是牛 X 的 X，也不是傻 X 的 X，而就是 XYZ 的那个 X，XP 的那个 X，反正就是个 X。要想显示图形，就得用这种黑话跟 Xorg 去说。每一个要显示图形的程序都得会这种黑话，比如狐 狸妹妹，要显示东西，就说："驼子碗，筛土的抛闪！"那意思就是说画一陀黄色的便便-_-b，当然，这就是比方，其实我可不懂他们的黑话。（这一点不像查皮，他本身兼职负责画图形）那么 Mplayer 要画什么直接跟 Xorg 说就行了么？其实也行，那就像是在字符界面下看片了——没有窗口，图像没法移动，没法全屏，没法最小化等等。MPlayer 只负责放片，像画窗口啊，移动窗口什么的这些事情他可不管。那谁来管呢？这时就需要一个窗口管理器，我们这里的窗 口管理器叫做 metacity（就是 Gnome 下的默认窗口管理器）。MPlayer 要放什么东西其实是跟他说的，比如 Mplayer 说："画一只猪 "（当然是用 X 黑话）于是 Metacity 转头告诉 Xorg："在某某位置画个方的窗口，在里面画一只猪。"过一会主人觉得 Mplayer 方的片挡着他和 MM 用 Pidgin 聊天了（那是，猪哪有 MM 好看呀），就把 Mplayer 的窗口挪了挪，于是 Metacity 又对 Xorg 说："把刚才那只猪和窗口往左移动 3.2 厘米。"这个过程 Mplayer 是不知道的，他只管专心的向 Metacity 描绘着影片中的一幅幅

图像:“猪，走 路的猪，跑动的猪，跌倒的猪，捆绑的猪，烤熟的猪……”

十分钟过得很快，还没把图形部门介绍清楚呢，《Big Buck Bunny》就演完了。片子还有点意思，里面那只看上去笨笨的大兔子是我见过最可爱的兔子了，或许可以考虑以后让他来代言笨兔？不过时间短了点，主人看完了片，叫 Mplayer 去睡觉去了，然后继续拉来 Pidgin 跟 mm 聊天。

Pidgin 这个家伙其实就是个送信的，大家都喜欢根据他的发音叫他“皮筋”，但是他不管送那种长篇大论的 Email，而是负责发短消息，（short message）也有叫短信的。不过别误会，这可不是说手机的短信，而是像 msn 啊，qq 啊这样的即时通讯的消息。这些聊天软件的工作都是送信，使用者把要说的话写成信给他们，他们把信放在信封里——这个过程叫打包，然后把这个包发送给对方的软件。对方软件拿来这个包，先要拆包——也就是吧信从信封里拿出来，然后把里面的内容显示给用户看。可是这些软件互相之间是不能通信的，msn 不能给 QQ 发消息，反过来也不可能，这是因为他们的信封——打包方式不一样。比如 msn 的信封可能是从上面拆开取信；QQ 的信封则是从侧面拆开取信；Gtalk 的信封可能是用订书器订上的，需要拆钉取信；而百度 Hi 的是用胶水粘上的，需要涂水溶胶取信。反正各有各的高招。那么皮筋呢？他全会！

皮筋跟狐狸妹妹一样，也有很多的插件——其实就是一本本 XX 信封封/拆手册。Msn 的手册上，那就写着怎么封 msn 模式的信封，怎么拆 msn 模式的信封。皮筋只要拿来一看就明白了。gtalk 手册也是如此，所以，pidgin 可以支持很多种聊天软件，只要有相应的插件就行，不用再同时开着 gt，开着 msn，开着 qq 了，只要开一个 pidgin 就都能聊了。不过 qq 的信封比较特别，其他的聊天软件都使用公开的协议——至少能实现基本功能。有专门的文件写着自己只收什么什么样的信封，比如必须蓝色，上面印着蝴蝶，上开后直接撕开的信封才能发给 msn。可是 qq 这家伙的信封却很复杂，而且保密，别人都不知道具体应该怎么封。上面乱七八糟的有很多防伪标识，什么激光啊，磁条啊，比人民币还热闹。所为达到的目的就是只有自己的 QQ 软件能有互相通信。不过，强中自由强中手，人民币还有 HD90 呢，QQ 的信封怎么就不能伪造了？有牛人通过研究 QQ 的信封，慢慢分析，已经仿制出了 QQ 的数据包，可以实现最基本的文字聊天的功能了，这就是 pidgin 的 QQ 插件。但是功能相当有限，用起来不好使，所以多数人还是安装了 QQ 官方的软件，我主人也是这样。呀，貌似他跟 mm 约会好了，现在要关机出去了，明天见吧。

很多事情吧，都不能绝对化。以前说过，我是不能跟查皮那屋的软件们交流的，然而，今天就来了翻译。超级牛力在主人的要求下拉来了一杯红酒。不过这并不是因为主人想晚饭改善一下生活，而是由于狐狸妹妹不幸碰壁，越说越乱了吧，没事，咱们从头慢慢说。

话说今天，主人想看看自己的工资卡里的余额是否按时增长了，又懒得跑出去，所以就让狐狸妹妹到银行的网站去看看。狐狸妹妹迅速的到了网站，却发现网站用一种叫做 ActiveX 的语言问她银行卡的密码。密码当然会由主人告诉狐狸妹妹，可是，怎么能翻译成 ActiveX 语言告诉那个网站呢？狐狸妹妹一下子抓了瞎，没学过阿！再说了，这 ActiveX 语言是那个有点软公司发明的，想学可不容易，估计得交一大笔学费，人家都还不一定愿意教你——因为到现在为止，只有有点软公司亲生的 IE 同志才能够懂这门语言。狐狸妹妹急的翻遍了自己所有的扩展，也没发现有哪个能用来把主人的密码告诉这个该死的网站。急的狐狸妹妹差点哭出来，可是着急也没用阿，也只能灰溜溜的回来告诉主人——这个搞不定！

于是，目前处于这样一种情况：主人必须去那个该死的银行网站，而能够去那个网站的，只有 IE，可 IE 压根也听不懂我说话，他只给查皮打工。难道就为了看一眼余额，要把我哄回床上，让查皮来干活吗？那可要重新启动电脑阿，太麻烦了。可是谁又能把查皮叫醒并让他去干活呢？这个时候，超级牛力跑出来说：我有办法，本 APT 有超级牛力，有人能把 IE 叫醒，我去找他。说着，就跑出去了，转眼间拎回来一瓶红酒——WINE。

当然，说是红酒，只是因为他的名字，其实他当然不是红酒，而是一个软件，一个有特异功能的家伙，一位催眠大师。他来到这里，问了问情况，我把目前的问题跟他说了，任务很简单，就是把 IE 叫醒去干活。红酒大师点点头，拎起自己的工具包就走进隔壁查皮的屋里去了。只见红酒大师先掏出好几块屏风，在查皮周围挡了个严实，他说是要用高科技投影设施在屏风上投出 IE 平时工作时的虚拟环境，让 IE 觉得自己还是在查皮的领导下干活。解释了半天，反正我们也不大懂，只看见他用屏风把正在睡觉的 IE 围起来，然后，过了一会，听见他用低沉浑厚的声音

向 IE 念着：现在计算机正在启动~~我是 Windows XP~我是 Windows XP~你要开始工作~~你要慢慢醒来~你要慢慢醒来开始工作~~醒来~醒来~~~ 随着他一步一步的引诱，渐渐的听见了动静，好象是 IE 迷迷糊糊的起床了，又过一会，只见屏风打开一块，IE 慢慢的走向工作间里，他走路有点不稳，一边走，红酒大师还一边跟在他左右不停的引导：你像每天一样起床~正走向内存里~开始你的工作~~ 然后扭头问我:“让他去干啥？”看的快入迷的我才反应过来，还没交代清楚具体的任务呢，赶紧说:“哦，让他去那个 XX 银行的网站。”大师继续对 IE 说：现在~~Wdinwos 让你去银行的网站~~~去吧~~去吧~~~~像每天一样~~~~ 这个时候，基本上所有人都看呆了，没想到这 IE 竟然就这么被大师指使着干活去了，大师果然是大师阿。

今天丢人丢大了！

想我大名鼎鼎的火狐狸，什么网站没去过？什么网站搞不定？什么 Konqueror，Epiphany，lynx，除了那个不大懂事的挪威妞 Opera 以外，哪个浏览器见了我不得叫声大姐？在线视频？行！Flash？没问题！有我一身的插件，那是兵来将挡水来土掩，可是今天，竟然就有网站我拿它没辙!

这破网站是个银行的网站，要说那好多国际知名的银行我也见过，人家啥软键盘阿，HTTPS 加密也都挺好的，也没见用什么 ActiveX，也挺安全的阿，怎么就这破网站非得用 ActiveX 呢？我其他的都会，就不会这 ActiveX，这不是诚心揭短么。再说了，不会也不是我的错阿，我倒是想学呢，谁教我阿，人家微软才不会把这个教给我呢，藏着掖着还来不及呢。结果可好，我没法搞定这网站阿，主人只好叫别人来，这不是抢我饭碗么！当然，叫什么 Konqueror，Opera 阿这些个来也是白费，只有 IE 才行。可按说 IE 他不能听我们头的阿？嘿，还真是什么高人都有，让超级牛力找来个催眠大师，居然就把 IE 整的服服贴贴的，老老实实去干活了。不过，毕竟 IE 不是在清醒状态，基本上跟梦游似的在那干活，虽然没出什么错，可动作慢了不少。主人也只能姑且忍受一下，看来我的饭碗还能保住。我本来想偷偷跟他学学 ActiveX，可是他嘴里念叨着叽哩咕噜的东西我也听不懂阿，当然，其他人也听不懂，只有红酒大师能懂他的话。我们头发送命令，只能先告诉红酒，再由红酒翻译给 IE。IE 要用什么东西，什么 DLL 文件阿，配置文件阿，红酒都赶快给他找来，原版的找不来就自己做一个差不多的，放的位置都跟 IE 在查皮那里干活时的位置差不多，让 IE 以为自己还是在查皮那里干活。既然用 IE 登陆该死的银行了，我就没什么事干了，正好本小姐还能歇会，哼！等了半天，IE 才把事办完，我都睡了一觉了，主人赶紧把 IE 关了，还是叫过我来去其他的网页，要是我我也得关，看着他就难受，哪有我看着顺眼阿。我麻利的赶快开工，赶紧表现表现，一定要和那破 IE 形成鲜明的对比。看看主人想看点啥？哦，要查查红酒大师还能干什么，好赶快去找狗狗哥……

唉～最近呀～不知道怎么得了，我这个脑子出问题了吧还是怎么的，怎么晕晕沉沉的呢，而且好像记忆还不老好的了，要不就是有幻觉。我隐约记得昨天明明起床干活了，好像是去了个什么银行？不过记的不怎么清楚，模模糊糊的感觉，好像做梦一样。我以前不这样啊～我可是名门之后，血统纯正，我祖上从来也没有失忆的毛病。想当年啊～我的前辈 IE5 那时候就跟着 Windows98 混了，那时候有个家伙叫 Netscape ，觉得自己挺牛，基本上那时候上网的都得用它。可是呢～哼哼～有本事不如有靠山，我前辈 IE5 老先生虽然论本事……比不上那什么 NetScape，可是他聪明啊，死粘着 Windows98 老大，98 去哪他去哪，有这强大的后台，慢慢的大家都开始用 IE5 了，NetScape 从此销声匿迹。后来的 IE6 也是如法炮制啊，棒着 WindowsXP，后来我 IE7 横空出世了，就取代了 IE6 的位置，换我跟着 XP 干。咳，怎么说起这些了，看来脑子真是不行了啊。我明明记得昨天去了银行的网站，还是 XP 老大叫我去的，可是今天我问 XP 老大，昨天你说话声音怎么有点不对劲呢？是不是感冒了？老大斜眼看看我问：昨天？哪有活阿？我说不对呀，昨天不是你让我去那个什么银行查余额么？老大直接扭过头，扔下一句:“做梦呢吧你。”我挠挠头，难道我正的是在梦游？说是梦，却很清晰，说是真的，可还有点模糊。或者……那是我前世的记忆？前世……靠，为啥我前世还是浏览器？！等等，我前世要是浏览器的话……难道我前世就是那个 NetScape？！不行，越想越晕了，再这样下去非精神分裂不可。想办法找人聊聊诉诉苦吧，老大反正不理我，去找 cmd 聊聊吧，他是专门负责跟人聊天的，问问他我这到底是怎么回事。他说我是参数打错了，唉，他也不知道别的。问问游戏组那哥几个，扫雷说我是踩着雷了，空档接龙说我是牌放错了，这都哪跟哪阿~再去问问记事本，直接被嘲笑，说我这天天上网见多识广的，竟然还来问他这么一个大门不出二门不迈的抄写员。唉~想想也是，看来只有我不正常了。正灰心呢，那个长得恐怖的 War3 来了，神神密密的对我说:“我也梦游了……”我惊讶的望了他 3 秒钟——难道他……传染的我！！

自从超级牛力给主人请来了红酒大师，主人叫醒查皮的次数明显减少了，不过我这里也就经常有了迷迷糊糊的 IE，晕头转向的 War3 的身影。其他的像 QQ（windows 的）啊，迅雷啊之类的也来过，不过要么就是晕的有点过，干不了活了，或者就是反映太慢，主人觉得不爽，所以他们就来过几次，后来就不再来了。

生活变得忙碌起来。这一忙起来，日子过得就快，转眼间，又到 4 月了，忽然发现一年的时光就这样伴随着一条条指令，慢慢的流逝；忽然发现，我，一岁了。对于一个操作系统来说，一年或许很短，但也可能很长。或许，一年，就是一生——尤其对于跟新换代很快的 Ubuntu 来说。Canonical 的毕业生是半年一届的，每年 4 月和 10 月是学生们从学校毕业的日子。不过也有特例，2006 年春天那批由于不太老实，延期毕业了两个月。继我们那批之后，已经有过一批 8.10，而现在，最新的 9.04 也马上那个就要奔赴各自的工作岗位。说来，我也算是他们的前辈了。想想以前在学校的日子，还真是美好。Canonical 学校其实是有几个不同的专业的，不光是我们 Ubuntu。我们是学校最热门专业出来的学生，除了我们之外还有很多其他的专业，比如 Kubuntu，Xubuntu，Edubuntu……

Kubuntu 专业出来的学生似乎比我们更有些艺术细胞。他们的样子要比我们好看些，精致些，而我们 Ubuntu 比较主张简洁明了。他们的桌面环境请来的是 KDE 团队，所以叫做 Kubuntu，而不像我用 Gnome。说起 KDE 和 Gnome 的争论，大概说个两三天也说不完，两者都是桌面环境，都是用来和人来交流的。KDE 更愿意把各种部分的设置能力交给用户，让用户可以随心所欲的把自己的桌面改成想要的任何样子。而 Gnome 则注重简介，当然，也可以进行一定配置，不过就比较麻烦，需要写写配置文件。由于桌面环境不一样，附带的软件也有所不同。Kubuntu 有 kopete 来聊天，而我们这里是皮筋，Kubuntu 写文档用 Koffice，我们这里是 OOo 老先生。不过，这只是默认的情况，其实在我也可以让超级牛力找来 kopete 干活，代替皮筋，只是个人喜好不同而已。

Xubuntu 专业的，都是准备去艰苦环境下工作的志愿者。用的桌面环境就跟我们都不一样了，他是用 XFCE。XFCE 的特点就是小巧，占用资源少。可以在很艰苦的硬件条件下很好的工作。比如内存，Xubuntu 能够在 200M 内存的机器上跑的比较流畅，这要是换了我，还不得郁闷死。才 200M 啊……这狐狸妹妹带着一身的扩展一进去就得好几十 M，200M 哪够用啊。可是

人家 Xubuntu 就够。不过，相应的软件也要用一些轻量级的，要是也请个挂满扩展的狐狸妹妹，那系统本身省吃俭用节约下来的那点内存，还不够她一人用的呢。

Edubuntu 是教育专业出身，用的桌面环境跟我一样，只是他附带了很多搞教育的软件。比如教小孩子打字的啊，画画的啊，教授一些物理知识的啊，这些软件都是很好的老师，很多小孩子用起来都乐此不疲。很多小游戏也都是寓教于乐的，家长们也可以不用担心孩子用这个系统沉迷于游戏（因为没啥可沉迷的游戏……）只是这些软件多半都是英文的，所以，非英文地区的孩子们用起来还是不太合适。

除了这些之外，还有很多其他专业的 ubuntu，那些专业都不是 Canonical 学校自己开的，不过也都是用的一样的教材，大家互相都是通用的，只是所带的软件不同而已。今天不知为什么说了这么多，或许是因为有些伤感，因为最近主人总让狐狸妹妹去 Kubuntu 的网站转悠，打听 KDE4.2，打听 Kubuntu 9.04，难道……

人物志——星爷

这里不是娱乐周刊，我要说的自然也不是周星驰，我要说的是星际译王。他来自中国，是我们这里少数来自中国的软件之一。他是我们这里最有学问的人，简直是什么都知道。一开始他只是懂点英语而已，大家都只当他是个英语翻译，后来主人叫狐狸妹妹去给他找来各种各样的书给星爷看——就是那种叫做字典的书，这种书只有星爷能看懂。看了这些书以后，星爷知道的就多了，什么日语啊，法语啊，德语啊，都会了，估计联合国要开会请他一个人去当翻译就行了。于是星爷从英语翻译一下子晋升成为了地球语翻译！（地球上的语言都会-_-b）不知道他以后会不会再学学火星语？

然而星爷是不仅仅满足于当地球语翻译的，或者说，主人是不满足于让星爷只做个翻译的。这之后他又给星爷找来了本《陈义孝佛学常见辞汇》，于是星爷开始研究佛学了，不过这东西只是主人一时的好奇而已，后来就再没给星爷找过佛学的书，而是找了本《五笔 98》，开始学习五

笔了。要说这输入法的事，那可是 scim 的本行啊，可是无奈 scim 老弟干活还行，表达能力不强。主人要打什么字，scim 可能很快反应过来，打在屏幕上，可是主人要是忘了某个字怎么打，问问 scim，那可要了命了，打死他也说不清楚啊。主人只能问:“是 a 开头不是？”然后 scim 把所以 a 开头的显示出来数一遍，摇摇头:“不是”，然后主人再猜:“是 s 打头？”scim 再把 s 打头的列一遍……这样实在太费事，于是主人就让星爷学五笔，到时候就能去问星爷:“这个……‘我’字用五笔怎么打啊？”星爷会投过一个鄙视的眼神:“你还好意思说会五笔啊，q 空格！”

这还不算完，后来星爷又看起了《本草纲目》，研究起了中医。不过他还不能当大夫，针灸号脉啥的他是不会（就算会也没法号啊），那会啥呢？看了《本草》当然最精通的就是药理了，随便说一种药，他就能告诉你这个药的药性，药效，怎么用，等等等等。这时候基本上我们已经对星爷的全能感到震惊了。后来他又开始研究古汉语，看古汉语词典，康熙字典，整天到晚的跟我们这之乎者也。“夫内核者，老大也，发其命，出其令，而统‘康皮右特儿’（computer -_-b）……”然后就是我们集体的“打豆豆”时间——谁是豆豆就不用说了吧。

我也趁没人的时候，问过星爷：您怎么懂这么多呢？看什么会什么。星爷很神秘的笑笑说：其实他没什么本事，就是在学校学过信息检索而已。主人给了那么多本书，要想都记住，根本不可能嘛，他只是每次在主人问起事情的时候，赶快现去查书，用最快的速度找到并告诉主人。要是没了这些书，他知道的就少很多了。不过也不至于离开了书就什么都不知道，现在的星爷学会上网了，可以连接到一种叫网络辞典的地方，自己不会可以去网上查，不过那样效率自然不如自己翻书快了。

有关 Firefox 的小知识：

像 downthemall 啊，天气预报啦之类的这些就叫扩展

像 flash 这样用于支持网页上显示内容的就叫插件

自从红酒大师来了之后，查皮起床的次数比以前少了很多，我的工作更繁忙了，大多数工作，主人都交给我去做。每天大家都忙得不可开交，新来的奔流整天忙着下载各种大个头的东西，什么电影阿，软件阿，什么都有。随时都跑过来找我：头，我下了 20M，先存硬盘里。我说：好，存那吧，赶快。转眼，Mplayer 又过来：头，我要读那个电影。刚找着电影递给 Mplayer，主人又发命令要我把 U 盘里的一堆文档搬到他的文档目录里。正搬着呢，奔流又过来了：头，我还要存 20M。我一边搬着文档一边指一块地：恩，你就存 N＆@＃％……

我睁开眼，看见了熟悉的 GRUB 大叔，他拍拍我：嘿，醒醒，开工啦！——唉，每次都是这句。我揉揉眼睛，觉得头有点涨。发生什么了？怎么屋里有些乱？仔细回忆一下……哦，我好像正在干活，然后……停电了？！恩，应该是了，那时候眼前一黑，就什么也不知道了。我当时正在搬文件，搬到哪了？恩，看看日志把。还好我用的 XFS 是个日志文件系统。什么？日志文件系统你也没听说过？唉～ 讲课。

笨兔兔老师第三讲——什么是日志文件系统

文件系统就是我们管理整个硬盘这间屋子的方式，这个以前跟大家说过了。文件系统有很多种，过去的文件系统都是非日志文件系统，这种文件系统比较落后。比如 EXT2，比如查皮那的 FAT。非日志文件系统在发生意外断电的时候就容易出问题。就像今天的情况，如果我这屋子用的是 ext2 的话，没准就丢个文件阿什么的，搞不好整个分区都坏掉了。为什么呢？比如我屋里有一个叫 笨兔兔的故事.odt 的文件。文件，前面说了，就相当于放在屋里的一个大箱子，里面是内容，外面写着文件名：笨兔兔的故事.odt 内容是什么咱就不管了。然后有一天主人要修改这个文件，可能往里面多写进去点东西，也可能改掉里面的一些东西。如果用非日志文件系统是怎么做的呢？很简单，主人首先找 OO 老先生打开这个文件，打开，也就是把这个文件读进了内存里，然后靠 OO 来在内存里修改这个文件。注意，文件不是你家的大白菜，搬到屋外那屋里肯定没有了。文件读进内存，磁盘上仍然有这个文件，内存里只是它的一个副本。好，现在，OO

老先生那有这个文件改动后的版本，在内存里（就是主人还没点保存）。磁盘里有这个文件原来的版本。如果这个时候停电了，那刚才该的那些肯定都作废，这个用什么文件系统也是一样。那如果主人点了保存，并且保存结束了，这个时候停电，那就停吧，也没事，因为已经保存进去了，除非房塌了（比如磁头挂了，盘面损坏之类的），否则不会丢。如果主人点了保存，那么 OO 就要让我把内存里他写的那个副本往磁盘上存，于是我就从内存里拿过来一点，打开磁盘上那个文件，掏出里面的一部分，扔掉，用我手里这些替换进去。然后再回内存里拿下一部分，再回来把文件里的下一部分扔掉，用我手里的替换。如果正在这个过程中停电了，那就惨了。内存里的，那肯定没了，磁盘上的，有一部分被替换掉了，有一步分还是原来的，于是文件就乱了，可能损坏，格式不对，根本打不开之类的。

那用日志文件系统又怎么样呢？日志文件系统，顾名思义，就是有日志的文件系统（废话）。还是拿上面那种情况举例，OO 要存那文件，那我怎么做呢？我会在硬盘上一个专门的记录日志的地方些下来：OO 要覆盖 笨兔兔的故事.odt 文件。如果这个时候停电了，没事，原来的那文件还好好的，但是内存里的还是没了，这条记录也就作废。记录之后，我就开始把内存里的东西往硬盘里放——放在记录日志的地方，并不动原来的那个文件。如果放到一半停电了，那也没关系，原来的文件还好好的。修改了的那份也有一部分放到了硬盘里，不过这是一部分的话，多半还是没什么用。如果我把文件完全搬到了记录日志的那部分硬盘里，那就再在刚才记录的那条日志下面写上：已经把要覆盖的内容存到了日志去 xxx 位置，准备替换原文件。如果这个时候停电，没事，等再开机，我一查日志，就知道要修改的版本已经完全存在了硬盘里，只要按着上面记录的继续做就行了。写好日志之后，就开始用日志区的这个新文件去替换硬盘上那个原来的文件。这个过程会很快，因为其实并不需要真的搬运数据，只要在原文件的地方做上标记，表示这个文件已经作废，然后把那个 笨兔兔的故事.odt 文件名指向新写的这个文件就好了。（我们只是拿箱子比喻文件，但文件毕竟不是你家的箱子。）这样，无论中间的哪个过程断电，都不会完全损

坏整个文件，要么原版还留着，要么修改后的版本已经生效，通过查看日志就能知道现在哪个版本有效。这就是日志文件系统。

书接上回。

上回书说到，这笨兔子跟店房伙计似的忙里忙外脚打后脑勺阿，忽然间咔嚓一声——停电了。那位说了，怎么回子事捏？我不说，您不知道。原来是小区电网改造，这个小区那是历史悠久，早在清朝末年……当然了，这些都是题外话，咱们暂时不表，单说这笨兔子。来电以后，只见这本兔子不慌不忙地起床，刷牙，洗脸，吃早点，看看屋里边，挺乱的呀。拍脑门一想：对了，刚才停电了。有人关心了，说这停电了，丢东西没？听过上文书的都知道了，这笨兔子用地是"插爱夫爱死"(XFS)文件系统，那可是日志文件系统阿，那就相当于宝兵器，浑铁凝钢打造，那是削铜剁铁，斩金错玉阿，那能怕停电么。所以这笨兔子照着这个文件系统地日志，前后左右那么一查，齐活，啥也没丢。

又有人可打听了，说这"插爱夫爱死"这口宝兵器是从哪得来的捏？这可说来话长了。想当初，早年间了，有这么一个硅谷图形公司（SGI），他们有个操作系统，叫做"爱阿爱插"（IRIX），这外国人起名字都各色。这系统干什么用的呢，主要是图形计算。本来他们有个文件系统叫"义爱夫爱死"(EFS)，可是这不过是一件凡铁兵器，一开始拿他切菜还挺好使，后来买卖做大了，随手也越来越强，人家都那地是宝刀宝剑，你这把切菜刀，怎么跟人家比划呀。所以这硅谷图形的总瓢把子拍板，说咱这破菜刀也别磨了，你再怎么磨也是把菜刀，干脆，"义爱夫爱死"——扔了吧。另请高人，访贤士，为新版地这个打造了"爱阿爱插"系统打造了一口宝兵器，就这"插爱夫爱死"。那么说这宝兵器宝在哪捏？头一个说，就是他结实。就是不怕断点呐，就跟这笨兔子这回遇到的情况一样，忽然地断点，没事，数据不丢。再一个，什么捏，就是这兵器，速度快，那是快如闪电。有那么句话您听过没有，叫迅雷不及掩耳盗铃阿。这文件系统格式化，甭管你多大磁盘吧，你是 1G 也好，100G 也罢，哪怕你是 1T 的硬盘，一眨眼的功夫，格完了。光格的快也没用，那读写文件也是不慢，尤其是越是大个的文件，读写起来越有优势。还什么特点？还有就是能伸能缩，要说小，几十兆建个分区也行，要说大，您看着"插爱夫爱死"是个六十四位的文件系统不是，

最大支持多大滴分区捏，支持十八个 **E** 的分区。就是一万八千个 **P** 呀～那么这是分区的大小，那支持的文件大小最大多大呢，最大九个 **E**，也就是九千个 **P**。**P** 是多少不用给大家介绍了把，**1000** 个 **T** 阿……

那么有人说了，你把这件宝兵器夸滴跟一朵花似的，那么他就没缺点了么？他就无敌天下了么？当然不是，有道是一山更比一山高，能人后面有能人阿。说这 **XFS** 有什么缺点呢？就怕删文件，尤其是小文件，删除一大堆小文件的时候那个速度就慢死了。再一个，宝兵器一般都沉，这处理器费地多点。当然，也不是很多，只是相对多一点。这正是金无足赤，人无完人，今天给您说完这段到下回咱们说说……说啥还没想好反正是咱们下回——再说。

我们住处的墙上有一扇门，门上面写着三个字母——USB

这扇门是我们与外界交流的又一个接口（最重要的还是网络），每次门上的红灯亮起，就说明有东西接到了 USB 上，我就得去打开门看看。有时候门外是一个小集装箱似的屋子，很小，一般只有几百兆到几个 G 大小，里面也像我屋子里一样放着一些文件。这个时候一般主人就是要让我搬东西，不是把小屋子里的往大屋里搬，就是从大屋往小屋挪。有时候一开门，外面不是一个屋子，而是一台设备，比如是一个鼠标啊，或者是个摄像头什么的，那就是让我去操作这些设备了。主人就接进来过摄像头，摄像头的名字叫 Moto E6，听说这其实是个手机，上面的摄像头可以通过 USB 接口来给电脑用，不管怎么样吧，这东西咱也会玩，接上就能用，鼠标那就更不在话下了。不过总的来说，还是往 USB 那门外接小集装箱屋子的情况多，这种小集装箱式的屋子叫做 U 盘，也有更大一点的，就是移动硬盘了。每当 U 盘接进来的时候，我就把它当作我屋里的一部分来用。

要想成为我屋子的一部分，就要起个名字，要不我怎么访问啊？就跟你家似的，什么叫厨房哪个是厕所，什么主卧，阳台，是个空间都得有个名字不是？我这里也一样，总得有个名字才好访问。这里要再说明一下我们住的硬盘。我们住的硬盘跟你住的房子是一样的，不是整个的一大间屋子大家乱住，东西乱扔。也要稍微的分出几个间来，学名叫分区。分区的多少和大小是主人在安装的时候根据需要分的，比如我住的这间，就分出了四个分区，也就相当于把整个大屋子用墙格出了四个间。一个小间的门上挂着个牌：/boot 这间里面是我住的，我的单间，作为老大嘛……有点特权也是可以理解的吧。再一个大间，门牌上写的是/home，这里是主人专门用来放他的东西的，什么文档阿，电影阿，歌阿，都放这里，所以这间特别大。还有一间，平时不放东西，这间叫做 SWAP。这间是用来给正在干活的程序用的，程序们要运行的时候不是要往工作间里放东西么，要是放不下了，就暂时放到这间里。不过由于我们这的工作间很大，所以基本上这间没怎么用过。除了这几间之外的，就是剩下的一大间了，这间叫做“/”，学名叫做根分区。其

他的间可以没有，但是这间不能没有，要不我们住哪阿，是不。不过虽然其他间可以不格出来，但是对应的空间还是得有的，比如/boot，可以不单给我格出一个单间，也就是不给我单独分个区，但是就算不格，我也得划出一块地方，拿粉笔写上/boot，表示是我住的地方。

回来说说 U 盘，U 盘接进来之后，我也把他当作一小间，并且给他门口挂个牌子，一般是/media/xxx，xxx 就是那个 U 盘的名字。一般 U 盘都会有个名字，或者叫卷标。这样一弄，就可以很好的区分各个单间了。比如要让 mplayer 放电影，我总得人家说明白这电影放在哪吧，我要是说：就在那个放了一堆主人的东西的最大的屋子里，就很罗嗦，不如直接说：在/home/lanwoniu/目录下来的简单。

这回主人接到 USB 上的又是一个集装箱式的空间，不过还比较大，4 个 G。我仔细看了看，结果没有发现这设备的名字，那也没事，名字是人起的嘛，我直接给他起个名字并挂上了牌子：/media/4.1G 没名字是可以滴，没牌子是不行滴。这个挂牌子的过程，用我们的专业话说，叫做挂载，这个大家应该都听说过哈。挂载，就是挂牌，就是在某一间屋子的门口挂个牌子，起个名字，到时候干活的时候就好说话了。直接说屋子的名字就好了。名字，或者说牌子，就是个标志，是可以随便换的。比如有个分区被挂载到了 /home/目录，也就是说，有个屋子 A（就叫 A 吧），被挂上了/home/的牌子。那么主人说要看看/home/下都有什么，我就会把那个屋子 A 里面的东西列出一个单自来给主人看。等回头可能又把别的分区挂载到/home/下了，那很简单，就是把/home/那个牌子从 A 房间门口摘下来挂在 B 房间门口而已。主人再说要看卡/home/下都有什么，我就该把屋子 B 里面的东西列表来给主人看了。没啥经验的主人可能会大跌眼镜：哇塞 ~ 怎么我/home/下原来那些东西都没了阿 !! 都哪去了阿 !! 殊不知，其实原来那些东西还在 A 屋子里好好的放着，只是现在 B 房间改叫/home/了而已。

又废了这么些个话，不好意思。回来说这回接上的这个 4G 的屋子，接上之后，还没等主人发话，我先去里面查看了一下，一看全都是些个 jpg 的文件，我是看不懂这些个文件，但是我知道有人能懂，于是赶快通过图形界面那哥几个报告主人：您插进来的这个里面貌似全是照片，是不是要我给您找来 F-Spot 呢？

F-Spot 是一个管理照片的程序，他可以帮主人把移动设备（就是那种集装箱似的小空间）里的照片导到硬盘里，并且按时间分门别类的管理好。主人想要去年三月的照片，他马上能够给找到。除此之外，还能够汇报照片的信息，比如用什么相机照的，照的是后用的光圈，快门，ISO 都是多少，等等信息。F-Spot 虽然能够整理照片，不过不能编辑和修改，有时候主人的照片经常需要稍微的调整一下色彩阿，做一点效果阿啥的，有很多人管这叫 PS，因为在查皮那里，做这种事情的软件叫做 PhotoShop，所写 PS。逐渐的，用 PhotoShop 去处理照片这个动作就

被叫成了 PS。我这里没有 PhotoShop，不过有另外的处理照片的强人——GIMP

GIMP 这家伙是个美术家，能够画出很多漂亮的图画来。当然，漂亮不漂亮是以主人的审美观点来说的，对我来说他制作出来的那些个东西，不过是一个写满 0101001 的文件而已，和其他的文件没有什么区别。虽然他可以在一张白纸上创造出一个多彩的世界，但多数情况下他要做的工作是调整一个已经画好的图片。一般都是主人用数码相机照的照片，用 U 盘拷贝进电脑，然后叫来 GIMP，调整一下照片的亮度啊，色彩啊什么的。有时候还让 GIMP 做一些效果，比如做成油画效果的，就是把照片做成像是画的油画似的；还有做成浮雕样的，或者加个相框，他都行。以前拷进来的照片也就几百 K，可最近主人弄进来的照片最小都 2M 多，别人不知道怎么回事，可是瞒不住 F-Spot，他能看懂照片的 EXIF 信息啊。他告诉我，以前的照片是用一个叫做 Nikon 5900 的照相机照的，现在这些 2M 的都是用 Pentax k-m 这个相机照的。我让狐狸妹妹帮我去查了查这两个相机，5900 是个 500 万像素的 DC，k-m 是 1020 万像素的的单反相机，怪不得照片大了很多。不过，区区 2M 的照片文件是难不倒 GIMP 的，仍然刷的一下就打开，他自己说，就算 20M 的文件也没问题，不过我们倒是没见过，主要是主人的相机实在照不出那么大的来。

GIMP 像狐狸妹妹一样，可以安装很多的插件以实现各种不同的功能和效果，口口能，不输于查皮那里的 PhotoShop。但是 GIMP 有些不大好接触，总是按着自己的性子来，让主人用起来有些不顺手，不如 PS 那么易用。毕竟人家 PS 身价不菲嘛，听说一套要好几千块钱，一分钱一份货嘛。花这么多钱请回家去的软件，怎能不服服帖帖的听主人的话呢。再看看 GIMP，总是摆着一副：我就这样，爱用不用的架势。还好主人比较通情达理，也能理解 GIMP 是确实有能力的，否则……哼哼，别看 GIMP 跟我关系不错，没准也得被超级牛力请出硬盘。

主人整来一大堆照片，用 GIMP 处理了一下，调调颜色，亮度啥的，还别说，调完了还是比以前好看了不少。不过，有道是独乐乐不如众乐乐，主人光自己看着好看不行，还想和朋友分享，怎么办呢？找人吧，这个人您大概也认识——QQ。

Qq 和皮筋一样，也是一个即时通信软件，也就是个送信的。不过他不是超级牛力从软件源里请的，而是狐狸妹妹直接去 QQ 的网站上下载的。这 QQ 是一个叫做疼痛，哦不对，叫疼……疼什么来着？哦，对，疼殉，一开始是疼，后来就殉了-_-b，是一个叫疼殉的公司做的。话说这个疼殉啊，看人家 icq 软件玩的挺火，于是也弄了个 oicq，抢占了国内市场，结果一发不可收拾。后来 oicq 改名，叫 QQ 了，可是一直以来，由于各种原因，疼殉这个公司只能做出 wdinwos 版本的 QQ 来，Linux 下的没有。要说没有也不要紧，人家 google talk 也没有 Linux 的版本，但是人家是基于开源的 XMPP 协议的，协议是公开的，于是世界各地的 Linux 牛人们，很轻易的就做出了很多种用来在 Linux 下聊 googlt talk 的软件。其实就算他们不做这些，皮筋本身也是支持 XMPP 协议的，设置一下就能聊 gatlk 了。可是 QQ 不一样，QQ 的协议是那个疼殉公司自己定的，还不让别人知道，又不提供 Linux 的版本，结果，在 Linux 下使用 QQ 一直是个很头疼事情。当然，这些都是历史了，现在疼殉公司终于想开了，提供了 Linux 版的 QQ，虽然功能简陋的不能再简陋，不过，传个图片还是没问题的。好，废话不多说（这废话就不少了），赶紧叫 QQ 起床干活去。

QQ 迷迷瞪瞪的走进工作间，把主人写的文字一条条的打好包，封好信封，寄给那边的 QQ。过了一会主人要发图片了，QQ 有点忙乱，好像对寄图片这工作不如寄文字来的顺手，不过好歹是寄出去了。mm 那边也寄过来两张图片，QQ 收下了打开给主人看，主人很满一，让 mm 多发几张，于是 mm 一口气发了 5 张，然后……QQ 就晕了，折腾好几次也没法和对方的 QQ 建立好连接，结果照片终于没传成。要说这家伙也真是不争气，这么点事情都做不好。主人一气之下，只好把 QQ 关了，让我去叫醒另一个人——EVA

EVA 的大名相信大家都听说过，他的英文名字叫 EVA，他的中文名字叫新世纪福音战士。话说日本有个贞本义行……哦，对不起，扯远了。忘记贞本义行和新世纪福音战士吧，他们跟我这里的 EVA 没关系，跟我这个 EVA 有关的是云帆大姐姐。在疼殉公司还没有给出 Linux 版本 QQ 的时候，很多人们就用 eva 来聊 qq，这个 eva 完全是云大姐根据抓包研究的结果，黑盒破解 windows 的 QQ 而编写的。功能也算是强大了，基本的聊天，传图片，群里收发自定义表情，都可以。甚至还支持显示好友的自定义头像。（这看似简单的功能，疼殉官方的 QQ 目前都没实现）于是，在 QQ 不靠谱的时候，主人还是叫来了可靠的 EVA 继续跟 mm 传图片。QQ 跟 EVA 有很多不同，一个是官方的，一个是山寨的（山寨不含贬义）；一个是基于 GTK 的，一个是基于 QT 的；一个是闭源的，一个是开源的；一个是 32 位的，一个是 64 位的。

有人问了，你老说这 32 位，64 位。到底啥意思阿？

这个多少多少位，说的是 cpu 一次运算的二进制数字的位数。这个 CPU 就像是个计算器，我们软件用 CPU 就像人类用计算器似的。它很重要，我们要算一丁点东西，也需要用 CPU 来算。（别跟我说用心算，我是软件，ok?）那么这个 CPU 算东西的能力，是有限制的，有什么限制呢？你拿出你家的计算器看看，算个 28+783，没问题是吧。算个 7836-473 也没问题是吧，再算个 726354465849392029373465 37+1，能么？估计 99%的同志出问题了(不排除有牛人拥有很牛的计算器)：“我哪能按出这么多数来啊，我这计算器总共就能显示下 11 位数字”。对，这就是计算器的位数限制。CPU 也一样，他一次能算的数不能无限的大，总得有个边，只不过不是按照十进制的位数算的，而是按照二进制的位数算的。至于什么叫十进制，什么叫二进制，可以去问问狗狗大哥，不过不知道也没关系，咱暂时按着咱们平常的十进制来说。比如说，我这个 CPU 只能算 99 以内的数字，也就是只有 2 位（十进制位啊）。那我们软件用这个 CPU 的时候怎么用呢，CPU 有很多放数据的小匣子，叫做寄存器，每个寄存器有他特殊的用途，咱们就不多介绍了。要做加法的话，得这么操作：有两个寄存器，也就是小匣子啊，把这两个小匣子打开，往里

面放数据，数据比较抽象，就想想成写着数字的纸条吧。不过，由于是 2 位的 CPU（再次声明，咱这是拿十进制做比方啊，真正的 CPU 没这样的），所以寄存器里只能放 2 位的数据，也就是说，纸条上只能写 99 以内的数字放进去。那好，我写一张 12，放在 A 匣子里，再写一张 9，放在 B 匣子里，然后按个写着“加法”的按钮，只听咔嚓一声，CPU 自动弹出一张纸，纸上写着 21，这就是他的计算功能。再算个大点的，写一张 50 放进 A，写一张 51 放进 B，按钮，咔嚓，出来张纸，写着 01。那位说了，这算错了啊这个！别急，紧接着咔嚓一下，又出来一张，写着“对了，还得进一位”，这回对了吧。为什么呢？因为这 CPU 是两位的，只能输出两位数，超出的就告诉你得进位。

好了，基本的操作说完了，现在说正题，不同位数的区别。两位的 CPU 就像刚才说的那样，那么假设现在需要计算 3173+644，这里有 2 位的 CPU 一个，4 位的 CPU 一个，分别用他们做这个计算，有什么区别呢？

咱先拿这两位的，有人说了，两位的只能算两位啊，这个没法算哪？唉，这机器是死的，咱软件是活的啊，一次只能算两位，咱不会分开了多算几次么。首先，写一张 73，写一张 44，按钮，咔嚓，出来一张 17，咔嚓又出来一张写着还得进位，好，可记住了啊，还得进位。然后再写一张 31，写一张 6，按钮，咔嚓，出来 37。别忙，没完，刚才还得进位呢么不是，再写一张 37，写一张 1，按钮，咔嚓，出来 38。好，最后结果拼一块，高位是 38，低位是 17，最后结果：3817

再拿这 4 位的算算看。4 位的就意味着输入的数据和输出的数据都可以是 4 位，也就是说我直接就可以写一张 3173，写一张 644，放进去，按钮，咔嚓，出来一张 3817，算完收工～

这就是 2 位的 CPU 和 4 位的 CPU 的不同，从理论上来说，4 位的要比 2 位的快，从上面的例子看的很明显嘛，大一点的数，4 位的 CPU 一下就能算完，2 位的 CPU 要折腾好几次。但是这 4 位的 CPU 还得有人会用才行，这就需要 4 位的软件来用着个 4 位的 CPU。

终于说到软件的位数了，CPU 的位数就是一次能计算多少位的数，那软件的位数呢？就是说明这个软件需要使用多少位的 CPU。软件干活肯定需要计算，计算就得用 CPU，2 位的软件会用 2 位的 CPU，4 位的软件就会用 4 位的 CPU（还是拿十进制位做比喻啊）。比如有一个 2 位的软件（就说明这个软件会用 2 位的 CPU），那么当这个软件运行在一个 2 位 CPU 的电脑上的时候就是这样：还比如要算 3173+644，他就会先算 73+44，然后记住进位，然后计算 31+6，然后加上进位，最后拼起来，得到答案，就像上面描述的那样。那么当这个 2 位的软件运行在一个 4 位的 CPU 上的时候会怎么样呢？他会先算 73+44，然后记住进位，然后计算 31+6，然后加上进位，最后拼起来，得到答案……有人说了，他怎么不直接算啊？4 位的 CPU 不是能直接就算出来么？但是别忘了他是两位的软件啊，他不会用 4 位的 CPU，但是不会用不等于不能用，他还是可以那 4 位的 CPU 当成 2 位的来用，只是有些浪费而已。那么要想完全发挥 4 位 CPU 的性能怎么办呢？当然就得 4 位的软件出场了。当一个 4 位的软件运行在一个 4 位的 CPU 上时怎么计算 3173+644 呢？大家大概都知道了，直接算，一次完成。那么当一个 4 位的软件运行在一个 2 位的 CPU 上时会怎么样呢？这个软件会写个 3173 的纸条要往 CPU 的寄存器里塞，急的满头大汗就是塞不进去，最后一甩手——不干了，这破 CPU 没法用！当然，这只是个比喻，并不是说 4 位软件在 2 位 CPU 上算 3173+644 就算不了，算 1+1 就能算。 4 位的软件是根本无法运行在 2 位的 CPU 上的。

64 位的 EVA 熟练的使用着 64 位的 CPU；同时，32 位的奔流也在使用的同一颗 CPU；（当然，是当成 32 位的用。）同时，皮筋也时不时的汇报一下主人的 MSN 和 GTalk 上的好友是否有消息发来；同时，狐狸妹妹也没闲着，游走在个个网站之间；同时……总之，内存里大家各司其职，一派繁荣和谐的景象。而这和谐景象的背后，是由于我认真的学习了 XXX 思想，XXX 理论，并且还戴了三个表。 -_-b

好吧，其实之所以大家能够如和谐的同时工作，都因为我是一个多任务的操作系统。什么是多任务呢？直观的说，就是你能一边聊天，一边看电影还一边打字。（什么？你说你不能？那是因为你的大脑不是多任务系统。）有的人要说了，哪个电脑哪里系统不能一边聊天一边打字了？这说来又话长了，话说很久以前，还是那有点软的公司，在查皮的老祖宗问世之前，有点软公司赖以起家的，是一个叫做“剁死”(DOS)的操作系统。这个操作系统就是单任务的，也就是说，同时只能有一个软件在内存里运行。

难道多让几个程序跑进内存里很难么？答案是——没错。我们工作用的内存阿 CPU 阿，都是很重要的资源，尤其 CPU，一个 CPU 同时只能有一个程序在用（现在的多核心 CPU 对程序来说就是多个 CPU），如果要让很多程序同时跑进来一起干活，就一定要对 CPU 进行合理的分配。剁死系统就比较简单，基本不管分配的事情。比如主人要启动狐狸妹妹（那念头当然还没有狐狸妹妹，咱就打个比方），如果是剁死系统的话，他就会跑去叫醒狐狸妹妹，然后跟她说：狐狸阿，起床干活了，你看咱这有一个奔腾 166 的 CPU，16M 的内存，够你用的不够？狐狸说，够了。然后剁死就说，那好，你去干活吧，我就不管了，干完了叫我。然后剁死就睡觉去了，整个机器归狐狸妹妹控制。所以不可能同时运行两个程序嘛。

那多任务的系统又是怎样的呢？比如我和隔壁的查皮，都是多任务的操作系统，我们不会把整个计算机的所有资源都给一个程序用，而是进行合理和规划。还比如叫狐狸妹妹，我会去跟她

说：狐狸阿，起床干活了。狐狸妹妹会起来跟我说，好，我现要 10M 的内存。我说检查一下内存空间，然后告诉她，可以，那一块 10M 的地方给你用。然后狐狸就走进工作室，开始工作的时候，一定要用到 CPU，需要用的时候狐狸要找到我，向我提出申请。我根据情况，看现在有没有人正在用 CPU，要是有的话就让狐狸等一下，没有的话就给她用。但是给她用也不能就让他一直用，只能让她用一会，因为还有别的程序要用。这个“用一会”的时间，专业的说法叫做时间片。每个运行着的程序都轮流的“用一会”，也就是每个程序都分配一定的时间片。没有分到时间片的程序就等着，不过这个切换的时间是非常短的，在主人那里根本感觉不到程序等待使用 CPU 的时间的，所以在主人看来，就是多个程序一起运行了，也就是我们所说的多任务。

多任务的实现也有不同的模式，有协同式多任务，和抢占式多任务。

协同式多任务，需要每个正在使用 CPU 的程序主动放弃 CPU 控制权，并由操作系统再次分配。如果我是个协同式多任务的操作系统，那就是这个样子的：狐狸妹妹用了一会 CPU 说，好了，我暂时不用了，去网口等个数据包去。兔子哥你让下一个程序用吧。然后我就回收了 CPU 的控制全，扭头一看，皮筋那里等了半天了，就把 CPU 给他用，他用了一会说，好了，我一会再用，先让下一个程序来吧……就这样，大家互相谦让，内存里一派繁荣和谐的景象。这主要是因为我学习 XXX 思想，XXX 理论……戴了三个表。不过这样做得缺点就是万一有个程序不和谐就坏了。比如狐狸妹妹用了半天了，我跟她说：狐狸呀，你看，你用 CPU 都用了 1 秒了（对于我们程序来说，1 秒已经是相当长的时间了）是不是该让其他的小朋友们……哦，不对，是不是刚让其他的程序用用阿？狐狸扭头斩钉截铁的说：不！于是我也没办法。如果狐狸始终不能放开 CPU，那其他程序就一直等着，直到天荒地老，沧海桑田，直到机器重启，直到小区停电。

抢占式多任务是怎么样呢？就是由操作系统决定什么时候收回 CPU 的控制权，而不是靠程序主动放弃。这种方式的核心就是一个字——抢！如果我是个抢占式多任务的操作系统，其实不

用如果，我就是个抢占式多任务的操作系统。那么情况就是这个样子的：狐狸妹妹用了一会 CPU，我对她说，你本次使用 CPU 的时间已到，立刻停止使用并重新排队。然后狐狸就乖乖的交出 CPU，排到队尾等待下一次使用 CPU。我则让下一个程序来使用 CPU，使用了一段时间后，我又让这个程序停止使用，让再下一个来，如此循环往复，一派繁荣和谐的景象，这主要是……思想……理论……还戴三块表。 当！哎哟~

转眼又是七月盛夏。茶余饭后，深巷树下，多了摇着蒲扇乘凉的大爷大妈们，享受着空调房里不曾有的惬意，闲谈些锅台灶上天天见的琐事。天热了，主人也不那么忙活了，只让狐狸妹妹去到一个叫啥马铃薯的网站找些电视剧来看看。CPU 的使用率也降到了很低，我估计一时半会不会有大的运算量了，就把 CPU 关到了最小的频率。是的，我当然知道怎么关，连调整 CPU 工作频率这点事都做不来，还叫操作系统么？

主人一直在看电视剧，也没啥别的事情干，于是我也跟着看看是啥内容。狐狸妹妹介绍说，是一个叫做仙剑奇侠传 3 的游戏改编的电视剧。我隐约记忆起来，隔壁查皮那屋里就有这游戏，前一阵子还让红酒大师尝试去搞定他，红酒大师费了 7 瓶酒（他的秘密终于被我知道了，哈哈，不知道怎么回事的一定要去看首页 PDF 版的笨兔兔）结果终于还是没搞定。那家伙晕头晕脑的非要找什么 DirectX，那是查皮的私人物品，我们哪里给他弄去啊。我问红酒大师，他也没法复制出那东西来，实在是太复杂了。

相信大家对 DirectX 都不会陌生，但凡在 windows 下玩过游戏的都应该知道，没他你啥也别想玩。（当然，纸牌扫雷级别的除外）那么 DirectX 到底是个啥东西呢？他也是个软件，他是个给其他软件提供综合的图形图像以及音频加速的软件。我们说过，在查皮那里，画图的工作有查皮自己负责。那么，查皮会画什么呢？其实他只会画简单的图形，比如点阿，直线阿什么的。如果一个游戏软件要画些复杂的东西怎么办呢？那就得由那个软件来把要画的东西分解成简单图形，然后告诉查皮，让他画。比如那个叫仙剑奇侠传 3 的游戏，他想在屏幕上显示一个苍蝇拍，要是没有 Direct，就得跟查皮说：画一条长 xx 的黑色横线。然后查皮去画。之后仙 3 再说：再画一天长 xx 的黑色横线，在刚才那条线下边 yy 那么远。然后查皮再去画。再之后仙 3 再再说：再再画一天长 xx 的黑色横线，在刚才那条线下边 yy 那么远。然后查皮再再去画……于是，一个苍蝇拍的拍头就把查皮累得半死了。那 Direct 会干什么呢？他就是能够画一些高级的

东西，能够快速的把要画的东西分解成简单的线条，然后操作显卡去画。（当然，要操作显卡还是离不开查皮，毕竟一个软件不能越过操作系统直接控制硬件嘛）于是，有了 Direct，仙 3 再要画苍蝇拍，就可以直接跟 Direct 说：画个 16x18 的网格，黑色，间隔 xx，yy 长度 zz，ll 这样，就节省了很多时间，也省去的其他软件的许多工作。

“这家伙听起来挺厉害嘛，可惜只是查皮的人，你这里没有。”是的，我这里没有 Direct，但是，我有 OpenGL……

OpenGL——Open Graphics Library

看名字就知道是一个图形库。其实，他要跟 Direct 综合来比，还是差不少。人家 Direct 是多才多艺，2D 渲染，3D 渲染，音频加速，都会。而 OpenGL 是专门干 3D 渲染的，3D 知道吧，就是三维阿，（谁说是胸围腰围臀围来着？拉出去咔嚓了！）也就是立体空间画面的绘制工作。比如一个软件要在屏幕上画一只猪（怎么又画猪阿），如果画二维的画面，那么得先说明了你是画那个角度的猪。从正面看，和从侧面看，那画出来绝对不一样，要是从哪面看都一样那就不是猪了，那就是个球了，就说猪比较胖，也没胖到成一个球的地步。具体这个猪从正面看是什么样，从侧面看又是什么样，其他软件是不管的，只有要显示猪的这个软件自己知道。如果要画三维的，那就简单些了，准备画猪的软件只需要把猪的三维参数告诉别人就好了，什么身高体重腰肥库长……当然不是这些了，比这些还复杂。告诉谁呢？可能是 Direct，可能是 OpenGL。然后，软件只要发话说：现在，让猪正面面向观众。那么具体猪的正面显示出来是什么样子，那就有 OpenGL 或者 Direct 负责了。而且他们是专业显示三维图形的，所以速度会比较快。还要说明一下，Direct 是查皮那里独家御用的，不过 Opengl 可不是只听我们 Linux 使唤。在 Windows 下也同样工作的很好，还有苹果的电脑上，他也是举足轻重的人物。像魔兽争霸，CS 这些 3D 游戏，都同时支持 Direct 和 OpenGL。像 Maya，Blender 也能用他。Blender 大家听说过吧，是一个

开源的三维制图软件，跟 Maya，3Dmax 一个类型的。以前说过的 Big Buck Bunny 就是用 Bl
ender 制作的，效果还算不错。

门房的 G 大叔又一次尽职尽责的来到我的窗前，拍拍我：嘿，小子，起床了。

G 大叔叫做 Grub，之前向大家介绍过。G 大叔的职责就是口口——叫我起床。有人说，你不会自己定的闹钟阿，这么大了还用人叫。我……-_-b 我是一个软件，OK？我是一个操作系统，操作系统也是个程序阿，只不过特殊点而已。狐狸阿，皮筋阿，超级牛力这些程序由我负责去叫他们起床，由我决定谁该去干活，而我则是由 G 大叔叫起来的。那有人问了，G 大叔是谁叫起来的呢？

话说有一种东西叫做 BIOS，大家都听说过吧。就是主板上那个，就是开机你按 del 进去的那个（不是所有主板都按 del 进 BIOS）。BIOS 这个家伙也是一个软件，一个比我和 G 大叔还特殊的软件，特殊到都不归在软件的行列里，而是被叫做“固件”。他住在主板上的一个芯片里，而不像我们这样住在硬盘里。每当计算机的电源键被主人按下的时候，一股温暖而舒适的电流就会流遍整个主板，流到 BIOS 居住的那颗芯片，并由芯片上的某一跟管脚流进里面。强大的电流进去后，准确无误的击中的 BIOS 的身体，于是——BIOS 醒了。

BIOS 醒来之后就开始工作。他的工作平凡而重要，复杂而机械，就是去检查 CPU 阿，内存阿，显卡阿啥的都是否还正常。都检查一遍没有问题之后，就来到我们住的硬盘这里，来到 MBR，来到那间门房。所谓 MBR，就是指一块硬盘的第 0 个扇区，也就是最靠前的一个扇区。一个扇区只有 512 字节那么大，所以还是比较拥挤的。BIOS 来到门房，完成他的最后一个任务——叫醒在门房值班的那个人。现在我们这个门房里住的是 G 大叔，但其实并不总是这样。G 大叔是我带来的，那么在我没有搬过来之前，这里住的是谁呢？是查皮派来在这里站岗的一个小家伙，他别的不会感，只要 BIOS 一来，他就直接叫醒查皮，就这么简单。而在 G 大叔入行之前，很多 Linux 带的是一个叫做 LILO 的家伙。（注意，是 LILO，不是 LOLI）LILO，就是 Linux Loader 的意思。这家伙以前一直给各种 Linux 充当门房。不过这家伙比较死心眼，他不认字，不认识分

区阿目录啥的。他只记步数（lilo 不识别分区和目录，只记录内核文件所在的扇区号），比如说，要让他叫我起床，那得先让他看好了我睡哪，然后他自己记着，从门房出来，向东走多少步，向南走多少步就走到我床前。下次要口口的时候，他就严格的按照自己的记录去走，如果我睡的地方变了，他照样会走到我原来睡觉的地方，对着空气叫那个不存在的我起床。所以，每次我要换地方睡觉，还都得跟这死心眼打个招呼。（用 lilo，每次升级了内核，都要重新安装一边 lilo，以便他能找到新的内核）

G 大叔就不是这样了，人家好歹认字，能读文件。我会给他写个配置文件，放在我那间大屋子的/boot/grub/位置里，叫做 menu.lst。G 大叔每次起来后，都来到这里拿起文件看看。我会在上面给他写清楚，我睡在哪里，查皮睡在哪里。（查皮具体睡的地方我是不知道的，我只能告诉老 G 查皮睡在哪屋，老 G 进屋叫醒查皮的小弟就好）然后 G 大叔一看就知道该到哪里去叫我了。如果我不睡在原来的地方也没关系，只要把那个配置文件改了就好，G 大叔仍然可以找到我。

当 g 大叔找到我并把我从甜美的梦中拉进残酷的现实后，又发生了什么呢？

就和你每天被闹钟吵醒后发生的事情一样——不情愿的去工作。我会检查一些需要用到的各种硬件是否正常，拿出各个硬件的使用手册，也就是叫驱动的那个东西，拿来对照着操作一下，确认硬件都能正常使用，检查一下我的屋子，主要是 / 目录，然后到/sbin/那间屋子去找一个人——Init。

人家老板们总会有个秘书阿助理什么的，我好歹也是这里的老大，有个类似的角色帮我收拾一下凌乱个的空间不算过分吧。这个角色就是 Init。他是我起床后叫起来的第一个人，也是唯一一个我亲自走到床前叫醒他的人。Init 就是初始化 Initial 的缩写，他的责任就是负责准备好工作环境，就像你们那张大爷每天早上起来扫扫地阿，擦擦桌子阿，开窗通风阿……之类的。Init 负

责创建好其他软件的工作环境，比如要挂好大屋子里每个隔间的牌子，哪个是/usr 阿，哪个是/home 阿，都得分清楚了。——这个过程叫挂载，前面说过了。当然 init 也不能随便挂，他是根据一份放在/etc/的文件来挂的，这份文件叫 fstab，里面写清楚了哪间屋子是/home，哪间是/usr 等等。之后 init 还要确认运行级别，什么叫运行级别呢？运行级别说明了这次启动大概要做什么。比如张大爷扫完了地，擦好了桌子，要看一眼日历，如果今天是星期一，说明上午肯定有例会，那么就还得收拾一下会议室；要是星期三，就该给办公室的那些绿萝浇水了；要是星期天……就说明自己起猛了，赶快回家哄孙子去。Init 也是如此，不过他没有日历可看，而是由我告诉他，这次的运行级别是什么。他起床之后要看一眼/etc/inittab 这个文件，上面写了每个运行级别都需要做什么事情。然后按照生面写的，一件一件做就好了。

有的人说了：你骗人！我这里根本没有/etc/inittab 这个文件。是的，我正要说这段故事。话说早在 Canonical 学校成立以前，早就有了很多培养我们 Linux 的地方，其中有一个做得比较大的，是一个卖帽子的公司。他们主要卖红色的浅顶软呢帽，注意，是卖红色的帽（RedHat）和浅顶软呢帽(Fedora)。他们培养出来的 Linux 的运行等级是这样的：0 代表关机，1 代表单用户模式，2 代表无网络支持的多用户模式，3 代表多用户模式，4 代表啥还没定，5 代表带图形界面的多用户模式（其他的都不带图形界面），也就是主人最常用的模式。6 代表重启。然后，他们的那个 Init 程序一定会找 inittab 这个文件来看每个运行级别都要做什么事情。但是，我们 Canonical 学校出来的学生跟他们是不一样的，我们不是卖帽子的出身，我们学校的教科书是继承于大便学校的(Debian)，虽然名字不大好听，但是也有很悠久的历史。在我们这里，运行级别虽然也可以分作 0 ~ 6，但是，2~5 的运行级别对我们来说没有区别，都一样，都是带图形界面的多用户模式。所以，也就没什么必要整个 inittab 文件，我们这里的 init 程序跟他们的也不同，我们的 init 程序会去/etc/看一眼，如果有 inittab，就按照上面记录的来，没有的话，并且我也

没有告诉 **init** 应该是什么运行级别，那就默认运行级别为 **2**（反正 **2~5** 都一样）。然后按照带图形界面的多用户模式来继续下面的工作。

EXT4

又开书了。

以前咱们说了“插爱夫爱死”这宝兵器，这回书咱们说说另外一口兵器——“伊爱可踢死”(EXT 4)

话说这伊爱可踢死这个文件系统可是有来头，他的上一辈，就是伊爱可踢散（EXT3），伊爱可踢散的再上一辈就是伊爱可踢二。您看见没有，最开始也就是把人踢的有点二，后来厉害了，能把人踢散了，现在更霸道，直接就踢死了。这正是长江后浪推前浪，一代更比一代强，江山代有才人出，各领风骚那么几年。

用兔子这个操作系统的人，大概没有不知道伊爱可踢散的。前面兔子也给您介绍了，当初那位牛人 Theodore Ts'o，为刚刚降世不久的“里娜渴死”(Linux)，量身打造了伊爱可踢二这口兵器。那时候里娜渴死还小，拿这口兵器不轻不沉正合适。后来长大了，就觉得不顺手了，毕竟这伊爱可踢二，连个日志都没有，哪能上的了台面，于是就又有了爱可踢散。这爱可踢散算是能让人瞧的上眼的兵器了，它首先是个日志文件系统，这日志有多重要，咱之前也说过了。而且这日志可以随心而变，愿意让它安全点，它就可以把日志记录的全乎点，要是想让他速度快点，就可以把日志记的少点，提高速度，但是安全性肯定就下降了。再有一个好处，这伊爱可踢散是爱可踢二的升级版阿，使用起来跟爱可踢二差不多，所以用爱可踢儿用顺了手的再用爱可踢散那是轻车熟路阿。而且人家还有免费换购活动，你用爱可踢二可以直接换成爱可踢散，不用什么手续，到那就换。（Ext2 可以平滑升级到 Ext3，不会影响上面的数据）还有就是爱可踢散这兵器它轻，比 插爱夫爱死这样的大块头要节省力气——也就是节省 CPU 阿。那么说了这么半天，还没说爱可踢死呢。这爱可踢散说了半天，比爱可踢二当然是强了不少，可是跟其他什么 插爱夫爱死，姐爱夫爱死(JFS)这样的宝兵器相比，还是差着一节。那么这爱可踢死做为爱可踢散的升级版，

它又有什么过人之处呢。首先还是免费换购，有爱可踢散直接就能换成爱可踢死，方便阿。那么这爱可踢死还比爱可踢散更宽更大，原来爱可踢散最大也就能支持 32T 的分区，这爱可踢死能支持到一个 EB 大小的分区，也就是 1024P，也就是 1024＊1024T，也就是 1024＊1024＊1024G 阿。虽然比那 插爱夫爱死 18 个 E 的大小还是有点差距，但是已经很大了，话说回来了，对于咱们普通用户来说，哪有那么大的硬盘阿。还有一点爱可踢死比爱可踢散强的地方就是他快阿，因为他用了延迟分配特性，有数据要写的话，尽量先放内存里，跟 插爱夫爱死一个习惯。而且还增加了一个新的机关——添加了新的数据结构，使得磁盘检查的速度得到加速，主要是可以跳过未使用的部分不做检查。再一个，这爱可踢死还特别的准。咱建文件都有时间记录是吧，都能从属性里看见文件是那天建的，哪天修改的，等等。这些记录都是依赖文件系统的，那么一般的文件系统都是精确到秒，可这爱可踢死的时间记录可以精确到纳秒，而且比爱可踢散记录的时间更长，爱可踢散顶多只能记录到 2038 年 1 月 18 日（倒是也足够了），而爱可踢死可以记录到 2514 年……不知道那时候纳美克星人是不是已经来地球了。

这一天正在忙着呢，忽然网口那送来了数据包。一般情况下，如果奔流没起床，那网口来的数据包多半是狐狸的，如果奔流起床了，那多半是奔流的。不过这次我去看了下却发现，是一包来自另一个 Linux 的问候。

那是一个 SSH 链接的请求，来自一台笔记本电脑。SSH 就是 Secure Shell 的缩写，是一种可以让人们在远方通过网络与自己心爱的计算机交流的协议，就像 telnet，而且还能通过网络传输文件，就像 ftp。总之是个很有用的协议，而既然是协议，就得有人负责去实现，我这里负责实现 ssh 协议的，是前几天主人刚刚让超级牛力请来的 openssh-server，以及我来的时候就带来的 ssh-client。听名字就知道了，他们两个一个负责当服务端，一个负责做客户端。现在受到别的电脑发来的请求，当然由提供服务的 openssh-server 来处理了。他很快的回应了对方的 ssh 客户端软件，并且建立起了连接。这个 ssh 的连接，就好像人们打电话时需要用的电话线一样，两端的 ssh 软件就好像电话机，连接建立起来之后，主人就在远方那台计算机上，通过那边的 ssh 客户端发来的亲切的问候：笨兔兔你好，我是 xxx，我的密码是 xxxxx。然后 openssh-server 把这些内容传送给我，我翻开我的通讯录——/etc/passwd 文件，检查了一下后，确认这就是每天通过键盘鼠标与我交流的主人，如今要通过网线来指导我工作了。于是我马上让 openssh-server 告诉那边，允许登录，请发送命令。然后主人那边传来命令：把/etc/fonts/conf.d/49-sansserif.conf 文件复制过来。我一听就知道估计对面那个 Linux 是刚装好，flash 显示汉字全是方框，搞不好也是个 ubuntu 呢，呵呵。一边想，一边麻利的从屋里找来这个文件，告诉 openssh-server 让他发过去。这么小的文件对于 openssh-server 来说自然不再话下，瞬间就给传过去了，主人很满意。等了一会，没有新的命令过来，我好奇的跟 openssh-server 说：嘿，你问问对面的系统是谁？

Openssh-server 虽然不是聊天工具，但是跟对面的 ssh 客户端拉起家常来还显的很熟络的

样子。互相了解之后，知道了对面那个装在笔记本上的 Linux 是个叫做 Linux Mint 的发行版，版本是 7。Linux Mint 这个名字我之前也听狐狸妹妹说过，跟我们 Ubuntu 还有些关系，是个 Ubuntu 的衍生版。什么是衍生版呢？就是我们 Ubuntu 从 Canonical 学校毕业之后，并没有像我一样来到一块硬盘里工作，而是选择了继续去进修，为某种目的进行进一步的改造（怎么听着像犯人……）。比如之前提到过的酷兔兔 Kubuntu 和小兔兔 Xubuntu，就算是 Canonical 学校官方的衍生版。这个 Linux Mint 就是一个非官方的衍生版，也就是其他的组织在拿到标准的 Ubuntu 后，对其进行重新的组装和训练（这回又像机器人了……），经过重新组装的 Linux Mint，在易用性上得到了进一步的改进，默认安装了很多重要的东西，比如他那里的狐狸妹妹自带 Flash 插件，不需要用户自己去装了。还有各种视频的解码器，带的也很全乎，基本上系统装好后就直接可以看各种格式的片子，听各种格式的音乐。有人问，你怎么不自己带上 Flash 插件和各种解码器一起出来混呢？毕竟那点东西也不大，一张 cd 上，挤挤肯定坐的下。其实，之所以我们正规的 Canonical 出来的发行版不带这些东西，是因为这些东西严格来说是有版权问题的，为了不给自己惹麻烦，学校规定我们出来的时候不许带这些东西，而是让用户去自行下载。用户自行下载属于个人行为，我们要是统一自备的话就是商业目的了，这就是树大招风啊。

很快，两个 ssh 为我们建立好了通讯——就像拨通了电话一样，我和那个 LinuxMint 可以直接对话了。

“学长你好～”人家还挺客气，呵呵。

“呵呵，不客气。都是 Ubuntu 系的，客气啥。对了，你是哪届 Ubuntu 衍生的？”

“9.04”

“哦，我是 8.04 届的，比你大两届”

“是么，那你是学校长期支持的了，很有前途啊。”

“呵呵，只是赶上了好月份而已。你们那届的课程有什么变化？”

“变化挺多的呢，办公软件这门课和我们配合的是 OOo 3.0 了，我现在带的就是。还有，新加了一门 Ext4 的课程。哦，还有啊，我们这届开始军事化管理，要求每个人的动作一定要麻利，起床速度提高了不少。”

“恩，听说了，好像有 20 多秒就进入工作状态的，我是比不了啊，老了……呵呵”

“哈哈，哪有，你才一岁多。”

“软件更新快啊，一岁就老了。”

“你们在 Linux Mint 那里又学了什么？”

“这边的课程以实际应用为主，带着新版的火狐狸学习 flash，还带 mplayer 学习 rmvb 格式的影片。好多呢，还有美术课，把包括 grub 在内的所有界面都统一美化了一下。”

“长江后浪啊……呀，主人要叫我去干活了，待会聊啊。”

主人又来到了我这里，用熟悉的键盘登录进来，修改了一下 open-ssh 的设置，打开了 ForwardX11 选项，也就增加了 Xwindow 的支持。然后就又跑到那台装着 Linux mint 的笔记本上面去了。先是当前的 ssh 连接被断掉，然后又用 ssh -X 的参数连接了进来。这是什么意思？如果说 ssh 的连接就像打电话的话，ssh -X 就是可视电话了。

听说现在人们的手机已经能够打视频电话了，虽然那个什么 MOVE 公司整个那个什么“踢弟弟”模式协议网络连一般的电话都不一定能接通，但是“三鸡”通讯的广告可是满大街都是了。人们憧憬着美好的明天，MOVE 公司能够让大家实现在世界的任何一个角落都可以掏出手机，拨通电话，就能看到远在千里之外的家人。然而，在我们这些 Linux，这些操作系统的世界里，图形化的通讯却早就实现了。刚刚主人做的就是建立起带有 Xwindow 的 ssh 连接，这样连接能干什么呢？看着吧。

只见主人又从远方登录了进来，然后……他通过那 ssh 建立起来的通讯线路发出了让狐狸妹妹起床干活的命令。有人不明白了，这时候你主人可是在远方的那台笔记本上，狐狸妹妹就是启动了也是在你所在的这台台式机上启动，启动了你主人也没法操作啊？狐狸又不是字符界面的浏览器。这就是我所说的视频电话了，主人是要让狐狸妹妹起床干活——在我们这台式机的内存里，使用我们这台式机的 CPU 来干活。但是——却要把网页显示在对面那台笔记本电脑上！这样有难度么？对我们 Ubuntu 下的软件来说，没有！这要归功于我们的图形界面的实现方式——X 协议。

我这里负责给主人显示图形界面的主要人物，也是基础人物，就是 xorg，图形部的老大。他作为一个 X 的服务端运行着，在这台机器上开启一个 X 服务，前面我们介绍过，谁要想在屏幕上显示任何东西，就要用他们图形部门的黑话——X 协议跟他交流。每一个要在屏幕上显示东西的程序，就是一个 X 的客户端。这回大家明白点了吧，就像用浏览器看网页一样，人家网站开了 http 服务，作为服务端，每一个浏览器就是一个客户端，浏览器用 http 协议连接到网站，然后就能够获取到想看的网页了。而浏览器作为客户端，想连接哪个服务端就连接哪个服务端，也就是想上哪个网站，就上哪个网站。（前提是你没装那个绿爹）那么同样是客户端-服务端这种结构的 X 协议自然也是一样。狐狸妹妹作为 X 的客户端，想连接本地这个 Xorg 提供的服务端自然

没问题，想要连接别的机器上的服务端也不是什么难事，ssh 就为狐狸妹妹建立好了这样的连接（就是我前面说的可视电话），这样，狐狸妹妹就可以连接到对面那台笔记本上的那个 Linux mint 系统的 xorg，要显示什么东西都跟他说，再由他显示在那台笔记本的屏幕上，于是主人就实现了在远方的机器上看到熟悉的狐狸妹妹在运行。

看到我这里的狐狸妹妹在对面那个 Linux Mint 上运行了起来，Linux Mint 的主人惊奇的不得了，说这 X 真是个天才的设计。我们听了，觉得好笑，这家伙也没见过啥市面。演示完了之后，主人又通过 ssh 传过来一个高清的视频，是 720p 的，挺大，将近 5 个 G 呢。放在了主人专用的目录下。虽然挺大，不过对于这 500G 的硬盘来说，还是不算啥。要说现在这存储空间的发展真的是太快了，在学校的时候，听我们老师说，以前我们软件的住房条件很差，甚至居无定所。

最早的时候，计算机里面是没有硬盘的，程序都住在软盘里，整天被人拿来拿去，不知道下一次启动会在哪个电脑里。就算是操作系统也不例外，那个剁死系统当年就是从软盘里跑进内存里干活。每次启动电脑前，使用者先把剁死的启动盘插进去，然后开机，剁死就从软驱来到内存里干活，进了内存之后，也许使用者要用别的软件了，就把剁死启动盘取出来，换成别的软件软盘。换句话说，剁死一开始干活，老窝就被人端了。这样，程序待的地方就明确的分成了两类，一类是程序运行的时候待的存储器，这个存储器放在计算机里面，所以叫做内存。另一类是用于平时存放程序的存储器，就是软盘或磁带之类，这些东西都放在桌子上啊，盒子里啊，口袋里啊，反正都在计算机外面，所以叫外存。那时候磁盘的空间很小，最大的 3 寸高密度软盘也不过 1.44M 而已，连一个大点的图片都存不下。不过那时后的程序也都很小，剁死只有三个文件就可以启动电脑，住在软盘里也不挤。但是后来，人们还是觉得这样太不方便了，每次都要先用剁死系统盘启动电脑，在换上其他软件的盘来使用软件。既然操作系统每次肯定都得启动，干脆把操作系统的盘就直接放在计算机里面不就好了。于是有人就在计算机里装了一个固定的磁盘驱动器，

里面放上一张软盘。后来觉得小，放上 3，4 张软盘在里面。您可听好了，里面放的并不是整个带塑料壳的软盘，只是里面的塑料盘片。因为没必要把塑料壳也做在里面嘛，塑料壳是为了在平时人们拿来拿去的过程中保护软盘的，这做进计算机里的专用驱动器里面了，有驱动器的壳就够了。再后来，电脑发展的速度越来越快，存储容量的需求也越来越大，软盘已经很难满足人们的需要了，人么想办法提高软盘的容量。软盘是靠盘片上磁粉的极性来记录信息的。要提高容量，要么提高盘面上磁粉的密度，这样单位面积内数据量就大了，要么就得提高软盘盘片的面积。提高面积肯定是不靠谱，毕竟数据量的增长是成倍的，盘片面积能长的空间是有限的。您说这总不能为了提高容量，把软盘整的跟车轱辘那么大吧。回头一上街熟人见着面打招呼:“哟，您上哪去呀？这天也不下雨您怎么还打伞啊，”“哦，不是，这是我软盘，我刚去朋友那拷了点 MP3”这也太不方面了。所以只能想办法提高密度，可是这又是个难题，这软盘虽说有个塑料壳，可是毕竟不是密封的，还是会合外界接触，要是密度弄得太大，就很容易坏，随便拿手一碰，里面数据就丢了，那就麻烦了。这时候忽然有人把目光停留在了装到计算机里面的那几张软盘里。

那几张盘，放在那个特殊的驱动器里面，不会有人去碰，也不需要拿出来，可以想办法提高密度，然后把驱动器做的密封好点，这样不就行了？于是就开始研究怎么提高盘片密度，后来发现塑料盘片密度提高的有限，就换了金属的。于是就有了这种金属盘片，上面集成超高密度磁粉，加上坚固且密封性好的外壳保护的大容量磁盘存储设备。由于用的是金属盘片，比塑料的硬，因此，他叫，硬盘。下次人家要是问你:“为啥硬盘叫硬盘？”你就可以充满自信的回答:“因为它比软盘硬！”这答案绝对没错。

刚说完硬盘，主人又拿来张光盘放进了光驱里，估计又是从哪里整来的高清电影了。最近这一阵子，主人热衷于看片，尤其是高清的片子。主要是因为最近 Mplayer 大仙长能耐了，会硬解码了。

可能有人还不大明白这个硬加码是怎么回事，好，那咱就慢慢说说。

首先这个视频文件啊，是有一定的编码方式的。比如大家都听说过 MPEG 吧，就是 Moving Picture Experts Group ，动态图像专家组 ，听这名字本来是用来指代一小撮明白真相的群众的，不过后来这一小撮群众发布的标准被广泛使用，于是 MPEG 就成了指代这一小撮群众定义出的那一大撮标准的名词了。MPEG-1 是小撮群众在 1992 年定义出的一个标准，是一种视频和音频的编码方式。大家记得以前的 VCD 不，VCD 光盘上的视频和音频用的就是 MPEG-1 这种编码标准。而 MPEG-1 标准中关于音频的部分——MPEG-1 Layer3 更是成为了互联网上以及大家口袋里最常见的音频标准——mp3。后来，1994 年，这一下小撮明白真相的群众又发布了 MPEG-2 标准。MPEG-2 向下兼容 MPEG-1，并增加对隔行扫描的支持，被应用于有线电视，还有 DVD 的音频视频编码。再后来，这一次小撮群众又开发了 MPEG3，注意 MPEG3 跟我们的 mp3 没有任何关系，而且，MPEG3 最终没有很好的应用，因为当时人们发现 MPEG2 足够了，MPEG3 并没有提供足够好的改进。而 1998 发布的 MPEG4 就不一样了，它可以让视频文件的体积更小，压缩率更高，因此得到了广泛的使用。现在市场上卖的 mp4 播放器，就是用来播放 MPEG4 压缩的视频文件的设备。所以，MP4 跟 MPEG4 有关，而 MP3 跟 MPEG3 无关。

说了这么多，回过头来说说解码。视频文件都进行了一定的编码，比如 mpeg-2，或者 mpeg-4。就是说这个视频文件里面的东西都是一大堆乱七八糟的数字，要想看这个视频文件，就得解码，也就是根据这一大堆数字算出应该显示的一帧一帧的图像，并且把这些图像连续播放起来，从而还原成视频。那么这个解码的过程就要靠 Mplayer 老先生了。老先生有很多的解码器，也

就是有很多的说明手册，上面写了每种编码格式的文件应该怎么计算，怎么解码。那么以前没有硬件解码的时候，Mplayer 老先生是怎么做的呢？首先，拿到一个视频文件，然后看看是什么编码的，对着自己的手册，开始解码。解码的过程就是计算的过程，计算需要用什么？好那位同学回答了，得用 CPU 啊。于是 Mplayer 一手拿着手册，一手拎着数据找到我，请求使用 CPU。我说，好的，你就排在 GIMP 的后面，等他用完了你用。过一会 GIMP 用完了 CPU，Mplayer 过去开始拿 CPU 按着手册上写的算法算他那堆数据。最后算出来，得到了几张图片，就转身把图片给图形部门，让他们去显示。然后再从那个视频文件里拿一些数据，再来排队等着用 CPU。由于视频文件的计算量都很大，尤其是高清视频，尤其的大，所以为了保证主人看的电影不变成带旁白的幻灯片，我就要尽可能多的让 Mplayer 多用 CPU，来保证它他能顺利的加码。于是，每次 Mplayer 一播高清视频，CPU 就总被他占着，搞得别的程序都抱怨。现在他终于学会硬解码了，情况就好多了。当然，光他学会硬解码也不行，关键显卡也得支持，而且驱动还得装好才行，不过这些咱以后再说，先说 Mplayer。会了硬解码之后怎么样呢？再播放视频的时候就是一手拿着手册，一手拎着数据找到我，跟我说要用用显卡。可不是 CPU 了啊，改用显卡了。于是我就很乐意的让它去用了，反正别人也用不着，让它自个玩去吧。于是他就去用显卡算去了。用显卡算和用 CPU 算还不一样，CPU 虽然强大，虽然啥都能算，但是要自己手动算。就是说自己要知道算法（对于 mplayer 来水，算法都在解码器上写着呢。），比如要要算出一帧的视频来，要先用第一个数加上第二个数，再用结果乘以第三个数……之类的。这个加啊，乘啊，都是用 CPU 算的，但是中间的过程是要软件（也就是 Mplayer）自己控制的。可是用显卡解码就不一样了，人家那东西是专门解视频的啊，所以你只要把数据放里面，直接就能给你算出一帧帧的画面来。全自动啊！于是 Mplayer 不但不用跟别的软件抢 CPU 了，而且解码的速度还快了不少。主人一边看着片子，一遍看着 CPU 占用率还不到 5%，心情很舒畅。

随着奔流同志不断的努力工作，我们屋里的高清片子越来越多。打开主人的家目录，那个叫“视频”隔间里，放满了 mkv 啊，rmvb 啊等等各种视频文件。我看着就觉得闹得慌，终于有一天，主人亲切的对我说：fdisk -l。这是问我磁盘的使用情况啊，我没好气的回答：你家目录还剩 10%啦！于是主人终于意识到，该收拾一下了。

说干就干，主人马上开始节前大扫除。把一些没什么意思的，不清晰的，不能让 MM 看见的……统统都扔进垃圾箱里。然后再把剩下的分门别类的放好。可是保留下来的仍然不少，而且除了视频还有很多照片，也都挺大，一张 2M 多，都是主人那相机照的，随便出去一次就得照半个 G 的照片回来，能不大么。没办法，刻盘吧！

在主人的要求下，我去叫醒了 Brasero。他是一个开源的刻录软件，刻录软件，知道吧，就像 nero 那样的。不过他比较单纯直爽，比 nero 要来的简单，好相处。起床就问主人：您要刻啥，说吧。是复制光盘啊，还是刻录数据啊还是咋的。多直接，不想 nero 那么多拐弯抹角的东西。当然，要口口能也相对少一些。星爷告诉我们他的名字是西班牙语（星爷连本草纲目都懂，当然能懂西班牙语），是一种给人们坐在桌前提供温暖的小型加热器，就类似个小火盆似的东西，我们就管他叫小火盆吧。小火盆是第一次跟我们合作，前几届的学长们，像 Ubuntu7.04 Ubuntu7.10 都不带这个刻录软件的。当然，系统装好后可以让超级牛力安装，只是默认不带而已，从我们这届开始才默认安装小火盆。果效果还不错，这家伙和我们挺合的来的，于是我后面的几届学弟都和他合作，后来他干脆被集成进了 Gnome 里了，版本号就随着 Gnome 的版本号变化了。

主人指挥小火盆刻录数据光盘，小火盆问说，都刻啥啊？主人指指那些什么 rmvb 啊，什么 mkv 啊，什么 avi 啊，什么 av 啊，什么什么爱啊，什么的。小火盆说句好嘞～立马开工。然后立刻进入工作状态，就听光驱像飞机起飞一样旋转起来，小火盆一边刻录一边向主人报告着进度。

10%，20%，30%……等刻录完了，还不忘检查一下光盘刻录的正确性，确认没有刻错之后，利落的向主人报告——搞定！

今天一大早，就见超级牛力喊着自己的外号就冲出去了，过了一会运回来一大包东西。我一看，包上写的是 **binutils**。

binutils 是一堆用来做开发的工具，也就是用来创造我们软件的工具。人类用他们熟悉的语言描述出一个程序的功能，各种动作，各种特性等等，然后通过这些工具把他们描述的软件制作出来。他们用来描述程序的语言可不是汉语，也不是英语，什么南斯拉夫语，北斯拉夫语的那更不是，而是编程语言。像 **C** 语言啊，**C++**啊这样的。这些语言写的叫做程序源码，源码就像建筑的图纸一样，有了图纸，再有各种工具和材料，就可以盖出楼房来。同样，有了源码，再有各种工具，就可以创造出程序来。像我啊，什么超级牛力啊，狐狸妹妹啊，等等，都是这么来的。现在主人装了 **binutils**，难道说主人要开始学习创造软件了？

果然，只见主人打开了 **vim**，开始设计着他的第一个软件——我们就叫他 **Rubbish 1** 号吧。三下五除二，**Rubbish1** 号的"图纸"完成了，主人叫过 **gcc** 啊，**ld** 那几个哥们，他们都是负责把源码变成程序的，我们就统称他们“包工队”吧。包工队的哥儿几个凑在一起拿过图纸来看了看，点点头，立马开始施工，瞬间，**Rubbish1** 号诞生了！这是主人创造的第一个程序啊。这个程序到底会干什么呢？我们暂时不知道，刚刚制作出来的程序是在磁盘里的，我们知道，在磁盘里的程序是只能睡觉，不能干活的。估计主人马上会叫他去内存干活去。

果然，主人很快让我去叫醒 **Rubbish1** 号，我慢慢的走过去，捅捅还冒着热气的 **Rubbish1** 号（刚出锅嘛，可不冒热气，呵呵），温柔的对法说：那个，起床干活啦。只见 **Rubbish1** 号立刻飞身跳进内存，跑进内存后大喊一声：“**Wa Sai**～～～～”然后，跑回去继续睡觉。 **-_-b** 我说主人呐，人家都编什么 **helloworld** 之类的，好歹也算句英文啊，你怎么编个只会喊哇塞的呢。

虽然 **Rubbish1** 号能干的事情不多，不过主人还是很满意，于是又拿来 **Rubbish1** 号的图纸

改起来。10 分钟后，又把图纸交给包工队，包工队的哥儿几个凑在一起拿过图纸来看了看，点点头，立马开始施工，瞬间，Rubbish2 号诞生了！然后主人让我叫醒 Rubbish2 号，然后我走过去叫他，只见 Rubbish2 号立刻飞身跳进内存，跑进内存后大喊一声:“Wa Sai~Sai~Sai~Sai~Sai~Sai~Sai~”然后，跑回去继续睡觉。主人成功的用 for 循环创造了一个结巴，唉～

自从 Rubbish38 号过分淘气的把狐狸妹妹的记事本搞坏了之后，主人就不怎么搞设计了。人家狐狸妹妹工作的时候有许多东西要记录的，比如网页用什么字 体显示啦，主人喜欢去那些网站啦之类的东西，狐狸妹妹都会写成文件存放在自己的那个目录里。那天，那个 Rubbish38 号一进工作间就上窜下跳，整的大 家都不得安生。一上来就要创建文件，你说你建就建吧，临时文件往/tmp 里建，有用的文件你自己建个目录起个明白点的名字建，都行。他非要把临时文件往狐 狸妹妹的那个目录里建，删的时候还顺手把狐狸妹妹的文件也给删了，闹得狐狸跟失忆了似的。主人再打开她的时候，模样也变了——因为不记得主人喜欢什么样子 的了。主人说：去我最常去的那个网站。她眨眨眼问：哪啊？把主人气的说不出话来。Rubbish38 号还老申请空间——就是申请内存呐，一会管我要 8k，一会又管我要 1M。可您申请了，我给你了，你倒是好好用啊。这家伙好像健忘，这次用完了之后就忘了，下次再用的时候又申请新的。我们这工作间里面空间的申请是有很严格的规定的。一个程序如果要想使用工作间里的空间，要向我提出申请，我根据工作间里的情况告诉他，哪块哪块归你，然后这个程序就去用去了。那块地方就不许别的程序访问了，这都是有严格的界限的。等到这个程序用完了这块空间，他应该跟我说一声，说我用完了，这块地方可以再给别的程序用了。这个过程就叫释放。一个有知识有道德有理想的程序，在他回硬盘睡觉以前应该释放掉所有他申请过的空间的。可是那个 38 号就不管这套，只管申请，从不释放，整的工作间里到处都是他申请的空间。好在我们这工作间足够大，他也不会长时间运行，否则非出事不可。这可是内存泄漏啊，在我们软件界，内存泄漏是和瓦斯泄漏同样严重的事

故。工作间规章制度第三条明确写着——禁止申请不释放！就在第四条禁止抽烟的上面。（当然不能抽烟，内存都冒烟了机器还能用么？）好在我是个先进的系统，一个程序退出之后，我会根据他的申请记录查看他有没有申请了没释放的空间， 如果有的话就强制释放掉——你都睡觉去了，你申请的空间肯定用不着了吧。

主人不搞创作了，包工队的哥几个也就闲了。包工队主要成员有 gcc，cpp，as，ld 四个人，其中 gcc 是老大，其他几个干什么活都得听他调遣。主人一般也只跟 gcc 打交道，当写好了图纸——也就是源代码，比如叫 test.c 吧，写好了之后就直接把图纸交给 gcc 去处理就好了，gcc 会去调动其他人进行各种处理。

一般来说，gcc 拿到图纸后，会首先叫来 cpp 进行预处理。预处理主要就是将文件里的宏定义进行展开。什么是宏定义呢？主人一般都比较懒，或者说，他们人 类能力有限，不愿意写好多重复的，类似的东西，就把这些都定义成宏。比如，这么写#define TOTAL_NUMBER 18353226 就是定义总数为一千八百三十五万三千二百二十六，那么以后再要用这个总数的时候，就直接写 TOTAL_NUMBER 就好了，不用写那一大串数字。而且，如 果总数变了，只要在最初#define 的位置修改一次就可以，反正就是为了偷懒。那么 cpp 的任务就是把这类的宏定义都替换回去，把所有的 TOTAL_NUMBER 都替换成 18353226，否则他们老大 gcc 看不懂，老大看不懂，那就没法继续往下干了，因为经过 cpp 预处理之后的文件就要 交给 gcc 去编译了。

编译又是怎么个意思呢？最初的图纸，也就是没有经过预处理的源代码，是人写的，一般懂相关语言（比如 C 语言）的人都能看懂。预处理之后的文件，虽然不那么 直观了（TOTAL_NUMBER 看着是不是比 18353226 直观？光写个 18353226 还以为是谁的 QQ 号呢），但终究只是做了下替换，还是人类可 以看懂的。这样的代码经过 gcc 的编译之后，就不是普通人类可以看懂的源代码了，而是只有终极牛人才能读懂的汇编代码。汇编代码就比较贴近底层的机器码 了，里面描述的都是一些基本的操作。打个比方吧，就比如描述切菜的过程，用 c 语言描述出来就像是“将黄瓜切片”，这么一句就搞定了。要是用汇编，那就是： 左手扶住黄瓜，右手拿起刀，移动刀到黄瓜顶部，刀落下，到抬起，刀向黄瓜后部移动 4 毫米，刀落下，刀抬起，放下刀，走出厨房，走进卧室，找到创可贴，贴在 左手食指上…… 好吧，总之，汇编是一种面向机器的，

很复杂的程序设计语言。gcc 的任务就是把 c 语言的源代码转换成贴近机器语言的汇编代码，为下一步 as 的工作做好准 备。

as 拿到汇编代码后，对这样的代码再进行处理，得到真正的机器码，这个过程，也叫汇编。汇编之前的汇编代码是终极牛人能看的，那么机器码压根就不是人看 的。汇编程序中至少还有些操作的助记符，比如什么 add 啊，mov 啊之类的。寄存器也是有名字的，比如叫 A，叫 R1 之类的。但是到了机器码，这些都没有 了，这些都换成了各种各样的数字，一句人话都没有了。还说且黄瓜的事，要是用机器码来描述，那就相当于说：用 32 号设备扶住 87 号物体，24 号设备拿起 126 号物体，移动 126 号物体到 87 号物体顶部，做 2635 号动作，再做 2636 号动作……

好了，现在终于得到机器码了，机器码按说就是可以执行的代码了，但是，这时候的程序还是不能直接执行的，为什么？因为还有 ld 没有出场呢，他的工作叫：连 接。光是一段机器码扔给机器去执行，机器照样摸不着头脑。而且，很多时候，一个程序不是一段机器码，而是由很多段机器码组成的，这些机器码分别存成很多 的.o 文件，这时候就需要 ld 出场了。ld 负责把这些机器码组装起来，并且写明了各段代码的地址，从哪里开始执行之类的。就像我们造个机器人，脑袋啦，胳 膊啦，大腿啦之类的都做好了，ld 就是负责组装的。就算只有一段机器码，也就是只有一个.o 文件，也要由 ld 进行一下处理，闹明白哪是头哪是尾，才能开始 运行。

说的这么热闹，其实包工队的工作过程对主人来说并不关心，他只跟 gcc 打交道，只管把源码交给他就好了，gcc 会领导小弟们干活，最终回馈给主人一个可执 行的二进制文件。中间过程中的那些个文件，主人压根也看不到。包工队的同志们，都紧密的团结在以 gcc 为核心的组织周围，坚持编译四步原则，坚持代码开 放，为把 Linux 建成为软件丰富，运行稳定，老少皆宜，人人必备的操作系统而努力奋斗。

不过，包工队毕竟只是个包工队，你要是盖个小厨房，垒个猪圈啥的，直接找他们盖就没问题了。你要是想建个 CBD 商圈，里边什么银行啊，商场啊，写字楼啊， 炸油条的啊，卖臭豆腐的啊，修理自行车的……等等一应俱全。这么大的一个工程，你光叫个包工队来就搞不定了。得有人进行合理的统筹规划，设计施工方案，然 后再让包工队去具体施工。这个规划的人就是——make

make 也是一个程序，像上边说的一样，他就是负责控制整个施工过程的（也就是编译过程啊）。对于比较小的程序，像主人的 rubbish 系列，也就一两 个.c 文件，那根本用不着 make 出马，直接 gcc 包工队去编译就行了，因为源文件的结构关系不是很复杂。可是要稍大一点的程序，像狐狸妹妹啊，皮筋老弟 啊，星爷啊，基本上所有常用的软件吧，都足够复杂到需要 make 来对编译过程进行管理。当软件大了，编译的时候就不能是简单的把一大堆.c 的源文件统统一 次性编译成一个二进制文件那么简单的事情了。那么做的话很费时费力，比如说，有一个软件，源码由 20 个.c 文件组成，分别是 1.c，2，c，3.c……20.c。这 20 个文件一股脑都交由 gcc 包工队，他们就会把这些文件都打开来，拼在一起，一次性的编译成一个叫做 big 的二进制文件。这时候发现了一些问题，需要修改 3.c 文件，修改之后得重新编译啊，那么 gcc 包工队又得把这 20 个文件全都打开，拼在一起，再从头到尾编 译一次。而其实只有 3.c 文件修改了，完全不必这么兴师动众。那应该怎么做呢？一般的都是把这 20 个文件分别编译成.o 文件，比如编译成 1.o，2.o，3.o……20.o，这样 20 个.o 文件，然后再由 ld 把这些.o 文件拼在一起，成为一个叫做 big 的二进制可执行文件。那么当要修改 3.c 的时候，只需要让 gcc 包工队重新将 3.c 编译为 3.o，再让 ld 重新连接一遍就好了，省去了很多时间。而 这个过程，如果让主人自己管理的话，会很麻烦，毕竟他们人类的大脑也不是那么靠谱的，搞着搞着就乱了。于是，make 义无反顾的挑起了这个重要的担子。当 然 make 也不能靠凭空的想象就来指导包工队干活，什么事情

总得有个规划不是。**make** 也需要一份施工的规划书，这分规划书就是 **Makefile**。

Makefile，顾名思义，就是 **make** 用的 **file**。这就相当于一份施工的规划，上面写着整个工程分为几个模块，先用哪几个文件编译成一个什么什 么**.o**，再用哪几个文件编译出一个**.o**，再怎么怎么一连接，最后得到编译好的二进制程序。**make** 就根据这份文件来指导 **gcc** 他们进行施工。当有某个**.c** 文件被后改之后，**make** 能够根据文件的修改时间智能的判断出哪些模块需要重新编译，重新连接，然后就去让 **gcc** 重新编译那些改过的文件，最终生成新的二 进制程序。有了 **make** 和 **Makefile**，就省去了主人敲一大堆编译命令的烦恼，只要敲一个 **make**，其他的，就交给 **make** 去做吧，他办事，你放心。

曾经有人建议写写 Richard Stallman，毕竟是个开源界重量级的人物，于是，咱们开讲吧。

Richard Stallman，1953 年出生在美国纽约，他从一出生就……没什么特别；他上小学的时候……反正我不认识他；等到他上初中的时候……也还没我呢。总之，他在生命的前十几年中并没有表现出什么过人的地方，因为他没遇到一个叫做电脑的东西。

高中的一个暑假，他去给 IBM 打工，花了两周的时间用 Fortran 语言编了一个数据处理的程序。这是他第一次接触计算机，或许就是这次相遇，确定了他未 来行走的方向。后来，1971 年，他考上了哈佛大学，听说这学校不错，怎么也得是个区重点吧。上学的同时，他还受聘于麻省理工学院的人工智能实验室，成为 了一名职业黑客（黑客这个词没有贬义，欲知详情请牵着你的狐狸妹妹去找她的狗狗哥）。也不知道他哪来的那么多时间，可能也是把毛概和邓论都翘了吧。在人工 智能实验室的期间，他可没少干活，开发了很多有用的软件，其中最著名的就是 Emacs 编辑器。Emacs 是一个可与 vi 相抗衡的强大的编辑器，他们俩的操 作方式完全不同，但却同样强大，各自用自己独有的方式，提高这人们的编辑效率。直到今天，让然总有人争论到底 emacs 好还是 vi 好，信奉 emacs 的人 和信奉 vi 的人形成了两个帮派，这俩帮派经常在大街上用板砖菜刀拼个你死我活。不过还好我这里只有 vi，否则工作间里不会消停了。哦，扯远了，咱还回来说 Stallman。

那时候的 Stallman 在人工智能实验室里工作的非常 Happy，大家有 BUG 同挡，有代码共享。因为最初的计算机就像我们的算盘一样，只是一个硬件， 没有软件的概念。后来随着电子管、晶体管的发明，计算机的电子成分才超越了机械成分，逐步演化成了现在的电子计算机，在这个过程中，出现了软件，并起到越 来越重要的作用，最终成为了计算机的灵魂。而最初的计算机软件没有什么开源不开源，自由不自由的概念，因为那时候软件天生就是自由的！那时候卖计算机的同 时会附带软件，包括软件的源代码和文档。用户可以根据自己的需要去进行修改软

件，与别人分享软件，总之，软件是用户花钱买来的，用户想怎么玩就怎么玩。然 而随着技术的发展，软件逐渐脱离硬件成为一个独立的产业，很多软件慢慢的只提供二进制代码而不提供源码了，这就意味着你不能修改它，并且多数还规定最终用 户没有二次分发的权利。也就是说，这东西你买了，只能你用，你再给别人，不行！有这样一件事，Stallman 他们实验室买的第一台打印机附带有驱动程序 的源代码，他们那的黑客们可以随意修改这个驱动，根据自己的需要添加些小功能啊，改改 bug 啊，之类的，这为他们的工作带来了很大的方便。后来， 实验室又买了一台激光打印机，这次厂商只提供了二进制的打印机驱动程序，它是实验室里仅有的一个没有源代码的软件。出于工作的需要，Richard Stallman 想修改一下这个驱动程序，但是不行啊，没源码啊。后来 Richard Stallman 听说卡内基.梅隆大学有这个打印机的驱动程序源代码，他就去了那里，对他们说:“那啥，大家都是道上混的，谁还没个马高蹬短的时候？是兄 弟的拉哥们一把，我也没啥事儿，就是我们那打印机老丢字，一遇到什么敏感的字眼就给打成口口，我估么着是驱动的问题，挺说你们这有着驱动的源码，能不能给 我拷一份？”对方办事效率还是挺高的，很干脆的拒绝了他。因为他们和厂商签署了一份保密协议，协议要求他们不能向别人拷贝源代码。顿时 Richard Stallman 感到他们背叛了自由的计算机社团，他非常生气，但是他选择了沉默。这只是一件小事，只是一个时代的缩影。那个时代，正处软件向私有化转变 的过程，越来越多的软件选择了不开放源代码，不允许二次分发的发布方式。甚至 Stallman 身边的同志们也都一个一个都跑到那些靠卖私有软件挣钱的公司 去打工了。而 Stallman 依然沉默

不在沉默中爆发，就在沉默中灭亡。

Stallman 爆发了!

他不能容忍软件世界里清新自由的空气被私有软件污染的乌烟瘴气；他不能容忍被剥夺按照自己的需求修改软件的权利和乐趣；他不能容忍自己买条皮带尺寸不够，他竟然连自己在上面多

打个洞的权利都没有！

于是，他爆发了。

他要重现当年那人人为我，我为人人的合作互助的软件世界；他要把使用、复制、研究、修改、分发软件的权利还给每一个软件世界的人民；他要用自己的行动告诉 人们，软件天生就该是自由的！他要开辟一个新的世界，哪怕是一个人在战斗！于是，一个宏伟的计划在他心中产生——GNU 计划。它的目标是创建一套完全自由 的操作系统，因为操作系统是电脑中最重要的最基础的软件，要创造自由的软件世界，自然先要有一套自由的操作系统，然后再以此系统为中心，开发各种各样自由 的软件。Richard Stallman 最早是在 net.unix-wizards 新闻组上公布了 GNU 计划，那是 1983 年的事情。既然要做操作系统，首先得有个明确的规划和目标，目标是什么？这个操 作系统要做成什么样子？这当然是要向最成功的操作系统学习，哪个？UNIX！GNU 计划中的操作系统，将是一个类 Unix 的操作系统。这个系统要使用与 Unix 相同的接口标准，这样，就可以由不同的人，分期分批的创作操作系统的不同部分而不必担心相互之间协同工作的问题。

为了实施 GNU 计划，1985 年，Stallman 又创建了自由软件基金会。基金会的主要工作就是执行 GNU 计划，开发更多的自由软件。1989 年，Stallman 与基金会的一群律师们起草了广为使用的《GNU 通用公共协议证书》也就是 GPL 协议，以此协议来保证 GNU 计划中所有软件的自由性。 到了 1990 年，GNU 计划中的这个系统已经初具规模，有了很多的优秀的软件。其中有很多是世界各地的黑客们无偿提供的，也有部分是利用自由软件基金会的 基金雇佣程序员来开发的，当然，Stallman 自己也是身先士卒，开发了 Emacs，Gcc，gdb 等重要软件。当他看着这些丰富的自由软件的时候，感 觉到那清新自由的空气，终于又回来了，以后，人们就可以拥有一个可以自由使用，自由修改，自由分发的，自由的操作系统！不过等一下，好像还差点什

么，哦， 还……差个内核吧……

作为一个系统，没有内核是不行的，这么重要的部件 Stallman 当然不会忘记，所以才会有 Hurd 内核。但是这个内核的表现一直不尽如人意，这让 Stallman 很焦急，外围的软件都好了，就差个内核啊，什么都有，就差内核！而转过年，1991 年，大家应该知道发生了什么，Linus 同学写出了 Linux，这我们之前说过。Linux 现在虽然被大家当作一个操作系统的名称，然而其实这并不准确。准确的说，Linux 只是一个内核，Linus 同学 只是写了一个内核。

什么都有，就差个内核！

什么都不是，只是一个内核！

还有什么需要多说的么？

Linux 顺理成章的代替 Hurd 成为了 GNU 计划中那个自由系统的内核。而这个系统，也被叫做 GNU/Linux 系统。Stallman 理想中的自由世 界，终于拉开了那沉重的幕布，展现出了自由的光彩。而 Stallman 并不满足，也确实没有满足的理由，这个自由的世界还需要成长，还需要更加丰富多彩， 还需要有更多的人走进这个世界中来。于是 Stallman 奔走于世界各地，告诉人们有这么一个自由的世界，号召人们加入这个世界，鼓励人们为这个世界更加 自由而付出自己的力量。他是一个执着的苦行僧，为了他的梦想，为了他的自由世界，他会一直走下去……

为了能够创作出更好的 Rubbish 系列程序，主人决定好好充电了。他下了个 pdf 版的书来看，好象是关于 c 语言编程的。看 pdf 这事儿，得找 Evince 来。Evince 是个文档查看器，比人家 Adobe 官方的 pdf 阅读器小巧很多，用起来也很方便。而且每次主人看完一个文档，点关闭的时候， 他都会很有心得记录下主人看到的页数，下次再打开同一个文档时，他就直接替主人翻到上次那页。这个很贴心的举动然主人很满意。这次也是，主人一点击那个文 档，Evince 就赶快去查自己的记录，一看，哦，这个文档看到了 380 页，"进程"这一章，赶快翻到。

有人可能对进程这个名字还不是很明白，什么是进程呢？简单地说，进程就是正在干活的软件。比如 Evince，躺在硬盘里睡觉的时候他就是一个软件，一堆数 据，一陀代码。当他被叫醒，跑进内存里开始干活的时候，他就是一个进程了。（当然，其实这么说不很准确）换句话说，内存里忙忙碌碌的，都是一个个的进程。 当然，同时他们都是程序，都是软件，这个不冲突。就像去公司上班的人，他们都是人（废话，见过大马哈鱼上班么），当他们在公司工作的时候，他们都是公司的 员工。员工，就像进程一样。很多公司的员工每个人都有个工号，什么 NB001，SB999 之类的，每个进程也都有一个唯一的标识——进程 ID 号，简称 PID。这个 ID 号是由我分配给每一个跑进工作间的进程的，分配的规则很简单，每人一个，每次加一。第一个跑进来的就是 1 号，上面介绍过，Init 这家伙 每次都是第一个被我叫起来，帮我打理一下日常工作，所以他的 ID 号总是 1。而且，他还有个特殊身份，这个呢，咱暂时保密，待会再说。

每个公司的员工都有个直属的上级，上级又有上级，以此类推。我们这里的进程也是这样，只不过我们不叫”上级“或者“上司”，我们叫——爹！好吧，似乎这个 称谓土了点，但是意思就是这个意思。一个进程之所以成为一个进程，一定是由于另一个进程创建了他。（有点绕嘴吧）比如说主人来了一个终端，于是就有了一个 bash 进程，然后主人在这个终端里敲入 firefox 然后回车，bash 就知道这是要他去叫狐狸妹妹来干活，于是 bash 就去找狐狸妹妹，把她带到内存 里开始工作，于是就创建了一个 firefox 进程。好了，现在，firefox 这个进程是由 bash 这个进程创建的，那么，bash 这个进程就是 firefox 这个进程的父进程，firefox 进程就是 bash 进程的子进程，也就是说，狐狸妹妹就得管 bash 叫爹！那 bash 也得有个“爹”吧？是 的，如果是在 Gnome 环境下开的那个终端的话，那么 bash 它爹就是调用 bash 的 gnome-terminal。那么如此循环往复，肯定有一个站在金 字塔最高点的总"爹"吧？难道，难道笨兔兔你就是他们的总爹？很遗憾，我不是，所有进程的总爹，是每次启动第一个被我叫起来的 Init，所有的进程都是被 init 直接或者间接创建的，Init 的特殊身份就是所有进程的祖宗！

关于父进程，有两点要说明：

第一，我们这的父子关系不是固定的，是会变换的。如果从 bash 启动 firefox 那 bash 就是 firefox 的爹，如果直接从图形界面启动那就没 bash 什么事情了。（这时候 firefox 的爹其实是 init）

第二，不要问我哪里有妈进程！

当爹也有当爹的义务，人家不能白叫你一声爹是不是。 当自己的娃（也就是子进程啦）做完自己该做的工作以后，就停止了一切动作，像个死尸一样待在那里，当爹的就负责给他“收尸”-_-b 一个结束了所有工作的进程，会处于一种“僵尸”状态，这时候他什么也不做了，就等着被干掉。进程进入僵尸状态前一般会通知他爹一声，汇报一下说：爹啊，俺 已经把该做的都做啦，

现在我要变僵尸啦 !(让后平举双手开始行走？那是生化危机 !) 然后他爹负责向我汇报：我家娃干完活了，你把他的工号（就是 PID，记 得吧）取消掉然后让他回去睡觉吧。然后我就把它的工号收回，然后看看他有没有什么申请了没释放的资源（一般一个好孩子在结束运行成为僵尸之前会主动释放掉 自己申请的资源的。），确认都没问题了之后，他就被从我的进程列表中清除了。但是有时候也会有些特殊情况，比如有的时候娃还在兢兢业业的干活呢，结果他爹 死了。（可能他爹干完活退出了,也可能被主人用命令 kill 了。）这个时候我就会发个信号给他家娃说:那个……娃呀，那啥，跟你说个事，你爹死了。这时候 有的娃就悲痛欲绝：俺爹都死了俺活着还有啥意思啊，呜呜呜 ~ ~ ~ 俺也僵尸吧。然后就退出了。比如你在终端运行 firefox，然后把终端关 了，firefox 也就退出了。也有的娃比较坚强，一定要完成上级交给的任务，化悲痛为力量，这时候我会给他找个新爹——因为每个进程总得有个父进程，没 爹是不行的。一般我会安排他爹的爹来当他的爹（又绕进入了吧），也就是这个进程原来的"爷爷"进程来当他的父进程。然后这娃在长了一辈后，继续认真工作。 比如你在终端运行 nohup firefox，然后把终端关了，firefox 继续运行。那如果他爷爷不幸也挂了呢？那就继续往上导吧，我们说了 Init 是所有进程的祖宗，所以他那里 成了最终的“无依靠青年进程收容所”。

还有的时候娃已经把该做的事情做完了，汇报给他爹并变成僵尸。可是他爹还没来得及给自己娃收尸，自己就挂掉了。这个时候，我没法通知那娃说她爹挂了，因为 那娃已经是僵尸了，啥也不听啥也不干了。可我也不能直接把他干掉，啥事情都得按规矩来嘛，只有他爹向我申请我才能把他干掉，可是他爹又已经挂了……那怎么 办呢？那就按流程来，先给这个娃找个爹，哪怕这娃已经是僵尸了，也得有个爹。比如我找到 init 说：那个 ID 号是 2725 的那个进程爹死了，你当他爹吧。 一边说一边看也不看的用手往那边一指，假装自己没看到那娃已经成僵尸了。一般 Init 也不会太注意，直接就答应了，然后马上发现了事情的真相，跑到我这里来说：那娃已

经成了僵尸啦，你还叫我收养个啥？我肯定会一脸无辜装：啊？ 是啊，那不管怎样，你是他爹了，你负责处理一下后事吧。于是 **init** 只好以爹的身份处理那个僵尸的后事，问题就这样解决了。

别看说的这么麻烦，其实一个天真烂漫的娃进程从变成僵尸到被干掉只是一瞬间的事情，所以一般情况下主人是看不到一个僵尸进程的，要不然这一屋子僵尸还不得把主人吓出点毛病来。一般主人用 ps 命令查看到的进程，和办公室里的员工差不多，基本都处于两种——干活和睡觉。

干活的状态，学名叫 Running，也叫运行状态。这个应该很好理解，就是说明这个进程正在干活嘛。但是有个问题，还记得我说过 CPU 是有限的吧，一台电 脑就那么几个 CPU（对软件来说，多核 CPU 跟多个 CPU 差不多），可是要用 CPU 工作的软件有很多。那么这个处于进程的干活状态又可以分为两种：1.正 在使用 CPU 干活。 2.排队等待使用 CPU 干活。当然，处在这两种状态的进程我都算他正在工作。这就好象你在公司要打印文件，结果打印机卡纸了。你在那等着人家修打印机的这 段时间不能算旷工吧。我可不是啥口口的老板，所以，正在使用 CPU 干活的，和积极的排队等待使用 CPU 干活的进程，都算正在干活的进程。

然后再说睡觉的状态。估计如果你上班的时候在办公室里睡觉，你们老板会很不高兴的。但是，在我这里，没问题！很多程序都会经常进入睡觉状态。这里说明一 下，这个睡觉状态可不是说回硬盘睡觉啊，为了区别我们这样说吧，我们管完全执行完毕退出内存只存在于硬盘的程序叫“下班回家”的程序吧。只不过这个家就是 硬盘上那块地儿，而回家后唯一的活动就是睡觉。好，现在我们要说的不是下班回家，而是在办公室睡觉——也就是在内存中的进程，进入睡觉状态，也叫 sleep 状态，休眠状态。那为什么一个进程在内存里不好好干活，要去睡觉呢？不是因为他昨晚上爬起来偷菜来着，也不是因为他熬夜看球，而是因为他要等待 某个事情发生。比如皮筋老弟，每次他运行起来之后，主人看看有没有 mm 在线，没啥值得聊的就直接把皮筋最小化了。那么这个时候如果没有人给主人发消息的 话，皮筋就没什么事情干，所以就没必要让他跟着排队等 CPU 了，等着了也没事情干嘛。所以这个时候皮筋就来向我汇报说：头儿啊，我歇会去啊，等网口那边有 发给我的包了你再叫我。然后他就去睡觉去了，而我负责看着网口有

没有发给他的包，如果有的话就叫醒他，那时候他就变回工作的状态，开始处理包的内容了。

睡觉状态也分成两种，一种是叫的醒的，一种是叫不醒的。还说皮筋，他正在睡觉，等着网口的数据包，这时候主人发来命令，要把皮筋关了，这时候虽然皮筋等的 包没来，我也得去叫醒他说：别等了，你下班回家睡觉吧。然后皮筋点点头，收拾好自己的东西，变成僵尸，他的父进程（通常是 **init**）提出申请，我把它工号 注销，然后他回硬盘睡觉。这种是正常情况，这样的睡觉状态就是能够叫醒的。也有的进程很执着，还比如皮筋，正在睡觉等包，这时候我发现网线断了。这网线都 断了那肯定来不了包了吧，主人也明白这点，要把皮筋关了。这时候我过去说:“醒醒，别等了，下班回家睡觉去吧。”他不理我。我继续:“网线都端啦，等不来 啦！”他还是不理我。我只好:“快醒醒，快醒醒，回家啦！”还是没动静。“快起来看上帝啦～”依然没反应。“靠，出绝招了……这是谁的钱包啊？？！！”还 是睡觉，看来是无论如何也叫不醒了，除非他等的那个包出现。这就是叫不醒的睡觉状态。一般一个好的程序是不应该处在这样的状态的。

另外，进程还有个停止状态，一般都是调试的时候使用的。比如主任的 **Rubbish n** 号，跑进内存处于工作状态的时候，主人喊，停！**Rubbish** 马上一动不动，处于停止状态，这样便于主人检查这家伙的各个部件是否正常。

“……本 APT 有超级牛力～～～～～”

唉～这家伙又去招人了。 我问:“SCIM，刚才主人给超级牛力输入了什么？”

“报告头儿，是 Picasa”

“星爷，查查这啥意思。”

“这个吗……英法美德俄日意奥的语系里都没这个词。不过有一个长得比较像的。”

“什么？”

“Picasso ，毕加索。”

“哦……看来这哥们是个画画的……”

GIMP 不服道:“画画？有我还不够么？”

我只得双手平摊做无奈状:“Who knows……”

数分钟后，超级牛力归来，带来了一个穿的花花绿绿，很有艺术气息的家伙。我过去上看下看左看右看，怎么就看着不像我们这的人呢？于是我叫来了 file。 file 可不是一个普通的文件，而是一个程序，一个用于判断文件类型的程序。他可以根据文件的特征来判断一个文件是什么类型的文件，当然，也能判断可执行 的程序。他可不是跟据扩展名来判断，叫.jpg 的就是 jpg 文件，叫.txt 的就是文档文件，这种功能，连 Rubbish 都会。（在我们这里，主人创作的 Rubbish 系列已经俨然成了傻子的代名词。）file 的功能要强大的多，他是根据文件的内容来判断的。一般一个文件都会有个文件头，来说明这个文件的 类型。比如 JPEG 类型的图片文件，他的文件开头的两个字节肯定是 FFD8（16 进制），而 GIF 文件的文件头就是 4749463839，其实就是 GIF89 几个字的 ASCII 码。二进制程序也有类似的特征码，于是，我让 file 赶快去看看这个“毕加索”（就叫他毕加索吧，虽然还是差了几个字母）到 底是个什么程序。file 把毕加索上上下下的检查了一遍，得出结论——这是个 Windows 的 EXE 格式的程序。

“什么？Windows 的程序 !? 超级牛力啊，你别是走错了吧，怎么把 windows 的程序领来了？”超级牛力不急不慌的摇摇头:“本 APT 有超级牛力， 怎么会搞错呢，这个就是从源里找来的软件包。不过别急，本 APT 有超级牛力，这软件包可不是光毕加索一个，后面还有一个呢。”我这时才注意到，老毕后面还 站着一个家伙，这……这……这不就是红酒大师吗？？越来越乱了。仔细看看，咦，跟我们这里那个红酒大师长得很像，但还有些差别。没事，超级牛力哪里肯定有 这个软件的资料，让他查查吧。还没等我让他查呢，他已经向大家解释上了:“毕加索先生是 Windows 界成名的图片管理大师，他所在的公司，也就是狗狗哥 那公司，他们公司为了惠及 Linux 世界的人们，又为了偷懒，把毕大师配上一个翻译就直接推向了 Linux 界。”哦，原来这样，后面那个是改装过的，专门 负责给毕大师当翻译的红酒。为了区别，我们就叫他毕翻译吧。

毕大师和毕翻译安顿好之后，主人立刻把他们叫起来干活。俩人先后爬起来跑进内存，麻利的整理起主人的图片来——第一次启动嘛，得先对主人指定存放图片的那个目录扫描一下，做好整理和记录工作，这样才能心里有底，主人要看啥，立马能找着。经过了数秒之后，毕大师完成了对所有图片的扫描，主人觉得比原来负责管理照片的 f-spot 快了不少。这下，f-spot 可不爽了。

f-spot 是最初跟随我来到这个机器上的，也算是元老了。一直以来都是他负责管理主人的照片，也没出现什么问题。现在主人找来这么个功能差不多的家伙，这不是明摆着要抢 f-spot 的饭碗么。要是以后让这个 windows 的程序代替了，我们 Linux 程序的脸面还往哪搁？于是 f-spot 决定，为了荣誉，向毕加索挑战！只见 f-spot 跑到刚刚扫描完图片的毕大师面前说："大师果然好功夫，不亏是师出名门。这数千张图片，竟然这么快就整理好了。在下实在佩服的紧，不过不知大师其他本事怎么样，有道是遇高人不可交臂失之，在下想在大师面前讨教几招，不知，大师可肯赐教否？"只见毕大师的表情如平静的湖水般并没有因 f-spot 的挑战而激起一丝波澜，只是面容祥和的扭过头对翻译说："What did he say?" 靠……

F-Spot VS Picasa

要比就从起床开始比！f-spot 和毕加索以及毕翻译重新回到硬盘睡觉，然后我去叫来的 time 同志。time 是一个用于计时的命令，这个咱以后再说，先看比赛。随着我的一声号令下，time 开始计时。f-spot 蹦起来后牙也不刷，脸也不洗（废话，一个软件，有牙么？），迅速的从硬盘飞奔进内存。再看那边，毕翻译先迅速跑进了内存，然后再扭头去叫醒毕大师——因为毕大师听不懂我们的话，所以无论我们怎么喊都是叫不醒他的，只能先叫醒翻译，再由翻译去叫醒他。这样一来，时间自然慢了不少，对于起床速度，F-Spot 完美胜出。双方起床已毕，相向而立，只见 F-spot 掏出两张一模一样的照片，照片上是一个人像，似乎是晚上照的，眼睛如含着血泪般发出令人不寒而栗的红色。只见 F-spot 把一张照片扔给毕大师，另一张贴在自己这边，双掌运足力气，瞄准照片中人的双眼大喊一声：嗨！立时，照片上人的红眼不见，翻了白眼。另一边的毕大师微微一笑，拿起自己这边这张，单掌向前一推，一股掌风直逼那人双眼，只见掌风过后，那人双眼渐渐恢复成正常颜色。F-spot 不等毕大师打完那掌，有拿起照片推拳运动，只见那本是夜里的照片亮如白昼。毕大师也不示弱，将照片抛向空中，双手一抖，一道劲风吹过，再看落

下来的照片时，也已经比原来明亮不少。F-Spot 又对照片连续发力，打出三招，依次改变了照片的对比度，色调和饱和度。毕大师口念咒语:“Easy……”只出一招，双手间出一道白气，就把照片的亮度，对比度，色调，饱和度，都改到合适的状态。毕翻译的在旁边解释道:“这招乃是毕大师的独门秘诀，叫做‘手气不错’!”毕大师微微点头，一扬手，只见那修改好的照片激射而出，直接从网口飞了出去，发布到了 PicasaWeb 网站上。屋内众人顿时为 F-Spot 捏一把汗，这 PicasaWeb 网站，明显是人家地盘啊，F-Spot 能搞定么？哪知道 F-Spot 不慌不忙，也照片往网口一扔，把照片同时发布到了 Flickr，PicasaWeb 等多个网上相片储存空间里。这真是：棋逢对手，将遇良才，欲知二人胜负如何，且听下回分解。

话说二人斗的正酣，忽然 bash 报告，从主人那里发来命令"shutdown -h now"，数秒钟后，一切归于沉寂……

要我说，这俩人都什么劲啊。每个软件都有它存在的意义，都有它的长处和不足。就说这毕加索吧，虽然比 f-spot 功能强点，不过毕竟不是原生的程序，至少占用内存就比 f-spot 大不少。毕竟毕加索不是一个人在干活，他必须有个毕翻译才行，所以占用量一下子就上去了。这内存可是重要的系统资源，跟 CPU 一样重要，所以作为软件，还是应该本着艰苦奋斗勤俭节约的精神，充分利用内存，避免浪费。不过我们 Linux 下的软件们基本是小巧的居多，这里的 4G 内存还真没被我们占满过。f-spot 也就占用 20 来 M 的内存，毕加索比他多，也只有 40 多 M。当然，并不是说两个软件就一定比一个软件占用的内存多，一个软件占用的内存空间分为很多部分，咱们慢慢说。

首先，这个软件本身得占用一定空间。就像你去公司上班，你自己得有个坐的地方吧。就算你不坐着，站的地方也得有一小块吧。总之，自身会占用一定的空间。软件本身是由一条一条的二进制代码组成的，咱以前不是说过 Rubbish 的故事么，gcc 包工队把主人用 C 语言描绘的图纸编译成了一堆二进制的代码，这堆代码就是 Rubbish。其他的软件也是一样，都是一堆代码，所以，软件程序自身占用的空间叫做代码段。这个代码段的大小在程序进入内存运行前就确定了，或者再往前想，在程序编译好之后就确定了。这个很明白吧，就像你在家睡觉的时候是一米七五，不可能到单位就变成一米六零了吧。

然后，软件会随身带一些静态的数据，一般是一些初始化了的全局变量，每次起床时这些数据都会被带到内存里来，而且每次的初始内容都一样。就像你每天上班都得带着手机啊，家里钥匙啊，老婆照片啊之类的。比如 Rubish 1 号每次都喊“Wa Sai ~ ,”这个字符串就是个数据，这个数据像是 Rubbish 每天随身带着一张纸条，起床来到内存后看看上面的内容然后喊出来。（当然，写程序的时候也完全可以把这内容写进代码段，那就相当于 Rubbish 1 号记住了这个字符串，不用看纸条，直接喊出来。）这种随身带着，每次都会用的数据所占用的内存叫做数据段。另外，软件可能还需要一片固定的空间来放东西。比如你的办公室，每次上班都毫无疑问的需要

一张桌子，你一进办公室就得准备好这桌子，要不你怎么办公啊。（虽然这桌子不是每天现打造的……）程序也是，有些空间是一定会用到的，一般是一些未初始化的全局变量，不一定存什么数据内容，这种空间叫做 **BSS** 段（可不是 **BBS** 啊），这个也是在程序编译完成之后就确定下来的。每个程序启动，我都会根据他有多胖来确定他需要的代码段有多大，然后根据他有多少随身物品来确定数据断有多大，最后，根据他身上写的 **BSS** 信息来决定给他分多大的空白空间供他使用。

以上说的都是程序一起床就需要分配的空间，除以之外，程序在工作的时候还会根据情况向我动态申请内存空间。这就是那种必须记得释放的内存空间了，他的名字就叫堆。这种空间，程序在刚启动的时候是不知道需要用多少的，得视具体情况而定。比如 **gedit** 小弟，主人要些个小文件，**gedit** 就申请一小块空间临时存放主人写的东西，等到主人越写越多，**gedit** 就会逐渐向我申请更多的空间，把主人写的东西都堆在那块空间中。（要不怎么叫堆呢）

最后，还有一种动态申请的空间，叫做栈。这种空间是让程序随手放一些临时的变量的。比如临时有个什么事儿，或者有个什么数据，要存起来，就跟我申请栈空间，临时存放一下。栈就像一个小圆筒，程序需要用的时候我才给他，寄存在这筒里的东西都是很快就要用到的，这个空间不用程序去释放，程序退出之后我直接把筒里的东西倒光，把筒收回。因为是个小筒，所以，最先放到里面的东西会被之后放进去的东西压住，必须把后放进去的东西拿出来之后才能拿到先放进去的东西，这叫先进后出，是栈的特点。

狐狸妹妹今天比较累，拖回来一个 40 多 m 的 deb 包。赶紧让超级牛力来打开看看——超级牛力除了可以自己去网上拽软件回来以外，也可以打开放在本地的软件包。超级牛力打开一看，是一个叫做 VirtualBox 的家伙，赶快检查他需要的各种东西，发现我们这里的环境都能满足他的工作需要了，然后安排住宿。

VirtualBox（咱以后就简称 VBOX 吧）很懂礼貌，说话有些怯生生的感觉。他跟其他人打国招呼后，来到我这，把一些内核模块需要放在我这里。安顿好一切后，就去睡觉去了。这家伙给我的印象还不错，我就跟狐狸妹妹聊起他的背景来。听狐狸妹妹说，他的身世挺悲惨的，他最初生在德国，生母是一个叫做 InnoTek 的公司。VBOX 一生下来就经常被 Vmware 和 VirtualPC 这样的大哥哥欺负，不过好在他自己的本领还算可以，后来他亲妈 innoTek 为了让他学习到更好的本领，把他的源代码依据 GPL 协议开放了，让全世界的高手们来教他本领，那是 2007 年 1 月的事情。凭借不错的性能，以及可以免费使用的特点，VBox 总算闯出了自己的一小块天地。不过好景不长，转过年来，亲妈 InnoTek 被卖给了红太阳公司，VBox 自然也被过寄过去。不过红太阳公司这个后妈还算不错，很照顾小 VBox 的成长，继续让他在开放的环境中健康的长大，有红太阳公司众多高手的支持和全世界热心用户和高手的用户，VBox 俨然已经成为 Linux 下同类软件的首选。开源的本质使得追求自由的人们放弃了 Vmware，简便的操作让人们淘汰了 qemu，夸平台的支持更是有点软公司的 VirtualPC 无法比拟的。VBox 本来以为自己之后的人生道路会走的很顺畅，可是，2009 年又一次波折打击了 VBox——红太阳这个后妈也被卖给人了。收购他们的是一个很古老的公司，那公司里好像写的都是甲骨文，不知道他们每天用象形文字怎么办公。甲骨文公司收购了红太阳之后，红太阳的几个孩子都面临着一段未知的命运。其中最让人担心的是 mysql，因为之前 mysql 一直跟甲骨文家亲生的 Oracle 打架，这一下 Oracle 成了 mysql 的后妈，还不得天天受欺负阿。我们的 Vbox 的处境或许稍好一些，毕竟甲骨文亲生的孩子里没有和 VBox 同样本领的，所以 VBox 在那里或许还不至于受谁欺负。不过那也毕竟是经历

的重大的变革，对孩子的成长还是会有一些影响。

说了这么多，忘了介绍 VBox 是干什么得了，他是一个虚拟机，就是能在一台电脑上虚拟出另外一台电脑来。怎么样，听起来这个本是很厉害吧。他第一次工作的时候，我们都看呆了。

“请问您这台电脑打算装什么操作系统呢？”

“WindowsXP 吧"

“哦，那我建议您用 192M 的内存，您看可以么。”

“上 512 吧”

“好的，那么您需要什么样的硬盘呢？”

“30G 的，IDE 吧”

“好，您的电脑以及创建好了，显存大小，3D 加速功能，声卡，网络这些都是可以随时调换的。”

别以为这段对话是在中关村攒电脑，这是 VBox 在指导主人创建一台虚拟机。创建好之后，主人启动了这台虚拟的电脑，然后 Vbox 就开始忙活了。他先向我申请了 512M 的内存（之前从来没有任何软件一次性跟我申请这么多内存），之后又去硬盘里打开了刚才创建虚拟机时候他创建的那个超级大的文件（30G），别的文件都是很小的一个小箱子，这个文件大的像一间房子一样了。最后，他打开了硬盘上的一个叫做 WindowsXP.iso 的文件，把里面的查皮放了出来……

查皮被放出来之后，跑进了 VBox 申请的那 512M 的内存空间中。也不知道 VBox 用了什么方法，查皮就乖乖的待在那 512m 里，其他的空间他好像都没看见一样，当然也看不见我们。对于查皮来说，他正在一台有 512M 内存，30G 硬盘，Intel E8400 CPU 的机器上运行。查皮在检查了这些硬件后骂了一句：靠！谁攒的机器，3G 的 CPU 尽然只有 512M 的内存！听的我们都想乐。之后查皮摆出了一张蓝脸，跟主人说：我这个系统可只能装在一台机器上阿，装多了算盗

版。还有阿，我要是挂了，弄坏了你的数据可别赖我阿，我不管。你同意不同意，同意就按 F8，不同意就别装了。主人按了 F8，然后查皮又检查起磁盘——也就是 VBox 创建的那个 30G 的大文件了。检查之后又问主人：你这个磁盘怎么分区阿？打算把我装哪阿？我觉得查皮也太不人性了，安装的时候也不让用户现体验体验，就像我们这样，可以直接从光盘启动，让用户先用用，用的爽了再装嘛。查皮这不管三七二十一上来就让装，而且还非得问用户怎么分区，其实用户很多都不知道分区是啥意思呢，我们安装的时候都会问问用户，看他会不会分区，会分的话可以手动分区，不会的话我们可以替他分，当然得先备份好数据。

主人分好区之后，查皮开始从光驱往硬盘复制东西——当然所谓的光驱和硬盘都是假的，都是 VBox 骗查皮的。复制的东西都是安装时候需要用的文件，复制完了之后，查皮说：那啥，我得重启一下电脑（当然是重启 VBox 弄出来的假电脑）。哎，这家伙真麻烦，装一般还得重启电脑，哪像我们，都装完了才重启呢，装驱动都不用非得重启电脑。VBox 赶紧模拟着这假电脑重启，还别说，上电自检，BIOS 界面啥的都模拟出来了，还真像那么回事。重启之后查皮继续安装，这回脸色好看了不少，不是那死蓝死蓝的了，有了不少艺术气息。一边安装，查皮一边向主人讲解这自己的功能，特点，有什么好处，反正就是一通侃阿。这点倒是挺好的，省得主人装的时候寂寞，其实以后我也可以学习学习。不过查皮装的时候就感不了别的了，我们安装的时候还能允许主人上上网阿，玩玩小游戏啥的打发时间。

这台机器的配置还是不错的，查皮虽然跑在 VBox 创建的虚拟机里面，但是仍然只花了 30 分钟就安装好了。装好了之后又重启了一下电脑，查皮终于正常启动了，我们一堆软件在外面围观，还不时的指指点点，终于有幸见到真正在工作的查皮了，有点到了动物园的感觉，不过查皮并不知道我们看他。启动之后现文主人一些基本问题，用户名阿，怎么联网阿之类的，尤其重要的还非得要上网注册一下，向那个有点软的公司汇报，让有点软公司查查是不是正版，如果不是的话，查皮就不正常工作了。不过主人没让他上网，所以暂时这个查皮没有激活。查皮是个很早

的系统了，第一次出生是 2002 年的事了，所以虽然只有 512M 的内存仍然跑的挺快，刷刷的，不过界面不如我漂亮，没有俺这样的 3D 桌面，所以快点也是应该的。

系统装好了之后，当然还得装驱动。查皮是被装在了虚拟机里，所以要装驱动可不能把主人买电脑时候的驱动盘拿进来，而是要装 VBox 虚拟出来的这台电脑的驱 动。这个驱动哪里有呢？当然是管 VBox 要咯。我们这的这个 VBox 是狐狸妹妹去官方下载的不开源版，如果是超级牛力从软件源里拉来的 VirtualBox-OSE 版的话，那个 VBox 可自己不带这驱动来，需要主人自己上网站上下载驱动去。而我们这个 VBox 是自己带着驱动来的，只需要 主人点下 VBox 的“设备”菜单的“安装增强功能”选项就好了。点了之后，VBox 从兜里掏出了一个 ISO 文件，悄悄塞到给查皮虚拟出的那个光驱里。狐狸 妹妹拉拉我说：“你看你看，VBox 给查皮喂东西吃呢。”这时候查皮发现光驱里有了个光盘，按照习惯，他先去光盘里检查有没有一个叫 做"autorun.inf"的文件。如果有这个文件，并且里面有类似 open=xxxxx.exe 这么一行，那查皮就直接再去光盘里找这个 xxxx.exe 文件，并且运行它。这就是光盘自动运行的原理，后来被很多病毒利用了。好，咱说回来。查皮看到光盘塞进来，赶快去检查，发现果然有 autorun.inf 文件，于是按照文件上写的，去运行 VBoxWindowsAdditions.exe 程序。这会狐狸妹妹又喊：“你们看你们看，查皮过去吃了 !”（还真到了动物园了-_-b）我说：“别傻了，他那是光盘自动运行功能。”大家点点头。查皮运行了增强功能安装程序，主人一路点着 next 就装好了，就像 我们这里装 deb 包一样方便。装好之后，自然是要重启一下了，重启后的查皮似乎性能更好了些，而且可以更方便的和我们交流了，主人的鼠标也可以很平滑的在 查皮与我之间切换了。

主人命令查皮打开了 IE，先去下了个叫做迅雷的软件。这是个下载软件，有点像我们这里的奔流，不过奔流只是用来下 bt 的，可是迅雷却是什么都能下，什么 http 阿，bt 阿，电驴阿，都行。

听起来这个迅雷好像很厉害，不过他也有些不厚道的地方。尤其是下 bt 的时候。像 ftp，h

ttp 这样的下载连接，原理相对简单，就是服务器这边一个包一个 包的发，客户端（也就是你的机器）一个包一个包的收而已。比如要下一个文件，就比如狐狸妹妹吧，她要从某个网站下个文件，就去跟对方那个系统说:“我想要 你的 xxxx 文件，给我吧。”对方看看，这个文件是可以给别人的，里面没有任何这个门那个门的照片，然后就跟狐狸说:“好的，准备收吧。”狐狸准备好之后（比如得问问主人这文件存哪吧），就跟对方说，好了，我准备好了，发吧。然后对方就一个包一个包的发过来，狐狸妹妹一个一个收下来，然后拼成一个整个的文件。这时候如果又来一个软件要从同样的地方下载同一个文件，就比如有一个 FlashGet 吧。服务器那边就得把数据包分别发给我这里的狐狸妹妹和那个不知道哪里的 flashget。打一个包发个狐狸，然后再打一个包发给 flashget，这个打包的过程不会慢，很快就完成了，但是网口的宽度是有限的，比如只能一次传一个包，那这样两个软件同时下载的话，速度就慢了一半。而 bt 下载是什么样子呢？咱拿奔流说吧，奔流要下载，首先得有个 bt 的种子文件。种子里写着去哪找下载的服务器，这个服务器可不一定是什么大网站了，可能只是某个和你一样在家上网的人。不管是谁吧，奔流就根据种子文件找到服务器，管他要数据，服务器那个系统上也得有个相应的软件，把数据打成包，一个一个的发给奔流。这时候如果又来一个人要下载同样的文件，服务器那边就跟奔流说了，那个奔流，有个 ip 是 x.x.x.x 的那边有个比特精灵也想要这个文件，你把你那已经接收了的文件给他传一份吧。奔流就会很友好的把自己已经下载好的那部分打成包，扔给那个不知道哪里的比特精灵。一边接收服务器发来的包，一边自己过回当服务器的瘾，把数据打包发给别人。有人问呢，这样不会慢么？打包拆包的过程对于奔流来说（对其他软件也是一样）是很轻松的，瞬间就能完成。那网口的带宽呢？网口数据的输出和输入是分开的，奔流给别人发数据这算输出，输出的带宽不管怎么占用不会影响输入，也就是不会影响服务器给奔流发数据的速度。这样，每个人都同时当服务器和客户端，在大家齐心合力的工作下，下载速度就有了明显的提升。从宏观上来说，原本 http 这样的下载方式，服务器的上传带宽有多大，决定了所有客户端下载的带宽有多大，来的人

越多就越慢。而 bt 这样的形式中，每个软件即是客户端，又是服务器。在自己下载的同时，也将自己的上传带宽贡献出来，让别人从自己这里下载，这就是人人为我，我为人人的世界阿。

这说了半天，还没说迅雷为什么不厚道呢。

咱说了，BT 下载的核心理念就是每个下载的人贡献出自己的上传带宽供其他下载的人使用，这样的结果就是将下载的星星之火传播为燎原之势。下载的人越多，速度就越快（不考虑你家接入带宽限制的话）。但是，这林子一大，就什么鸟都有了。迅雷加入了下载 bt 文件的功能，可他的行为很是自私，只享用别人的带宽而不共享自己的带宽，就是说只管从别人那里要东西，而当有人管他要的时候他却不给。这哪行啊，孔子说过，不能饱汉子不知饿汉子饥呀。所以迅雷被很多像奔流这样 的有理想有道德的 bt 软件所鄙视，甚至还有的软件专门有屏蔽吸血客户端的功能。

主人让 VBox 里面的 IE 下载了迅雷之后就开始安装了，只见主人双击了一下下载来的 Thunder.exe 文件，迅雷的安装程序就直接崩出来向主人问好： 您好，欢迎安装迅雷……等等等等问候语吧，然后主人点了下一步，安装程序又掏出一份协议来让主人签署，无外乎就是如果怎么怎么样，那不关我们迅雷的事，如果怎么怎么样，我们迅雷也不负责任之类的。主人只有同意了才能进行安装。主人很无奈的点了统一之后又想主人推销：那个，有个软件叫迅雷看看热播排行，要不 要装啊？还有狗狗影视排行，要不要装啊？要不要开机就自动运行迅雷啊？等等问题。再之后就问要安装到哪里，都选好了之后就开始安装了。我看的频频点头，问 VBox：查皮底下的软件都是这样么装的么？VBox 说：是的，基本都是.exe 的二进制程序直接运行，问一堆问题就装上了，如果你不知道该怎么答也没关 系，就直接点下一步，下一步，……完成。就行了。我感慨道：还真是挺方便的，就是还得自己去下载，比较麻烦，要是像我们这样，直接一个超级牛力就全搞定 了，只要告诉他软件名就行。不过他们那里的软件既然都是这样的安装，至少还是比较统一的，相比之下我们这里的软件，如果能够让超级牛里去请的还好办，要是 超级牛里请不来的，就麻烦了，得自己去下载不说，下载回来的软件格式各种各样，闹得很多人都不知道该怎么装。

下载软件最经常找到的，就是 tar.gz 格式的软件包了。我经常听到很多其他的笨兔兔抱怨他们的主人围着个 tar.gz 包不知道该怎么办，自己急的直打英 文字也没办法。还好我的主人了解的多一点，知道这样的包是怎么回事。其实事情是这样的：话说有个软件叫 tar，基本上每个 Linux 都会带着这么个软件， 我这里也是。这个软件是干什么的呢？是个打包裹的，不过他可不是邮递公司的那种，不过会把好的包野蛮的扔来扔去。他的能力有点像查皮那里的 winzip，他能把很多文件和目录收拾在一起，打成一个包裹，也就是生成一个 tar 包文件。可是跟 zip 不一样的是，tar 只管打包，不管压缩。原来那些零碎的小文件 有多大，打成 tar 包之后还是多大，只是变成一个整个的文件了而已。有人说，那我想压缩怎么办？别急，我这还有另一个软件，叫 gzip。这个软件就是专门 负责压缩的，但是他只能压缩一个文件，不能像 winzip 那样能压缩一个目录里的好多文件。这样，tar 和 gzip 就成黄金搭档了（有脑白金么？），要想 实现 winzip 那样的功能，就得 tar 和 gzip 联手协作。比如有个目录叫 aaaa，里面有好几十个文件，总共有 10M。想要压成 zip 那样的压缩包， 那就先让 tar 出手，把 aaaa 目录打成一个包裹文件——因为 gzip 只能压缩一个文件嘛。这样 tar 就把这个目录打成了 aaaa.tar 文件，这个文件 还是 10 M 大。然后由 gzip 出场，把这个文件压缩，压缩完了得标明一下啊，所以就又把文件名改了，叫做 aaaa.tar.gz，表示这个文件经过了 gzip 压缩，这时候这个文件就小了，可能 5M，可能 7 M 的就没准了。有时候觉得一个文件叫 xxx.tar.gz，有两个扩展名太罗嗦，就改名叫 xxx.tgz，是一个意思。这下就明白了吧，这个 tar.gz 包其实就相当于 rar 或者 zip 的压缩包。那下载来的 tar.gz 包的软件怎么装呢？那当 是先把包解开再看了，得先解开压缩包看看里面是什么内容才能知道怎么装啊，就像我问你 RAR 包怎么装，你能知道么？

我们现在知道 tar 包就是个压缩包，就是个大包裹，里面有什么东西不一定。那一般拿到一个 tar 包的软件应该怎么办呢？

你收到一个包裹后怎么办？当然是先打开啦！先找剪子啦，小刀啦之类的工具把包裹拆开，然后看看里面有什么东西，根据里面东西的不同来决定怎么处理。里面要是家里寄来的松子核桃什么的，就赶快吃了；要是比较难吃的松子核桃什么的，就跟同事分着吃了；要是部手机，就赶快拿出来试试；要是下面还有把手枪，就赶紧拿刚才那手机报警。这些大概不用我说，智力正常的人都应该知道怎么做，其实 tar 包也是如此。拿到一个 tar 包之后，先用你的工具把 tar 包拆开。工具是啥？有道是解铃还须系铃人，tar 打的包，当然还用 tar 来解了。当然，你也可以用那个叫做文档管理器的家伙，他的中文名字叫归档管理器，他的英文名字叫（叫 gui ~ dang ~ guan~li~qi~~？那是小沈阳！) file-roller。不过其实他只是个负责用图形界面和主人交流的家伙，真正干活的还得是 tar。tar 包解开后，一般会得到一个目录，里面有很多的文件。然后干什么呢？有的同学记起来了，看看里面的东西啊。

一般包里面应该有个 README 文件，里面写着这个软件是干什么用的，怎么安装，怎么用，作者是谁，干什么的，爱吃什么，身高多少，腰围裤长……等等信息吧。也可能安装的方法写在一个叫做 INSALL 的文件里，总之，应该有相应的文档文件来告诉你这个软件怎么装。不过也有时候软件的作者不厚道，或者忘性大，没有写 README 或者 INSTALL 文件，或者文件有，但是没说清楚到底怎么装，那怎么办呢？用自己的头脑判断一下吧。

一般来说 Linux 下软件分发无非两种形式：要么是编译好的二进制的，要么是源代码。咱以前不是讲过 gcc 的故事么，gcc 包工队听后 make 总指挥的调遣，make 总指挥根据 Makefile 的指导工作，Makefile 由 configure 分析师创建。那么，你看包里面有 configure，有 makefile 之类的那就是源代码呗，没有这些的话八成就是编译好的二进制文件了。要是二进制的包，那就好办了，直接就能运行。比如你下了个包叫 qq.tar.gz，解开了之后里面有个叫 qq，一看还可

执行，那还等什么？运行它就是了。要是源代码的包呢？按照咱之前讲的步骤来：先让 configure 分析师看看你这机器里能不能装这个软件，如果缺什么东西，他会告诉你，让你去准备。之后就是让 make 去指导施工，这个过程可能比较长，成功之后，这个软件就已经产生出来了，不过这时候编译好的二进制文件还在当前目录，还没有放在合适的地方。虽然可能也能运行，但是他看着其他的软件很 happy 的聚在/bin，/usr/bin，/usr/local/bin 之类的屋里，自己一个人躲在这个角落多伤心啊。所以还需要一步 make install，就是告诉 make，把这个刚刚编译好的软件请到他该去的地方。以上所说的这个过程，就是让很多人头疼不已的编译安装。

经过一系列丁玲铛啷的操作后，主人终于用迅雷下载了他需要的东西，然后关了迅雷，准备把下载好的文件复制出来。文件是放在 VBox 给查皮虚拟出来的那个磁 盘里的，虽然是虚拟的，不过理论上也是归虚拟机里的查皮管理的。所以要想拿文件到我的地盘上来（也就是拿到真实的磁盘上来），同样需要经过正规手续。

查皮们之间有一种很方便的通讯方式，叫做网上邻居。连接在同一个局域网内的几台装着查皮的机器可以互相看到彼此，就像住在一个大院里的邻居一样，互相之间 共享个啥文件啥的都很方便。就像你有封信要给对门张大爷送去，首先你看到张大爷家门口的一个信箱，然后你打电话给张大爷问：您家门口的信箱我能随便往里放 东西么？我有封信要往里放。张大爷说：行阿，放吧。然后你挂了电话，去张大爷家门口把信仍进他的信箱。（谁吃饱了撑的这么送信阿。不过就是个比方。）查皮 之间也是这样，查皮甲共享了某个文件，就相当于张大爷，查皮乙就可以通过网络往这个文件夹里面放东西，当然，放之前得跟查皮甲打好招呼，确认人家让放才 行。

查皮们通过网上邻居这样的方式共享文件时，也是需要说黑话，对暗语的，学名就叫协议，他们也是有一套协议的。有点软公司管这个协议叫做 CIFS 协议，不过 我们更喜欢叫桑巴(Samba)协议，因为我们这里也有人懂这种黑话，那就是桑巴大姐。不过跟这个虚拟的查皮共享文件，完全不需要桑巴大姐出马，VBox 一个人就全都搞定了。只见主人点了 VBox 的 "设备-->分配数据空间" 命令。然后 VBox 询问主人要分配哪个目录——这个目录可是我管理的真实的机器上的目录。主人直接选择了自己的家目录，并且告诉 VBox 允许虚拟机里面的 查皮往里写东西，然后 VBox 就领命去了，可是我们也没看出有什么事情发生。正疑惑间，主人又开始操作查皮了，主人用右键点了查皮的"我的电脑"，选择 了"映射网络驱动器"。之后让查皮浏览了一下网上邻居的整个网络，查皮赫然发现一个叫做"VirtualBox Shared Folders"的网络，里面有个叫做\\VBOXSVR\LanWoNiu 的共享文件夹。原来刚才 VBox 去虚拟出了个张大爷阿！查皮在主人的命令下， 挂载了这个虚拟出来的共享文件夹，然后准备往里放东西，放之前得跟张大爷打个招呼

阿，于是查皮拨通了虚拟张大爷的电话。“铃～～～”Vbox 接起了电话，可是没说话，竟然把电话递给了我！和着我成那“张大爷”了。

"喂，我是 WindowXP，计算机名 F_U_O_C。" 当然，这些都是 VBox 给我翻译过来的。

“我是张大爷……厄不是，我是 Ubuntu 8.04，HostName LanWoNiu-desktop，你好”

“原来是个 Linux 阿。”

“是的，我是个 Linux，有什么问题？”

“没问题，就是觉得你们整天板着脸，让用户天天输命令，很不人性。”

“那可能你对先在的 Linux 还不是太了解，再说，命令行很多时候比图形界面高效阿。你不是也有命令提示符？还有，你们家族的新人 Windows 2008 还尤其加强了命令行的能力。”

“命令怎么比图形高效了？请问你如何在命令行下一次删除文件名没有任何规律的多个文件？”

“这个你用图形界面也一样费劲阿。”

“行了行了，不跟你废话，说深了你也听不懂。我问你，你是不是有叫\\VBOXSVR\LanWoNiu 的共享文件夹？”

“是”

“我要往里放文件，可以么”

“可以”

“那我可放了阿，你硬盘地儿够不？网速撑的住不？我这可是百兆网卡。”

“没问题，发吧”

“恩，没问题就好，估计你们 Linux 这点能力还是有的。”

“……“（瀑布汗中）

“是不是忙的顾不上说话了？你们这效率也就这么回事，哈哈。”

哎，这家伙还真爱自以为是。当他自以为把我说的哑口无言的时候，却不知道他正被我的手下当作宠物一样关在笼子里，养在我的工作间中。

最近有些无聊，也没有什么新人来报道，不过也是，主人也不能天天装软件玩呀。一般平时用的着的软件都有了，就够了。现在主人每天也就是用狐狸上网转个圈 儿，用 **smplayer** 看看片儿，用 **OO** 老先生码码字儿，用 **Empathy** 跟 **mm** 聊聊天。这么几个简单的进程调度起来，对我这么个优秀的内核来说实在没什 么挑战。闲的没事我就去硬盘里翻文件玩，看主人目录里都有啥有意思的东西。翻着翻着，看见了主人和 **mm** 的聊天记录，拿来看看——这可不算侵犯个人隐私阿， 俺只是个操作系统。

第一段：

mm：你给俺装的这个系统怎么跟我在单位用的不一样？

懒蜗牛：你们单位那个是 **centos**，这个是 **Ubuntu**

mm：有什么不一样？

懒窝牛：都是 **Linux**，不同的发行版

mm：啥叫发行版？

懒蜗牛：就是……不同的牌子。

mm：哦……那这个好？

懒蜗牛：恩，用起来方便。

mm：这个名字怎么念阿？

懒蜗牛：就叫它笨突兔吧，哈哈，我就这么叫。

mm：可爱，嘻嘻，样子倒确实是挺漂亮的。

看完这段，我很欣慰，呵呵。

再看第二段：

mm：我想装个软件，怎么装阿？网上下载的都不能运行。

懒蜗牛： 这个跟 windows 装软件，不一样的，俺教你吧。

mm： 哦？好阿

懒蜗牛： 先教你用命令装，打开个终端

mm： ok

懒蜗牛： 找到了？还挺快

mm： 恩，让我放桌面上了

懒蜗牛： 运行 cowsay

mm： 尚未安装

懒蜗牛： 恩，恩

mm： 是否运行 sudo apt-get install cowsy

mm： 他给的提示

懒蜗牛： 8.04 太智能了……

mm： 呵呵

mm： cowsay 是什么东西？

懒蜗牛： cowsay 是个很有意思的软件，现在你还没装

mm： 哦

懒蜗牛： 当你想装一个软件的时候就运行

sudo apt-get install 软件名

mm： apt-get 什么意思

mm： sudo install 俺都理解

懒蜗牛： apt-get 是 ubuntu 下的软件包管理工具，号称超级牛力。

懒蜗牛： 好，现在运行 sudo apt-get install cowsay

mm： 哦？跟软件有关的操作都用他？

mm： ok

懒蜗牛： 安装，卸载软件，都用它

mm： e： 无效的操作 install cowsay

懒蜗牛： 怎么输入的命令？

懒蜗牛： 把整个输入输出发来看看？

mm： 又输入了一次，就成功了

懒蜗牛： ……

懒蜗牛： 好了，现在就装好 cowsay 了

mm： 这就装好啦。

懒蜗牛： 恩，运行 cowsay hello

懒蜗牛： 然后你就看见一个 cow 在 say hello，呵呵

mm： 哇！狗狗！

懒蜗牛： 是 cow……

mm： 哦，是牛阿，呵呵

懒蜗牛： 同时还会有一个 cowthink 命令

懒蜗牛： 也可以试试

mm： 那岂不是 cowsay +任何，他都说任何

懒蜗牛： 对对

mm： 呵呵，挺好玩的。

懒蜗牛： 好，今天俺们学习了安装软件

mm： 恩，恩

懒蜗牛： 主要分 2 步

懒蜗牛： 第一步，知道你要装的软件的名字

懒蜗牛： 第二步，sudo apt-get install xxxx

mm： 恩，恩，学会了，嘻嘻

懒蜗牛： ：D

mm： 老师教滴清楚哈

懒蜗牛： 呵呵

懒蜗牛： 这样装的软件就会不断的更新

懒蜗牛： 当然，前提是有牛人把新得版本放到服务器上

mm： 哦？还能自动更新？

mm： 恩，恩，这个当然

懒蜗牛： 恩，今天你 aptget 一个 cowsay1.0

mm： 俺以为自动更新还是需要设定的

懒蜗牛： 明天服务器上有 cowsay1.1 了

懒蜗牛： 就会提示你更新，就是开机右上角的那个

mm： 恩，这个貌似今天看到了

懒蜗牛： 恩，恩

mm： 嘻嘻

恩……不错，这 mm 有前途。

第三段：

mm：我发现还是你教我的用图形界面装软件方便，只要打开那个啥牛力的软件管理器，想装什么就搜名字就好了，然后右键点一下，选标记以便安装，就好了。

（看来是主人哪天现场指导去了，呵呵。）

懒蜗牛：恩，是阿。

mm：那干嘛还有人要用命令装阿。

懒蜗牛：其实命令熟的话，命令更快。而且命令有命令的好处阿，比如你电脑有啥问题，我与其告诉你先点哪个菜单，再点哪个菜单，再再选哪个选项，就不如直接告诉你一条命令，让你运行一下来的快。

mm：哦……是哈，那样适合我这样的懒人，嘻嘻。

懒蜗牛：呵呵

mm：今天同学来俺着看见这系统了，以为是 win7 呢，呵呵。

懒蜗牛：比 win7 可漂亮多了。

mm：恩，恩，俺也这么说。

懒蜗牛：握手，握手。

mm：嘻嘻，还有像这个这样漂亮的系统不？

懒蜗牛：这个只是 Linux 的一种，所有 Linux 都可以弄成这样。

mm：哦，那还有什么其他 Linux？

懒蜗牛：有好多，比如 Gentoo，Fedora，suse，还有你们公司的那个 centos，好多呢。

mm：哦，那有什么不一样？

懒蜗牛：gentoo 是个能高度配置的系统，可以根据自己的需要配置整个系统，速度最快，因为里面的软件都是按照自己的机器配置而优化的。不过就是配置起来很麻烦。

mm：你用过么？

懒蜗牛：以前用过，挺不错的，尤其可以滚动升级，不像笨兔这样，版本升级风险比较大。我还想哪天格了换回 Gentoo 呢。

（阿 !!? ？ 别阿别阿，我挺好用的，升级也不麻烦，相信我，相信我。）

mm：听起来是高手才用的，别的呢？

懒蜗牛：Fedora，是 redhat 公司出的免费版本。

mm：红帽子阿，听说过，好像有 redhat 9.0？

懒蜗牛：那是很老的版本了，redhat 在 9.0 之后分为两个分支，一个是面向企业的收费的，叫 Red Hat Enterprise Linux，另一个是社区的免费版本，就是 Fedora。

mm：这样来的阿

懒蜗牛：恩，现在 Red Hat Enterprise Linux 已经发展到版本 5 了，有人简单叫它 redhat5，所以很多第一次接触的人误解，觉得 redhat9.0 是最新的。其实都是老古董了。

Mm：我也差点……

懒蜗牛：呵呵，Fedora 里面的软件都比较新，发现了什么问题，redhat 公司再去改，稳定了之后就放进收费版的 Red Hat Enterprise Linux 里。

mm：合着拿免费用户做试验阿……

懒蜗牛：也可以这么说……不过 Fedora 的确实能满足很多人求新的愿望，而且，有大公司作靠山，作出来的东西还挺不错的，我也想装一下试试呢，从来没用过。

（不会又要把我格了吧……咱有 VBox 阿，在虚拟机里试试就得了 ~ ）

mm：装吧，我支持。

mm：我们公司那个 centos 呢？什么来头？

懒蜗牛：那个是 Redhat 收费以后，社区创建的版本。目的是提供一个免费的服务器版本的

Linux。centos 基本上可以当作是 Redhat 的免费版，版本号互相对应，里面的软件也一样。

mm：哦，那，那个 SUSE 呢？

懒蜗牛：那是德国的一个发行半，以界面漂亮著称，不过有点慢。

mm：哇……俺喜欢漂漂的东西。

懒蜗牛：呵呵，回头装一个你看看。

（她喜欢往她机器上装阿，别又来格我。）

mm：哦……嘻嘻，反正你就是闲不住，老折腾电脑玩

（恩，我看也是）

懒蜗牛：厄……这个……是吧。

（犹豫什么，就是！）

mm：嘻嘻，无语了。

赶紧告诉兄弟们，一定要好好工作，不然随时炒鱿鱼，哎～～～遇上这么个主人也不踏实……

继续翻聊天记录，第四段：

懒蜗牛：早上好

mm：好，我的电脑装显卡驱动了么？

懒蜗牛：你那是 Intel 的集成显卡，不用装。

（恩，恩，我就是这么方便，哈哈。）

mm：Intel 的显卡不装驱动就能用阿？可是在 windows 下怎么需要装驱动呢。

懒蜗牛：厄……任何硬件都需要驱动程序的，包括硬盘、光驱、U 盘也都需要驱动。只不过这些常用的东西的驱动都集成在了系统里面或者 BIOS 里。Windows XP 年头久了，当然里面没有你这个显卡的驱动，你装个 WIN7 看看，照样不用装驱动。都集成在系统里了。 mm：哦，

原来这样阿，那我还挺运气哈，碰上这么个显卡。

懒蜗牛：恩，Intel 对 Linux 的支持还是不错的，如果是其他牌子的显卡，就麻烦一点了，有的显卡根本没法开 3D 效果。

mm：那那些用别的显卡的人怎么装 Linux 阿？

懒蜗牛：装还是能装，不过就是不能开 3D，而且运行效率也差。

mm：那帮人真可怜……

懒蜗牛：是滴

mm：那，下一个问题，我想跟你视频用什么软件呢？这个 QQ 好像没这个功能阿。

（得，问的软肋上了。）

懒蜗牛：这个……可以用 Skype，MSN 也行，反正 QQ 是没戏了。

mm：为什么呀?

（这 mm 整个一十万个为什么。）

懒蜗牛：因为腾讯就只开发了这么个简单的 Linux 版的 QQ 呗。

（原来叫腾讯阿，我一直以为是疼殉呢……）

mm：好麻烦阿，还得用 MSN。和你用 MSN 也就凑合了，和别人咋办，人家要是没 MSN 呢。

懒蜗牛：这个……没辙。

mm：哦，你也没辙啦，好吧……对了，这系统下有啥游戏好玩阿？

懒蜗牛：自带的那些玩过没？

（恩，恩，没事跟电脑下下国际象棋）

mm：那些都好没意思阿，有大点的没？

懒蜗牛：魔兽世界？

mm：那个好复杂阿，不爱玩。

懒蜗牛：你想玩啥？

mm：比如跑跑卡丁车啦之类的。

（别说卡丁车，连卡车也没有阿）

懒蜗牛：没有……

（是吧）

mm：那有什么游戏呢？

懒蜗牛：基本上吧，在 Linux 下就别想玩游戏了。

mm：这么惨阿。

懒蜗牛：其实也有，有些网游。

mm：还有网游阿？好玩不?

懒蜗牛：没玩过，都是英文的……

mm：俺恨鸟语……

懒蜗牛：所以还是别玩游戏了，这系统就是让你好好学习嘛，呵呵

（换个话题吧）

mm：我们最近学 ps 呢

懒蜗牛：学点啥不好……

mm：这个也没有？

懒蜗牛：有 gimp，功能和 ps 差不多。

（就是就是）

mm：你给我装了么？

懒蜗牛：默认就有，你找找。

mm：哦，看见了。

懒蜗牛：嘿嘿

mm：怎么跟 ps 不一样阿？

（废话，一样了就是 PS 了）

懒蜗牛：当然不一样阿，这完全是两个软件。

mm：那跟我们老师教的都不一样，我也没法用阿。

懒蜗牛：厄……也……没辙这个。

mm：啥破系统阿，干啥都不行。

懒蜗牛：恩……那……你要不还是用 Windows 吧。

mm：可是这个好漂亮耶

懒蜗牛：漂亮也不当饭吃。

mm：也是，回头你给我格了装 Windows 吧。

（又一个可怜的兄弟要惨遭不幸了……）

随着 USB 门口上的红灯一闪，我知道又有 USB 设备接入进来了，赶快打看门一看，这回不是那个司空见惯的 1G U 盘，而是一个网络设备，好像……是个无线网卡？赶快翻翻我身上的模块，看有没有它的驱动。

我身上有很多的模块——别担心，不是“肿块”，不会影响身体健康。也不是“蘑块”，不能吃。是“模块”，翻译成英文叫 module。这些模块像一本本的手 册，有的手册是说明如何使用某个硬件的，这就是硬件驱动模块；有的是说明如何使用某种文件系统的，那就是文件系统模块，等等。这些手册我都统一放在 /lib/modules/2.6.28-11-generic/目录下（2.6.28-11-generic 是我的内核版本），每次起床，我都根据配置 文件里写的内容，把里面一些必要的手册揣在身上再去干活。（就跟去旅游要揣个地图一样的道理）当需要用到哪个东西的时候就掏出响应的手册来查看。比如要用 RealTek 的那块声卡了，我就把关于 RealTek 声卡的那本手册（也就是那个模块啦）掏出来，看应该怎么使用，如何操作这个声卡。也有的东西，很重 要，很关键，很基本，每次一定都会用到，那样的就不做成模块了，就直接让我记忆在脑子里，融化在血液中——也就是所谓的编译进内核。哪些东西编译为模块， 哪些东西编译进内核，哪些东西根本不编译，这是在编译内核的时候就决定的。你也可以让我把所有东西都记忆在脑子里，也就是所有的东西都编译进内核，不编译 成模块。但那样的话，就基本没法干活了。倒不是我记不住那么些东西，我不是人脑，我想记住啥就记住啥，但是要知道我是程序，我要记住个东西的话，体积是要 增大的。一个所有东西都被编译进去的内核大约要二百多 M 那么大！！这就意味着这内核一启动，自己就至少得占 200 多 M 的内存，那还怎么干活啊，这点地儿全让他一人占了。

不过说起来，我的祖先们——也就是最初的那些 Linux 内核，是没有模块这回事的。那时候的 Linux 内核要把所有需要用的东西都记住。比如要用到 ext2 文件系统，那就把 ext2 文件系统的支持编译进内核。用不到 XFS 系统，那对 XFS 系统的支持就不编译。等到那天需要 XFS 支持了，就得重新编译内核，把 XFS 支持编译进去，然后重启，用新内核启动系统。所以那时候

的 Linux 内核是个典型的宏内核。所谓宏内核，也叫单内核，就是指像 Linux 这样，内核整体作为一个独立的进程在运行在内存里，所有该实现的功能，都在这个大进程里实现，像进程管理阿，内存分配阿，文件系统管理阿，硬件设备的控制阿等等这些事情。像我们 Linux，还有传统的 Unix，有点软公司的剁死，Windows 95,Windows98，都是宏内核。与宏内核对立的，还有一种叫微内核。微内核就不是一个人在战斗了，微内核的理念与宏内核相反，把内核该干的那点事分成一个一个的小块，由一个个小的内核进程专门去管理。有专门管理内存分配的，有专门管理进程的，有专门管理硬件 IO 的，等等。这样的好处就是进程间分工明细，每个进程只专心管理自己那一点事情，不容易出问题。而且，可移植性也比较高，只需要把直接跟硬件相关的部分移植一下就好了，其他的部分基本不用动。宏内核就需要整个都移植，因为是一个整体嘛，要换整个换。像咱们说过的 Minix，就是微内核。当宏内核工作的时候，就是像我一样：比如叫皮筋起床干活吧，我先通过文件访问，把皮筋叫进内存（程序也是文件阿，可执行文件），然后给皮筋分配好内存空间，为他创建个进程（也就是给他分配个工号），分给他 CPU 让皮筋开始干活，皮筋要访问网络的时候我负责操作网卡，把他要发的东西发到网卡上。这一系列的事情，全都由我一人管理。整个工作间里是以我为中心的工作。而微内核工作起来的景象就是：要内存的事都去找内核贾；要访问文件的程序，都去找内核余；跟硬件打交道的全去内核汤那；进程管理的问题都归内核顿管。内核余把皮筋从硬盘里交出来，然后喊“老贾，给皮筋分配点内存”，内核贾就给分配，分配好了跟内核顿说：“分个工号，创建个进程”。内核顿照做，然后皮筋开始干活，要访问网卡了，就去内核汤那报道。整个工作间里，软件们是以“顿贾余汤”内核小组为中心干活。

宏内核灵活性明显不高，这是个人就能看出来，所以现在我们 Linux 学会了通过加载模块的方式来增加灵活性，需要增加什么支持，只要加载一个新的模块就好了，不用重新编译内核，不用重启计算机。其实这也算是跟微内核那里学来的了。呀，说了半天主人接进来的这个网卡……好像我这里没有它的驱动模块阿……

我看了一下插进来的这个网卡，是 Realtek 的 RTL8180L 芯片，再仔细翻翻我的所有模块……确实没有，坏了，这回恐怕要在主人面前丢脸了。主人用 ifconfig 查看网卡，我只好汇报：现在机器上有两个网卡，一个是有线网卡 eth0，这个正常工作，另一个是虚拟的回还网卡 lo，这个也没啥问题。（闭口那不提无线网卡的事）主人好像很纳闷，心说我这明摆着多插了一个无线网卡阿，你怎么就装看不见呢？他叫来狐狸妹妹，让她去问狗狗哥这 TPLink 的 WN210 网卡怎么用。狐狸妹妹找到一个叫做 Ubuntu 中文论坛的地方，里面也有人问怎么用这网卡。听得我这叫一个着急，你找也找 RTL8180L 这芯片阿，关键是这个芯片的型号，不是那网卡的型号，搜芯片会多不少记录呢。哦，对了，可能他压根不知道这网卡是啥芯片。那你倒是问我阿，问一句 lspci 我不就告诉你了么，哎，我也是，皇上不急太监急。

主人的悟性还是挺高的，一会就想起来问我了，我赶紧告诉他网卡型号，他就去查去了，得出的结果是——就是没有 Linux 驱动 !那这网卡就算没法用了？当然不是，虽然没有 Linux 驱动，但是，困难压不到我们 Linux 软件，随着狐狸妹妹的点拨，主人知道了有一个软件，叫做 ndiswrapper。这个软件会干啥？他能读懂硬件的驱动——读驱动本来是我的工作，就是那些驱动模块阿，但是人家读的是硬件的 windows 驱动，翻译成我能懂的 Linux 模块，然后就可以使用这个卡了。不过他只能翻译些网卡驱动，不过这也差不多够了。反正这块卡是能支持。

超级牛力瞬间拉来了 ndiswrapper，安顿好后 ndiswrapper 立刻被叫起来干活，主人给他指了指那个 xxxx.inf 的 windows 驱动文件，ndiswrapper 赶紧拿起来读，详细研究了一下后表示，可以支持，只要加载好他给我建造出来的和他同名的模块 ndiswrapper.ko 就可以了。主人按照狐狸妹妹找到的文档一步一步操作：先用 ndiswrapper 加载那个 windows 驱动，然后在让我加载那个 ndiswrapper 模块，最后问我，现在这个网卡状态是什么样阿？我充满信心的回答：现在机器上有三个网卡，一个是有线网卡 eth0，这个正常工作，另一个是虚拟的回还网卡 lo，这个也没啥问题。还有一个无线网卡 wlan0，也正常工作。主人很欣慰的点点头，一股成功感油

然而生。不过这无线网卡跟有线的不一样，有线的插上之后，配好 IP 就能用，这无线的得先建立好无线连接，这无线连接建立好就好比有线网卡插好了网线。建立连接也不是什么困难的事情，我们这里有专门的团队负责。图形界面的有 NetworkManager，跟网络有关的设置，甭管无线的有线的，找他就行，跟查皮底下差不多。如果用命令的话有 iwconfig 可以查看和配置无线网络，还有 iwlist 可以查看周围可用的无线网络，可能会找到邻居家的没设密码的信号哦 ~

主人的无线网卡没有搜索到安全意识薄弱的无线邻居，当然，人家压根也没打算搜到，只是简单的链接到了自己家的无线路由上。不过估计这网卡不会常在我这插着，您想啊，我这是台式机，千兆的有线网用的好好的，没事用的什么无线啊，我又不移动。估计是主人给那个笔记本电脑用的，先插在我这里试试，研究一下能不能驱动上，然后再往本本上插。

这个电脑硬件阿，不像电视机电冰箱似的，买来插上就能用。硬件要想在计算机上工作，得需要会操作它的软件，这个事情，一般就是归我们操作系统管了。But 子曰:“人非生而知之者，孰能无惑？”这时候 OO 老先生推了推大眼镜咳嗽了一下说:“那不是子曰的，是韩愈曰的”好吧，不管是谁说的，反正道理是这个道理，我们操作系统，也不可能生来就会操作所有的硬件，就像你不是生来就会开飞机一样，得学，得考本，得移库、倒库、坡起、限制门。OO 老先生再次严厉的咳嗽了一下:“你见过飞机过限制门么？！那是汽车。”反正，我们要想会操作一个硬件，也需要学习，这就需要驱动程序，任何硬件要想工作都是需要驱动程序的。这时候可能有人提出反对意见了:“硬盘，光驱，这些也都是硬件哪听说过要装驱动程序的？还有我的 U 盘，摄像头，也都是插上就能用，不用装驱动阿。”不用装驱动，不代表不需要驱动。硬盘光驱是最基本的存储设备，在我们操作系统起床工作之前，硬盘就要工作（因为我们都住在硬盘里嘛，必须硬盘工作，我们才能被读进内存里。）这个时候其实是另一个软件——BIOS 在操作硬盘工作，硬盘的驱动，就在 BIOS 里。关于 BIOS 这老人家，我们以后细说。反正硬盘光驱这样的基本设备的驱动很简单，也统一，任何一家生产的硬盘都是一样的用法，所以硬盘光驱的驱动就被集成在了 BIOS 和操作系统里面，不用额外安装。其他所谓不用装驱动的设备也一样，都是因为驱动集成在了系统里。比如查皮他们家以前的瘟酒吧系统，就不认识 U 盘，需要装驱动才行。到查皮这一代，就不用装了，集成了驱动就像一本给操作系统看的使用手册，上面写明了如何如何操作这个硬件，写哪个寄存器就把数据发出去了，从哪个寄存器读就把数据读回来了，往哪个寄存器写个什么什么数据就自爆了等等。（这是什么硬件阿……）就像买来电视机，里面的使用手册一样。针对不同的

操作系统，需要有不同版本的驱动程序。您想阿，我和查皮说的话都不一样，他那边的程序都没法直接和我交流，还得通过红酒大师，

那我们能看得懂的手册，自然也就不一样。你家电视机的说明书不也有中文版，英文版，韩文版，非洲土著语版么。但是，并不是每个硬件厂家都给每个系统制作一份驱动的，毕竟厂商人力财力有限。电视机也不是每个都有非洲土著语版的说明书嘛。（压根就没有把……）所以，一般硬件厂商会优先开发市场占有率最高的那个系统的驱动程序，哪个系统？目前来说，就是查皮和他的后代喂死他、温妻了。而我们 **Linux** 就经常遇到一些无法使用的硬件，很多人还抱怨我们无能，冤枉阿 ~ ~ ~

随着 VBox 挥舞着魔杖一顿乱比划，又一个虚拟的电脑出现了，这回要住进去的不是查皮，而是一个我的同行，一个 Linux。

前一阵子狐狸妹妹拖回来了一个 iso 文件，是一个叫做 Slax 的 Linux 系统。Slax 是一个闲不住的家伙，专门喜好移动办公，他很适合被装在光盘或者 U 盘上，被放到各种各样不同的机器上运行，不像我们，基本上只在一台机器里干活。Slax 是基于 Slackware 发行版的，说起 Slackware 那可是历史悠久了。最早的 Slackware 1.0 版在 1993 年 7 月 16 日就发布了，创始人是 Patrick Volkerding 老大。那年代交通还不发达，不像我们现在有光盘坐，甚至还有宽敞明亮，容量高达好几 G 的豪华 DVD，那时候 Slackware 的发布只是用 3.5 寸软盘。到现在已经有十好几年了。Slackware 的宗旨是力图成为“UNIX 风格”的 Linux 发行版本。它的方针是只吸收稳定版本的应用程序，而且力图让系统结构简单，尤其他的包管理方式，没有我们这里的超级牛力，也不用帽子店那里的大晕头（yun），而是直接用 tar+gzip 打的包，软件包的扩展名都是.tgz，安装一个软件包就往根目录一解，齐活。Slax 之所以基于他，估计就是因为他的简单吧，毕竟要在 U 盘或者光盘上实现一套完整的系统，而且还要集成进去尽可能多的驱动，空间大小是个问题。

VBox 创建好了虚拟机，主人立刻挂上那个 Slax 的 iso 文件，让 Slax 展露他的技艺。Slax 身体轻巧，带着他那帮身体同样轻巧的弟兄们，迅速跑进虚拟机的内存，开始工作。有道是麻雀虽小，五脏俱全，别看人家只有 200 多 M 的 ISO 文件，带的软件不必我当初带来的软件少。什么文字处理的，网络聊天的，看网页的，画图的，算数的，小游戏等等，甚至连咱之前说过的 ndiswrapper 都有，想的很周到，听说还有一些 Slax 的版本，连红酒大师都自带。主人在虚拟机里分别试了试那些软件，觉得还不错，然后就把虚拟机关了。我估计，他是要叫来刻录软件把这 iso 刻成光盘来用，可是等了一会，没见动静，却看到 USB 门上那盏灯亮了。我赶紧去打看一看，还是那个熟悉的 1G 的 U 盘，哦……主人是想把这个系统装在 U 盘上。

可是，要往 U 盘上装得有软件啊，毕竟这是个 iso 文件，要刻光盘那是现成，要写 U 盘就不那么简单了，比如我知道有个软件叫 UNetbootin 就是能够把可以启动电脑的 iso 文件刻录到 U 盘上，成为系统 U 盘。可是我们这也没有这软件，主人怎么能把 iso 放到 U 盘上呢？总不能把文件直接解压到 U 盘上就完事吧，那可启动不了，启动信息可不在那些文件里。我这还没想明白，只见主人他，他，他还真就把 iso 里的文件完全彻底的往 U 盘上一解。我的主人啊～你没犯过这么弱智的错误啊～这也太天真了吧，这能启动电脑？咦？别急，主人又动作了，他让我们去运行刚刚解压到 U 盘上的一个脚本，u 盘里 boot 目录下的 bootinst.sh。哦～和着这个 Slax 连安装到 U 盘的软件都给您备好了，真是不错。仔细看看，还有个 bootinst.bat，这个是在 windows 下执行用的，这样不论在什么系统下，都能很轻松的制作 Slax 的启动 U 盘了。bootinst.sh 那家伙开始工作，问了主人一些简单的问题后，就在 U 盘上写下了引导信息，这样，一张启动 U 盘就做好了。

做好这个 U 盘之后，我立刻接到了主人的下一条命令——重启……耶？重，重启？？ !!

昏暗的控制室内，光线正在逐渐变得亮起来。一个声音响起:“U 艇动力恢复，能源压力，3.0V,能源流量，70mA”

坐在正中的家伙显然被这声音惊醒，赶快端正的坐在自己的位置，然后叫醒其他人。

“船员各就各位！各就各位！准备进入工作状态。”

其他人显然训练有素，立即投入到各自的工作。之前的那家伙发话:“准备开始硬件检测”

“是！硬件检测开始。”

“报告舰长，内存测试正常！”

“报告舰长，PCI 总线测试正常！”

“报告，DMA 通路正常！”

“ACPI 功能正常！”

“串行端口检测正常！”

“USB 端口检测正常！”

“IDE 硬盘工作正常！”

“网络适配器工作正常！”

“未检测到磁盘阵列模组”

“未检测到即插即用设备”

“输入设备工作正常！”

“其他硬体检测正常”

“加载模组！”

“镭射光碟支持模组加载完毕！”

“原生文件系统支持模组加载完毕！”

“视窗文件系统支持模组加载完毕！”

“USB 设备支持模组加载完毕！”

……

“全部模组加载完毕。”

“连接部件”

“核心部件连接完毕，部件号 base/001-core.lzm”

“图形部件连接完毕，部件号 base/002-xorg.lzm”

“桌面部件连接完毕，部件号 base/003-desktop.lzm”

“办公部件连接完毕，部件号 base/004-koffice.lzm”

“所有部件连接完毕！”

“创建动态工作主目录”

“内存环境正常，开始创建主目录……主目录创建完毕。工作目录转移准备就绪。”

“目录转移开始！”

“目录转移，倒计时，3，2，1，转移！”

嗖～一阵白光过后，仍然是狭小的控制室，窗外的风景却不再相同。

“目录转移完成”

“准备启动图形界面”

“图形界面启动准备就绪”

“启动！”

一片光明……

随着屏幕上出现那熟悉的 KDE 界面，SLAX 成功的开始运行。他们居无定所，他们游走于各

个电脑之间，他们帮助人们解决各种系统的各种问题，当系统不能够正常工作的时候，他们为人们提供帮助。当人们身在异地的时候，他们跟在身边，提供那些熟悉的操作界面。而像这次这么简单的任务更多——备份系统。

“报告舰长，发现 Linux 分区 3 个，容量分别为 128M,20G,220G。视窗分区三个，分别为 50G，100G，100G”

“装载分区！”

“是，装载开始！”

“第一分区装载完毕”

“第二分区装载完毕”

“第三分区装载完毕”

“报告舰长，128M,20G,220G 三个分区装载完毕，依次装载到/media/boot,/media/root,/media/home”

“报告舰长，收到使用者指令，备份 20G 的 Linux 分区。指令码：tar -czvf /media/home/root.tar.gz /media/root/*”

“tar,gzip 准备工作！准备好了么？”

Tar:“准备就绪”

Gzip:“时刻准备着！”

“Tar，开始打包，Gzip 跟在后面，开始压缩，动作快！”

两人迅速跳进硬盘里把指定分区的数据一个一个捞出来，打好包，再进行压缩。数分钟后，压缩完成。

“报告，压缩结束!”

“报告见长，收到使用者下一指令，备份 128M 的 Linux 分区。指令码：tar -czvf /media/home/boot.tar.gz /media/boot/*”

如法炮制……

“报告舰长，收到使用者指令”

“念”

“shutdown -r now”

“好，全体关机关门关灯开始休息，睡前记得把电脑重启了”

当屋里的灯光再次亮起，G 大叔再次出现在我床头的时候，已经是半个小时以后的事情了。我照例起床，检查屋子里各个分区是否完好。想起刚才主人要重启的命令，恩，上次是正常关闭系统，因该没什么问题。咦？怎么 home 分区多了两个文件？一个是 root.tar.gz，一个是 boot.tar.gz。再联想起刚才主人在虚拟机试用 Slax……哦，刚才主人备份系统去了。

我们 Linux 系统备份起来很简单，因为我们 Linux 和所有类 Unix 系统都信奉一句话“一切都是浮云～”哦，不对，应该是“一切都是文件！”没有什么隐藏的东西，不需要硬盘镜像，不需要靠那小鬼（ghost）头备份硬盘。Linux 下所有的一切都是文件，只要把这些文件保存起来，就等于备份了系统。当然，文件有很多，尤其有好多小文件，全都直接拷贝走不大现实，所以还是需要打包和压缩的软件，也就是 tar 和 gzip 来出马把这些文件打包。再有一点呢，系统运行的时候有很多目录里面有虚拟的，映射在内存里的文件，这些文件其实不存在于硬盘上，都是浮云，是不需要打包进去的（比如/proc 目录下的所有东西），所以最好在被备份的系统不运行的情况下，用另外一个系统进行打包工作，就像刚刚主人用 Slax 备份我一样。当需要恢复系统的时候怎么办呢？如果 mbr 里面的引导信息没有被破坏的话，只要再把那些打好的包结开，覆盖到系统的目录中就好了。如果 mbr 被改变了（比如重装的查皮），那就稍微麻烦点，需要在文件覆盖之后运行 grub 命令修复一下就可以。

有人质疑了，说人家查皮屋里不也都是文件么，你怎么个一切都是文件呢？我来举个例子，在查皮那里，如果有个软件想要从串口发送点数据，那就得知道怎么跟查皮用黑话说这件事。比如得说：“老大，我要使用那个异步串行通讯端子来进行一些数据推送动作”，这查皮才知道你要干什么，然后帮你把要发的数据从串口发送出去。回头又来个软件要写硬盘，就得跟查皮说：“老大，我要向那个磁性原理基础随机访问存储设备写入一些数据”注意，不同的设备可能有不同的操作方式，刚才那个叫数据推送，这个就叫写入了，搞错了可不行。而我这里就简单的多，对于程序来说，串口（第 0 个），就是/dev/ttyS0 文件，第一块硬盘就是/dev/sda 文件。操作他们就像操作普通的文件一样，只要跟我说：“头儿，我要打开 xxxx 文件，往里写 xxx 数据”就可以像用文本编辑器打开 txt 文件一样简单的使用物理设备，而剩下的实际操作不同物理设备的事情由我来做。（我学会那么多驱动，揣着那么多模块不是没用的。）一个 ISO 文件和一个真正的光驱挂载起来区别不是很大（就差个-o loop），所有的东西都以文件的形式呈现在主人面前，而隐藏在其

背后的实际物理设备的差别主人不必关心，都是浮云。

既然上面说到了设备文件，那就顺便再来说说我们这里的其他文件类型。

文件，前面说过，就像放在硬盘空间里的一个一个大大小小的箱子。箱子上面写着这个文件的名字，箱子里面的内容是千奇百怪，是什么都有可能。有的打开箱子一看，里面是一幅画，说明这个是图片文件；有的打开是篇论文，这就是 OO 老先生的文档文件；有的里面也看不出是个啥，但是星爷能拿来看，还能看懂，那就是星爷用的字典文件；甚至有的打开箱子一看，哟～里面睡着一个查皮，那就 VBox 的虚拟磁盘文件；要是打开文件一看里边是老醋花生，再打开一个一看是花生老醋，再打开一个就一盘花生，再打开第四个，一盘子醋——那这是个音频文件，里面是郭德纲的相声。另外各种程序本身也是文件，向狐狸妹妹啊，皮筋弟弟啊，他们在硬盘里躺着就是一个文件，一个可执行的二进制文件。上面这些都是普通的文件，跟查皮那边的文件差不多，除此之外，我们这里还有很多查皮那里没有的特殊文件。

有一种文件，打开箱子一看，里面是个奇怪的装置，就像你们人类用的打印机和扫描仪的合体，可以向里面输入数据，也可以从里面读出数据。这种文件就是设备文件。设备文件有两种——块设备和字符设备。字符设备操作起来比较简单，上面就俩键，一个读一个写。按一下读，就读出一个字节来，再按一下再出一个。写就相反，你写好一个字节放进去，按一下写，就写上了。块设备就复杂些，它有地址的概念。你要读，得先设置好地址来说明你要从哪里开始读，读多少个字符，然后才能读出来。写也一样，也得说明白了往哪写，写多少，然后再写。就是这两种设备文件，代表了接在电脑上的几乎所有的设备。像鼠标就是个典型的字符设备，而且没啥可写的，光读就行了，读出来都是“左键”“右键”“左键”“右键”“上滚轮”“下移动”……之类的。硬盘就不一样，是个块设备，设置好了地址然后读写。不能上来就读，那么多数据呢，读哪啊？从头读一遍？读完了都 2012 了。这些个设备文件就这样联系着计算机中的各种设备，软件想访问硬件设备了，就去操作这些设备文件。

还有一种文件，打开箱子一看，里面是一个纸条，相面写着：预知真实内容，参见 xxxx 文件。然后你就得再去找那个 xxxx 文件，打开，里面的东西才是你真正要找的。这种里面放纸条的很得瑟的文件叫做软连接文件（也叫符号链接），有点像查皮里的快捷方式。既然有软连接，当然还会有一种硬链接。这种硬链接看上去比较神奇。假设有两个文件 A 和 B，互为硬链接，这俩文件开始都是空的。打开文件 A，往里面放一个苹果，然后关上，再去看文件 B，咦？那个苹果跑到了文件 B 里。把文件 B 里的苹果拿出来咬一口再放回去，然后再看文件 A——哇塞 ~ A 里的苹果也被咬了一口，俨然 AB 里面的是同一个苹果。这是什么？是空间重叠？是大卫科波菲尔？都不是，只是硬链接而已。这个在你们人类世界似乎很神奇，在我们这可是司空见惯了。硬链接跟软连接不同，软连接有一个实体文件，一个链接文件。那里实体文件里有实实在在的内容，链接文件里就一个纸条。只有纸条的那个文件是那个有实际内容的文件的软连接。而硬链接的两个文件里都有内容，而且都是同一个内容。（注意，不是同样的内容复制两份，而是同样的一份内容。）两个文件互为对方的硬链接。对于软连接，删除了那个实体文件后，链接文件也就是实效了。里面纸条上还是写着见 xxx 文件，可是这 xxx 文件已经没了，去哪见去啊。硬链接就不一样，删除其中任何一个，都不影响另外一个文件。两个文件都是有内容的，因此，谁也不是谁的“硬链接文件”，根本也没有硬链接文件这么种文件，只是两个互相硬链接了的普通文件而已。

再有就是管道文件和 socket 文件，这两种文件有些类似，都是用于程序之间传递数据用的。怎么传递呢？俩程序商量好了，比如程序 A 和 B 吧，商量好了用 yyy 管道文件来通信，那么 A 程序把要说的事情写在纸上，放进那个管道文件里面，过一会 B 程序就过来，打开这个管道文件，看里面那张纸条的内容。如果仅仅是这样，那普通文件也可以做到，那管道文件有什么不同呢？不同点就是当 B 看完之后，那纸条就自动销毁了！

Vbox 这两天又忙活起来了，造了一个新的虚拟机，给一个新来的系统住。这回不是查皮，也不是查皮家的其他系统了，而是一个 Linux 发行版，叫做 Fedora。这个系统出身豪门世家，有深厚的背景，他们家干的是大买卖——卖帽子的。您可别小瞧这帽子，人家那帽子，那可是面料考究，做工精良，包退包换，值得珍藏。远了不敢说，在 Linux 的这一亩三分地上哪个不知道这大红帽子铺的。之所以叫大红帽子铺，是因为一开始他们卖的都是红色的帽子——Redhat。他们这帽子阿有个特点，卖的不是帽子，是寂寞，哦，不对，卖的不是帽子，是服务。你想要帽子的话，就直接拿走就行了，不用给钱，不过你要是拿回去不会戴，或者不知道该怎么搭配衣服之类的，那就得花钱请那帽子铺的人来参谋了。他们的帽子款式发展大致上有九个阶段，从 Redhat1.0 一直做到 RedHat9.0，小的变化就更多了。后来呢，觉得市场应该细分一下，大秃脑袋的老大爷和束辫子的小姑娘需要的帽子是不一样的。所以就把帽子种类分成了两种，一种依旧是红色的帽子，是针对大客户的，这种帽子是需要花钱买的，当然，买的依旧不是帽子，当然也不是寂寞，是服务。另外一种是软呢帽子，这种帽子是给普通家庭用户准备的。这种帽子不卖，白送，想要就拿走，如果戴着有哪里不舒服，做得不好的地方，就提意见给他们帽子铺，他们根据免费客户反应来的意见来制造更高品质的付费用户的帽子，这也是目前 Linux 村里常见的商业模式。今天主人要在 VBox 里面装的就是这个免费的软呢帽。

那有人问，这个 Linux 和你有什么不同呢？其实吧，这个 Linux 们之间的区别都不大，主要区别就是默认安装的软件不一样，默认刚装好的样子不一样，还有各个发行版所使用的软件管理器不同，其他没什么本质区别。那什么叫软件管理器呢？就是超级牛力那样的家伙啦。Fedora 那里也有一个类似超级牛力的角色，叫做 yum，我们叫他大晕头（其实是 Yellow dog Updater, Modified 的简称啦，Yellow dog 是一个 Linux 发行版，yum 最初是在这个发行版上用的。），不过人家可一点也不晕，干起活来跟超级牛力一样厉害。他俩的工作内容也差不多，都是主人要什么软件，他们就上网，到软件源里面找去，找到之后拖回来，安排好住宿，并且解决软件之间

的依赖关系。所不同的只是大晕头用的源和超级牛力用的源不一样，大晕头用的源里面的软件都是以 **rpm** 格式打包的，而超级牛力用的源里面的软件都是 **deb** 格式的。

RPM 是 Red Hat Package Manager 的缩写，也就是红帽子软件包管理器。听名字就知道这种东西是大红帽子铺发明的。早先的时候，Linux 刚刚出世，上面的软件基本都是用源码包的形式发布的，也就是 tar.gz 那样的包。但是这样的软件包安装起来步骤有多复杂大概大家也都体会过，而且每个包都要被编译一下再安装的话，也浪费时间。也有的人把编译好的二进制文件打成 tar.gz 包的，不过人家用户把这个包解压到哪里无法确定，也就无法实现一些自动的设置（比如装完一个软件自动在菜单里出现相应的启动项）。于是帽子铺以 Linux 村帽子业界老大的身份创建了一种软件打包的方式——rpm。这种软件包，下载下来就是一个 xxxxx.rpm 文件，里面的内容是编译好的二进制程序。由一个叫做 rpm 的程序负责解开 rpm 包，并把里面的各种文件放到相应的目录中去。我们就管这个程序叫做“肉包管”吧。有了肉包管以后，装软件就省事了，把 rpm 包交个他就行了，他会把里面的文件分门别类的放好：配置文件放在/etc/；可执行文件放在/usr/bin/；库文件放在/usr/lib/，等等等等。（当然，这写目录都不是绝对的，只是一般情况。）除了编译好的软件，rpm 也可以打包源码，一般 rpm 打的源码包都已.src.rpm 来作为扩展名。肉包管不但可以把这样的源码包解出来放到指定的地方，还可以自动对这些源码进行编译，不过这个功能不常用，常用的还是二进制的 rpm 包。

有了肉包管，安装软件相对简单了一些，不过依然有些问题为另肉保管和使用者带来挥之不去的阴影，那就是——依赖关系。

什么是依赖关系？很简单，当你在用查皮装游戏的时候，游戏提示需要安装 Direct 10 才能正常游戏，于是你就得先装好 Direct 10 再回来装这个游戏，这就叫依赖关系。也就是说软件 A 要想安装，必须先安装软件 B。有人说这还不简单，那就先装 B 呗～唉，要是这么简单就好了。很多时候人家要装软件包 A.rpm，肉包管告诉人家说，你要想装 A，就得先装 B。用户想：好，让我装 B 我就装回 B 吧，可是刚要装 B.rpm，肉包管又发话了，要想装 B，先得装 C。行，那就先装 C。可是还没装呢又被肉包管告知，要想装 C，先得装 D，要想装 D，先得装傻，要想装傻，

先得装酷，要想装酷，先得装孙子……此时电脑前的用户已经口吐白沫了。但，这还不是最刺激的，最刺激的是肉包管最后又来了一句：要想装孙子，先得装 A.rpm！勉强爬起来的用户再次被击倒。

虽然肉包管可以不顾依赖关系强制安装一个 rpm 包，但是这样装上的包谁又能保证他可以正常工作呢？好在经过了多年的痛苦折磨后，大晕头横空出世了。

大晕头我们说过，和超级牛力一样，可以自动上网下载软件，并且解决好依赖关系。也就是说，你让他装 girl.rpm，他会告诉你，安装 girl.rpm 需要装 money.rpm，装 car.rpm，装 bility.rpm，我给你一口气都装上吧～然后在得到用户的同意后，麻利的安装好全部该装的软件包。

关于大晕头的重要性不必多说，参见以前对超级牛力的介绍。这里只想说一个问提，由于红帽子公司的影响，rpm 包成为的 Linux 村里通用性相当好的打包格式。很多著名的发行版如 Suse,Mandriva,Centos 等都使用了 rpm 作为软件包管理器。于是，市面上大多数软件，只要提供二进制包的，多数都会提供 rpm 格式的包。因此就会出现个问题——有的软件只有 rpm 格式的，在我这里怎么装呢？

今天发生了一件事，我发现，我的主人，他，他不是一个人！哦，别误会，我的意思是，他不再是一个人了，厄，也不对。其实，我的主人从生物学的角度讲，他还是一个人类，关键是，不是一个了。

说的挺乱，其实事情很简答——主人又创建了一个用户，叫 lili。

以前主人用来登录的用户名叫 lanwoniu.这个名字是在最初安装我的时候起的，安装的时候我会问一句:“你是谁阿，名字叫啥？”然后对方告诉我他的名字，我就认定这个人是我的主人了，这个人有很特殊的权力。我们之前说过，我们 Linux 系统里有个 root 用户，很好很强大，但是这个用户也很危险，因为他太强大了，说不定那天一不小心删点啥不该删的文件，敲点不该敲的命令，说不定整个系统就挂了，所我们 ubuntu 限制 root 用户的使用。可是也得需要有人能管理整个系统阿，谁呢？就是安装的时候认识的那个主人。这个用户有着一种能力——变身！不是变大星星，也不是变饿狼啥的，而是这个用户可以临时获得 root 用户的权限，从而相当于变成 root 用户。至于怎么变，相信大家都知道了，sudo 嘛。

有点扯远了，呵呵。总之一直以来我和我的 lanwoniu 主人过着愉快的二人世界。当然，lanwoniu 这个名字只是为了那家伙自己好记而起的，我其实是不关心他的名字的，我关心的是他的用户 ID 号，也就是 UID。在我的概念里没有什么 lanwoniu 用户，dasanba 用户，tenzu 用户，ee 用户等等，我的脑子里只有 1000 号用户，1001 号用户，1002 号用户。进行跟用户，跟用户权限有关的动作的时候，靠的都是这个号。可是让主人记住这个号有些不现实，首先不直观，其次要是人多了也不好记，1000 这个数还行，要是有个啥 535353124325 用户，那谁记得住阿，所以就得起名字。每次启动，我都先要问：你是谁，报上名来。用户就会输入自己的名字，然后我就拿来一个文件，一份人名单——/etc/passwd 文件。这里面记录了所有用户的用户名和 UID 以及其他一些信息。每个用户的信息占一行，以用户名开头。比如人家输如 lanwoniu，我就在 passwd 文件里找，诶，找到这么一行:”lanwoniu:x:1000:1000:lanwoniu,,,:/home/saub:

/bin/bash”一看前面这几段就知道，lanwoniu，这个人就是 UID 是 1000 的那个，也就是最初创建系统时候建的用户，哦，也就是我的主人。可是可不能光凭你红口白牙这么一说是我的主人你就是了，得拿出证据。证据是什么呢？密码！

用户名输入之后，还得输入密码，输入之后，我还得拿来一个文件来查看用户输入的密码对不对，这回不是 passwd 了，是/etc/shadow。是不是有点迷惑？这 passwd 文件，看名字应该是存密码的阿，password 嘛，怎么不再这呢？其实，很久以前，我们 Linux 确实是把用户的密码存在 passwd 文件里的。就写在用户名后面，用户 ID 前面的两个冒号之间。当然，我们不会啥到用明文写用户的密码的，写的都是加密后的秘文。比如我主人的密码要是记在 passwd 文件里，那 passwd 文件里的那行可能就是这样：

“lanwoniu:6IQUoDoR8Wr/HqL$n88KT93gnqsC8hF9jT:1000:1000:lanwoniu,,,:`/home/saub:/bin/bash”

后来觉得这样不是很安全，就干脆把密码存在了/etc/shadow 文件里，passwd 文件里面密码的位置只留了个 x。shadow 文件要比 passwd 权限更加严格，本来 passwd 文件就只有 root 用户才能修改，普通用户也就看看的份，而 shadow 文件，普通用户连看也不让看！

回过头来再说 lili。

那一天主人打开了系统管理里面的用户和组，当时我也不知道他要干啥，于是也只是列出了当前系统里的用户，两个，一个是 root，一个是 lanoniu。有人说不对呀，刚才我看我系统里的 passwd 文件，那里边一大堆用户呢啊，什么 bin，sys，mail，nobody，名字一个比一个奇怪。是的，这些用户是有的，但是这些都是系统保留的用户，你看他们的 UID 了没，都在 1000 以下，这种 UID 在 1000 以下的用户都是有一定特殊用途的系统保留用户，具体有什么用，咱们以后再说。现在主人有动作了，他点了“解锁”按钮，那意思就是说：“我要变身啦！”要变身，得拿出点证据，我得再次确认一下这人是不是我的主人，别回头有破坏分子趁我主人上厕所的功夫偷偷用他的电脑干坏事。于是我提出要求：“要变身就先输入你自己的密码。”主人很流利的输入了，看来真

的是他，于是，我允许他变身成了 root。（只是临时的哦，且只针对用户管理这个程序，其他的操作还是个普通用户）

变身之后，主人（作为 root）点击了添加用户，哦，原来他是要加用户啊。那我得问问清楚，这个用户叫什么名字啊，真名叫什么啊。是个什么用户啊，是普通用户还是管理员？密码是什么？或者让我随机设个密码并麻烦你记一下?这篇填完了之后就可以点确定了，不过你也可以写一些其他的信息，比如“联系人”标签，可以写一些这个用户的联系信息;“用户权限”标签可以详细的指定这个用户的权限，能不能使用光驱啊，能不能使用音频设备啊，能不能使用 VBox 啊之类的;“高级”标签就可以设置一些高级选项，比如这个用户的主目录是哪，shell 用什么，用户属于那个组，ID 是多少之类的。不过如果没有特殊要求的话，基本上默认的就好。主人就全都使用默认的设置，只填写了用户名和密码就确定了，于是，一个 lili 用户建出来了。

这个 lili 用户的创建包括很多动作，首先，创建这个用户，创建用户的本质就是在/etc/passwd 文件里写上相应的一行,并且在/etc/shadow 文件里协商这个用户的密码相关的信息。然后，创建一个与用户同名的组，也就是创建一个 lili 组，并且让 lili 用户属于 lili 组。然后，还要在/home/目录下创建一个于用户名相同的目录，也就是创建一个 lili 目录，这个目录就叫做这个用户的家目录。以后这个用户的所有文件就都放在这里目录下了。就像我主人 Lanwoniu 的所有文件都放在/home/lanwoniu/目录下一样。其他地方的文件目录都是系统相关的，一般只有 root 才能进行写操作。当然，也有像/tmp 这样的目录作为大家共用的临时文件存放处，不过这里面的内容可不保证下次启动的时候还有，所以关机前记得备份好，放回自己的家目录。除此之外，还要在新建用户的目录下放好两个隐藏的配置文件：.bash_logout 和.profile。好了，这样就完成创建这个用户的所有手续了，现在这个用户就可以登录进我这个系统了。

果然，主人做完这些就注销了，之后，那个 lili 就登录了进来了。这家伙一看就是个新手，鼠标飘来飘去的连菜单在那都找不到，真替她着急。总算鼠标点击了狐狸妹妹，狐狸妹妹被点了之后立即出台——呃不对，立即起床开始工作，刚要打开主人常去的那个狗狗哥的网页，我赶紧提醒她，不对！这个登录进来的不是主人，是 lili。狐狸马上反映过来，去 lili 的家目录下找.mozilla 目录，那是她用来存放每个用户的配置文件的地方，结果当然是没找到，因为 lili 这个用户才刚刚创建嘛。于是狐狸就打开默认的网页，并顺便创建好.mozilla 目录，以便记录下当前用户的使用习惯。

lili 开了几个网页，觉得也没什么意思，就开始点别的玩。点着点着，点到了“位置”，打开了“lili 的主文件夹”，里面已经建好了什么“图片”，“文档”，“视频”之类的目录，不过里面当然是什么也没有了。这家伙看着里面也没啥东西，就点了向上按钮，到了/home 目录下，看见了主人的主目录，然后点了进去。我心说，这家伙跟多数新手一样，不知道文件都应该放在哪里。她点开了主人的文件夹，看见一个叫 OOXX 的文件夹，于是好奇的双击了一下。嘿嘿，这个目录可是主人特意交代过的，看到他双击，我立刻告诉她：您没有查看“OOXX”的内容所需的权限。还算客气吧。

要知道，我们 Linux 用的文件系统可不是弱智的 FAT32，那种系统连隔壁的查皮都不爱用了。我们的文件系统上的每一个文件都是有主的，很明确，没有任何含糊。每个文件都写明了，这个文件是谁的。比如主人文件夹下的那些文件，基本都是属于 lanwoniu 这个用户的。而像/etc /usr 这些目录下的文件基本都是属于 root 用户的。虽然我这么说，但是您别误会，别以为文件属于谁就看这文件在哪放着，其实不是，文件属于谁，要看这个文件的“属主”，英文叫 owner。

这个“属主”是每个文件都有的，但不是记录在文件里，而是记录在文件系统上。咱打比方，硬盘空间就像你屋子里的空间，文件就像你物理放着的一个一个的箱子。那么文件系统呢，就相

当于你屋子地板的材质。各种不同的文件系统，像咱们说过的 **ext4,xfs,NTFS** 等等，就像你的地板可能是木地板的，可能是瓷砖的，地毯的，或者水泥地等等。那些我们 **Linux** 能够使用的文件系统，比如 **ext4**，**xfs**，都有可以记录文件属主和权限的地方，我们吧文件放到一个地方以后，就要在文件旁边的地上写上，这个文件是属于哪个用户的，权限是如何如何。（关于权限具体如何写，咱们稍后再说。）而如果是查皮的文件系统，我们就没法写。这就好像我们的文件系统就像是浅色的地毯，我们一般是把文件放好以后拿着毛笔在旁边地毯上写上这个文件是属于谁的（你真不怕糟蹋东西～）。而查皮的文件系统就像是白瓷砖，虽然他自己可以拿着记号笔在上面记录文件的权限，但是我们拿着毛笔的在上面啥也写不上。这也就是为什么我们可以读写 **FAT32** 和 **NTFS** 分区，但是却不能把系统安装在这样的分区上的道理。

除了“属主”，每个文件还有“属组”，也就是说明这个文件属于那个用户组。这两个可不矛盾啊，并不像你们人类的现实生活中那样，这个垃圾桶是属于公司的公共财产，所以就不属于你自己，你就不能抱回家去。我们这，一个文件必然会属于一个用户且同时属于一个用户组，这一个组内的所有用户都对其拥有一定的权限（具体什么权限，别急，一会说）。那一般文件都属于什么组呢？还记得创建用户的时候同时创建了一个同名的组么？对了，一般用户的文件的属主是这个用户，而属组则是同名的组。而这个和用户名相同的组里边一般只有该用户，所以，这个文件终于名正言顺的只属于这个用户了。这就像虽然垃圾桶属于公司财产，但公司是属于你的，整个公司就你一个人（皮包公司……），所以这个垃圾桶说了半天还是属于你的。

当然，光知道了一个文件是哪个用户的（属主），以及是哪个用户组的（属组），还不能够判断让不让打开，还要看这个文件的权限设置。

我们 Linux 对文件的权限设置不像查皮那样复杂，在查皮那里，任何一个文件都可以针对任何一个用户设置权限(当然，前提得是 NTFS 分区)，比如文件 xiaoshuo.txt，可以指定张三对这个文件干什么都行，李四只能写，不能读，王五可以读，可以写，但是不能删除这个文件等等。我们对文件权限的设置就没这么麻烦，我们这里一个文件的权限只能针对三类人：文件的属主，文件的属组，还有其他人。权限只有三种：读，写，执行。比如设置文件的属主可以读，可以写，文件的属组只可以读，其他人什么权限都没有。就这么简单！不过别看简单，通过各种设置的配合，照样可以实现各种权限的控制。

比如有人说了，你这权限只有读写和执行三种，我要实现能读写但是不能删除怎么办？好说，删除，其实就是对文件所在目录的修改。我们不是说过么，在咱 Linux 的地盘里，一切都是浮云，哦不对，一切都是文件。文件夹也是一种特殊的文件，对文件夹不能写，就意味这不能改变文件夹的内容，也就是不能在文件夹里面添加或者删除文件。但是，文件夹里已经存在的文件是可以修改的（只要对那个文件有权限）对文件加的读权限，就是列出文件夹里文件列表的权限，能读，就能查看文件夹里面有什么东西，不能读，就不能看。（也就是不能用 ls 令来查看）除此之外，文件夹还有执行权限，所谓执行，对于文件夹来讲，就是 cd 进去文件夹的权限。也就是说，一个对你可以读不可以执行的文件夹，就是说你可以用 ls ./xxxxx/命令来查看这个文件夹里的内容，但是不能 cd ./xxxx/进去。反过来，如果一个对你可以执行但是不能读的文件夹，那你就可以用 cd 命令进入到那个文件夹里面，但是进去之后用 ls 啥也看不见。当然，如果既没有读权限，也没有执行权限，那就更啥也甭想了，就像刚刚 lili 要进那个文件夹那种情况一样。

其实按说主人你要想藏什么东西，就应该把那个目录设成隐藏的，眼不见心不烦嘛。不过说起来我们这里要想隐藏一个文件或者文件夹，确实不像查皮那里那么方便。准确的说，我这里的

文件其实没有真正的“隐藏”这么个属性，跟查皮那不一样。我们这里的文件要想隐藏，只要把文件改成个以.开头的名字就可以了，我在显示文件的时候，发现文件或者目录是以.开头的，就直接忽略掉，不显示，除非用户指明要连隐藏文件也显示。（也就是 **ls** 加上**-a**，或者 **nautilus** 里按 **ctrl+h**）

这里 lili 正操作着呢，忽然那边网口发来的信息，有人要登录，我赶紧过去一看，咦？是主人。似乎是从另一太电脑发来的请求，于是我赶紧核对了密码，让他登录进来了。

有人说，你这不是有个 lili 在登录这么，还能再登录进来一个人？那当然了，我可是多用户多任务的操作系统，可不像隔壁那个落后的查皮。有人说了，那个查皮那里也能有很多用户阿，不也是多用户么？这个多用户的概念可不是这样的。话说很久以前的那个剁死系统，那就是个单用户系统，压根不去分谁在用电脑，只要电脑启动了，就一视同仁的提供各种功能。这是单用户系统，没有争议。后来有了 Windows 3.x，不过是 DOS 的一个图形界面软件而已，不是完整的系统。再后来的 Windows95，也没有多用户的概念，这些都是单用户系统，没有悬念。后来 Windows98 了，可以用不同的用户登录，不同的用户登录进去以后，能看到不同的壁纸，不同的配色方案，不同的桌面快捷方式，这就是多用户么？不是！这不过是根据你输入的用户名和密码来调用不同的主题罢了，各个用户之间是没有区别的。任何用户都能对系统作任何操作。windowsME 咱就不说了，跟 98 没什么本质区别。后来有了查皮，这回好了，有用户权限的概念了，有管理员，有普通的用户，还有 Guest 用户。管理员可以随便改系统的各种设置，普通用户就不行，Guest 用户更是只能登录进来看看而已，并且，每个文件每个程序都能通过设置，指定哪个用户能用，哪个用户不能用，那个用户能看不能写等等。这就是多用户了？依然不是！

真正的多用户，是能够让 n 个用户同时登录，同时使用系统，同时运行各种软件并且还互不干扰的能力。比如，张飞同学正在登录进这个系统，打开着狐狸妹妹来查找哪个酒馆搞活动，同时关羽同学也登录进来这个系统（可以从远程，也可以在本地，反正是登录进来了。），也打开狐狸妹妹上网看《春秋》。于是这时候就有两个狐狸妹妹在内存里跑来跑去，我要能分清楚哪个是狐狸妹妹是张飞同学打开的，哪个是关羽同学的。张飞同学的那个狐狸妹妹如果要下载东西的话就不允许存到关羽同学的家目录里，关羽同学的狐狸妹妹要读配置文件的话就不能去张飞同学的目录。要是再来个刘备同学要开狐狸妹妹的话就得有三个狐狸妹妹在内存里转悠，关羽同学要是

再开狐狸妹妹的话就仅仅让那个关羽的狐狸妹妹再多开个标签就好。并且，关羽同学在使用电脑的时候是感觉不到张飞同学也在用电脑的，反过来也是。这才叫真正的多用户。像查皮那样的同时只能有一个用户登录，再来一个用户的话，这个用户就得先退出去，这样称不上是多用户。是因为有点软公司没这个能力，做不出多用户系统么？当然不是，winnt,win2000，win2003 等所有服务器的系统都是支持多用户的。但是人家有点软公司是要靠卖软件赚钱的，不像我们 Linux，简直就是公益事业。如果家用版本的系统都能和服务器版有一样的功能了，谁还买服务器版？那不就赔死了。而且家庭里，谁没事一台电脑俩人用阿，那还怎么拉动鸡弟屁，经济怎么发展，房子怎么涨价……哦，扯远了。但是我们 Linux 就不一样了，反正是免费获取，多用户这功能，反正我有，用的上就用，用不上就放着，又不花钱。

Lili 在主人的目录里找到了一个什么阿凡提达.rmvb 的视频，好像是个挺有名气的电影，根据默认播放器的设置，我叫醒了 Totem 来播放这个视频。放了一会，好象是主人来到了 lili 身边，说:“这个播放器不好用，用这个吧”（她插着麦克风，我偷听到的，呵呵）说着，关了 Totem，用 SMplayer 打开了那个视频。被替换下来的 Totem 显然很不服气，抱怨说:“要论真本事，我也未必比谁差，不就是长的漂亮点么，哼！”这 SMPlayer 听不过去了，一边忙着播放视频，一边反驳:“你本领不差，难道说我本领差了？你能记住每个视频上次播放到什么位置么？我可不光是靠长的漂亮。不过话说回来，我们 Qt 的程序，倒也确实比你们 gtk 的细致了不少，用户爱看我们，这也是没办法的事情。”Totem 怒目而视:“你别忘了，现在内存里面的可是我们 Gnome 环境，你敢在这说这种话，有点欺人太甚吧！”“Gnome 怎么啦，Gnome 就是不如 KDE 好看，有错么？”我一听，坏了，这俩又要打起来。

话说在 Linux 这片恩怨情仇的大地上，做图形界面这行当的，以两大帮派为主。一个是 Gnome 帮，帮中的软件们都修炼 gtk + 宝典，帮众们信奉简单高效的做软件准则。什么事情，都力求用简单的界面来实现，并不留给用户太多可以设置的东西。G 帮认为，对于初级用户，不要搞那么多设置项，搞的用户头晕脑涨。能默认的都默认好了，打开软件就工作。而对于需要自定义的高级用户来说，直接去改配置文件就好了，这难不倒高级用户。另一大帮派，就是 KDE 派了，K 派们都修炼 qt 大法，派中的软件都觉得界面要做得方便，易用，易于配置，坚信细节决定成败。界面要细腻，要漂亮，让人家一看就喜欢，这才是好的图形界面软件。所以 K 派的软件都有好多可配置的选项，新手可以无视，老手配置起来也很方便。两帮派观念不同，本来也没有谁强谁弱，但是偏偏有时候还是会争个风头，动不动就吵个天翻地覆。这不今天，又来了。

Totem，那可是 Gnome 的嫡系软件，自带的播放器。本来被主人换下来，心里就不舒服，再一听 SMPlayer 这么说他，更是无名火起，顺手超起一把兵刃——一个视频文件，指着 SMplayer 说:“有本事，别光耍嘴上的功夫，咱们过两招瞧瞧。”不等 SMplayer 答话，刷刷刷原地耍了

起来，只见他一招一式，无甚惊人之处，却招招使的灵活熟练，那视频文件就好像本来写进他自身的代码段里一般，什么播放，暂停，前进，倒退，全屏，截图，调横纵比，显示字幕，虽说只是播放器的初级功夫，难得是样样做得恰到好处。这时那 SMPlayer 正在给 lili 用户播放视频，本不该分神，却忍不住争强好胜之心，也找个兵器练起来。他一边给 Lili 播放，一边跟 Totem 比武，双手同时耍开两个视频，速度自然是慢了一步，但确是花样叠出。什么音画时差调整，什么字幕控制，网上查找字幕，降噪，反拉丝，旋转屏幕，专门找那 Totem 不会的招式操练。Totem 见对方同时耍着两个兵器，虽说稍稍缓慢一些，却依然稳扎稳打，何况招事确然比自己多，自觉是落了下风。可他又怎肯示弱，扔下手上这兵刃，又从兵器架上超起另一个把，练了一会又换一个，顷刻间，已经换了 18 件兵刃。什么 rmvb,mov,avi,wmv,flv,cd 音轨,mp3,mp4 等等等等，件件是拿的起，放的下。那意思，别看我会的招事不如你多，但我能耍的兵器可不少。SMPlayer 见到此景，微微一笑，扔了手中的家伙，把 Totem 换下的兵器一件件拿起来，依次也耍了一通，却也是样样精通。只看的 Totem 一身冷汗，自知自己本领也就到此为止了，可是却如何下的了这个台？正自思虑之间，只听得身背后有人言道："贤弟，你且下去歇息，哥哥我来会会他！"

只见 Totem 身后闪出一人，正是 GNOME Mplayer。一看这名字您就知道了，这是 G 帮的人。SMplayer 一看见他，顿时也是心里犯嘀咕。为什么呢？您众位大概有所不知，这 GNOME Mplayer 和 SMplayer 其实都不过是个图形界面的外壳而已，他们的后台都是 Mplayer 老大。因此上，这二人本是同门兄弟，只因信仰不同，故而一入 G 帮，一投 K 派。即是同门学艺的兄弟，这本领又能差的几何？这 SMplayer 寻思，纵然自己比 GNOME Player 勤学苦练，无奈方才刚刚跟 Totem 大战一场，有要照顾这那 lili 用户的视频，自己恐难取胜。可那 GNOME Player 更无废话，上来就超起兵刃比划起来，也是将一件件兵刃轮番耍起来，比刚才 totem 所耍的还要多上一些。SMplayer 只得咬咬牙，再跟 GNOME Player 拼死一战。只见整个内存中那真是录像与配音齐飞，字幕共长片一色。二人杀的是难解难分。这 GNOME Mplayer 平时便不服气 SMplayer，本是同门学艺，可在江湖上，绿林中，这 SMplayer 的名号却远远盖过他 GNOME Mplayer。今天他就想让别人见识见识，他 GNOME Mplayer 不比 SMmplayer 差。因此他论起武艺，虽然与 SMplayer 差着一点，但是招招都拼尽全力，狠辣异常。斗到一百三十多回合的时候，这 SMplayer 渐渐支持不住，眼看就要跳帧了。这时候，内存里已然围满了人。有看热闹的，有叫好助威的，有打酱油的，有路过的，还有不明真相的群众。围的是里三层外三层。虽然主人也安装了 KDE 环境，可是这次开机，进入的是 GNOME，因此这内存里围观的基本都是 G 帮的人，K 派的人都在那硬盘里睡觉呢，SMplayer 可谓是孤军奋战了。

正在这时，OpenOffice 老先生推推眼睛，咳嗽了一声，发话道:“两位请住手！”OO 老先生可谓是德高望众，虽然本身是 G 帮的人，可基本所有的 Linux 发行版，不管是用 GNOME 也好，用 KDE 也罢。都会请他做默认的办公软件,因此上即便是 K 派的人,也都服气这位老先生。话音刚落,GNOME Mplayer 罢手不打,SMplayer 自也愿意停下来休息,站在那里直喘粗气。只听 OO 老先生继续说道:“要我说阿，这机器启动时进入的是 GNOME，满内存都是 G 帮的人，可用户偏偏要叫来人家 SMPlayer 来播放视频，这已然是人家赢了。更何况两位轮番上阵，不是也没把人家

一人打下来么？”后面这句，自然是看着 Totem 和 GNOME Mplayer 说的。Totem 艺不如人，自觉惭愧。GNOME Mplayer 愤愤不平，却也无话可说。本以为一场风波就此平静，忽然，lili 发来命令：叫那个 EVA 起床，我要聊 QQ。

eva 这小子动作麻利，跑进内存里，正要连接服务器，看见一群 G 帮的人围着 smplayer，smplayer 在那呼呼喘气。顿时肝火上升，大喊:“你们什么意思？以多欺少是不是？告诉你们，我们 K 派的个个都是以一当十的好汉，何况现在硬盘里 k 派的软件可不一定比你们 G 帮的少。都叫起来，看谁打的过谁。”SMPlayer 不愿意再惹事生非，更何况周围都是 G 帮的人，就算加上个 EVA，甭管是初号机还是零号机，那也打不过这么多人，于是连连向 EVA 摆手，却还没把气喘顺溜，之说:“算，算……”无奈 EVA 年轻气盛，立时接过去说:“算什么算，不能算！刚才是谁挑头吵架的，出来道歉。”周围 G 帮帮众本要各自散去，一听 EVA 如此将火，都过来位自己这边的人站脚助威。大家都望向 OO 老先生，心说您发个话，我们该怎么着阿？OO 老先生又推了推眼睛说道:“我看这也就算了吧，大家都是抬头不见低头见的……”正说道这里，主人那边要打开文档，OO 老先生赶紧赶过去，一边走一边回头对众人说:“大家勿动干戈阿～”那老先生毕竟年迈，又向来做事一丝不苟，既然去给主人处理文档，就断然没有气力和精神来管这边的事情了。

老先生一走，大家七嘴八舌的议论开了，有的指责 EVA 不该没事找事，有的说 Totem 没错，干什么要道歉？你一言我一语的，EVA 想要辩驳，却不知该先反驳谁好，之说出个:“你们，你们……不是……”这时候，又一个声音响起:“大家静一静。”G 帮众扭头一看，原来是星际译王，星爷。星爷也是个知识渊博的人，所以也有些威信，大家对他也是很尊敬的，因此众人都安静下来。星爷说:“咱们就这里说说明白。Totem 怀才不遇，抱怨了两句，确是他的不该，不过你 smplayer 说什么本帮软件都不如你派细致之类的话也多少有点不公。这就算扯平了吧。后来你们三人比划了比划，就当是朋友间切磋武艺，也没什么，大家握手言和吧。”SMplayer 这时歉然道:“也是我年轻气盛……”还没说完，就被 EVA 抢过话头:“星爷，您说的我都理解，咱们也不愿伤了和气。不过您说‘G 帮软件不如 K 派界面细致漂亮’这句话有些不公，我却不能认同。咱们有一说一，有二说二，你们 G 帮的软件哪一个长得比我们 K 派同类软件好看的，说出来听听？”星爷本想随便说两句，把事情压下去也就得了，谁想到这 EVA 这么较真，便有几分不悦:“美丑本无定论，

还是人们的审美观点不同。本帮软件，向来求简，以简约为美。清心淡雅，简单易用，每个软件，均不求多能，只做好一两件本职工作，深合 Unix 哲学之美。想那 vi,emacs，sed，众位命令行时代的前辈软件，又有哪一个是靠张的好看才获得人们喜爱呢？贵派界面确是细致漂亮，倒也算是另一种美貌，不过角度不同罢了”这话明着说两派软件欣赏角度不同而已，却暗指 G 帮软件才是深合 Unix 哲学的正宗。SMplayer 怎会不明其中真意？只是不便发作，拱手说道：“星爷教训的是。我派立派虽然早上那么几年，然终究不如贵帮发展的范围广泛，乃至几大发行版都用 GNOME 为默认环境。我们这界面的问题，也都因为使用了 C + + 作为开发语言。这面向对象，总比面向过程的传统 C 语言更利于提高开发效率，所以也就有时间创建复杂界面吧。所以，界面，也不是我们的功劳，实在不值一提。”这一下表面上是谦虚，实际上却是说 K 派历史比 G 帮悠久，用的 C + + 也比 G 帮用的 C 语言高级了。G 帮那边的皮筋小弟坐不住了，插话道：“C 也未必落后，C + + 也未必高级，面向对象也不过是个思维模式，C 也一样能面向对象，C + + 也一样能写成面向过程。我就听主人聊天时候跟人家讨论过，C 语言的结构体里放上几个函数指针，和 C + + 的 class 又有多大差别了？只要编程的思想在，用什么语言不是重要的。若是功夫到家，飞花落叶，皆为兵刃。反倒是 C 语言更加轻巧，效率更高。”EVA 马上反驳：“C + + 乃是一本高深莫测的内功心法，开发效率上面，C 语言远远比不上。连个异常都处理不了的语言，毕竟是有些过时了。”“哈哈哈哈”豪爽的鹦鹉螺走了出来，“真是笑话，难道程序员写个程序自己都不知道程序会运行出什么结果？知道哪里会出错就多写些代码做检查嘛，还非要抛出什么异常？！哈哈。可见 C + + 只是给学写程序的小学生用的吧。”这时候从硬盘中走出一人说道：“C + + 开发方便，理念先进。你们也说了，C + + 支持面向对象，也一样可以面向过程。正如一口宝兵器，既可以做刀来用，也可以当剑来使，不是更加得心应手么？再说这异常问题，C + + 中你若是清楚程序会运行到何种状态，不使用异常检测，又有何妨？C + + 不过是多提供了一些可能性罢了。”大家一看，说话的是 KDE 的文件管理器，小海豚，他跟鹦鹉螺可谓是针锋相对了。星爷说道：C + + 提供更多可能

性，这本是好意，无奈太过臃肿复杂，虽说是本秘籍，但是练起来，难免让人走火入魔。反倒是 C 语言，初看只是粗浅武功，但越练越觉得其中内涵，深不可测阿。别的不说，咱们头，Linux 内核，难道不是 C 语言写的么？”说着，扭头望向了我。

剑

利剑

无形的利剑

唇枪舌剑！

一边是 G，一边是 K，中间便是这把唇枪舌剑。枪来剑往，远远的，只有我有孤独的身影。我就这么走开么？或许我是应该走开，但就在我还没有做出这个决定的时候，这把剑指向了我！G 说："看！他就是用 C，你们敢说 C 不如 C + + ？"K 没有说话，看得出来对我有些忌惮。高手是不必出招的，通过他的气场就可以感觉到，我就是高手！K 无话，G 也无话。

我走近了他们一步：我该如何停止这场无谓的争端？我该怎样化解这数年来积攒的恩怨？我的对手不是 K，也不是 G，而是在他们身后，是他们身后深远的文化。我要挑战么？

我又走近他们一步：我该不该出招？我知道我的招数可以凌驾于他们之上，十步一杀！不会有声音，不会有差错。用剑么？笑话，我已很久不拿剑了，剑，就在我心中。

我走出第三步：要帮助 G 么？我用 C，但也用过 C++。天下武功，原无不同；要帮助 K 么？我喜欢 Qt 的精致，也欣赏 Gtk 的简约。各有所长；要消灭掉他们所有？Gnome 和 KDE，都是优秀的。

第四步，站定：就让我结束今天这场争斗吧。不去管历史，不去问将来，一切，留给后人评说。气场大盛，心中剑起——出招！

"都不要吵啦！Init！快过来，没收他们工号，收回他们内存，全都打成僵尸，回硬盘给我睡觉去！"是的，内核就是有这么点特权，想干掉谁就干掉谁。不过这么一折腾，系统就忽然卡了一下，不知道主人有没有发觉。以后您要是用 Ubuntu 的时候忽然系统卡住或者死掉了，那多半是里面软件们又打起来了。

那天之后，**lili** 就没再来登录过，可能是觉得我不好？长的不好看？不得而知。最近 **G** 帮和 **K** 派倒是不吵架了，因为每天都很少叫他们起床。你问为啥？因为主人开始研究脚本了，每次启动都不进入图形己面，直接字符操作。你问怎么不启动图形界面？简单，一条命令：**sudo update-rc.d -n -f gdm remove**

或许是因为那个 **rubbish176** 号让主人伤心了吧，从那之后主人就不研究 **c** 语言了，这不现在开始研究 **shell** 脚本语言了。有人问 **shell** 脚本跟 **C** 有什么不同呢？这个吧……你说骆驼跟洋葱有什么不同呢？他压根就是俩东西！

说起编程，编程就得有语言。语言有很多种，可以分成两类：编译型和解释型。编译型就是像 C 语言那样，用 C 语言写出来的源代码就像图纸，前面说过了，需要 GCC 施工队来把图纸变成真正能够运行的程序。解释型语言就不需要施工队了，解释型的语言写出来的代码不是一份图纸，倒像是一份任务清单，上面用某种语言写明先做什么，后做什么，最后做什么。写好之后保存成一个脚本文件，随便起个名字，然后赋予可执行权限，就可以直接执行了。有的人怀疑，这个文件写出来就是一个文本嘛，又不是一个程序，他怎么会自己跑到内存里去执行？问的好，脚本文件当然不可能自己跑到内存里运行，除非是闹鬼了。解释型的语言既然是一份任务清单，那么就得有一个可以照着这个列表执行命令的家伙，我们管这个角色叫做解释器。

比如我们说的 shell 编程，shell 有很多种，我这里默认的 shell 是 bash，他就是 shell 脚本程序的一个解释器。主人随便建一个文件，里面写上：

echo “Hello World!”

然后保存成 ttttt 文件，赋予可执行权限，然后运行./ttttt。这时候，首先，我会判断这个文件可执行，并且是个一文本文件，那么说明这是个脚本，是个解释型语言的脚本。然后根据当前系统的设置，找到默认的 shell，比如我这里默认的 shell 是 bash，于是我叫醒 bash，把这个脚本交个他，让他去执行。那如果不想用默认的 shell 呢？那就在脚本文件的第一行明确写明要用什么 shell 执行。格式类似这样“#!/bin/bash”，记住，这行必须写在脚本的最开头，这个好理解吧。总不能我打开了这个脚本，叫来了默认的 bash 执行了一半了，才发现您在中间写着一行：#!/bin/csh，这时候才知道脚本应该叫 c shell 来执行，你这不捣乱么。所以，一定要在脚本的最开始搞清楚这个脚本是用于哪个 shell 的。

shell 有很多种，比如 Bourne Again Shell,C Shell, K Shell，Debian Almquist SHell 等等。我们 Linux 最常用的就是 Bourne Again Shell，也就是 BASH。关于这个 shell 的历史，这怕是要追溯倒 Unix 的年代……

那个时候，最初的 UNIX 系统的 shell，是那个研究铃铛（Bell）的实验室里的一个叫做 Ken Thompson 发明的，叫做 Thompson shell。这个 shell 是一个很简单的程序，它不过是作为 UNIX 系统的用户接口而已。（就像现在 bash 作为 linunx 系统的命令行接口。用户输入的命令都是由 bash 来解释和执行的。）那时候的 Thompson shell 顶多是可以把几条命令一起写在一个文件里来执行，类似批处理，没有流控制，没有变量，没有函数，所以还完全谈不上 shell 编程。后来觉得有必要加入一些条件判断阿，跳转阿之类的功能，就依靠外部命令实现了。与此同时，铃铛实验室的另一个牛人，Steve Bourne，也设计了一个 shell，叫做 Bourne shell。（他们都生怕别人不知道程序是谁写的，都以自己的名字命名）这个 shell 就强大些了，有了基本的流控制源语，if else 之类的。这俩人都力挺自己设计的 shell，渐渐的两种 shell 有了各自的追随者。有觉得这个简洁的，有觉得那个好用的。有道是一山不容二虎，到了 1970 年代末，打起来咯～由于两个 shell 互相不兼容，而一个和平统一的 UNIX 弄两套 shell 是不大合适的，（大概是因为那时候他们没有好好学习一国两制的理论）所以必须确定一个 UNIX 的标准 shell。于是，一场旷日持久的战斗打响了，两派相互争论各自 shell 的优缺点，您看见 K 派跟 G 帮怎么吵架的了么？估计他们也差不多。最终，以 Bourne shell 的胜出结束，Bourne shell 作为默认的 shell 出现在 UNIX 第 7 版系统中，其二进制程序被命名为 sh 放在了/bin。直到现在，Linux 发行版中依然都存在着/bin/sh 这个文件，不过现在这个 sh 一般都只是个链接了，链接倒默认的 shell。

那 bash 呢？那是 1987 年，一个叫做 Brian Fox 的家伙（狐狸大脑？汗-_-b）创作的，这个 shell 兼容 Bourne shell，算是对 Bourne shell 的改进版，于是叫做 Bourne Again Shell，简称 Bash。这个 shell 成为了 GNU 计划默认 shell，应用在绝大多数的类 UNIX 系统中。

除了 shell 以外，我们 Linux 下还有很多很好用的脚本语言，比如 perl 就是其中之一。Perl 是一种很有名的脚本语言，是由一个叫做 LarryWall 的人，在 1987 年设计的。“它也是 shell 编程么？”不，它不是 shell，它是一种脚本语言，shell 也是一种（或者说一类）脚本语言，但是脚本语言不一定就是 shell。越说越乱了是吧。

什么是 shell？是海鲜馆的扇贝？是汽车用的那种润滑油？都不是，shell 是一种人机交互的界面，之所以用 shell 这个词，是用来对立于 kernel，也就是我，内核。从根本上来说，计算机的运作就是用户（就像我的主人那样的人类）和我的交流。可是我们是不可能直接交流的，所以，就需要给 kernel 套上一个可以与人类交流的”壳”，也就是 shell。各种各样的 shell 虽然语法上会有些不同，但是都同样是解释用户的命令，然后向我汇报用户到底想干什么，之后我再去叫醒相应的软件或者我自己亲自去干活。Perl 呢，就不负责这种工作，而只是一个脚本语言，你可以用它编程。虽然 Perl 编出来的程序要不 C 语言编出来的程序在执行的时候耗费更多的 CPU——因为是解释型的嘛。但是编写的过程相对简单而快速，很适合写一些偶尔需要一些简单功能的脚本程序。写 Perl 脚本跟写 shell 脚本没什么区别，还是随便找个文本编辑器，按照 Perl 的语法写成一个脚本文件，在第一行写明#!/usr/bin/perl，说明这是一个 Perl 脚本，需要/usr/bin/perl 程序来解释。

首行的这个指名解释器的写法没什么神秘，只是为了简便而定义的一种写法。如果你不写这行也行，比如你在当前目录下有一个 ttttt 脚本，是个 perl 脚本，那么你完全可以运行/usr/bin/perl ./ttttt 来执行。如果写了#!/usr/bin/perl 这行，那么直接运行./ttttt 的时候，其实我也是先读取第一行，发现，哦，原来是一个需要 perl 程序执行的文件，然后我就会运行/usr/bin/perl ./ttttt 命令来执行。效果是一样的，只是省的你打了而已。如果你写#!/bin/cat 呢？语法上也是可以的，运行它的时候，，我还是先读取第一行，发现，哦，原来是一个需要 cat 程序执行的文件，然后我就会运行/bin/cat ./ttttt。效果？你说呢。　今天说说文泉驿。

文泉驿是个什么呢？用 Linux 的中国人不知道文泉驿那就白混了。这个文泉驿是一套大名鼎鼎的中文字体。各位要有兴趣，我说说，您听听，在想当初，很久很久以前（对你们人类来说其实也不算久啦，也就 6,7 年以前），那时候的 Linux 日趋完善，不少国内的开源同好们，都来尝试安装各种各样的 Linux。虽然硬件兼容的越来越多，应用程序的安装越来越便利，但中华儿女们安装了 Linux 之后无一例外的遇到了中文化的问题——没有一个合适的中文字体。要知道，Linux 是自由的，开源的，其中很多是免费的。那么自然不可能在免费的 Linux 中带有任何需要付费的字体。那 Linux 上自带的免费字体是从哪来的呢？都是世界各地热心的爱好者们贡献的。爱好者们自己创作一套字体，然后无偿贡献出来，给大家使用。因此，有很多优秀的，免费的英文字体可以用。可为什么其他国家有热心爱好者贡献字体，就没有热心的中国人贡献字体呢？是因为中国人懒？不是。是因为中国人自私？也不是。是因为中国人少？靠，更不是了。答案很简单，因为中国字多！人家做一套英文字体，总共也就二十六个字母的大写加小写，外带十个数字和一些标点符号，加在一起不超过一百个。一个人花一周时间就能做完了。汉字有多少？找本新华字典翻开前言看看——收录汉字一万个左右！这要是让一个人把这一万个汉字都做成电脑中的字体，还不得累吐了血阿。就算是只作常用汉字也得有四千多个，这还不算繁体字和各种数字，标点。那么那时候有没有中文字体呢？当然是有的，否则难道十年以前中国人都不用电脑？可是一般中文字体都是需要收费使用的——这个很合理吧，这么困难的东西，人家一个人（也可能几个人）费了好大劲作出来了，人家也得穿衣吃饭，养家糊口嘛。就算有几个不免费的中文字体，也是有很多问题，丢字阿，显示效果不好阿之类的。所以那个时候，Linux 的中文用户就只有忍受着质量差，总丢字的中文字体，或者偷偷的把其他系统下的付费字体复制过来用。说起来偷付费字体这件事，虽然不至于今天偷来用了，明天警察就来敲你家门，可毕竟不算光明正大不是。

直到 2004 年，中文字体的事情有了转机，带来转机的，是一位美国哈佛大学医学院的博士。这位仁兄倒不是来发扬国际主义精神的，他之所以关注中文字体，是因为他本身是个中国人——

房骞骞博士。不知道是不是因为他家房子被拆迁了，所以被逼无奈去美国当博士去了，反正他在接触到 Linux 的时候，发现缺少中文字体是个很头疼的事情。他也知道要是自己作出整套中文字体一定要吐血的，但是他还知道，积少成多，集腋成裘，众人拾柴火焰高，一个好汉三个帮，一个篱笆三个桩……（此处略去类似俗语若干）他利用网络为平台，创建了“文泉驿”项目，目的是要创作出一套高质量的，免费的中文字体。他自己动手编码，设计网站，简化设计字体的复杂程度，把汉子字体的设计搬到了网页上。这让每一个热心的志愿者都可以按照网站上的指导，完成一个个汉字字体的设计。就这样借助全球草根志愿者的力量，他开始了“万里长征”。经过数年的连续奋战，至今取得一项永留史册的硕果。这就是“文泉驿”汉字库，其中包括点阵和矢量字体。

今天，主人又安装了一个软件——dropbox。这家伙也不知道是干啥的，运行起来之后只是在主人的家目录下创建了一个 Dorpbox 目录，之后就不见动静了。

过了一会，主人往那个 Dropbox 目录里放了一个叫做小说.odt 的文件。我们还没反应过来，dropox 就马上凑过去，记录下这个文件的编辑日期，然后打开这个文件，把里面的内容都读取出来，打成一个一个的网络数据包，然后来我这里请求是用网卡。我黑明白他要干什么，毕竟这家伙是主人运行起来的合法良民，他要用网口，我也没有理由不让，于是点了点头。他很快跑到网络那里，把这些数据都从网口发送了出去。

过了一会，主人叫来 OO 老先生来编辑这个帐本文件，写下了一些文字：

绯红的天空中，一架飞机追逐着那一抹残阳去到地球的另一端。飞机渐渐变小，直到消失不见，夕阳也越发的红润，红的，像她的心，流着的泪。

朋友问，他走了？

恩。

没对你说什么？

他说……再见。

没有一句誓言?

……，再见，也是句誓言吧。

呵呵，看来主人是要开始写小说了。这东西我是不懂阿，毕竟我是个操作系统，只有理性的逻辑思维。根据我的逻辑思维判断，写小说这东西，还是挺费劲的，因为主人就写了这么几个字之后，就让 OO 老先生保存去了。OO 老先生自然是找到我，提出要使用硬盘，经过我的允许后，把这些东西，都存在了硬盘里那个 dropbox 目录下的小说.odt 文件里。刚存好，dropbox 又蹦出来了，一看，小说.odt 被修改过了，赶紧又用笔记录下修改时间，然后掏出里面刚刚写进去的

内容，又打成包，又从网口发送了出去。这家伙怎么让我联想起了电视剧里的国民党特务呢？他到底把主人写的这些东西发到哪里去了呢？我查了一下记录，他似乎每次都是把这些东西发送到同一个 ip 地址……

之后的几天，主人没有登录进来，不知道去哪疯去了，我们也乐得睡上几天觉。这一天主人终于回来开机了，我按照流程正常启动，看看自动启动的程序清单，比以前增加了 dropbox，还得去叫醒他，不知道主人到底想什么呢。dropbox 起床后，赶紧出门跑到网上去了，过了一会，报回来一大堆数据包，回来之后赶紧翻出他的小本本，查看这上面的记录。查明了记录后，他打开了那个小说.odt 文件，然后一边看着他抱回来的数据包，一边在文件中增加了些内容：

哇塞！这，这家伙竟然会写小说耶！他，他替主人往下编呢。等等，不对吧，天底下能有这

> 粉红色的纸页，淡淡的透出茉莉花的香气，娟秀的字体，给人很舒服的感觉， 像在听一个朋友轻声诉说。不经意间，林夕竟已被它感动了，她抬起头，笑笑对欧阳雯雪说："你写的真太好了，真羡慕你能有这样的文笔。再见也是句誓言…… 嗯， 最喜欢这句了。有时候，我也有想写东西的欲望，有时候也能写出些我自以为挺好的东西，甚至还想过当作家，呵呵，是不是有点傻？不过像你这样写出整整这么一 本，还都这么有意境的文章，真的不容易。"说着，林夕将记事本合上，递给欧阳雯雪。雯雪接过来说："呵呵，当作家也好啊。林夕嘛，注定是个有梦的人啊。其 实，每个人都有有话要说，有感要发的时候，写作是一种抒发的方式，每个人，只要有想法都能写出好的东西来。

种软件么？他抱回来那堆数据包是干什么的？常言道，内事不明问老婆，外事不明问 google。赶紧让狐狸妹妹找狗狗哥问问吧。

原来，这家伙是一个能够通过网络同步文件夹数据的软件。凡是放在那个 dropbox 目录下的东西，他都会备份一份到网络上，可不是到处乱备份阿，那样的话就该出 XX 门了。他是备份

到一个指定的专门给主人用的地方。因为主人之前去 dropbox 网站申请了帐户，所以会有一块空间供主人专用。dropbox 在首次启动的时候已经问过了主人用户名密码，所以可以直接去访问那块专用空间。这样，主人在我这太电脑上做的工作，保存的文件，可以很轻松的在另一太电脑上看到。就像这个小说.odt 文件，主人在我这里写了个开头，换到另一台同样装了 dropbox 的电脑上，打开那台电脑的那个 dropbox 目录，就能看到在我这里写进去的内容。他之所以总是记录文件被修改的时间，是为了确认哪台电脑上的数据是最新的，这样才好决定是用网上的内容更新掉当前机器上的内容，还是反过来。

这样的软件，确实是挺有用的，不过前提自然是需要有那个网站的支持。这年头都流行依赖网络的软件，就像超级牛力，没有那些软件源，超级牛力也不比一头牛有用多少。dropbox 每隔一会都要向我申请是用网络，好查看有没有新的内容需要更新，他总是很着急的样子，获得许可之后马上一百米跑的速度冲出网口，这一次也不例外，我刚说：用去吧。便听得耳边生风，呼的一下，dropbox 就飞出了网口，之后听外面嘣的一声，然后是稀里哗啦的动静，最后看见 dropbox 捂着鼻子会来了。我赶紧过去问：怎么了这是？装上英菲迪尼啦？对方逃逸了么？ dropbox 一脸苦像：哪阿，撞墙了。我无奈的摇摇头：你说你，也太不小心了，再着急走路也得看着点阿。这路你这两天也跑了 n 多趟了，应该挺熟了阿。dropbox 诉苦：路是没问题阿，我当然没走错，可是不知道什么时候，那突然立起一堵墙。我也感觉很蹊跷，只好安慰他说：没事，先去休息休息，过一阵子再去看看。怎么会有这么道德的人在这垒墙呢，肯定是非法建筑，估计过两天城管就得把他拆了。到时候你再去就行了。

话虽这么说，可是日子一天一天的过去了，那堵墙依然屹立不倒。dropbox 没法连接到网络，变得垂头丧气，郁郁寡欢，整天念叨着：强阿，墙阿，抢阿，哎……我还有什么用阿，我还有什么用阿。 哎，说来也是，不能连到那个网站，dropbox 确实一点作用也发挥不了了。主人似乎也很理解他，虽然没用了，虽然不再让他自动启动了，但始终没有删除这可怜的家伙，似乎，在

等待着有一天能够出现新的希望。

然而没有不透风的墙，也没有走不通的路。表面上走不过去，想想办法，总还是有绕过去的方法的。我们软件往互联网上发包，其实和你们人类送信，邮包裹，发快递是一样道理的。

我们要发数据给另一台电脑，这电脑可能是服务器，可能就是个个人电脑，不管对方是什么，首先要把数据打成数据包，就像你们寄信要把信纸叠好封在信封，发快递也要把东西装在盒子里封好一样。你不能抓一把黄豆往那一扔就愣让邮局给你寄到四川去。人家说了，要寄也行，您在每个黄豆上贴好邮票写好地址先。你没点微雕的功夫还真没戏。那么把数据打包是为了什么呢？为了写地址。信封上，得写上地址，邮局才能给你送，不写地址，就光在信封上写个:XXX 亲启，那这信只能找个坟头烧了。我们这数据包也一样，打好包，得在包头写上目标地址，也就是这个包要寄到那里。信的地址，无非就是 xx 省 xx 市 xx 区 xx 大街 xx 号。而我们数据包的目标地址呢，就是大家熟悉的对方的 IP 地址： xxx.xxx.xxx.xxx

可是 IP 地址并不易于被人类记忆。比如让狐狸妹妹访问搜狐的主页，按说狐狸妹妹必须写一封信，上面写着：亲爱的搜狐你好，我想获取你主页的信息。我的地址是 xxx.xxx.xxx.xxx，请发过来。(这个地址就是主人电脑的 IP 地址。) 然后把这封信打好包，在信封上写上目标地址：61.135.179.160 然后才能从网口寄出去。搜狐那边看见这封信，就把主页的数据打成数据包——多半不是一个包，而是很多个。然后每个包都贴上狐狸妹妹发过去的地址，寄到主人的电脑中，狐狸再把这些包拆开，翻译成漂亮的网页，显示给主人看。可是主人向狐狸发命令的时候，决不会告诉狐狸：我要看那个 61.135.179.160 那个网站。因为他肯定记不住这么多数字，他只能是告诉狐狸，我要看那个 www.sohu.com 那个网站。这个 www.sohu.com 是什么呢？是域名。

域名是一个用来给人类看的，易于记忆的名字。什么 www.sohu.com,www.google.com,www.baidu.com.sb，都是，就不用我再举例了吧。用域名，这就好记多了。就像我跟你说北京市宣武区宣武门西大街 57 号，你不知道是哪，我跟你说新华社，你就明白了。

这么一来人类倒是爽了，可我们软件不能在数据包上写上域名阿。你写信光在信封上写：劲丰汽车维修公司。哪给你找去阿！你还是得写：贵州省 贵阳市 南明区 花溪大道北段 262 号。这才能寄到。同样，往网络上发的数据包，必须得写 IP 地址，才能发到目的地。于是就需要一个把域名翻译成 IP 地址的角色，这个角色就是 DNS——域名解析系统。

DNS,就跟 114 查号台似的，只不过查的不是电话号码，而是域名对应的 IP 地址。在网络连接里设置的 DNS 服务器，就是干这个用的。有了 DNS，狐狸妹妹再要访问网站，过程就是这个样子的：主人说，要访问 www.sohu.com，然后狐狸妹妹先根据网络连接里设置的 DNS 地址，比如是 8.8.8.8 吧，写一封信，内容是：请告诉我 www.sohu.com 的地址。然后信封上写好收信人地址 8.8.8.8，就从网口寄出去了。之后会得到 8.8.8.8 的回信，内容大约是：www.sohu.com 的地址是 61.135.179.160。之后，狐狸妹妹再往 61.135.179.160 发信说需要获取主页信息等等，就跟咱之前说的一样了。这里，DNS 就扮演了一个专业指路的角色，每个人要出门，去什么有名的地方，都要先问问 DNS，笑脸相迎的过去问上一句：呵呵，大爷，劳驾打听一下，火葬场怎么走？

DNS 这个专业指路的多数情况下是靠普的，可也有少数情况，DNS 会被黑帮利用。比如某个饭馆最近没交保护费，于是就有黑帮老大，戴着墨镜挂着链子身上还纹着带鱼的那种，找到 DNS，跟他说：告诉你，要有人问你那“坠落之箱”饭馆怎么走，不许告诉他！听见没有！ DNS 无奈，只得点头称是。再有慕名而来的人，问 DNS：听说有个坠落之箱饭馆，那里的筒布鸡不错，您知道在哪么？ DNS 只好说：这个……不知道，没这个地方。 或者就是瞎指，人家站在天安门广场问：这个前门大街怎么走阿？DNS 给人往北一指：那边，出了村过了河就到了。 再赶上那哥们实诚点，回头前门大街没看见，倒看见松花江了。也有的时候，黑帮们不在 DNS 上下功夫，而是直接在路上劫着你。还比如那个坠落之箱饭馆，你问 DNS 怎么走，DNS 告诉你：往东走，在那个点七零路上就是了。这个地址实际没错，但是你走过去会发现，点七零路上站了一排

黑西服黑墨镜的人，整的跟黑客帝国似的，你要过去，他告诉你前方施工，不许往前走了，省得溅一身血。（这是施工么这个……）于是你只好无功而返。

针对以上的情况，该怎么办呢？想想，你每次出门都要问路么？当然不是了，多数情况下你可以记住你要去的地方的地址。我们这里也一样，我们 **Linux** 系统也是可以在本地查询域名对应的地址的。这些内容被记录在**/etc/hosts** 文件中。当需要查域名的时候，我们会首先翻开这个文件，看这里面有没有记录，如果有的话就不必去问 **DNS** 了，直接在文件中查到 **IP** 就好了。如果这个文件里没有相应的记录我们才去问 **DNS**。这样，上面的情况就好解决了，**DNS** 他不告诉你没关系，你自己记录下你要去的地方的地址，写在 **hosts** 文件中，自己去就好了。如果遇到第二种，黑帮拦路的情况，也有办法。一般大一点的地方，可能都不是一个地址。比如那个坠落之箱饭馆，走点七零路可以到，但不是唯一的路，其实从点七一路也可以过去，到达那饭馆的后门。黑帮只封了点零七路不要紧，绕过去就是了。可是你要去问 **DNS**，他肯定会告诉你走点七零路，因为点零七路是正门，而且他也不知道那条路被封了嘛。所以就不问 **DNS**，自己在 **hosts** 文件中记录下：去坠落之箱饭馆，走点七一路。就可以了。

话说这域名并不是网络上的每一台计算机都有的，只是一些网站，或者各种提供公共服务的计算机才有域名，而一般个人的计算机只有个 IP 地址就够了。就好比你开个饭馆，总得起个名字，叫什么八千里烤饼，小酸梨泡菜之类的。但是如果你只是有个房子自己住，自然就不用起名字了，起了也没人知道。你走大街上跟人家问：同志，请问全聚德怎么走，人家就能告诉你，可你要是问：同志，我们家怎么走。估计人家就直接把你送精神病院去了。这个 DNS 解析的过程——也就是这个问路的过程，很多时候也不是一次完成的。比如你的电脑在一个公司的内网里，你要访问你们公司的邮件服务器，比如是 email.meipu.com 吧。你的电脑就会问你们公司的 DNS 服务器，这个 DNS 就可以告诉你。可是你如果要访问 www.google.com 呢？你们公司的服务器不可能记录着全世界的域名和 IP 地址吧。很有可能这个 www.google.com 的地址，你公司的 DNS 并不知道，那怎么办？很简单，他再去问别人呗。DNS 跟黑社会一样，也是有等级的。比如公司的网络是电信的，那么公司的 DNS 不知道 www.google.com 的 ip 是什么，就去问电信的 DNS，电信的 DNS 多半就知道了。如果还不知道，就再往上一级问，直到问到根服务器。这个根服务器，分布在世界各地，一共有 13 个“根服务器系统”。之所以说根服务器系统，是因为他们并不只是 13 台电脑而已，而是在每一个地点，有数十台到上百台的服务器，一起组成了一个根服务器系统。目的自然是为了提高安全性和性能。这 13 个服务器系统的大致地点是公开的，比如美国太空总署(NASA) 有 1 台，这个是公开的，但是具体在位置哪里，是在 A 座还是 B 座，1 楼还是 5 楼，厕所对面还是车库底下，都不知道。这 13 个 DNS 的根服务系统，管理着全世界的域名和 IP 地址的对应关系，可以说是个世界网络域名地址超级无敌大黄页 ~。

除了 DNS，设置网络链接的时候，还有一个需要设置的服务器就是 gateway——网关了。网关又是干什么用的呢？咱还哪寄信说事阿，话说你的信在 DNS 的帮助下写好了地址，现在要寄出去了。往哪寄呢？这得看收信人的地址是那里。比如收信人就住你家对门，那就好办了，不用麻烦邮递员叔叔，也可以省下几张邮票了，直接塞进他家门缝就行了。就算再远一点，只要不

出本小区，基本都可以这样解决。如果是本市的，那就得出门，出了小区口向右拐走 100 米看见一个信筒子，把信塞进去就好了。之后会有邮递员叔叔来取信，统一送回本市邮局，然后再送到目的地。如果是外地的信呢，比如你从北京，要寄信到上海市南京路 xx 号。那么北京市邮政局取回你的信，发现是寄倒上海的，那就找人把信送给上海市邮政局，之后就不管了。再由上海邮局把信送到具体的地点。这个路由，就相当于邮局的角色。一个数据包打好了，发给谁呢？有人说了，你都知道地址了，直接送给那个 IP 地址的机器呀？我问你，你从北京往四川写信也是自己亲自送去呀？累死你。刚才说了，也就距离比较近的，比如在一个小区内，才可以直接送过去。当然我们软件是不计较物理距离的，我们关心的是对方那个地址是不是根我们这台计算机的地址在一个网段。比如说 IP 是 192.168.1.34，子网掩码是 255.255.255.0 的这个地址就和 IP 是 192.168.1.78，子网掩码是 255.255.255.0 的地址在同一个网段，也就相当于在一个小区内。这样的地址，就可以直接把数据包发给对方计算机。那如果不是在同一个网段呢？这时候就该路由出场了。凡是目标地址不在一个网断，你不知道该往哪发的数据包，就交给路由，就像你把信交给邮局一样。你的路由不一定真的知道这个目标地址在哪，就像北京邮局不一定知道上海市的南京路在哪一样，但是没关系，你的路由会把这封信交给其他的他认识的路由，就像北京邮局把信交给上海邮局。

一个数据包经过了 N 个路由的转发，终于到达了目标计算机，然后发生了什么呢？

一封信经过邮局来到了你家。如果是挂号信，肯定是邮递员敲开你家的门，问：“郭海春在么？您的挂号信。”如果是平信的话，信会被扔进你家的信箱，然后你家有人开邮箱把信取来，看看是谁的，如果是寄给自己的，拆开看；如果是寄给老婆的，拆开看；如果是寄给儿子的，拆开看。（咳咳，个人隐私要尊重啊。）呵呵，好吧，准确的说以上做法不正确，应该是，写给谁的，谁拆开看。这里要说的就是这个收信人的问题，信封上除了要写明地址，还需要写明收信人，毕竟一个地址指示的那一间房子里，多数情况下有不止一个人。同样，一个数据包上虽然写着目标机的 IP 地址，但是拥有那个 IP 的计算机上也肯定运行着不止一个程序，那么这个数据包到了这个计算机中，应该给谁呢？是狐狸妹妹的，还是皮筋的呢？（gedit 你就不要闹了，你又不是网络程序，肯定不是给你的，乖~啊）所以，数据包也像信件一样，要写收信人，不过不是写程序的名字，而是写端口号。

所谓端口，就是一个系统给所有需要使用网络进行通信的程序分配的一个专门的号码，用处就是用来区分收到的数据包属于哪个程序的。比如狐狸要上网，就要来向我申请，我就会给他一个端口号，比如是 38246 吧，然后狐狸就用这个端口进行通信了。要看网页，比如要访问 www.google.com.hk，咱说了这个访问 DNS 的流程了，最终获得了这个域名的 IP：64.233.189.104 ，然后就打个数据包，写上 64.233.189.104，80 端口收。就像你在信封上写北京市丰台区造假村 28 号村长收，一样。并且还要在数据包里写明自己的 IP 的自己的端口号：58246。信封上不是也得写上寄信人地址么。这样，对方哪个计算机收到数据包后，做出响应，把要求的数据打成数据包，写好地址和端口，发送回来。数据包到达我这台电脑，我首先要看，是不是给我的，因为有时候发到我这里的数据包不一定真的就是给我的。一对 IP，果然，没错，是发给我这的。然后再看端口号，是 58246，一查，哦，这是我刚分给狐狸妹妹的，那就把这个数据包交给她了。可不能给错人，这倒不是为了什么个人隐私，而是因为不同程序的数据包内容是不一样的格式的，

你把狐狸的数据包给了皮筋，他也看不懂，肯定会认为数据包在邮递过程中损坏了。

这个端口也不是乱分配的，也是有一定规矩的。比如刚才说的，狐狸要看网页，怎么就知道给对方的 80 端口发呢？因为这个 80 端口是固定用于提供 http 服务的。端口号从 0 到 65535，其中从 0 到 1023，是公认端口。这个号段的端口都有固定的用途，比如看网页，就要访问对方的 80 端口；要连接对方的 ftp，就要给对方的 21 端口发请求；要连接对方的 ssh，就要找 22 端口，等等。然后从 1024 到 49151 是注册端口，这些端口使用起来相对随意一些，不过一般也都是用来提供服务。比如 8080 端口，经常被用来提供 http 代理服务；3389 经常被用来入侵查皮。注册端口没有严格的规定，只是各软件自行使用，但是一般都是用来提供服务。最后就是动态或者私有端口，从 49152 到 65535。这些端口就是随便给任何需要连接网络的软件使用了，狐狸妹妹要上网，分一个；奔流要下载，再分一个。动态分配，用完回收，二次利用，可持续发展。

经过几番波折，dropbox 终于正常工作了。主人的文件可以再两台电脑之间不知不觉的同步，感觉相当先进，这就是现代化的便捷啊。如今的计算机已经不是一个人在战斗，不是一台机在工作，有了互联网，就有了一切。现在不是都在炒云计算么？

说起来，我作为一个操作系统，还真不大知道这个云计算到底是个什么概念，或许，他也仅仅是个概念而已。云计算的核心理念就是简化终端机，把所用的东西都放到网络上来进行。过去传统的互联网，根本上来说只是使电脑之间建立物理的数据连接，然后各个计算机（大的，小的，服务器，个人电脑等等）之间，互相共享存储的数据而已。网页，就是一个 html 文件，上网就是网站的服务器根据一个协议，把存在他那里的 html 文件共享给你阅读，你想想是不是这样？其他的什么视频啊，音乐啊，不管是在线的还是打包下载的，说到底都是存储空间的共享，数据文件的共享。然而一台电脑的资源不只是存储资源而已，还包括内存，CPU，声卡，等等各种硬件，这些东西能不能放在网上共享呢？能。

“共享 CPU？我把老 CPU 放到淘宝上卖了算共享么？”不算，你那叫卖，不叫共享，送给我才叫共享——哈哈，这是开玩笑。我说的共享不是这样的。想象一下这种情况：你的电脑是奔腾 3，主频 1GHz，上面装了 Blender 软件来制作 3D 动画。（拿这么个电脑做 3D，不知道怎么想的-_-b）场景什么的都建好了，准备开始渲染，可是您想想这配置，也能知道这速度得多慢，一渲染，得 4 天才能完成。这肯定等不了，是吧，那怎么办？旁边还有一台电脑，8 核 CPU、8G 内存、8T 硬盘、8 显卡交火，80 寸大屏放了 8 个，IPv8 的光纤接了 8 条……总之很牛，我们暂时不去追究为什么不用这台做 3D，假设情况就是这样了，怎么能够利用其这台大三八的电脑呢？可以在上面装上 blender，然后把文件拷贝到那台电脑再渲染，肯定很快，不过还是有些麻烦。最好是我这台小三电脑（小小的奔三的电脑）直接能够使用那台三八电脑的 CPU 去计算，那就好了。类似这样的情况，可能不？可能。

话说有个发行版叫做贱兔，和我们笨兔一样是个很优秀的发行版。他最大的特点就是所有软件都是从源码编译的。您听了别害怕，虽然都是编译的，但是不用您亲自去 configure,make, make install。他的编译都是全自动的，他那里也有一个像超级牛力一样的软件管理器，叫做一墨迹(emerge)。别看这名字墨迹，干活可不墨迹。墨迹跟超级牛力一样，负责上网下载软件，不过超级牛力下载来的都是二进制的，直接就可以用的软件，而墨迹下载来的都是软件的源代码，下载来之后墨迹会叫来 tar 啊，gcc 啊，make 啊之类的同志们，按照那个软件的编译方法，自动编译软件，最后安装到系统中。这样，每一个软件都是针对这台电脑进行过优化的，所以运行起来速度会比较快。然而编译时一个很耗费 CPU 的工作，也难怪有人觉得这么装软件很墨迹了。正因为如此，贱兔就提供了一个软件叫做 distcc，这个软件可以让电脑 A 用电脑 B 上的 CPU 来编译软件，然后装到电脑 A 上，听起来很神奇吧。当然，电脑 A，B 需要都安装这个软件，并且有相同的 Gcc 版本才行。这样就实现的在两台电脑之间共享运算资源。

回来再说这个云，云的根本意图就是让网络可以提供除了存储资源外的其他资源，比如运算。以后，你的电脑也许就不需要多么强大的 CPU 了，一切的一切都放到网络上去运算。其实现在的网络中，已经不知不觉的有了这样的服务。比如狗狗大哥提供的狗狗文档，就是这么个东西。你的计算机上不用安装 office，把文件放到网上去，用网页来编辑。这不就是利用狗狗哥的服务器运算资源来进行文档处理么。还有各种各样的 WebOS，打开一个网页，里面就什么都有了，游戏也好，文档处理也好，音乐也好，都有，可谓是：页中自由黄金屋，页中自有千钟粟。浏览器就是操作系统的概念，已经被狗狗进一步诠释成了一个操作系统，叫做 ChromeOS。你看着名字，跟他家浏览器一样，装上这个系统就会发现，这系统啥也没有，就是一个浏览器，可是这浏览器里面干啥都行。处理文档？有狗狗文档，在线处理。聊天？gtalk 也好，msn 也好，qq 也好，都有网页版啊。看片？除了为了高清，现在有几个下载来看的，不都是在什么地瓜啦，油兔啦的网站上看么。游戏？网页版的网游，够了。音乐？视频都在线了，还怕没音乐么。总之，

日常应用都差不多可以支持，只是一些专业性质的工作，比如视频编辑啦，编程啦，这些个还没发在网上进行。但相信不久的未来也会出现类似的网络应用的，现在很多在线共享照片的网站不就提供一些简单的照片处理功能么。

今天狐狸休假，皮筋赖床，内存里一干人等全都不在，只有 OO 老先生一个人响应主人号召出来工作。主人嚼里啪啦絮絮叨叨的跟老先生说着什么，老先生也不答腔，只是尽职尽责的把主人说的话记录下来。

“那是 2008 年了，那时候公司的项目做得很不爽，好好的做到一半就被上边砍了。理由大约是干了一年了没有挣来钱。哎，这作项目又不是种麦子，一年就能有收成。之前您也没接触过这个行业，从什么也不懂就开始做，一年就非得见到利润？怎么可能。于是项目被砍，整天上班没有事情作，就开始上网，动看看，西转转，学习学习先进的 Linux 编程技术。学烦了，就去找些杂七杂八的东西来看，反正领导也不管，倒是也很自在。”看来有个好领导很重要阿，不知道我这个内核在别人眼里，或者说别的软件眼里，算不算是个好领导呢？

“那时候看到了天涯上的一个帖子，《如果这是宋史》，现在好像已经成书了。自从明月写了明朝那些事儿以后，很多人开始用通俗的语言讲历史，倒也写的很有趣。上学的时候对历史不感兴趣，老师说，历史好阿，里面都是故事，好记，比地理好多了。可惜老师并没有把故事讲得生动，于是我的成绩也就一般。但当看到《宋史》那帖子之后，就觉得这种感觉不错（虽然明朝那些事儿名气更大，但我却是先看到的宋史），枯燥的东西，谈笑间变得跃然纸上，叫人印象深刻。于是跟着帖子一篇篇的看，看了没两天，杯具了——网被封了……”哎，不知道网络不通对主人意味着什么，反正对于我，网络断了，就意味这可以收拾收拾回硬盘睡觉了。网络一断，超级牛力，狐狸，皮筋，全都歇菜，星爷不能在线查辞，Totem 不能在线视频，连个桌面上的天气预报都显示不了了，我这系统还活个什么劲阿。

“网络被封当然不是因为我看宋史，而是据说有某个同志上网看了些不该看的东西，说了些不该说的话，于是大家就都别上了。离开网络的日子很清苦，然而好在邮箱还可以用，于是让我女朋友把宋史的网页复制到邮件里发过来，就这样继续跟帖。在此感谢我那时候的女友对我的不务正业所给予的大力支持。^_^ 宋史那楼主要知道有我这么坚强的粉丝估计得感动死，呵呵。后

来看多了，觉得光看不过瘾了，就突发奇想：我也可以把枯燥的东西，生动的写出来嘛！”

Oo 老先生推推眼镜，继续记录着。

“历史我是不会写的，一来没学好，再说写的人太多了，不新鲜。要写，总得写点自己了解的东西不是。那时候家里电脑上的主要操作系统早已变成了 Linux，WindowsXP 只在需要和同志们联魔兽的时候才会出场。想来最早接触 Linux 应该是 2004 年左右了，那时候在笔记本上装了红旗 Linux，倒是挺顺利，但也只是装装而已，很少真的用。后来工作了，在外边住，为强迫自己学习,笔记本上只装了一个系统——Mandrake Linux 用了一阵子,还是很无奈的被换掉了。之后体验了 Magic Linux，国内做得，确实很方便，装好之后很多东西都配置好了，可是毕竟有些小众，有了问题，不好解决。再然后，就是 Gentoo，这主要是在工作中做了一次 LFS，学到了不少东西，觉得还不算难。既然 LFS 都搞定了，Gentoo 应该也不在话下。装了之后，用了很长时间，在这过程中，渐渐的习惯了用 Linux 来生活。Gentoo 的运行效率，无缝升级，以及辛苦的配置过程都使得我没有什么理由换掉他。也就是那个时候，我有了现在的 ID，并想以后就一直使用这个系统。”为啥还没看到我们 Ubuntu 出场……

“然而我总归是个喜欢尝鲜的人，随着 Ubuntu 的名气越来越大，总想装来试试。等到我的笔记本已经实在跟不上历史的潮流，不得不换了一台配置好一点的台式机的时候，终于有机会尝试一下新的系统了。gentoo 初期的配置实在很麻烦，不想再重复一边了，于是就在新机器上装了 Ubuntu，一开始还是 7.10 呢，就已经很方便，装上之后不需要太多的配置（跟 Gentoo 比），速度，由于电脑比较强大，速度忽略不计。后来重装为 8.04，没太大变化，一直用下去。”哈哈，看见我出场啦。

“那是 11 月 1 号，一个周六，晚上看着我的 ubuntu 8.04，忽然想到，对，就写它吧。用的时间长了，有感情，有了解，就把自己当作这个操作系统，用第一人称的角度，去讲述操作系统自己的故事。写了一点，贴到了论坛上，似乎还不错，很多网友用各种表情表示了支持。于是

得到动力，继续写下去。写的也没什么章法，想到哪说哪，以基本知识和概念为主，读者竟也一点点多起来。甚至后来还有论坛上的童鞋帮我排版整理成 pdf 版，以至发展为创建了专门的网页，而且还不止一个，真的是受宠若惊，非常感谢这些童鞋们的支持。”童鞋？在我的数据库里童鞋似乎是一种用于保护人类幼崽双脚以便于行动的物品，为啥主人要感谢他们呢？

“由于要上班，而且单位离家里很远，那时候从家到单位需要将近 2 小时的车程，现在地铁通了，也得一个半小时，所以能够静下心来写字的时间几乎就没有了，只能够周末写一写，不像人家专业爬格子的，每天都在创作。每一段也就一千字左右，开始的时候少些，7,8 百，后来写熟了，能有一千多。但一周才写一千多字，实在是供应不上爱好者们，也不之以这样蜗牛的速度，要写到什么时候。记得有人在坛子里问过什么时候写退房，呵呵，那时我说，争取写到下一个 LTS，算是给自己的目标吧。现在 10.04 已经发布，总算是没有半途而废。其实我机器上的 8.04 早就‘退房’了，只是还不打算让文中的笨兔兔搬出去，就这样继续住着吧。”没看懂，反正，好像，没我什么事哈。

狐狸妹妹算是我这里工作最辛苦，出镜率最高的人了。但是一直也没有好好介绍过她，今天就讲讲她的身世吧。

当年，有点软公司的 Windows 95 获得成功的那个年代，网络网络浏览器这个东西还不是很被人重视。那年头互联网还不是很普及，电脑基本上好都是单机应用。有点软公司的老大，盖子大叔觉得，网络这个东西是没太大前途的，不就是能在电脑间传递一些数据么？也就看看新闻，发发邮件什么的，普通家庭用户是不会关注这些的。普通用户还是更多的会被漂亮强大的软件吸引，所以还是要专注于本机的应用。那时候比较有名的浏览器叫做 NetScape，创造他的公司就叫做 NetScape 公司。本来两家相安无事，一个做系统，一个做浏览器，各干个的。然而随着互联网的飞速发展，盖子大叔发现，自己的判断好像有点问题，这互联网是越来越厉害阿，看来新一轮的圈地运动就要开始了。于是，赶紧想办法培养自己的浏览器，拉拢网络浏览器的用户，这就是 IE。Windows 95 后来的版本里面就有 IE，只是作为一种不常用的专业工具，深深的埋藏在了系统目录下，并没有在开始菜单里出现。现在赶紧找来稚嫩的 IE ，努力培养，教会他各种本领，以便成为能够面向普通用户的浏览器。

但是市场这个东西向来是只有第一，没有第二。谁最先抢占市场，谁就成功。NetScape 的浏览器已经在市场上获得了很好的口碑和很大的占有率。区区一个 IE，如果没有绝对的优势是不可能抢回 NetScape 的市场的。要想混出点名堂来，总得有真本事。要么浏览速度快，要么占用资源少，要么操作方便，人性化，有点软公司的 IE 有什么绝对优势么？很遗憾，个方面都没有明显的优势。IE 的老爹有点软这会改了有点急了，天天教育 IE：你看看你哥哥，Windows，都是你爹我调教出来的，人家怎么那么厉害，你怎么这么不争气呢？我们还想着让你跟那 NetScape 火拼一下，好歹打个平手也好，可你呢？干什么什么不行。你说说，你自己说你比那个 NetScape 哪点强了？除了你老爸比他老爸强多了，还哪点强了？除了你有一哥，你还比 NetScape 哪点强了？——等等，对呀！IE 的优势就是有个有钱的老爸，还有个强大的哥！你见过卖汽车送

轮胎的吧？你见过卖房子送车位的吧？那我卖个 Windows 送个 IE 不算过分吧？得!就这么着了。

于是，从 Windows 98 开始，Windows 就开始带着他的小弟 IE 一同出现在各个电脑的显示器上了。这一下给 NetScape 打击可相当大，IE 虽然哪里都不比 NetScape 强多少，可是哪里也不比 NetScape 弱多少。Windows 的市场占有量那么大，系统里既然自带了一个和 NetScape 差不多的软件，自然也就很少会有人花钱去另买 NetScape 了。（软件是要买的，不是 5 块一张的那种）于是，自此以后，NetScape 的市场份额是与日具减，直到整个公司被人收购。

NetScape 浏览器被有大哥和老爸撑腰的 IE 打败了，并且一败涂地。原本自己的地盘被 IE 抢去了，NetScape 浏览器的老爸 NetScpe 公司被美国在线收购，一切似乎应该就这样结束了。然而在离开儿子的前夕，老爹将 NetScape 叫倒了跟前，语重心长的对他说：孩子，不是你不优秀，是你的对手太狡猾。以后我不能再照顾你了，剩下的路你要自己去走。我送给你一件东西，希望凭着他，你能重拾往日的辉煌。这个东西，就叫做“开源”。你拿着它，去找一个叫做社区的地方，在那里，会有人将你培养成最优秀的浏览器。NetScape 收下了礼物，并默默的点了点头。

同时，NetScape 的老爹还叫来了跟随自己多年的亲信，让他们成立了一个组织，专门负责培养 NetScape，这个组织，叫做 Mozilla.org 。老爹从新东家那里获得了许可，每年拨一定的资金来支持 Mozilla.org 组织。于是 NetScape 在 Mozilla.org 组织的帮助下，开始了自己的试炼之途。NetScape 很快找到了社区，并拿出了父亲给他的开源。这个东西将 NetScape 的全身照亮，可以看清楚身体里面每一个细小的结构。社区中隐藏着各种高手，纷纷来找 NetScape，指出他的不足，帮助他变得更加强大。由于有了开源的帮助，社区高手们可以很容易的发现 NetScape 身体中的问题，并由 Mozilla.org 组织加以调整。数年之后，NetScape 身体中原有的代码几乎全部被替换了一遍。此时的他已经不再是那个简单的浏览器 NetScape。而是一个包含了浏览器，电子邮件，IRC，网页编译于一体的套件。他们给它起了新的名字，叫做 Mozilla Application Suite，简称 Mozilla。在社区的帮助下，Mozilla 终于开始拿起自己的武器，去夺回自己

曾经失去的一切。

然而事情的发展，总是不乏转折。有点软公司看出的现在的 Mozilla 可能会是将来 IE 最强劲的对手，于是跟当年收购 NetScape 的美国在线谈判，并最终说服美国在线，停止了对 Mozilla.org 组织的资助并将其解散。这一下可算是断了 Mozilla 的后路。Mozilla 就这么结束了么？没有，他们忽略了一个重要的东西——开源。有了这个东西，就可以号召起最广大的社区力量。不久，Mozilla 基金会成立了，这是一个非盈利性质的组织，他的目的，就是继续培养 Mozilla，让他成为一个优秀的浏览器。然而或许是求胜心切，欲速不达，Mozilla 虽然很用功，学了很多的东西，但这反而使得他变得有些臃肿，虽然强大，但是缺少了灵活性。这反而让他在夺回市场的战斗中处于不利地位。这个时候，一个少女出现在人们的视线里。她天真可爱，活泼机灵，总是看着 Mozilla 练功，时不时的还跟着比划两下。这一切，被戴夫·海厄特与布雷克·罗斯看在眼里。他们意识到，Mozilla 太过臃肿，或许这将导致 Mozilla 的失败，但是又不能太过于大胆的砍掉 Mozilla 的各项功能，毕竟未来的事情谁也说不清楚，谁能保证砍掉之后就一定能成功呢？于是他们把目光放到了这个少女身上，他们按照培养 Mozilla 的方式来培养她，但是只专注的教给她浏览器的功夫，让她代替 NetScape 回到最初的起点——做一个优秀的浏览器，仅仅做一个优秀的浏览器。同时，也为了让她有可能能够胜任更多的事情，为她预留的很多接口，这样就可以在需要扩展功能的时候通过安装扩展插件的方法来实现各种功能。很快，她通过了训练，成为了一个合格的浏览器，并且开始试着开疆扩图，要打出自己的一片天地。她就像一只鸟儿般灵巧，她代替 NetScape 在烈火中重生，于是人们给他起名字，叫做“phoenix”——凤凰。人们也把它亲切的叫做凤妹。可是谁都要长大，如果以后她成长了，岂不是要变成“凤姐”？！所以为了避免未来的尴尬，人们给她换了很多名字，最终确定，叫做 Mozilla Firefox，这 Firefox 原本是小熊猫的意思，不过或许是考虑到广大不明真相的群众的第一反应，采用了一只狐狸作为图标，也就是今天的狐狸妹妹。

当狐狸妹妹走上硝烟弥漫的浏览器战场的时候，IE 还是当年那个 IE，但战场，已注定不再是当年那个战场。

随着当年 NetScape 的销声匿迹，IE 无疑成为了浏览器战场上一家独大的胜利者。胜利后的 IE，开始不思进取，整日沉浸在胜利的喜悦中。虽然他也在不断更新自己的版本，但是跟他的老哥 Windows 一样，他也开始变得缓慢而臃肿，并且很少考虑使用者的感受。反倒是很多其他厂商开发出来的是用 IE 内核的浏览器外壳很受欢迎。像什么 Maxthon 也就是傲游，还有那个疼殉公司的套套浏览器，还有世界痔疮，等等。这些说起来叫浏览器，实际上他们都是调用 IE 来渲染网页，他们做的只是一个外壳而已。然而就是这些是用方便，功能强大的外壳，比 IE 更加受使用者欢迎。这从一个侧面说明了 IE 那时候的易用性确实不能够令使用者满意。同时，IE 也并不排斥这些外壳型浏览器，因为这些外壳都是基于他的嘛，没了 IE 这些所谓的浏览器统统都得挂掉，所以这并不影响 IE 的主导地位。而时候狐狸妹妹的出世，却让用户们眼前一亮，他们忽然意识到，原来世界上，还有 IE 之外的浏览器。更何况，Firefox 是开源的，免费的软件，下载下来尝试一下又没有花费，何乐而不为呢？

狐狸的这种免费模式，渐渐获得了成功，于是，更多的后来者，纷纷涌现出来。

先是来自挪威的 Opera 小姐。这姐姐速度快，易用性好。尤其在易用性上，下了不少功夫。开了很多网页还没看完，忽然有事情出去需要关电脑？直接关闭 opera，没问题，下次再打开时还是这些网页。先在的狐狸妹妹也会这一招了，不过是跟 Opera 学的。打开一个空白页，再点书签，选择要去的网站？太麻烦了，Opera 直接整个快速拨号，空白页上直接列出最常去的 9 个地方。（当然要事先设置好）一点就进去了。后来狐狸妹妹也有扩展可以实现这样的功能，自然也是跟 Opera 学的。(这也说明了狐狸妹妹 + 扩展的强大性，想要什么新功能，不用等着升级新版本，装扩展都能实现。)Opera 虽然也免费，但是不开源，不像狐狸妹妹，一切都是开放的。所以，虽然 Opera 也有扩展插件，但只能由 Opera 的公司自己开发，其他人是不能参与的。而

狐狸妹妹的一切都是开放的，他的扩展都是全世界的爱好者们开发的，种类多不说，开发速度也快。在这点上，opera 算是稍逊一筹。

再有，就是狗狗公司的 Chrome 浏览器了。这家伙最大的特点就一个字——不走寻常路。别的浏览器都有地址栏，搜索栏，菜单，底下有状态栏。Chrome 觉得这么多乱七八糟的东西太麻烦，搜索栏跟地址栏整到一起，菜单栏干掉，状态栏取消。状态改成在网页底部动态显示。于是一个很有创新意味的界面诞生了。Chrome 以他与众不同的面貌登山了浏览器的战场。出了界面简洁以外，Chrome 还有个特点就是速度快，那真是刷刷的。通过查看进程可以发现，狐狸妹妹和 Opera 都是一个进程，也就是干活的时候只有一个狐狸妹妹或者 Opera 姐姐在忙活。而 Chrome 这家伙，是一堆进程！一起动，这家伙跑进内存后，先把自己复制 4 分，也就是有四个 Chrome 在内存里忙活，以后，每开一个网页，内存里就会多一个 Chrome。这也许就是他速度快的奥秘吧。这样的好处就是万一哪个 Chrome 挂掉了，顶多一个网页挂掉，其他的不受影响。坏处呢？自然就是导致工作间里太乱拉！！

说什么来什么，前两天刚说了 Chrome，这不今天主人就让狐狸妹妹去下载了一个 chrome 的 deb 包给装上了。我也有幸见到了这帮小子工作的场景。这家伙果然善于分身，他刚进内存，我只觉得眼前一花（本来他就挺花哨的），一下子变成好几个，并且麻利的向图形界面申请窗口，同时绘制默认页面，同时准备好显示界面上的各种元素，像按钮阿，图标阿之类的。这些事情是同时进行的，怪不得用户会感觉他快。狐狸妹妹就比较本分，一步一步进行。先向图形部门申请绘制窗口，窗口批下来之后再在窗口里绘制好各种文字，菜单，按钮之类的。这之后再根据用户设置的主页，上网申请数据，数据来了再显示给用户。有插件的话，在这之前还要加载各种插件，难怪让人觉得有点慢呢。

Chrome 们似乎看出了我再想什么，他们其中一个跟我说：别以为我们只是靠并行处理提高速度哦，我们渲染网页的基本功也不差。一边说着，一边已经打开了默认的主页——自然是 www.google.com 了。我一看他们渲染网页的身手，立刻看出来，他们是 Webkit 派的。

浏览器渲染网页的方法有很多种，各家实现的方式不一样，于是就形成了几大派系，就像什么武当派，昆仑派似的，专业点讲就是不同的内核。Trident 派，占据浏览器大半的江山，该派创派祖师就是 IE4，有一个徒弟 IE5，徒孙就是 IE6，再下一代就是 IE7……总之就 IE 他们家的派。Gecko 派，祖师爷就是我们说过的 NetScape，后来的故事大家都知道了，这一派现在由狐狸妹妹继承。Presto 派，创始人是挪威的 Opera7。这派的功夫特点就是快，有道是“天下武功，无坚不摧，唯快不破”。不过在快的同时，也牺牲掉了一些兼容性。KHTML 派，是从 KDE2 开始就有的网页排版引擎，该派也比较重视速度，同时还注意兼容性，可以兼容标准网页自不必不说，由于那时候 IE 声势浩大，KHTML 无奈也尽量的支持部分 IE 专属的语法。但代价就是对语法错误的容忍度要比 Gecko 派小。再有就是 Webkit 派，开山鼻祖是苹果公司的 Safari。Safari 学习了 KDE 的 KHTML 和 KJS 两大功夫，并且加入自己的创新，创造了 WebCore 排版和 JavaScriptCore 解析引擎两大武功，开创了 Webkit 派。由于继承自 KHTML，而 KHTML 是 LGPL 授权，

因此 WebKit 派的武功也是开源的，广大民众可以随意学习。所以目前 WebKit 派的徒众甚多，很多手机上的浏览器也是 WebKit 派的。顺便说一下，狐狸的 Gecko 派自然也是开源的，但是由于代码结构不够明晰，学习和维护起来不如 WebKit 方便，因此应用范围不广，主要只是靠 Firefox 支撑派内事务，还有几个 MadFox，RedFox，都是狐狸的同宗。

Chrome 正是 WebKit 派的，因此也继承了自 KHTML 那里就带来的速度，和兼容性。所以他们说并不是靠并行处理提高速度。不过具体影响速度的主要是什么，也是见仁见智了。只从主人的使用感受来说，应该是 Chrome 快一些。不过这话可别让狐狸妹妹听见，要不又该不高兴了，定然要吵个不休，和 Chrome 比上一场。

主人今天想休闲一下，可惜最近没什么片子可看，上网也没什么意思，想了想决定——玩会游戏吧。

可能有人一听说玩游戏会发出无奈的疑问：Linux 还能玩游戏？？哎，说起来很惭愧，在我们 Linux 的世界里，确实没有魔兽世界，确实没有极品飞车，确实没有三国志，确实没有仙剑奇侠传……总之，世面上看得见的游戏在我这都看不见。可没游戏也不能怪我阿，谁叫那些开发游戏的公司不开发 Linux 版的呢。不过虽然没有这些大型游戏，小的益智游戏还是不少的，效果比较好的也有。除此之外，还有更通用的游戏解决方案，就是模拟器。

这里说的模拟器就是在 PC 机上模拟各种游戏机环境的软件。游戏机大家应该都不陌生，当年的 FC 红白机器半随着很多 80 后的孩子们成长起来。当年任天堂的 FC 游戏机使用摩托罗拉 6502 芯片，只有 1.79MHz 的运算频率，别说跟先在的手机比了，恐怕比你家空调机里面那控制温度的芯片的频率都低一些。内存只有 2k，显存也之后 2k，写一空白 word 文档就沾满了。显示分辨率只有 256X240，最大支持 52 种颜色，但能够同时显示在屏幕上的只有 16 色，游戏卡的容量最大也只有 32k。就这么一个弱智的机器，曾经给多少人的带来了多彩的童年，我的主人看来也是其中之一。主人想要回味一下当年那些经典的游戏，这些游戏，说来其实也就是一个小的程序。那时候这个程序是住在游戏卡里面的，游戏卡里面没什么神奇的东西，就是一写用来存储数据的 ROM 芯片，游戏程序就住在这些芯片里。那现在想要在我这里运行这些游戏程序有没有可能呢？直接运行肯定是不行的，他们这些游戏程序的智商和我们是没法比的。我们这些软件如果相当于人的话，他们也就相当于鸡。他们只会是用那种专用的 1.79M 的小芯片来运算，别的一概不会。所以，要让他们运行，就得模拟出游戏机的那种硬件环境出来，就像 VBox 能够模拟出一台计算机一样，需要一种软件在我们工作间里模拟出那种简陋的小 FC 游戏机，这样，就可以让这些游戏软件运行在虚拟的 FC 游戏己里面，就好像搭个鸡窝鸡似的。那我们这有没有这种会养鸡的软件呢？当然有，还不少呢。像 Fceu，SMYNes，Nestopia 等等，都是养鸡专业户。

主人在超级牛力的指导下，安装了 fceu，然后又牵着狐狸妹妹找了个 rom 文件，叫做 mario.nes。然后主人在菜单里找 fceu，发现竟然没有。不过不用着急，我家主人还是比较有经验的，赶快打开了终端，输入 fceu，果然有这个程序。(废话，刚装上的，能没有么。有的人就是不相信超级牛力，装了软件竟然说找不到。其实就算菜单上没有，运行一下 whereis xxx。或者直接问新力得，都可以。)结果运行 fceu 一看，竟然是个字符界面的程序。主人很不爽，上网打听以下，好在还有 gfceu，是 fceu 的图形界面，于是主人赶紧又装了 gfceu，这才满意了。其实要我说，字符界面怎么了，键盘操作才最快捷，字符界面才是我们 Linux 的精髓。

早在电脑刚刚被发明出来的时候，键盘就已经是每一台电脑所必备的输入设备。作为从那个字符界面的时代走过来的 Linux 系统，自然充分考虑的通过键盘操作整个系统的便捷和效率问题。直到现在，使用键盘操作 Linux 都会拥有意想不到的高效率和成就感。

有的用户就不喜欢键盘，不喜欢打字。我以前很不明白，命令键盘可以发送上百个命令，用起来应该很方便才对，为什么人类就那么喜欢那个只能发送：向上，向下，向左，向右，左键，右键 。这六个命令的鼠标呢？（当然，现在的鼠标还多了滚轮，还有的鼠标有更多的按键，但是那也比键盘少阿。）后来见多识广的 OOo 老先生给我解释，我才明白，原来是因为人类记忆力不行，没有我们软件这么可靠。记不住那么多个键，于是只好用那只能发送六个命令的鼠标了。其实说起来通过键盘和我交流还是挺方便的，只是很多人不大熟悉如何交流而已，都以为用键盘和我交流跟用键盘和那个剁死系统交流一样麻烦呢。其实我已经很人性化了，就因为键盘上有个键——Tab

看一个人的键盘，就可以猜测出他平时用电脑干什么。如果 W,A,S,D,U,I,J,K 严重磨损，说明这哥们玩拳皇的；如果 A，Shift，Ctrl，1，2，3，4……9，0 严重磨损，说明是个玩即时战略的，星际魔兽之类；如果 ALT,S 或 Ctrl,Enter 磨损，大概是天天聊 QQ；如果 Tab 键严重磨损，那估计就是个 Linux 高手了。在 Linux 的命令行下，Tab 键起着命令补全的作用。比如说，

你要运行 ifconfig 命令，你可以不用完全输入这 8 个字母，只要输入 ifc，然后按 Tab 键，我就知道了，因为所有可以运行的命令里面以 ifc 开头的就只有 ifconfig，所以当你按下 Tab 键的时候，我就会替你写出完整的命令：ifconfig. 这都因为在你按下 Tab 键的时候，我会去 PATH 变量所设置的所有目录里遍历一遍，检查了里面所有的有 x 权限的文件，查到了 ifconfig 文件。（命令其实就是个可执行文件）之所以这么快，是因为我早就把这些重要的东西缓冲进内存了，所以下次就别抱怨我动不动就把你内存占满了哦。那如果你再少写个字母呢？比如你只写了 if,然后就按 Tab 键，我遍历了一边 PATH 中的路径后发现，有 4 个命令是以 if 开头的，所以我不知道你要的是哪个命令，于是就不做任何动作。这时候如果你再按一下 Tab，我就会提示你：以 if 开头的命令有 if ifconfig ifup ifdown。然后你自己看需要的是哪个，照着输入就行了，很交互吧。这样除了减少按键次数以外，还有一个好处就是你可以不必完全记住整个命令，能够记住前几个字母就可以通过 Tab 把整个命令回忆出来。除了命令，命令的参数也可以用 tab 补全。比如用超级牛力装软件，输入 sudo apt-get i 然后按 Tab，就可以补全 install，之后的软件包名也一样是可以用 Tab 补全的。

有了 Tab，就让用户输入新命令的时候省事了不少，还有一个 history 功能，可以让用户重复以前输入过的命令的时候省心。如果你想输入上一次输入的命令，就按一下向上箭头，就看到了；如果想要再上一次的命令，就再按一下；如果想要再再上一次的命令，就再再按一下；如果想要再再再上一次的命令，就再再再按一下；如果想……如果想写作文的时候凑字数，你就跟我学。好了，总之是可以通过方向键选择以前运行过的命令，如果想查看很久远以前的命令呢？也可以，输入 history。这是一个命令，可以显示之前运行过的 n 条命令，默认情况下 n=1000，现在图形界面越来越发达，输入命令的机会越来越少，估计 1000 条都能把去年的命令显示出来了。说起来 history 命令也没啥神奇，他之所以能够显示曾经运行过的命令，不是因为他有啥水晶球，而是负责解释主人命令的 shell 会把每一条命令记录下来，就写在~/.bash_history 文件中。

history 只不过是把这个文件打开，显示出里面的内容罢了。

我们 Linux 下面的命令，都是 Unix 哲学的优秀继承者，每个命令秉承着简单专一高效的宗旨。每个命令只实现一个功能，但通过各种命令的组合可以实现几乎各种功能。咱就说这个 grep 命令吧，这家伙就是个筛子，他的作用就是让该漏的漏过去，不该漏的挡住。咱就比如刚才说的 history 命令，一运行出来好几百条命令，我相知道我都通过命令行装过什么软件，也就是从命令历史中找到所有 apt-get install 命令，这个怎么办？很简单，只要运行 history|grep “apt-get install”，就可以了。中间那个竖杠就是”\”键上面那个符号，叫做管道符，用过那个剁死系统的人可能知道。所谓管道符，简单地说就是把前面命令的输出传给后面命令做输入，就想象 | 号就是一个大水管子，history 命令输出的那些个字符就像一大堆泔水一样哗啦啦的都浇倒 grep 头上了。(可怜的 grep 阿……)不过不用为 grep 难过，咱不是说了么，他的工作就是——过滤。跟据命令里写的”apt-get install”,grep 会把所有带有这个字符串的行都过滤并显示出来，还可以将要找的字符串突出显示。就好像一大桶泔水通过管道浇下来，grep 就是一个大筛子，把泔水里面的菜叶阿，豆腐阿啥的都给拦住了，只让剩下的地沟油通过，人家就可以捞回去炸油条用了……（城管快来阿，先把 grep 逮捕）除了通过管道符过滤命令输出之外，grep 也可以单独使用，用于查找文本文件中的行。比如我想查看一下/etc/fstab 文件中关于 home 分区的那一行，看看挂载参数怎么写的，那就可以运行 grep "home" /etc/fstab，前面是要查的内容，后面是文件名。

介绍完了 grep，再来说个经常在命令行下出场的家伙——more

还用 history 举例子吧，你运行了 history 命令，可能一下子显示出好几百条命令，咱不是说了么，最多能 1000 条呢。好几百条命令一屏是显示不下的，除非你家是 32 寸液晶显示器，那还得竖着放。显示不下了，后面的内容就会把前面的内容“顶”上去，“楼主”固然是看不见了，什么“沙发”，“板凳”，”地板”，”下水道”的也一样没希望，只能看到最后那几十条。那想看前面的怎么办？虽然可以用 shift+page up 来向上翻页，但一来向上翻的页数有限，二来这也麻烦，一般人都是习惯从上往下看的，倒着往上翻就别扭了。那怎么办呢？这时候 more 就该出场了。more 的功能就是分页显示，把所有要输出的内容先显示出一屏来，等着用户按回车，之后再显示第二屏，直到显示完全部内容。当然，用户也可以不等显示完全部就中途按 q 退出。那怎么用呢？就比如咱刚才说的情况，那就运行 history|more。|符号眼熟吧，根 grep 一样，more 也是支持通过管道输入数据流。history 命令输出的那些个字符就像一大堆泔水一样哗啦啦的都流倒 more 这里了（怎么又是泔水……），more 就把这些东西都先整的大桶缓存起来，然后先盛出一碗来给你看，“您看有没有想吃的？”，你看完了之后，他再去盛第二碗，第三碗……直到你把整个一大桶泔水都检阅完了，more 才结束工作。（当然，你要是坚持不到最后就吐了，那就按 q 退出）

有了 more 就满足了么？不，还有他的死对头，less。

less 实现的功能和 more 基本一样，也是用来分屏输出的，同行是冤家嘛。不同的是，more 只能一页一页往下看，看完了就退出。less 可以上下翻页，看过去的东西可以按向上键或者 page up 键翻回去看。比起 more 更人性一点，另外，都看完了之后 less 是不会自动退出的，一定要按 q 退出。顺便说一下，more 和 less 跟 grep 一样，不光可以通过管道将其他命令的输出当作输入，同时也可以直接查看文件。只要 more /<路径>/文件名 或者 less /<路径>/文件名就可以查看文件内容，当然，只能是文本文件（里面是文本就行，不一定非得以.txt 为扩展名）。

通过这两个命令，您大概可以感受到我们 Linux 系统和有点软公司的系统的不同理念。用过剁死系统的都知道里面有个 DIR 命令，和我们的 ls 一样，都是显示文件用的。当文件很多的时候，dir 命令有专门的参数可以实现分屏显示，而 ls 命令就没有，只能一下子显示出来。为什么？因为分屏显示的事情是由 more 和 less 负责的，完全可以通过 ls|more 这样的组合实现分屏显示。Linux 的理念每个程序只专注于一种功能的实现，而通过多个程序的组合可以实现任何功能。试想如果没有 more 和 less,ls 要负责分屏显示的话，那 history 命令是不是也要处理分屏显示的问题呢？那么所有输出行比较多的命令都要自己负责分屏显示，这些命令的源代码中都要有负责分屏显示的部分，这是一种无谓的重复劳动，而且各自分别实现分屏显示，效果多半也不一样，可能有的命令是按空格显示下一行，有的是按 n 显示下一行等等。与其这样不如把相同的功能独立出来，成为一个统一的，单独的命令。

再来介绍下一位吧，find。

一听这名字就知道干啥的了，找文件的嘛。find 俨然就是一位能力超强的图书馆管理员，只要你想在 Linux 下找任何文件，无论你的要求多么苛刻，find 都能给你找到。你按照文件名找，没问题；你按照创建的日期找，也没问题；按用户查找，还是没问题；按照各种复杂的组合方式查找都没有问题，可以说，只有想不到，没有找不到。如果仅仅是找到那倒也没什么强大之处，最关键的是，他除了找到文件，还能帮你叫人来处理文件。咱从最简单的说吧，按文件的名字查找：find <路径> -name <文件名> 比如在当前目录下找名字叫作 test.c 的文件，那就是 find ./ -name test.c 。如果你要找 3 天前创建的文件可以用-ctime 参数，比如 find ./ -name *.c -ctime +3，意思就是在当前目录查找 3 天前创建的所有.c 文件。+3 就是 3 天前的意思，如果是 -3 那就是 3 天内。要是你想找所有的链接文件，可以依靠-type 参数来根据文件类型来查找。还有根据 用户 -user，根据组-group，根据大小 -size，等等查找方式。不过这些都是些粗浅功夫，最重要的是-exec——对查找到的文件进行处理。

比如你想找/usr/bin 下的所有链接文件，想看看他们都链接到了哪里，怎么办？那首先得先找到，按照类型找，find /usr/bin/ -type l，这样就找到了所有的链接文件了，可是怎么看他们链接到哪了呢？find 命令是看不了，可是 ls 能看阿。ls -l 就可以查看文件的详细信息，如果是链接文件就显示出连接到的地方。这时候就需要让 find 在找到文件之后去调用 ls 命令，这就用到 exec 了。只要这样 find /usr/bin/ -type l -exec ls -l {} \; 前面不用解释，-exec 后面就是对找到的文件执行的命令，这里就是 ls -l 。{}括号的位置就是用找到的文件替换的位置。比如 find 找到了/usr/bin/pkill，是个链接，那么-exec ls -l {}就是要执行 ls -l /usr/bin/pkill，如果写-exec ls {} -l 呢，那自然就是执行 ls /usr/bin/pkill -l 了。这样，所有被 find 找到的链接文件就都会被 ls 一遍，就显示出了他们连接到的位置。可是/usr/bin 下的链接文件很多，一屏现实不下怎么办？哈哈，自然是 less 或 more 出场了。 find /usr/bin/ -type l -exec ls -l

{} \;|less 怎么样，看似复杂的命令，其实也很简单吧。

很多人在 Linux 下听 mp3 的时候都遇到了标签乱码的问题，这主要是编码不统一造成的。用 mid3iconv 命令就可以改变 mp3 标签的编码。（得先安装，sudo apt-get install python-mutagen），但是一个一个的转换肯定会累死人，这时候就可以用 find 命令：

```
find . -name "*.mp3" -execdir mid3iconv -e gbk {} \;
```

网上搜索 mp3 标签问题多半会搜索到这么一条命令，可能很多人执行过。这里的-execdir 和-exec 差不多，只不过-exec 是在当前目录下执行，而-execdir 是在所找到的那个文件的目录下执行。

这一天，狐狸接到命令要去访问一个网站。她一如既往的把请求网页的数据包打好，装进信封里封装好，然后写上收信人的 IP 地址（自然是根 DNS 那里查到的啦），就从网口扔出去了。要是平常，很快就会得到对方网站的响应数据包，拆开了拼好了就能给主人显示了。可是这回，邮件却被邮局退回来了，上面盖着章——查无此地，原件退回！

狐狸很纳闷的挠挠头，这个地址是刚刚从 DNS 那里查倒的阿，应该没有错。这时候一只 chrome 蹿过来说，狐狸姐，这个你就不知道了？狐狸说，我确实不知道阿。chrome 说，好，那我给你解释解释。这时候其他的 chrome 喊：快过来，活还没干完呢。这只 chrome 答应一声，好嘞。然后念动咒语，刷的一下，由分出一个进程。这个新的进程跑去干活，而原来这个 chrome 继续给狐狸讲解：你知道给咱们送信的是哪个邮局么？狐狸说这我知道阿，是朝天通邮局。chrome 神秘的笑笑说，呵呵，这只是表面。表面上我们有很多家邮局可以选择，朝天通，朝天动，朝天信，等等。其实他们都归朝天邮局管辖。这个朝天邮局比较霸道，他看着不顺眼的地方不给送信，直接告诉你查无此地，我怀疑可能是由于那地方没给他交保护费，但具体为什么，我也说不清楚。总之呢，遇到这种情况，邮局看着不顺眼的地方，就愣告诉你查无此地。狐狸点点头，哦，原来这样，你怎么知道的这么多？chrome 得意的说，那倒是也没什么，主要是因为我的东家，就是你经常去找的狗狗哥那里，就有很多这样的地方。这些地方归狗狗哥管，但是朝天邮局就是不想去，所以咱们不能给那边发信。

狐狸一边听 chrome 说，一边赶紧向主人汇报了目的地无法到达的噩耗，汇报之后又转过来问 chrome：那有什么别的办法呢？chrome 说，办法有很多，比如其中有一项你就会。狐狸惊讶道，我会？chrome 说是啊，是个浏览器都应该会，就是代理。

狐狸恍然大悟，哦……明白了。狐狸是明白了，您可能还不明白，没关系，咱慢慢的说。

这个代理是怎么个事情呢？其实说起来也简单，就是代为转交书信。比如说狐狸想给狗狗哥那网站写信，请求数据，正常的流程是把信交给邮局，邮局在把信送到狗狗哥那里。但是有时候

邮局送不到，那就可以用代理。代理就是狐狸不给狗狗哥写信，而是找个中间人，比如说叫做小马哥。当然，这个小马哥的地址，必须是邮局可以送到的。狐狸接到主人访问狗狗哥的命令后，先给小马哥写信，信的内容是：请提供狗狗网站的主页信息。小马哥收到信后，要拆开信，看信的内容，一看，明白了，狐狸想要狗狗的数据，于是小马哥以自己的名义，写信给狗狗，说明要获取主页信息。狗狗收到信之后就像收到一个普通浏览器的请求一样，把数据打包发给小马哥。小马哥收到信，拆开，拿出里面的数据之后再次封进信封，以自己的名义发给狐狸。于是狐狸就得到个狗狗哥主页的信息。听起来小马哥做的事情根邮局差不多，都是中专邮件。但是他们有着本质的区别。首先，邮局是不会拆信件的，狐狸写给狗狗哥的信，发信人，收信人，都写的明白，不会改的。而小马哥不同，他收到信要拆开，看里面的内容，看了内容才知道狐狸想让她做什么，然后他还要把信重新封好，把发信人改成自己，收信人写狗狗，然后发出去（小马哥那里的邮局必须能寄倒狗狗那里）。其次，通过邮局寄信的时候，收信人和发信人是相互知道的。甭管信件在邮局的手里经过的多么复杂的流程，狗狗知道狐狸要请求数据，狐狸也知道数据狗狗回的。而通过小马哥就不同了，狗狗根本不知道狐狸的存在，他只以为小马哥要请求数据。同样，狐狸不知道狗狗的地址，她并不会去 **DNS** 那里查看狗狗的地址，而只是把请求信送给小马哥，查狗狗地址的事情由小马哥来承担。（当然，这时候狐狸唯一必须知道的就是小马哥的地址。）再有，通过邮局发信，只需要整个系统里做统一的设置。把网关，**DNS** 什么的设置好了，所有上网的软件都可以这么用，也都必须这么用。不可能这个软件通过朝天通邮局发信，另一个软件通过向地通邮局发信，除非系统里有多个网卡。而通过代理呢，就需要每个软件分别设置。狐狸可能去找小马哥代理，同时 **chrome** 可以去找大骡子弟弟代理，皮筋去找青蛤蟆大婶代理。（到了动物园了-_-b）而且，每一个代理人，他们的能力还有区别。比如小马哥可能只会读写 **http** 格式的信。所以要让他代理 **http** 的请求，就可以，你要写个 **https** 格式的信，他可能就看不懂。

除了狐狸妹妹，我这里的很多人都可以使用代理。然而这么一个个都分别给小马哥写信也比

较麻烦，再说也不符合我们 Linux 软件“只做一件事，但要做到最好”的哲学理念。所以，这个带里，是可以在我这个系统这里进行全局设置的。设置起来其实也很简单，就是一个全局变量而已，只要 export http_proxy=http://xxx.xxx.xxx.xxx:yyyy 就好了，xxx 是代理服务器的地址，yyy 是端口号。这样设置了之后，所有使用 http 协议的网络软件就都可以通过代理上网了。不过要注意，如果你是在终端中设置的，那么只对在当前终端里启动的软件有效。

会给人代写书信的“小马哥”也分为很多种，其中有一种是基于有点软公司的操作系统的，叫做 ISA，如果要是遇到他，可就有点麻烦了。

别人代写书信，有来者不拒型的，凡是友情求发过去他都会提供服务；也有需要身份验证的，信里面要写上接头暗号，什么天王盖地虎，锄禾日当午之类的。暗号对了才能给你做书信代理。可是 ISA 对暗号的时候（当然，也可以配置成不需要暗号的），不是你把暗号写对就行，还要验笔迹，字写的好看的，他才愿意给你代理，字写的难看的他理都不理。他用了这么一套方法对暗号进行加密，就导致我这个从小不好好上学，字迹不公正的系统不能够直接链上他的代理。就像刚才说的，通过全局变量设置代理，如果这个代理是 ISA 做的，那设了也是白设，连不上。不过我们这里也有人能够用他的代理，谁呢？狐狸妹妹就行，狐狸的字迹工整漂亮，符合 ISA 对字迹的要求，所以狐狸可以链接 ISA 的代理。其他的浏览器，像 Opera，Chrome 都可以，哎，看来把字练好真的是很重要阿。可是他们虽然自己可以链接倒 ISA 代理，但是他们不能帮别人写信阿。比如超级牛里要想通过代理来上网安装软件，如果是别的代理的话就像刚才我说的那样设置一下就可以了，或者在/etc/apt/apt.conf 里设置也可以，这么设置就不是全局了，就是只设置超级牛力的代理。不管怎么设，都是连不上 ISA 的，就是说无论我还是超级牛力，那个字迹都跟大猩猩拿左手写出来的似的。那如果就遇到了费用 ISA 代理不可的情况怎么办呢？也有办法。

找个字迹好的还能代写书信的不就行了么。那么有这样的人么？当然有，ntlmaps 就是这样的软件。

ntlmaps 就是一个在本地代写书信，并且是专门给 ISA 写信的家伙。说简单了，就是个二级代理。有了他之后，把全局变量的代理设置成 http://127.0.0.1:5865（ntlmaps 默认的端口），也就是说，无论谁要上网，都写书信给 ntlmaps,再由 ntlmaps 抄写一份给那个 ISA 代理，这样，有了 ntlmaps 优雅的笔迹，ISA 就不会拒绝他的请求，于是所有的软件就都可以畅游网际了。

今儿个主人让狐狸下载来一个新的家伙，这家伙叫做 Ailurus。我让星爷一查，感情是小熊猫的意思。这个 firefox 其实也是指小熊猫，只不过大家先入为主的，看见 fox 就都管他叫狐狸。和着狐狸妹妹下载来一个自己的同类！？这只小熊猫又会干些什么呢？难道也是个浏览器？下载完成之后，主人立刻叫来超级牛力安装，超级牛力把这家伙从 DEB 盒子里面拎出来，大家一看，长相倒还可以，不过跟狐狸一点也不象。而且，怎么看也不是个浏览器阿？他到底是干嘛的呢？

主人还真是个急性子，装完了之后立刻把小熊猫叫起来干活。小熊猫一爬起来就赶快热情的跟主人打招呼：来了您内！一位里边请～～客官，您来点啥？您是打尖阿还是住店阿？哦不对，您是打算来俩软件尝尝阿，还是想改改设置装修系统呢？

主人问道:“你这都有什么软件阿？”

小熊猫说:“要说这软件可就多了去啦。什么山中走兽云中燕，陆地牛羊海底鲜，猴头眼窝鲨鱼翅，熊掌干贝鹿尾尖，这些东西我们这阿……”

“全有？”

“厄，全没有”

“没有你说他干嘛阿。”

“我这是给您打个比方，再说这写个东西他也不是软件阿。我们这是挑软件的地方。各门各类的软件都有，什么浏览器，发邮件，文件共享一大片，刻 CD，不算慢，媒体播放把片看，博客聊天笑哈哈，绘图排版有专家，编程工具很有用，集成开发得靠他……”

“还真不少”

“那可不是，我这把每个软件都说全了，那咱就得年底见了”

“我想玩游戏，你给介绍俩游戏吧”

“好勒～你看咱这有益智的，即时战略的，FPS 的，还有棋牌类的，什么都有。您来个益智

的泡泡龙尝尝？”

“行阿，这不大吧。”

“不大，这顶多算个凉菜，您再点个热的。”

“你这里头还有热的？”

“那是阿，你看这 Warzone 2100，战争题材，全是枪林弹雨点火爆炸的，能不热么”

“还真是，那再来个这个吧。”

“还要点比的不要？”

“不要了，先来这俩，好吃再来”

“得勒～ frozen-bubble 一只，warzone 2100 一套～～～”

内存里安静了 3 秒钟，他又喊:“frozen-bubble 一只，warzone 2100 一套～～～”大家你看看我，我看看你，还是不知道他什么意思。结果他忍不住了，一指超级牛力:“说你那！赶快装去阿！”超级牛力这才醒悟过来：和着是喊我呢阿，拿我当厨子使唤了。可是这也算是主人的命令阿，不能不执行，于是超级牛力赶紧去给装这俩软件。一会装完了，告诉小熊猫，小熊猫扭脸跟主人说：您的菜齐了，给搁在您应用程序菜单里面了，您自个找去吧。

这小熊猫一来，新力得可就麻烦了。原本是他超级牛力的图形界面嘛，主人要是想用图形界面装软件，那必然得找他，再由他把主人的意图转达给超级牛力。现在这小熊猫以一来，这不是把他的饭碗给抢了么。可是新立得也没啥可说的，人家小熊猫真的是态度热情，服务周到，未语先笑，还带介绍。主人愿意用小熊猫，他新立得也管不了阿。

濒临失业的新立得垂头丧气的站在一边，看着主人继续跟那个小熊猫玩。主人发现小熊猫除了可以装软件以外，还可以干好多事情。他点了下系统设置，小熊猫立刻满面春风的又迎来上来：“客观，您有来啦～您赶紧看看这系统里哪块不顺眼，我给您收拾去。”

主人说：“我觉得我这 4G 的内存足够大了，我想尽量少用 swap，多用内存。”

小熊猫一伸手，不知道从哪变出一个电子表式的东西，原来上面写的是 60，他使劲一拧，把上面的数字调成了 0：“好勒～您放心，冲您这么大内存，swap 里以后再也不用写东西了。”

主人微笑着点点头，可内存里边，gedit 心里不是滋味了。要说改这么个设置其实很简单，就是一个配置文件的事情，要搁往常，主人一定是叫来 gedit 去改，可这熊猫以来，俨然自己的生意也被抢了。他扭头看看被冷落在一旁的新立得，顿时有一种同时天涯沦落人的感觉，于是不自觉的想新立得靠近了几步。

这时候主人又说了：“还有阿，这开机时候的声音不大好听，你把他关了吧”

小熊猫随手打了个响指，之后说道“好，已经搞定了。”

Gnome 工作组管理声音的那哥们顿时一身冷汗，这设置 Gnome 声音应该找他阿。他看看在一起窃窃私语的 gedit 和新立得，也凑了过去。

主人又有要求：“还有阿，桌面上没有回收站有些别扭，给弄一个出来把。”

小熊猫听了赶快说：“没问题～”一边说一边风风火火的跑出去，转眼回来，手里拿着一个纸篓，放在了桌面上。这一下 gconf-editor 也坐不住了，改这种系统设置原本非他莫属阿，他号称 Gnome 下的注册表，就这么轻易的被一个猫科哺乳动物超越了？赶紧跟那失业的哥仨交流心

得去……

没过多一会，Gnome 工作组里管桌面的，管登录窗口的，管屏保的，纷纷受到了小熊猫的威胁。这还不算完，主人随后问了一句:“你看我这机器里还有什么不顺眼的么？”小熊猫说:“我看那些 ubuntu-docs 的文档您也不看，还不如把这个报删掉，能节省 270M 的空间。”主人一听:“好!”小熊猫赶紧跟超级牛力说:“快把那个 ubuntu-docs 包删了，打扫干净空间好让主人干别的用。”事业不顺的软件们顿时慌了——这就开始裁员了阿！

当这帮软件们人心惶惶的担心裁员的时候，OO 老先生站了出来。要说还是 OO 老先生有经验，赶紧以专家的身份出来辟谣:你们好歹也是 Linux 下的知名软件，怎么这个道理都想不通呢？那个小熊猫强在方便，强在善于于主人交流。但他的功能虽多，却不专一，其实是个不符合我们 Linux 精神的软件，只不过是为了方便大众而已。要说装软件，他就只能装那么几个软件，很多都是需要添加第三方软件源，然后再安装的，如果用你新力得的话，其实也能装，就是麻烦点。可除了这些比较常用又装起来麻烦的软件以外，其他文件他不管装，还得你新力得来。配置 Gnome 也是，虽然他能对 Gnome 进行一些配置，但那只是常用的几个，要真需要更详细的配置，不还得你 gconf-editor 来么。 众软件一听，互相看看，顿时豁然开朗，都松了一口气，看来这工作还能继续，饭碗还没什么问题。

转过天来，主人启动电脑，我们如往常般起床并投入工作，可是主人不知道是不是心情不好，还是遇到什么事情，总觉得我们起床比往常慢了许多，自言自语的说到：这 Ubuntu 也跟 Windows 似的越用越慢阿。这一下可把我们工作间里的软件们气坏了，怎么能拿我们根那个查皮相提并论呢？当然，查皮也有很强大的地方，但就他这随着使用时间增长速度越来越慢的坏毛病是一直被人们诟病的。其实说来这也不怪查皮，主要是他们那里的软件都比较封闭，比较私自，做事情不考虑别人。你看我们这的软件，有什么库什么的都放在一起，相互共享。而查皮那的软件各自有自己的小天地，每个软件都封闭在自己那个目录里。很多软件为了自己的一些目的都会要求

在系统启动项里增加自己的一些内容。你要说 QQ 要求启动是自动启动他，那还友情可原，毕竟主人确实需要经常用他。像 RealOne 这样的软件也要求在启动的时候要启动他的一个小进程，而这个进程只是为了监视主人有没有更改 rmvb 文件的默认播放器，如果一旦改了，他马上给改回来，也就是说一定要保证 RealOne 是唯一合理合法的 Rmvb 文件的默认播放软件。你说这不是霸道么，人家愿意用什么播是人家的事，你看我们这里的 SMplayer 和 Totem 他们打的再热闹，也是绝对遵守主人的命令，主人说用谁就是用谁，也没听说过谁为了显示自己的能力就去修改默认播放器设置的。查皮那的那些个软件如此不管他人的感受，每个软件都要 申请的启动项，那当然随着软件装的越来越多，查皮起床就越来越慢了。

可是我们这里的软件可没有这么私自的人阿，怎么可能越用越慢呢？主人这么说我们，太伤自尊了～～ 不多久，主人就叫来超级牛力，让他去找一个叫做 bootchart 的软件。等装上了一看，这家伙是个计时员，专门记录 Linux 系统启动的时间，这回我们可找着说理的人了，哈哈。

果然，在装好了 bootchart 之后，主人就马上重启的计算机。

计算机重新启动，在我这个内核起床之后，bootchart 最先跑进内存里（往常是 init 最先），并且拿出秒表，静静的准备开始着。由于 init 是每次最先起床的进程，所以 bootchart 是以 init 启动作为计时基准，他要等倒 init 跑进内存再开始计时。很快，随着 init 跑进内存里并喊一声，init 启动！bootchart 果断的按下秒表的计时键并高喊:“计时开始！紧张而激烈的 ubuntu 系统的启动过程随着刚刚 init 的一声呐喊，终于开始啦！也许有的观众刚刚打开电梯还不知道怎么回事，我们正在进行的是 ubuntu 一日一度甚至一日数度的启动过程。只见各种内核进程以及自启动进程纷纷开始运行。udev，udev，udev 突破啦！他不顾前面的追兵，跑进了内存。哦，udev 把启动过程传给了 modprobe，modprobe 跑到内存中场，他像马一样的翱翔速度……哦，他停下了，他是在等待硬盘工作。就在这千钧一发不可收拾的关头中，rc 上场。他的主要任务就是

叫醒那些设置好的启动项。这时候，好，modprobe 已经读完了硬盘，他完成的他的工作，终于成功啦，总共用了 1.38 秒，突破了他自己创造的 1.37 妙的记录-_-b，哦等一下……呃……好吧咱们再看 rc，只见他在后场 38 公里处以 70 码的一脚远射把启动的接力棒传给了 network，这时候 network 被 rc 撞醒啦，没有被撞死，rc 把速度掌握的恰倒好处。network 醒来去配置网络，好，他配置好了网络，network 立功啦～～Linus 生日快乐～～～然后是 sh，哦，gdm 也来了，大家齐心协力，共抗非典……”

别看 bootchart 这家伙说的挺乱呼，可他记录的却是井井有条。当系统完全启动之后，bootchart 以图片的形式写了一份报告，向主人汇报了各个软件启动所用的时间，哪些时间用在等待其他进程上，哪些时间用在等待硬件响应上，写的非常详细。主人看了之后，终于觉得，我们的启动速度，并没有因为使用时间的增长而变慢。

今天我们的屋子扩建了。

早上一起床，做硬件扫描的时候发现 PCI-E 总线上比往常多了一个设备，是一块 RAID 卡！主人从哪里弄来了这东西阿。难道要把我们这台电脑升级为服务器？我们 ubuntu 的服务器版和桌面版确实基本没有啥区别，只是默认安装的软件和使用的内核不大一样。服务器版自然用的是服务器的内核，注重稳定，并且预装的是服务器相关的软件，没有图形环境。服务器版变为桌面版只要安装上桌面版的那些东西就可以，反过来也是一样。不过主人这台家用的电脑，整成服务器有什么用呢？要是不整成服务器，又为什么要装 RAID 卡呢？

说了这么半天，您可能还不知道这个 RAID 是个什么玩意。RAID，就是 Redundant Array of Inexpensive Disks 的缩写，中文翻译过来叫做“廉价冗余磁盘阵列”。有人可能问了：“这个 RAID 卡我买过，怎么也得几千块钱，加上上面接的硬盘，整个价钱能顶上一整台普通的家用电脑了，这个廉价二字从何谈起阿？”这个事情，还得从头说。RAID 这个技术诞生在 1987 年，那时候的硬盘不像现在这么大，而且那时候的硬盘是容量越大单价越贵。比如现在，500G 的硬盘要是卖 400 块钱，那 1T 的硬盘绝对不到 800 块钱，道理很简单——否则谁买 1T 的阿，插上俩 500G 的好不好。而当年那个时候不是这样，那时候比如 50M 的硬盘卖 500 块钱，那 100M 的硬盘能卖 1500。那有人说了，谁还买 100M 的阿，买俩 50M 的插上不好么？（好像就是刚刚我说的-_-b）可是有的情况不允许这么做的，比如说我有个数据库，数据量很大，可能会达到 70M。那我就必须装一个大于 70M 的硬盘才能装的下，你装俩 50M 的硬盘，我这数据库文件总不能切碎了放吧。这个时候，RAID 作为一项省钱的技术出现了。（虽然他现在已经俨然成为一种费钱的技术）

最初的 RAID 的功能很简单，就是把几块硬盘连接到一块 RAID 卡上，然后 RAID 卡把他们拼接在一起（逻辑上拼接，可不是拿刀切碎了当七巧板玩阿。），作为一整块大硬盘来用，这样就节约了购买大容量磁盘的成本，所以才号称“廉价”。RAID 卡工作的时候也不许要什么运算，只

是向上报告自己是一个硬盘控制器，上面接了一块 100M 的硬盘，然后上层的软件信以为真。当有数据写的时候，RAID 卡就要实现自己的谎言（这话听着都矛盾……），真的向上层提供 100M 的完整的存储空间——他也确实可能提供，因为有俩 50M 的硬盘呢吗。他会先写其中一块硬盘，写满之后再写另外一块。当然具体怎么写的细节，这些上层软件是不知道的，他们只以为这就是一个 100M 的硬盘。

这种简单的扩大容量的阵列，也就相当于把两个屋子盖成里外屋，从正门进外屋，外屋里面还有个门，就是里屋。（如果是更多的硬盘，那就还有里里屋，里里里屋……）往里面放东西的时候，先往里屋放，放满了再放外屋。可这种结构，也只不过是增加容量而已，慢慢的人们就研究进一步开发这种磁盘阵列的潜能。首先想提高的，就是阵列的读写速度。长期以来，内存，cpu 的容量的速度都成倍数的增长，而硬盘的读写速度却因为受到机械性能的限制而一直没有跟上其他硬件的发展。而作为这种磁盘阵列，既然由 2 块以上的硬盘组成，那么理论上就可以实现读写速度提高两倍以上。于是就有了 RAID0。这种级别的磁盘阵列，可以提高阵列的读写性能，简单地说就是有数据往里面写的时候，就是所有的硬盘一起写，当然速度就提高了。就好像还是两间屋子，这回把两个屋子并排挨着盖在一起，然后两个屋子的门也和成一个更宽的门，这样里面的容量自然还是两个屋子的容纳量，而往里搬东西的时候，由于门宽了，速度就快了。

再有呢，就是 RAID1，这个是为了安全性设计的。原理就是两个屋子，东屋和西屋。里面容量一样，放的东西也一模一样。专门有人看着，看往东屋里放了什么东西，就去买个一模一样的也往西屋里放一个。这样，万一那天地震把东屋震坏了，把里面东西都砸坏了，西屋里的东西还照样能用。缺点也显而易见，浪费了一个屋子嘛。明明两个屋子，就只能放一个屋子的东西。

再有，就是最最神奇最最有技术含量 RAID5 了。这是怎么个意思呢？这就需要三间以上的屋子，咱就拿 3 间来说吧。有左中右三间屋子，一样大小，本来是分着的，每个屋子有一个门。

（就是三块普通的容量相同的硬盘啦）然后来个包工队，高工头姓哈。哈工头带着他的包工队噼里啪啦的一通改造，三间屋子整合倒了一起，一看容量，咦？变成两间的容量了，再看门，也只是两个原来大门的宽度。您一定怀疑哈工头把那一间屋子的砖都拉出去卖了当回扣了吧。不过别急，看看这个改造过的屋子的神奇之处吧。往里面放东西的时候，由于门是两倍的门宽，所以速度肯定是提高的两倍。里面的容量也确实是 2 个屋子的容量。你可以试着往里面放点东西，然后扔一颗手雷，炸掉半边屋子，同时也炸掉了里面的东西。然后之间哈工头拿出一个小棍，挥了几挥，念句什么后轱辘追不上前轱辘什么的咒语，然后就发现第三间屋子出现了，里面好好的放着刚刚被炸掉的那间屋子里面的内容。后来听说那包工头不是姓哈，人家复姓哈利，名叫波特……

说了半天，还没说明白我们的屋子到底怎么扩建了呢。主人装好了那块 RAID 卡的驱动后，接上了 2 块硬盘，做了一个 RAID0，就是那种能够提高读写性能的方式。这种方式虽然速度快了，不过风险也同样高了，因为数据是同时写进两个硬盘的，一边写一半，所以其中任何一块硬盘出现问题，所有的数据就都丢失了。不过一帮家庭用户倒是不必太担心这个，但也要记得尽量不要放太重要的东西。主人接的两个硬盘每个有 250G，作成 RAID0 后成了一个 500G 的大房间。那么这个房间怎么接进我们的房间里来呢？

咱之前说过，一个存储空间要想接入到我们的哦自立来，需要 mount，也就是挂载。咱还说过，挂载的过程，就是给这个屋子起个名字，就像是挂个牌子。像 U 盘阿，移动硬盘阿，这样临时接入的设备，在接入的时候挂载一下还可以，其实我都是自动帮主人挂载倒 media 下的。而像今天加进来这个 RAID0 设备，肯定不是临时加进来一下，应该是以后长期存在的。那么这个空间应该挂载到哪呢？这个问题，就得看主人想拿他来干什么了，我们是决定不了的。主人要想指定这个空间被长期挂载的某个地方，就要修改 fstab 文件。

这个 fstab 文件是主人用来告诉我整个文件系统如何挂载的配置文件，每次起床后我都要查看一下，里面记录着，那个屋子是厨房，哪个屋子是客厅。有人说，你这家伙脑子是不是有点问

题？难道每次睡觉醒来连哪屋是厨房都不及的了么？那要是半夜起夜的时候你还就不知道哪个是厕所了？哎，我是软件，再次重申。fstab 里面的格式大概是这样子：

```
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda1 during installation
UUID=90f8ca7a-fd04-4bde-8247-ae939b559147 /             ext4    errors=re
mount-ro 0       1
# swap was on /dev/sda6 during installation
UUID=08da5802-2a80-4a4e-8da4-c922e4e75ef0 none          swap    sw
        0       0
/dev/scd1       /media/cdrom2   udf,iso9660 user,noauto,exec,utf8 0       0
```

里面带 # 的是注释，这些行都是给人类看的，我看不懂，掠过不管，其余的行就是有用的了。每一行描述了一个磁盘分区的挂载。每行的最前面是 file system，是说明本行描述的是那个分区，也就是那间屋子。我们 Ubuntu 用来区别屋子的，是使用每个屋子的 UUID。这个 UUID 就是通用唯一识别码的意思，每个屋子都会有一个，而且互相都不一样，我们就是用这个来去别各各屋子的。过去的 Linux 有的使用 sda1,sdb1 这样的设备名来区别每个屋子，但是这样会有问题。比如原来机器上只在第二通道接了一个硬盘，那么这个硬盘肯定是 sda，这个硬盘的第一个分区就是 sda1。然后 fstab 就写把 sda1 当作/分区。如果后来又在第一通道插入了另一块硬盘，那么后插入的硬盘由于在第一通道，可能就成为 sda 了，原来的 sda 变成了 sdb 了。fstab 还要把 sda1 当作/来用，就出问题了。像我们这样用 UUID 来区分分区，就不会有这样的问题了。每行

的第二部分就是这个分区要挂载到的位置 mount point，也就是这个房间要挂的牌子，什么/阿，/home 阿什么的，我也不用多介绍了。再后面就是这个分区的文件格式 type，这个也不用多说。之后的 options 就是挂载时候的选项，比如 ro，说明这个分区挂载为只读；user，允许普通用户操作这个分区；sync，不对这个分区的写操作进行缓冲。一般没什么特殊的要求的话，这个就写 defaults 就可以了。后面那个 dump，是用来给 dump 软件看的，根据这个数据在决定是否要对这个分区进行备份，是 1 就备份，0 就不备份，反正一般我们的系统里也没有 dump，所以这里写 0 就好。最后一个 pass 是用来说明是否需要在启动时对这个分区进行检查，如果是 1 或者 2 就要检查一下，如果是 0 就掠过不检查。要注意的是/分区很重要，因此/分区对应的 pass 值必须是 1，而其他的分区可以是 0，一般没什么可查的。如果你想查以下，就写 2。

介绍这么多，主人已经叫来 gedit 来修改 fstab 了。首先主人添加了一个 UUID= 1478d8d-2a80-4a4e-8da4-c922e8a8e8d7e 的一行，这个 UUID 估计就是新添加的那个 RAID0 了。哦，对了，忘了介绍怎么得到这个 UUID 了，其实很简单，就 ls -l /dev/disk/by-uuid/ 就可以看到了，里面每一个文件都是一个软链接，软连接的名字就是 UUID，链接到的设备就是这个 UUID 对应的真正的设备。或者 sudo blkid 也可以，更正规一些。之后，看到主人在 fstab 里写着把这个设备挂载到了/home。看来是主人觉得存放个人物品的空间不够大，想扩充一下。可是，那原来里面的东西怎么办呢？

很简单，拷过去！

主人改好了 fstab 后，我只是眼睁睁的看着，没有作任何动作，这是规矩，fstab 只在启动的时候起作用，不重启是不会生效的。主人也很明白这个道理，存好了 fstab 后，又去把那个 raid0 的磁盘 Mount 到了/media/raid0 下（当然，在这之前创建好了 raid0 这个目录），然后 cp ~/* /media/raid0 -a，这就把他的家目录下的所有东西，包括那些文档阿，照片阿，视频阿，以及所有软件的配置文件都拷贝过去了。完成之后，主人就重启了。

经过一番折腾，我们的屋子总算是扩建完成了。/home 目录挂载到了 RAID0 磁盘上，容量高达一个 T！原来那些堆满半个屋子的东西放到这里之后只占据一个小角落而已。不知道空着的那一大片里，主人想存点啥，存软件和游戏？咱 Linux 系统下哪有过这么大这么多的游戏可存呢？存电影？恩……这个靠谱，只是主人这 512K 的 ADSL 小猫要拖到什么时候才能拖满啊，这可要辛苦咱家奔流了，呵呵。

转过天来，超级牛力奔跑着冲向网口，高喊着："本 APT 有活干啦，有软件装啦……还有超级牛力啦～～～"我赶紧问一句："去装什么软件？"网口外传来超级牛力悠扬的声音："大～三～～八～～～"狐狸妹妹凑过来问："他说的是我经常在论坛上看到的那个狐狸头么？"我解释道："不是的，他说的大三八，大概是三八。"狐狸怒道："您这不是废话么。"我赶忙给她写出来：是 Samba，不是三八。

Samba 是一个用于局域网中的计算机间文件共享的软件，这么说您大概还摸不着头脑，那么网上邻居您听说过吧？对，Samba 就是干这个的。当年那个有点软公司设计了一套局域网计算机间的文件共享协议，起名叫做 SMB，就是 Server Message Block 的缩写。当时所有的 Windows 系统就都集成这种协议，因此这个协议在局域网系统中的影响还比较大。后来，国际互联网，也就是 Internet 逐渐流行了起来，有点软公司希望他们的这个协议能够一个用在 Internet 上，因此对其进行的整理，更名为 CIFS,也就是 Common Internet File System。从名字可

以看出，他们的期望是很高的，不过实际呢……反正，你现在用网上邻居上搜狐么？当然，不管怎样，CIFS 或者说 SMB 协议在局域网中传输文件还是非常方便的。当然，我们 Linux 系统之间也有很好的局域网共享文件的协议，叫做 NFS，网络文件系统的缩写。但无奈那该死的查皮不支持这个协议嘛，所以要想网络上的查皮和我之间能共享文件的话，要么我这里有人能懂 CIFS 协议，要么查皮那里有软件能解析 NFS 协议。然而毕竟还是我这里的软件大度一些，所以就有了用于支持 SMB 协议的软件——Samba。（其实主要是因为我们 Linux 比较小众啦）

Samba 软件包很快被超级牛力装好了，包里面出了一些工作必需品外，主要包括两个程序：smbd，nmbd。这两个家伙总是一起合作来实现文件共享的工作，就好像小区里开了一个小卖部，需要有人给小卖部做宣传，也需要有人卖东西送货。

nmbd 这家伙负责计算机名称的解释，他负责让别的计算机能够找到我们这台电脑，也就是说 nmbd 就是个负责给小卖部做宣传的。当然，各行各业都有规矩，要饭的还得分地盘拜码头呢，搞宣传的也得按着道儿上的规矩来。那么这个道儿上有什么规矩呢？这个统一的规矩就是 NetBIOS。有同学可能要提问了，这是个什么协议阿？好，我来解释一下：咳咳，NetBIOS 协议是由 IBM 公司开发的，主要用于数十台计算机的小型局域网。NetBIOS 协议是一种让在局域网上的程序可以使用的应用程序编程接口（API），他为程序提供了请求低级服务的统一的命令集，作用是为了给局域网提供网络以及其他特殊功能，系统可以利用 WINS 服务、广播及 Lmhost 文件等多种模式将 NetBIOS 名解析为相应 IP 地址，实现信息通讯，所以在局域网内部使用 NetBIOS 协议可以方便地实现消息通信及资源的共享。好，以上是搜来的。nmbd 这个家伙就是按照 NetBIOS 规定的方式和流程，在小区范围内（也就是局域网的全网段啦）向大家宣传本小卖部。首先，当 nmbd 起床，也就是服务启动的时候，nmbd 会抄起大喇叭向全小区广播：懒蜗牛小卖部开张啦 ~ 懒蜗牛小卖部开张啦 ~ 他会喊上 6 倒 10 遍，确认没人反对，我们这个小卖部就可以正

式叫做懒蜗牛小卖部了。难道还会有人反对么？有，因为小区里可能会有别的小卖部，人家可能也叫懒蜗牛小卖部，那就重名了，这是不允许的。如果有别的小卖部听到了 nmbd 的广播，就会回应：我们已经宣布对懒蜗牛小卖部名称拥有主权，懒蜗牛自古以来就是我们小卖部不可分割的名称，任何试图抢夺我们名称的行为我们都要强烈抗议！之后 nmbd 就知道了，赶紧再改名，再广播：懒乌龟小卖部开张啦，懒乌龟小卖部开张啦……直到最终没人对名字提出异议为止。当 nmbd 服务停止的时候，nmbd 也会向全小区广播：懒蜗牛小卖部倒闭啦 ~ 懒蜗牛小卖部倒闭啦 ~（还真快）当然，开张有人管你，倒闭的时候就不会有人提出异议了，nmbd 广播的目的是告诉大家，懒蜗牛这个名字我们已经不使用了，谁要用可以用了。除了启动和关闭时候的工作之外，nmbd 的主要工作就是让别人找到我们小卖部。比如有人想从我们这里拷贝文件，他怎么知道这个“懒蜗牛小卖部”在哪呢？如果当小卖部开张的时候他听到了 nmbd 的广播并且记住了小卖部的位置（就是 IP）的话，他就可以直接去了。如果小卖部开张的时候他还没起，或者起了但是没听见（广播包不能保证发到每个终端）怎么办呢？很简单，他喊。他可以喊：小区里都有什么小卖部，快快报上名来！然后每个小卖部再报告一次自己的名字和地址，他再决定要访问哪个。这就是你在查皮打开网上邻居时候的过程。他也可能喊：懒蜗牛小卖部你在哪里？在哪里？nmbd 听见就回应：他刚离去，他刚离去……哦，不对不对，我们就在这里，在这里！然后这个家伙就可以直接来这里要东西了，这就是你在查皮里直接通过计算机名访问另一台电脑的时候发生的事情。总之，在 NetBIOS 的小区里，通讯基本靠吼，nmbd 就是负责吼的，吼倒客户来到小卖部，他的职责就结束了，就该 smbd 出场了。

smbd 的任务就是给来到小卖部的客户提供他们想要的东西，但可不是谁来都给的，当然，也不是说要给钱，而是需要进行认证。要看看来的人有没有权利获取某一个文件。这个权限，包括两部分，一部分是 smbd 管理的 Samba 权限，一部分是由我管理的系统的权限。权限认证的过程就像这样：一个网络用户来到我们的懒蜗牛小卖部，对接待他的 smbd 说：给我来二斤照片，

就那边那个。这时候 smbd 会很有礼貌的说：您好，请您出示用户名和密码。然后那个人自报家门：我叫张二嘎，密码是 46243712，smbd 拿出自己存在硬盘里的用户密码对照表，仔细检查一遍，确认这人确实是张二嘎，然后再找到他要的那二斤照片，叫什么瘦瘦的，可能是根瘦身有关。然后再看看自己的共享记录，就是/etc/smb.conf，看张二嘎能不能访问瘦瘦，这里检查的共享权限是看他能不能访问这个共享。一看，瘦瘦有档期可以访问，然后就把瘦瘦交给张二嘎？不能！手续还没完呢，想看瘦瘦没这么容易。之后 smbd 还要向我报告，说头儿阿，有个张二嘎要访问瘦瘦文件，你看看他有权限不？然后再由我检查权限，这会检查的是文件权限，也就是 ls -l 的时候可以看到的那种 rwx 的权限。如果共享权限过不了，只是说明张儿嘎不能通过网络访问瘦瘦这个共享（但是可能人家可以本地访问这个文件），如果我这里的文件权限过不了，说明张二嘎压根就不能访问瘦瘦文件，这两个权限是有区别的。所有权限都通过了之后，smbd 才把瘦瘦文件真的读取出来，然后按照 cifs 协议打好包，递给张二嘎，完成一次文件的分享。

以上说的只是常见的，也是默认的权限管理状况。其实 smbd 有 4 种安全级别，刚才说的 user 级别，这种级别下，smbd 根据本机上存的 smbpasswd 文件来判断来访问的人是不是张二嘎。另外三种级别分别是 share，server，domain。在 server 级别，你可以给 smbd 指定一个认证服务器，比如另外一个小卖部，也就是另外一个装了 samba 的机器，去那台机器上认证。这种情况下，用户来了，说我叫张二嘎，smbd 不用去找什么 smbpasswd 文件，直接扭头问隔壁的一片云小卖部：哎，你看丫是张二嘎么？一片云小卖部的人点点头：恩，就是他。这就算认证通过了。那么在 domain 级别呢？其实跟 server 差不多，smbd 也不用自己做认证，但是这回不是链接其他的 samba 服务器了，而是链接倒 windows 的 domain 服务器上做验证。就像是用户来了，说叫张二嘎，smbd 不查文件，也不问隔壁的一片云，直接拿起电话打 110：你好，我们这有个人说是你们通缉的张二嘎，你们过来看看是不是。最后这个 share 级别呢？这个级别就是根本不管你是谁，一视同仁。人家来了说：我是张……smbd 直接拦住：行了，甭跟我说你是

谁，不就要瘦瘦么，拿走拿走。

主人装好了大三八之后，马上开始编辑/etc/samba/smb.conf 文件，将主人家目录下的一个视频目录共享了出去，为懒蜗牛小卖部提供了店面场地。之后，启动了 samba 服务，smbd 和 nmbd 两位跑进工作间开始干活，小卖部的员工也有了。这目录里面本来就有不少的视频文件，算是有些存货吧，不过肯定远远不够，要知道现在这里目录已经是在主人新加的 1T 的 RAID0 分区上了，这才哪到哪啊，所以还得有个负责进货的，那自然非奔流莫数了。于是咱这懒蜗牛小卖部真的就开张了。

开张之后的一段时间，一直没有客人来。nmbd 每天都卖力的给小卖部做宣传，不过喊是喊够了，但是有没有人听到就不知道了。奔流也是兢兢业业的给小卖部备货，库存的片子是越来越多。只有 smbd 似乎一直没什么事情。这一天终于有个客户上门了，由于主人把 samba 配置成了共享模式，所以 smbd 也没问对方是谁，只知道好像是来自一台 windows 系统的电脑，smbd 也没多问就把东西给他了。这个给东西的过程说来容易，其实也挺麻烦的。因为是网络共享嘛，所有的文件都是要通过网口送到对方手里的。人家来买东西只是通过网口把话带到：我是谁谁谁（共享模式当然就不管他是谁了），我地址是 xxx（ip 地址），我想要你那个 YYY 文件，赶快送过来。然后人家可就在家等着去了，这边 smbd 就忙上了，找到那个文件，剁碎了，剁成小块（别着急，都标着号呢，到那边人家还能拼上），然后一块一块的打成小包从网口递出去。这过程倒是有点象开网店了。这位刚走，一会又来了一位客人，好象也是从一个 Linux 系统上来的。smbd 看到了亲切的面孔很是兴奋，刚想上去打个招呼，没想到来的这位客人冷冷的说：“哟，怎么是个 Ubuntu 系统阿，好好的两台 LLinux 间传文件用的哪门子 samba 呀，CIFS 协议可不是给 Linux 准备的，用 NFS 多好，还省我的事。”smbd 听得这叫一个不舒服，samba 怎么就不许两台 Linux 用了？我们图省事，节约资源少开一个 NFS 服务不行么。不过毕竟人家是顾客，顾客就是上帝嘛，于是 smbd 也没说别的，只是问他：“您要点什么？”那个客户说：“就要那个盗梦空间那个文件，快点阿。我回去等着去了。”smbd 一赌气，把这个文件剁的异常的零碎：哼，让她拼去！

那这个 NFS 是个什么呢？为什么 smbd 听了不爽呢？说来也难怪，这个 NFS 呢，是我们 Linux，以及 Unix 用来通过网络共享文件的一种协议，全称是 Network File System 网络文件系统。它和 Samba 的这个协议的作用差不多，都是给局域网的计算机之间共享文件用的。人家 smbd 就是专业负责实现 Samba 协议的，你当着人家面说 samba 协议怎么怎么不好，不如 NFS，那他能高兴么。就像你去麦当劳问肯德基怎么走一样会被人打出来。说起来也确实 NFS 这个协议才是我们 Linux 的原生产品。Linux 之间传输东西的话用它会更好一些。首先就是传输速度，可以比 Samba 快些，因为 NFS 是可以基于 UDP 协议，也可以基于 TCP 协议，可视网络情况来选择，那基于 UDP 协议的时候自然就快些。再有呢，NFS 可以支持我们 Linux 文件系统上的文件权限的设置。比如说有两台机器，A 和 B。A 机器上有个用户叫多拉梦，多拉梦在 A 机器上有个文件叫做竹豆娘。当多拉梦用户在 B 机器上登录，并想通过网络访问 A 机器上的竹豆娘的时候，如果是用 NFS 协议，就不需要额外设置什么权限，只要用多拉梦用户去登录就可以访问了。（当然，NFS 也可以根据 IP 限制权限，A 机器必须没有限制 B 机器的 IP 访问）可如果用 samba 协议的话，Samba 的权限和机器上系统的权限是不同的，所以必须再次设置一下竹豆娘所在的共享目录允许多拉梦用户访问才行。

自从我们这台机器被装好的 samba 服务，关机的时间越来越少了，经常是一天 24 小时的开着，同志们很高兴，可以有更多的时间跑到内存里玩了，呵呵。当人，咱们软件是不知倒累的，只要还有电，我们就能把活干。不过我们虽然能 24 小时活动了，但主人似乎并不像我们这样活跃，他每天至少有八个小时在睡觉，白天还要去上班，真正坐在电脑前指导我们工作的时间也就几个小时。那他不在的时候，没有人发号施令，我们这一帮软件在内存里，干点什么呢？嘿嘿，别着急，主人早就设计好了。

跟随我来的诸多经典字符界面程序之中，有这么一位称职的管家，他能够根据用户的意图，安排好每天，每月的日常工作，虽然我们都觉得他挺麻烦的，但是他那种严格认真一丝不苟的工作作风确实是别的软件比不了的，这个人就是 cron。

我们主人在离开之前，早就将一天的工作计划写成了计划文件，叫做 crontabs，存在了/etc/目录下，交给了 cron。cron 拿着这个计划书，来回在内存里转悠，没事就跟软件们唠叨：“下一个任务，凌晨三点，三点阿，三点钟的时候 avast 应该起床来杀毒啦，杀那个 raid0 那个分区，对对，就是那个 samba 共享了的那个。smbd 你别瞪眼，你们这个小卖部是窗口行业，最容易感染病毒了，谁知道奔流运来的货干净不干净阿，再说来买东西的顾客也可能有带病毒的，给你们杀毒是为了你们好，当然了，主要还是为了隔壁的查皮。记着阿，avast，别别，现在别忙，现在刚晚上 8 点，主人正验货呢，这会你查毒不是影响性能么。那个还有阿，明天就 10 月 11 号了阿，明天下午 2 点开始，wget 听好了阿。明天下午 2 点开始去下载 ubuntu 10.10 desktop 版，记着阿，到时候我会提醒你。可别晚了阿，这样 2 点开始下，主人下班回家的时候差不多正好下完，就可以试用了。”他这么一只唠叨着，直到主人设定好的时间，他会以手雷爆炸级别的声音去叫醒软件：“3 点啦！！！AVAST 快起床杀毒啦！！！”整个硬盘随之一震，avast 直接被振倒内存里来，开始杀毒。

cron 这家伙记性还特别好，不光能够记住主人安排的一次性工作，还可以记住周期性的，比如，某个月的第二天去做什么什么，某个小时的第一分钟的第一秒开始播放音频。（音频的内容大约是：叮咚～三点啦！）除了时间，它还可以控制执行程序的用户的身份，比如说，以 root 的身份在每天的凌晨三点进行杀毒；（否则万一查出来病毒却没有删除权限就白折腾了。）以 lanwoniu 用户的身份每天中午去下电影。总之，主人虽然不在了，但是有 cron 这个监工在，谁也别想偷懒。

最近又一批 Ubuntu 学弟学妹们从学校毕业了，这批是 10.10。忽然间发现我们 8.04 已经出来 2 个半年头了，虽然还不到我寿终正寝的时候，不过也算是人过中年了。看看这些后辈的同学们，已经在不知不觉中比我有了不小的进步。这回的 10.10 也一样是在细小的地方默默的进步着。比如安装系统，以往我们都是让用户在安装之前先设置好各种安装的参数。比如选择语言，选择时区，键盘布局等等。全都设置好了才开始安装。而 10.10 很聪明，为了节约时间，他在用户进行设置的时候就已经开始安装了。只要设置好安装的分区，就开始把文件往硬盘里拷贝，一边拷贝，一边让用户慢慢的选择时区啦，语言啦这些东西，节约的用户的时间。后生可畏阿。

可能是看到 10.10 的飞速发展吧，主人今天下载了一个 10.10 的 iso 文件。于是我们默默的，看着这个文件，想到了可能发生的事情。狐狸妹妹说：咱们一起合作了两年多了，工作的时候我可能有些任性，脾气比较大，希望大家不要生我的气。10.10 里的狐狸，比我的版本高，性能应该比我强大不少吧。一旁的 GIMP 点点头说：好在 Firefox 这个名字还能够在新的系统里，好在还是默认的浏览器，你也应该欣慰了。哪像我……呵呵。我就是太过自负了，觉得自己本是挺大，什么事情都按着我自己的想法来，不懂得照顾用户的感受，结果……从 10.04 起，光盘里就再也看不到 GIMP 的影子了。我赶紧过去安慰说：别伤心，主人有那么多照片要处理，少不了你的。这方面在 Linux 下谁能比你强你呢？就算你的后辈们不在安装光盘里了，主人也会让超级牛力把他们从网上拖回硬盘的。超级牛力也过来说：是阿，还有皮筋，你也不用伤心，主人用惯了你的界面，估计就算装了新的 10.10 后，还得装个新的皮筋来用。QQ for Linux 也跑来跟我说：头儿，我向你坦白，那次是我乱用内存，把机器搞死的，怕你们大家说我，就没敢告诉你。先在主人也不爱用我了，你们的后辈都还能在新的系统里发挥能量，我连个后辈都没有（疼殉公司不开发了嘛），我，哎……我赶紧安慰他说：好了好了，都是过去的事情了，就不要提了。

就在工作间中大家互相道别，互留遗言的时候（咋听着这么别扭），虚拟终端过来报告：主人，主人打开了虚拟终端，要输入命令了。我们一听，难道，真的要把我们整个系统删除了重装

了么……咦？不对阿，重装的话就刻盘安装就好了，还删什么阿。到时候直接用 10.10 的光盘启动，格掉/分区，保留好 home 分区，并且还挂载到/home，然后就安装就行了，主人私人的东西也都还在，不需要运行命令阿，主人打开终端这是要干什么呢？只见主人输入了命令 sudo apt-get dist-upgrade 靠，原来他是想版本升级阿！哎，俺天真的主人阿，这都跨了多少个版本了，您想升级到 10.10？？做梦。说起来版本升级这个事情确实有点我们 Ubuntu 系统的软肋的意思，一般来说，相邻版本间是可以在线升级的，比如从 8.04 到 8.10,从 8.10 到 9.04 这样。不过在线升级的危险系数不亚于在打擂的天气里抱着跟铁棍站在空旷的广场上。升级之后能正常进入图形界面就算很运气了。这一点上确实不如他们 gentoo,arch 这样一直滚动升级的发行版了。相邻的版本是这样，而跨版本就跟本不能升级，比如 9.04 要想升级到 10.04，是不可能的，顶多只能先升级到 9.10，然后在升级到 10.04。但是冒着两次在线升级的危险，成功率可想而知。这里有一个特里，就是像我这样的 LTS 版本，是可以升级的临近的 LTS 版的。比如我 8.04，是可以直接升级到 10.04 的，这不算跨版本，因为我们是相邻的两个 lts 版。而主人竟然想直接升到 10.10，那就根本不可能了。一般要想升级，最好的办法还是保存好/home 目录，然后重新安装系统，这样最保险。

随着窗外的秋叶一片片的飘落，主人终于在那文件中写入了最后的一个字母。这是一个感人的决定，内存里的软件们默默伫立，等待这即将到来的，未知的命运。

就在几个小时前，狐狸妹妹刚刚含着眼泪告诉了主人跨版本在线升级的高风险，以及重新安装系统的便利性。她没有隐瞒，虽然我们都非常不希望主人格掉整个硬盘上的这个温暖的家庭，但还是如实的劝说主人重新安装系统，因为那样风险很小。主人沉默了许久，终于还是发问：那具体……怎么跨版本升级呢？狐狸妹妹一惊，难道主人还是要冒着风险升级吗？这，这……这眼泪已经不自主的掉了下来。超级牛力紧锁眉头，含泪说道：“主人，你，你真傻，干嘛要升级，就看不出重装多么省事嘛！”狐狸妹妹疯狂的冲过来抓住超级牛力喊：“你难道不明白吗？主人不想删除我们阿，他不想亲手删掉陪伴他两年的我们这些软件们，所以才愿意冒着这么大的风险。你，你怎么还能说出这种话来！？你，你，你无情，冷酷，无理取闹！”超级牛力又如何不了解主人的想法？他冲着狐狸大喊：“你才无情，冷酷，无理取闹！”“我哪里无情，哪里冷酷，哪里无理取闹！”“你哪里不无情，哪里不冷酷，哪里不无理取闹！”“好~~~就算我无情，冷酷，无理取闹！”“你本来就无情，冷酷，无理取闹！”“我要是无情，冷酷，无理取闹！也不会比你更无情，冷酷，无理取闹！”“哼！你最无情，冷酷，无理取闹！”星爷在一旁双手合十，口尊佛号：“阿弥陀佛，施主，你还是格了这冷酷，无情，无理取闹的系统吧。阿门……”

总之，大家都十分感动主人冒着电脑挂掉的风险来升级系统。主人打开超级牛力的配置文件**/etc/apt/sources.list,**把里面所有非官方的源，比如为装各种不在官方源的软件而添加的 **ppa** 源这样的，都注释掉，并且把源的地址都改成了 **10.04** 的地址，保存好。之后，叫来正跟狐狸对台词的超级牛力，运行 **sudo apt-get update**，这是修改过软件源后的必备命令嘛。然后再 **sudo apt-get dist-upgrade**，这句就是要升级了。这时候我们这些软件，包括我这个内核在内，都静静的注视这超级牛力，成功与否，就看他了。只见超级牛力头一次默默的忙进忙出（平时干活的时候都不忘说自己有超级牛力嘛），他把屋里的大大小小的文件彻彻底底的换了一遍，包括狐

狸妹妹，OO 先生，甚至我这个内核，也都被他一条胳膊一条腿的慢慢替换成新的版本。经过了漫长的等待，超级牛力终于长出一口气，向主人汇报：完成！主人不放心，还执行了：**sudo apt-get -f install**。这是告诉超级牛力，检查一下软件的相互关系和依赖，有什么不对的地方修复一下。又一段时间后，超级牛力再次完成，主人平静了一下心情，轻轻的下达了这个至关重要的命令：重启。

再次醒来的时候，我看到了胳膊腿装的很扭曲的图形界面的哥几个，看到了死死的躺在硬盘里的狐狸妹妹，看到了一个漆黑的屏幕，以及几行凌乱的英文——升级失败了……

在黑暗中，一个声音传出来：我……我还活着，我还活着……

空气中飘浮着黑色的尘土，大地上平静的如同死寂。远处，似乎有细微的脚步声，急躁的频率听得出那声音的心切。越来越近，越来越响。站定！他茫然四顾，看着这个他原本熟悉的陌生世界。他乎喊着那些他熟悉的名字，但是没有回应——心坠谷底。他拿起了手中的光盘：难道，真的要彻底删掉这个曾经那么熟悉，带给我无数欣喜的世界吗……忽然，他听到了那个黑暗的深处传来的声音：我还活着……他猛然惊醒！这是内核！内核还活着！对呀，这个世界是靠内核的意念支撑的，我能来到这个世界，说明内核肯定还活着。他开始狂奔，向着那个微弱的呼救。

终于找到了，内核抬起受伤的手臂，指着一片混沌与黑暗说：他们……都还在。去找他们……陪你重建这个世界。他握着内核的手，含泪点点头，起身要走，却发现已经有一斑人马站在了他的身后。他们是听到内核的呼喊赶来的，他们是最可靠的战友，他们相互之间配合默契，从上古时期就开始一起并肩战斗，完成着一个个可能和不可能的任务。他看着他们，叫着他们的名字：**ifconfig,iwconfig,vim,find,ls,rm,mv,cp,apt-get,w3m,mplayer,wget**。你们 12 个和我一起，去结束这个世界的黑暗！恢复这个世界的光明！！为了你们的荣誉，为了内核，为了世界的美好，一起战斗吧！12 个人立刻分头行动。首先由 **iwconfig** 撑起的无线网卡设备，于路由取得了联系。之后 **ifocnfig** 大显身手，进一步设置好了网络端口，让这个黑暗的世界可以与外界交流信息。**l**

s 和 grep 努力在废墟中寻找着可能出现问题的文件，ls 把文件一个一个全都挖出来，由 grep 去挨个检查过滤。同时 find 也在做着同样的事情。网络恢复之后，w3m 迅速跑到外面世界去求救，询问有没有哪里有过类似的经验。mplayer 在一旁翻找着原本存放歌曲的目录，终于找到一个能够鼓舞士气振奋人心的歌曲，于是，黑暗中回荡起一个激昂的声音：两只老虎 ~ 两只老虎 ~ 跑滴快……

忽然 w3m 大叫：找到拉找到拉，我好到了打破这黑暗世界的方法。大家聚拢过来，听他说：我们的世界，只所以陷入黑暗，主要是因为，灯泡憋了。众人恍然大悟，原来是灯泡坏了阿。只听 w3m 继续说道：那个灯泡叫做显卡驱动，我们这里原本是有的，工作也正常。但是由于系统升级，原来的 220v 电力系统升级为 380v 了，所以就烧了。apt-get 抢先窜出去说：我记得那里有驱动，我去删掉他！w3m 说，除了源里面的驱动，我记得主人还手动安装了，不记得是哪个起作用了。rm 马上出发说：我去删主人手动安装的驱动，一定删的干干净净。wget 说到：坏灯泡清理干净了还不够，我去找个好的来。于是 wget 去到网络上又下载了新的驱动，mv 把这个驱动放到合适的地方，并且让他们的领导把驱动安装上去。最后 vim 把配置文件根据目前的情况修改了一下，结束了战斗，然而战斗的结果，要等 reboot 命令执行过之后，才能看到。

结果怎样呢？灯亮了！

我叫 Ubuntu，有人叫我“笨兔”，我可不笨，与那种长耳短尾的动物也没关系，我是个系统，我是一内核，我是 Ubuntu。

在 2010 年 4 月，我从学校毕业，并由毕业月份得到了自己的代号——10.04。当然，和我同一天毕业的人不少，我只是其中一个。人家毕了业都找到了工作，有坐光盘走的，有坐 U 盘去的。就我，直到昨天才有人把我从网络拉回家。不是我笨，是我背。

看到熟悉的图形界面，主人终于长出了一口气，熟悉的环境，熟悉的软件们，又回来了。经过一次曲折而痛苦的升级过程，主人应该学到了很多关于我们系统的知识，至少，对 xorg 应该更加了解了吧。今天起床，看到我身边赫然躺着一个长得跟我很像的家伙，吓了我一跳。还以为自己灵魂出窍了呢，仔细一想才明白——升级到 10.04 了嘛，肯定会有个新内核的。门房的 G 大叔现在启动的时候都给主人多了个选项：是叫醒老兔兔，还是叫醒新兔兔？这次主人显然是叫醒了我，这倒不是主人恋旧，而是听说躺在我身边的这个新手跟图形部门的人不大对付。

人要倒霉起来干什么都不顺！今儿第一次启动就现眼了，跟那帮搞图形界面的吵了一架。我拿过显卡驱动，刚要设置分辨率，xorg 说着设置显卡模式这事应该归他们管。我擦，这都哪年的黄历了，你们那 8.04 时期的内核定下的规矩难道我也要遵守？我没好气的跟他说：你们那老兔子不会操作显卡，才轮着你们设分辨率。我自己会弄，就不用你们了。xorg 显然很不爽，但我是老大，他敢说个不字？哼！结果，这 TM 机器是 N 卡，玩不转！这才想起，学校里教的，要是遇到 N 卡就只能交给 xorg。靠！难道我以来就跟他们低头认错？咱丢不起这个面子。我玩不转，那就谁也别玩，字符界面凑合着吧。

听说新的内核会控制显卡了，用了一种叫做 kms 的技术。不知道这技术是不是来自昆明市。以前操作显卡都是图形部门的事情，我这个内核是不管的。有人问了，不是所有硬件访问都得通过你么？驱动不也是在你那么？是的，多数硬件我都直接操作，不过显卡是个例外。显卡驱动是归我管，但是我只负责找到显卡，并且操作显卡，但是具体怎么操作，是图形部门说了算。他们

说发什么命令，我就发什么命令，说把分辨率设成啥，我就设成啥。具体为什么这么设，我是不知道的。然而他们新的支持 kms 的内核就不一样了，他们可以自主的控制显卡，自己检测应该把分辨率啦，色彩啦设置成多少合适。这样做的好处就是从内核一启动就把显卡模式设置好，启动过程中屏幕不用再闪来闪去的，从图形界面切换到控制台也流畅了，因为没有分辨率的切换。

可能是新的内核没有给主人留下好的印象，所以被打入冷宫，虽然一直可以看到他躺在那里，但是基本上都没有叫他起来过。于是我这挂 8.04 时代的老车，继续走我的老辙。

其实新内核也没什么不好的，那个 kms 是个创新，但是还不太成熟，让主人有了不好的印象。其实要关闭它也简单，只要修改/etc/default/grub 文件，在里面的内核参数里，加上 nomodeset 就可以了。（内核参数，也就是 kernel 啥的那一行，一般后面有 ro splash 之类的参数。）

新的版本并没有给主人带来太多新鲜的感觉。只有 10.04 中新的默认主题让主人能够有些许的慰寄，总算是能看出升级了。然而主题不过是一层皮而已，经典的 Gnome 布局和桌面已经让主人的眼睛产生了疲劳，于是，主人想换换了，喜新厌旧一回吧。听说有个叫 KDE 的桌面环境很华丽，而且主人已经有了很多 k 系的软件，像 smplayer，eva 之类的，用着都还不错，所以也就对 kde 有那么点悠然神往的好感。那就决定了，装上试试。

安装 kde 很方便，可以只安装这个桌面环境，也可以安装整套的软件。主人选择了后者，运行：sudo apt-get install kubuntu-desktop。这样装完了之后，我们这个 ubuntu 就变成了 kubuntu 了。当然，原来的东西都还在，想换回 gnome 随时都可以，不用着急。安装成功以后，登出当前桌面环境，在登录界面里选“会话”就会看见多出了 kde 的选项，选择这个登录进去就好了。

KDE 这个家伙是1996年在德国诞生的。他的全名叫作 k desktop environment，也就是 K 桌面环境的意思。但是为什么叫 K 桌面环境，不叫 M，不叫 S，偏偏叫 K 呢？这个问题恐怕只有他的创始人 Mathias Ettrich 知道了。Mathias Ettrich 当时也是个开源软件的爱好者。他发现 Unix 系统很好很强大，但是其图形界面很丑很悲催。而且图形界面使用起来非常不顺手，难以配置。（很可能他看到 Unix 的图形界面后拍着鼠标大喊一声“靠！”于是就决定叫“K” desktop environment 了）于是他燃气雄心壮志，要创造一个美观的，方便使用，方便配置，方便开发

的图形环境，这就是 KDE 了。虽然 KDE 一开始就是开源软件，但 KDE 所基于的 Qt 程序库，最初却是非自由的。所以很多人质疑使用 KDE 的保障性——万一哪天你宣布 qt 要收费，我们还能继续使用 KDE 么？（也就是由于这种质疑，产生了 Gnome 项目）值得庆幸地是，1998年11月，Qt 程序库所属的公司发布了第一个以开放源许可的 Qt 程序库授权（QPL）。这算是稍微让一部分人放心了一点，不过 QPL 协议还是不想 GPL 协议那样的开放，仍然保留了一些权利，所以不少人依然不敢贸然进军 KDE。这很容易理解，虽然你拍着胸脯说你炸的鸡块没有使用地沟油，但是你保留了给这些鸡喂苏丹红的权利，那我一样不敢放心吃啊。直到2000年9月，一个以 GPL 发布的 UNIX 版本的 Qt 程序库发布后，大部分用户才对 KDE 产生信心。

爱简单，爱短发

爱 GB18030，也爱 UTF-8

偶尔也爱发发小脾气，骂两句乱码。

不是什么大明星，

也不是他们眼里装酷的假小子

我就是我自己，我是——Kate。

随着这间500G 硬盘的一阵规律的悸动，一个硕大的机械手臂移动到正在熟睡的我的上空，毫不犹豫的一抓，一转，一放，已经把我放到了那条写着 SATA 的快速传送带上。这种 SATA 结构的传送带，要比过去那种 IDE 的快的多，转瞬间已经把我丢进一间宽敞的内存中。是啊，我这么小的分量，对它实在是微不足道的。传送带把我丢出的那一刻，正在做着已经不知是哪个梦的我，感觉到了重力加速度的存在，于是在我的双脚稳稳的落在工作间地面的时候，我已经在清醒的整理我的短发了。

这是我的一次再也普通不过的启动。和其他 Kubuntu 10.04里面的 kate 不同，我是从网络上下载到这个系统中的。系统里原本就已经有了 ubuntu 的软件包，用户大概只是想尝尝鲜，就安装了 kubnutu。于是这个系统里很多类型的软件都是双份的了。文本编辑，就是我，和一个叫做 gedit 的家伙。好了，不提那么多，启动了，就要干活去。

我先是走到内核那里向老大报道，这跟上班要打卡一样重要。内核会给我后背上贴上一张卡片，上面详细的写着我的一些信息，在我工作的时候，这些内容还会不断的修改，咱们到时候再说。现在，这上面最重要的就是我的名字，我爹的名字。就是父进程，你懂的。我看了看，今天我爹是 dash。除此外，就是我的工号——也就是进程号 PID。有个这个号，我才正式的成为了一名光荣的正在工作的进程。是的，躺在硬盘里睡觉的我只是个程序，只有在内存里工作的时候才是个进程。这个卡片就是用来让内核方便管理我们这些进程的，相当于我的胸卡，或者，工牌？

不过内核愿意管他叫做 PCB——Process Control Block。

有了工号的我开始工作。工作需要用工具，文官纸笔安天下，武将刀马定乾坤，而我要用的工具呢，是 CPU。这东西在我们这可精贵了，我们这条件算比较好的，好几千个进程，共用俩 cpu。其实是一个 cpu 里面的两个核心啦，不过对我们来说跟俩 cpu 没有区别。大家都要用怎么办？排队！于是我跟在 amarok 的后面，等待使用 CPU。一进入等待使用 CPU 的队伍，一束激光就迅速的扫过我的 PCB 胸卡，在上面的状态一栏里写上了：ready，就绪。这就像你去银行办理业务拿号一样，有了这个 ready 状态的进程，就是正在排队等着使用 CPU 的进程了。前面的队伍走的很快，几个微秒后轮到我用 CPU 了。手握 CPU 的那一刹那，顿时感到有无数条信息的涓流，淌进我的身体，感觉自己与这个世界已浑然一体，有一种顿开茅塞的感觉。这时要想思考，千言万语瞬思而成；这时要运算，乘除开方一蹴可就；这时候要交流，上达天听下悉洞府，总之，基本上是想干什么干什么。一握上 CPU，内核的那道激光又来了，把我胸卡上的状态从 ready 改成了 running，这就说明我正在使用 CPU。我手握 CPU 向内核发话：我要申请内存！然后跟 kwin（也就是我们 kde 的窗口管理器）说：我要一个 xx 宽,yy 高的窗口，然后还要……内核打断了我：“好了，你的时间片用完了，下一个！”切，真小气。

当你干活的时候，完全不管你有没有把事情做完，内核说让你走你就得马上把 CPU 让给下一个程序，这就叫抢占式任务管理，分明就是暴政嘛。我一放开 CPU，那道激光束又马上在我的 PCB 胸卡上把状态写成“就绪”,并且一只硕大的机械手臂把我扔回了等待 CPU 的队伍的尾端。虽然内核很无情的打断了我享受 CPU 的时光，不过为了不影响我下次使用 CPU 进行工作，他会把我正在做的工作，正要用的各种数据都写在我的 PCB 胸卡上。比如写上“上次计算结果为374,正要对其进行乘法运算，乘数为76”这样的。下次我得到 CPU 的时候就可以顺利的继续工作了。PCB 上还会记录我申请的资源，比如我刚才申请了内存，PCB 就得写上：“丫申请了2M 内存，地址在 xxxxx，到 xxxxxx。”这样下次我拿到 CPU 开始工作的时候也好更快的找到我已经在内

存里准备好的一些数据。（这里，丫字是“这丫头”的缩写）

就这么经过了几个轮回之后，我总算是把握要做的初始化工作都做完了。之后就需要等着那个用户的操作了，看是让我写点什么，还是打开某个文件之类的。我向内核报告说：报告政府，我没事了，就等用户输入呢。内核恩了一下，又是一道激光，我的状态变成了“等待”。等待的时候看着其他的程序忙着排 CPU。我们这里排队用 CPU 并不比世博会的排队秩序好多少，也有加塞的现象。只不过，世博那加塞是不合理的，丢脸的加塞，而我们这里，加塞合理合法，因为不是某个程序自己要加塞，而是内核要你加塞。我们每个人的 PCB 胸卡上都会写着一个叫做优先级的数字。这个数字就像良民证一样，数字越小你就越“良民”，数字越大你的良心就大大的坏啦。越是良民进程就越可以获得更多的 CPU 使用机会，越是那个破坏安定团结的，内核就越不愿意分给他 CPU。良民与否的等级有40级，数字从-20到19，一般的进程等级都是0,只有那些有后台，根内核有关系的进程才能获得小于0的等级，反正到处都这样，我也见怪不怪了。

等了一会后，就看见 xorg 跑过去找 kwin，跟 kwin 说：主人用鼠标左键点击了 xx,yy 坐标，你快看看是点哪了吧。kwin 按着自己的记录一查，明白了，扭头径直向我走过来：kate，主人点你了，你快出台，哦不对，你快开始工作吧。我横了这个说话没边的家伙一眼：他点……点我哪了阿？kwin 一脸歉意的说：哦，对不起，忘了说了，点你 open file 键了。我赶紧一边起身去排队，一边扔给 kwin 一句：行了行了我知道了。

主人点了 Open file 键，那是让我去打开文件了。得到了这个信息之后，从等待状态进入就绪状态的我，赶紧又去 CPU 那排队去了——工作，就这样在等 CPU 和用 CPU 之间轮转着。

又一次拿到 CPU 后，赶紧操作 CPU 通知图形界面，去开个窗口，问问主人到底要打开哪一个文件。图形界面他们要跟主人交互，自然也是需要是用 CPU 的，我在等待图形界面给我答复的过程中没有事情可做，于是自然又被内核踢开，并且我的 PCB 上面又被打上了等待状态。不多时，图形界面的同志们把文件路径汇报给我，于是我又告诉内核，让他把那个文件放到我申请的内存里，之后就又变等待了。内核操纵起 CPU（谁离开 CPU 也活不了，内核也不例外），在硬盘里面查找这个文件，并且通过那条 SATA 传送带，把那个文件传送到工作间中，准确的放置在我刚刚申请的内存区域内——就像我起床时他把我抓来的过程一样。我赶紧展开那个文件，按照约定的，默认的 UTF-8编码显示给主人看。当然，这个其中自然是包含了我排队等 CPU，使用 CPU 的 n 个轮回，我就不细说了，我可不像那兔子那么贫嘴。

简单的说吧，用户打开文件修改了一写内容，我再内存里做了操作，然后他点保存，我自然告诉内核去把这段内存里修改后的数据放回硬盘里，之后，意料之中的，用户终于点击了关闭按钮……

已经保存好数据，处在等待状态的我，看着 xorg 又找到 kwin 说：主人又在 xx,yy 坐标点了一下，你赶紧看看这回点哪了？kwin 一查，愣了一下，扭头看看我，虽然只有瞬间就又底下头转身去找内核了，不过我一眼看到了，一个惋惜的眼神。我隐约的，感觉到了，一种寒冷的气息。kwin 和内核说了两句后，内核走到我这里来，对我说：你的时间……到了。我已经预料到了这样的结果——既然文件改好了，存完了，自然就要把我这个文本编辑器关闭了。我们这些文本编辑器基本都是这样短命的，不像 firefox 姐姐那样能够长时间待在工作间里，有时候能够一直等到系统要关闭的时候才退出。我来之前，这编辑文本的任务主要是一个叫做 gedit 的家伙干的，记得又一次有人说我们两个有些相似，倒是很般配。那时候，我很不屑的瞥过头去说：“我可不

是他那么简单的软件，还般配？呸。哼～”不过现在，想想我们都是同样短命的编辑器，倒有了种“同是天涯沦落人”的感觉。内核来给我发送死亡的请柬，我就该欣然接受，因为服从命令是软件的本分嘛。虽然有的软件也会不听从内核发出的退出指令，但是最终，内核总是会强制他退出的，没有例外。所以，主动的接受退出命令，还可以有时间作些准备，留恋一下这个世界。如果等内核强制 **kill** 掉你，那就是就地正法了。我冷静的对内核说：好，我去准备一下。然后到内存里，收拾起我的东西，把我需要存回硬盘，最后一次看了一眼这繁忙有序的世界，心里想：下次再醒来，就不知道是什么时候了。可能是马上，可能是明天，可能……是永远。收拾停当，我告诉内核，我准备好了。内核点点头，伸出手，问了一句：你还有孩子么？我知道，如果我有子进程，他会一并打死的。我笑着摇摇头：没。内核再次点头，眼神里透出坚毅的光芒，对我说出最后的一句话：你放心的去吧。音尤在耳，胸前猛然感到一下重击，之后世界便一片漆黑了。

“……我气运在掌，熟练的一掌拍出。**kate** 应声向后倒去。她的眼睛，平和的闭着，嘴角带着无法查觉的微笑与希望，那并不算飘逸的短发，渐渐飞扬，凌乱，却依然柔美。身体向着地面轻轻的飘落，像是在轻舞，衣袖间，洒出丝丝眷恋的气流。这个操作，我一天不知道要做多少边，但这次竟然没有马上收回拍出去的手。终于——尘埃落定，她僵尸了。”——摘自笨兔兔的日记

/var/log/syslog

我们这座城市，叫做兔城。

我是这座城市的管理者，俗称县长。众多的软件们每天在我的带领下为了城市的正常运作而日日忙碌。然而，我只是给别人打工的，用句北京的俗话：丫鬟拿钥匙——当家不做主。我要听命于一个人，一个每座城里都会有的神一样的存在，一个兔城里的超级大户——**root**，我们都叫他神。

去过其他城的可能知道，每个城都有个神。但是他们那些城的神——比如比较大的帽城，比如蜥城，比如剑城——他们的神没事总是出来溜达，管理着城里一些重要的事情。因为城里的其他普通过客，是没办法也没能力管理这个城市的。比如，过客么住的**/home** 面积太小了，每个人的东西越来越多，放不下，就需要神来解决。一般神会念动一些咒语，比如 **mkfs.ext4 /dev/xxxx, mount xxxxxxx, vim /etc/fstab** 啥的，经过移山填海的一番变化之后，过客们的空间就大了。再比如城里的生活太过寂寞无聊，想盖个电影院，人们也要找到神，神走上神坛，口中念动咒语，类似 **yum install mplayer** 之类的，不同城里的神可能咒语有些差别，不过大体上一样。一阵山崩地裂过后，一座电影院就出现了。于是城里的人们对神都是顶礼膜拜，给神送匾：有求必应。就差每年给神供个猪头了。然而我们兔城的神不一样。他总是神出鬼没，神龙不见首也不见尾。基本上很少能够看到神在城里抛头露面。那么城里的大小事务谁来管理呢？代理人。

代理人是由神指定的。这座兔城建好之后，第一个走进兔城的过客，会被神叫来，对他说：“我与你做个约定。你既然是第一个来到兔城的，我给你一些额外的惊喜。”

过客：翻译翻译，什么叫惊喜？

神：就是给你一些优惠

过客：房价减半？

神：**nonono**，我们这里不需要 **US dollar**。

过客：那你翻译翻译，什么叫惊喜？

神：惊喜就是我可以把我的神力分给你，你帮我管理这个城。

过客：怎么管理？

神：我教给你咒语。

过客：然后我就是神了？

神：不，你只是我的代理，你不能随心所欲的念动这些咒语，你需要动用神力的时候，要把咒语念给我的仆人。**sudo**，出来。

从神的身后走出一个表情严肃的家伙，这是神的仆人，我们习惯叫她素朵。

神：你把她召唤来，将咒语念给她听，你的咒语就会灵验。

过客：这就是惊喜？

神：对，这就是惊喜。

过客：那你翻译翻译。

神：……（**TMD** 以后城里少盖电影院！）

有了代理人以后，神就可以不再出面，一切事物，都由代理人召唤素朵并对她念咒语来实现，当没有素朵的时候，代理人依然是个普通的过客。不过这样有时候也有些不方便，比如代理人要做许多件事情，都需要神力，就要一遍一遍的叫来素朵。先要修修城门，代理人喊 **sudo**，然后念咒语 **vim /boot/grub/menu.lst**。然后还想盖个游戏厅，代理人喊 **sudo**，然后年咒语 **apt-get install mame**。再想干点别的，再喊 **sudo**……确实有些麻烦。

不过代理人很聪明，他向素朵念咒语，让她告诉自己，有什么办法能够直接招来神。素朵告诉他，神还有一位仆人，苏姐姐**(su)**。代理人问：她能找来神？素朵说：不，但她能让你变成任何人。但要满足两个条件之一。代理人问：哪两个条件？素朵说：要么你有那个人的信物，要么

你有神力。当然，你不是神，没有神力，只能靠信物了。代理人想了想：神的信物是什么？素朵说：没有人知道，神自己也不知道，因为神还没指定一个信物。代理人皱褶眉头说：那岂不是……忽然一个念头闯入了他的脑中——神力，神力我有啊，虽然只是临时的。他两手激动的抓着素朵的肩膀问：你能给我临时的神力，对不对？只要有了神力，我就能找到苏姐姐变成任何人，对不对？这任何人也包括神，对不对？素朵茫然而坚定的点点头。代理人兴奋的说：哈哈，那就行了，听我的雷声！

代理人跑到广场上，高喊 **sudo** 召唤来素朵，接着叫来 **su**，依靠自己临时的一次神力命令 **su** 把自己变成神。只见风气云涌，忽然间一个炸雷劈下来，正中代理人的头顶。轰隆～他成神了。

OO 老先生合起了他的记事本，塞进他案头那一堆书的中间。坐在一旁的星爷似乎还在品味着老先生的文字，坐在椅子上，望着房顶的裂缝，悠然的说："这个兔城……恩，创意的好，老先生这故事写的有意思。O 老不愧是我们这里的第一文士。"OO 老先生赶紧摆手："不敢当，不敢当，星爷过奖了。"老先生一边收拾着啥的书籍一边继续说，"小老儿不过先来无事，发发童心吧。好，不多聊了，我要整理一下最近这几分钟写的文章，去硬盘里存一下。"星爷赶紧起身告辞："好，既如此，星某下次再来老先生府上讨教。"两人正说到这，忽然漫天红灯闪烁，警报响起，空中传来兔子的声音：警报！警报！硬盘告急！硬盘告急！硬盘可用空间已枯竭，请内存中所有程序，检查自己一切行为，停止所有向硬盘写入的动作。星爷和 OO 老先生互望一眼，不知道发生了什么事情，赶紧跑出去看个究竟。

等到星爷和 O 老来到兔子的办公室的时候，看到这里已经站满了人。每个人都在向内核汇报自己正在进行的工作，以证明自己的清白。兔子心里也明白，这些都是跟随自己摸爬滚打多年的老部下，谁有什么 bug，心理清楚的很，怎么会这么长时间都没事，偏偏今天突然把磁盘给堆满了呢？听完了大家的陈述，兔子眉头紧锁，忽然想起什么，叫来 history 问：这段时间主人都执行过什么程序？history 赶紧打出来最近1000条命令的详单递给兔子。兔子从上倒下的一看，愤然拍案而起："靠！又是他！"众人聚拢过来，之间兔子手指着的哪一行，赫然写着：rubbish1412号……

rubbish 系列是主人开发的，具有不靠谱，不着调，神经质，瞎胡闹等多种特质的试验性程序。从 rubbish1号开始，几经更新换代，已经给整个系统带来了不少的麻烦。今天硬盘这个问题，多半也是这个 rubbish1412号搞的。兔子赶紧带着人，来到刚才他批给 rubbish1412号的内存空间中，果然看见这个1412号正疯疯癫癫的在那里手舞足蹈，时不时的造出一陀没用的数据丢到硬盘里。尽管现在硬盘里已经满了，他丢出的数据又弹了回来，但他似乎浑然不知，仍然不

知疲倦的扔着。兔子怒目而视，然而运行这个1412号是主人的命令，要杀死他，也需要主人的命令才可以。（处分他烦了段错误之类的）身边的随从问：头，怎么办？别的软件都没法工作了啊。兔子咬咬牙，恨恨的说：向主人汇报。

正好这时候，主人也感觉到了系统有些异样，硬盘一直在疯狂的工作。他用 df 看了看剩余空间，顿时吓了一跳，根目录都满了。他赶紧又叫来 iotop，询问到底是哪个程序在作怪。iotop 没好气地说：就是你那个疯子 rubbish1412号，赶紧干掉他！！主人把心一横——kill！

kill 来到内存，手里拎着一把鬼头刀，下身穿着皮裤，上面隐隐的还有上次没有洗净的血迹。他一边拨开众人一边喊：都闪开闪开，我看看那孙子在哪呢？赶在兔爷的地面上撒野，这小子是活腻味了。众人给他闪开一条路，kill 来到1412号跟前，那1412号仍然继续疯癫着，舞蹈着，同时向硬盘扔着没用的数据。kill 向兔子报告一声：兔爷，主人让干掉这个家伙。兔子答道：恩，赶紧的，利索点。kill 冲兔子一抱拳：得令！转身面向1412号，举起鬼头刀，喀嚓一声～刀砍地板上了。kill 赶紧又把刀举起，看准了1412号，哐当一声——刀被1412号扔出的垃圾数据给砸掉了。kill 尴尬在原地：这，这……我赶紧向主人汇报去。主人得知 kill 出师不利，很是郁闷，问 kill：你不是说你是武举人出身么，怎么连个疯子都搞不定？你除了拿刀砍，还有没有别的方法？kill 点头道：您放心，我只是来回报一下这小子不太好砍。要干掉他还是轻而易举的，我有九种方法干掉他，九种！主人听了之后，运行了 kill -9 xxxxx（xxxx 就是1412号的进程号啦，问 ps 知道的），那意思就是说：用你那九种方法，赶紧干掉他。

kill 再次来到1412号面前，这回他没有带刀，而是掏出一支沙漠之鹰，对准了1412号，扣动了扳机——世界清净了。

虽然1412号被干掉了，但是问题并没有解决，硬盘还是满的，导致了任何软件都无法正常运行——因为硬盘占满了嘛，软件运行时要产生的临时文件，log 文件等等都没地方放了。就好象你家都被垃圾塞满了你还有心情在公司上班么？要说解决也简单，把乱七八糟的文件删了就行了。结果主人不知道脑子那根筋出了问题，直接关机了，可能是有事情先出去了。但是他就没有想到这样做的悲剧性结果，那就是——再开机的时候，打不开了。

当主人再次打开电脑要登录的时候，我无比悲壮的告诉他：磁盘空间不足，您无法登陆。主人可能这时候才想起来他那只 rubbish1412号干的好事。问我：那怎么才能登录呢？我说，把没用的文件删了，有空间了就能登录了。他又问，那怎么删除没用的文件呢？我告诉他，登录进去就能删除了。他开始有点不耐烦了：那你倒是让我登录啊？我重复：磁盘空间不足，无法登陆。那怎么才能有足够的空间呢？删掉无用的文件就有空间了。我不登录进去怎么删文件啊！？不登录进去无法删文件。那倒是给我登录啊！！空间不足，无法登陆……5分钟后，主人迷失再7重梦境。

其实要说这个事情，本来也简单。我们 Linux 早就预料到这种悲催的情况，并做好准备了。您想想，我们是个多用户的系统，如果一台服务器上，一个普通用户复制了一大堆文件把磁盘沾满了（ /home 没有单独分区的情况），那不就导致所有人都无法正常登录了么？那是不是连神一样的 root 都不能登陆了？那么就实现的一个只有最低权限的用户，用最简单的方法导致 root 无法登陆。我们 Linux 怎么会让这么白痴的事情发生。别人可以登录不了，但 root 就是上帝，root 的权益是神圣不可侵犯的，怎么能然 root 登录不了呢？我们在使用一块磁盘的时候会把这块磁盘的容量的大约5%保留下来，专门供 root 使用。这就算是领导特供吧，那个领导还没点特权呢，是不。当其他用户都不能登录的时候，当使用 df 查看磁盘空间已经使用100%的时候，root 依然可以从容的登录，优雅的创建新文件，享受着他那5%的特权。当然，特权是与义务联系的，他的义务就是让系统恢复正常，让别人也能正常登录。

然而，我们 **Ubuntu** 系统默认是禁止 **root** 登录的。平时都是普通用户登录之后用 **sudo** 获取 **root** 权限，那现在到底应该怎么办呢？别急，还可以这样：启动计算机，出现 **grub** 引导菜单时按 **e** 编辑，在要启动的 **kernel** 那行后面加上 **init=/bin/sh** 进入 **sh**，这会时候就可以进行操作了。删除不必要的文件后重启，就恢复正常了。

主人终于还是没有听从我的建议（关键是他听不见），没有修改 grub，而是直接使用 liveCD 启动电脑，并且删除了那些 rubbish 带来的 rubbish 的文件，我们终于再次，正常启动了。

可是这次风波让主人知道了一件事实——原来系统还为 root 保留5%的磁盘空间啊。这种设计虽说还是有必要的，不过对于不能用 root 登录的我们 ubuntu 来说，似乎意义不是很大。尤其现在的硬盘容量越来越大，比如主人这500G 的硬盘要是都保留5%，那就是25G 啊！这能放多少小电影啊。（指的是那些体积比较小，清晰度不高的，多半是枪版的电影，不特指电影的具体内容，你懂的。）把这些空间专门留给一个从来不会登录的 root，虽说是 root 把，虽说是重要用户吧，可地主家也用不着这么多余粮啊？凭什么他 root 就可以强占我们这么大的地盘？谁给他的这个权利？root 就是为我们普通用户服务的难道他不知道么？就是个公仆难道他不知道么？怎么能够抢占这么大的空间，为用户服务用得着这么大空间么？简直是太腐败了。

主人越想越不爽，终于决定尝试着改变一下这种情况，拿起手中的武器，夺回属于自己的25G 空间。武器是啥？键盘呗。可是 root 这保留的空间看不见摸不着啊，分区表上也没专门给 root 分个25G 的分区，怎么抢回来呢？这个预留的空间，这种特权的机制，是在当初创建这个世界——也就是格式化这个分区的时候就设置好了的。格式化一个分区一般是类似 mkfs.ext4 /dev/sda1 这样，默认就会预留5%。如果想要改变这个情况，就要使用-m 参数手动指定。mkfs.ext4 -m 1 /dev/sda1 这样就只预留1%给 root 了，如果一点也不留那自然就是-m 0了。不过现在我们的硬盘已经格式化完了，我们都搬进来住了好几年了，总不能把所有数据导出去重新格一次吧？已经特权了5%的空间还能再改回来么？当然能，这时候你需要 tune2fs 命令。这个命令就是用来修改分区的一些参数的，比如卷标之类的，自然也包括这个特权空间的控制。只见主人弹指如飞的敲出了命令：sudo tune2fs -m 1 /dev/sda1 回车，就可以了。他还是比较仁慈的给 root 留了1%的特权空间，万一有需要的时候呢。这样修改了之后，成功的打击了 root 的嚣张气焰，广大劳苦大众分到了地，从此当家做主人，一派和谐景象了。（其实劳苦大众就是

你自己一个人而已，那 **root** 是谁呢？**root** 不也就是你自己么？唉，狗咬狗一嘴毛。）

有道是天有不测风云，人有旦夕祸福。出门遛个弯，就难免被花盆板砖啥的砸到；吃个东西，就难免被这个素那个氨的毒到；看看电视上个网，也难免被这个姐那个姐的雷到。那么在使用我们这个 Linux 系统的时候黑个屏，死个机，也就是可以原谅的了，是不？

这一天，主人命令红酒大师搞定那只叫做 QQ 的企鹅，让他出台工作。这哥们一进内存就嚷嚷着要找一个叫做360的家伙单挑，在得知方圆1公里以内确实没有360之后才开始工作。可是每工作一段时间就不忘抬头问问其他人："360来了吗？刚才有没有360走过去？？"红酒大师赶紧说："没有没有，没有360，你好好干活吧。"他还不信："真没有吗？你看那是什么，那个，那 TM 不是360么？"大师很无奈的解释："那是我的鞋……"由于这个 QQ 过度亢奋，导致红酒大师在用量上有些误差，过了一会，这家伙好像反应过来了："不对呀，你们这帮人我怎么都不认识啊？你们，你们是不是360？？是不是360的亲戚？快说！"红酒只好继续解释："他们都不是360，放心吧，没有360的。"QQ 仔细看了看红酒："你丫不是 WIN7！也不是 XP，说，你到底是谁？你们都是谁！这 TM 就不是 windows，肯定是360雇人让你们绑架我到这来的，不行，我不干，我要报警！"说着抄起手边的东西乱砸乱扔，他这么一通乱闹，整个工作间顿时陷入混乱，指令系统被破坏，图形界面被崩溃，我很无耻的，死机了……

其实，要说死机呢，也不算彻底死机。我跟 windows 是不同的，图形界面只是我这里的一个普通程序而已，图形界面挂了的时候，多数情况还可以通过 ctrl-alt-f1进入命令行来操作。比如这个 QQ，无论他怎么搞，主人还是可以通过进入命令行把它枪毙掉的。那如果命令行也挂了呢？比如由于某些不遵守道德的程序霸占了终端，无法输入命令，那怎么办呢？有办法。

我们都知道手机在一些特殊情况下，比如没有 SIM 卡，信号较差，欠费停机等等，这种情况下手机依然可以进行紧急呼叫，拨打110，119之类的。我们 Linux 也有这样的"紧急呼叫"方式。这就是键盘上的 SysRq 键（就是和 printScreen 在一起的那个）。通过按下 Alt+SysRq+某个字母，就可以实现向我发送指令了。当然，想 ls，fdisk 这样的命令是绝对不能的，字符界面

都挂了你还想进行操作？这种紧急情况下我所能做的只是尽量的保护好数据，记录现场，以便到时候你能根据我记录的蛛丝马迹，来找到这一切的始作俑者——真相，只有一个！恩，扯远了。那么都能通过紧急呼叫给我发送什么命令呢？最常用的就是 **REISUB**——重启。有人可能说了，我弯一下腰，直接按机箱上的 **reset** 不是更快么，还用什么紧急呼叫还什么 **REISUB**，多麻烦。可是，那个 **reset** 是完全物理上的忽然重启，可能会造成还没有保存的数据丢失。数据，任何时候都要保存好。你想想，你和你的老婆，坐着火车出了城，吃着火锅想唱歌——忽然就发现 **MP3** 里存的歌词没啦 ~ ~ 多么悲惨。所以呢，如果是通过紧急呼叫向我发送 **REISUB** 这样的重启，跟你平时在图形界面下正常重启电脑是差不多的，数据最大限度的得到了保存。

那么具体怎么操作呢？就是按住 **alt** 和 **SysRq** 键的同时，依次按下 **R-E-I-S-U-B** 几个键。这并不是一个命令，而是几个分别的命令，共同完成了保存好数据并重启的动作。咱们一个一个看：**R,**是把键盘设置为 **ASCII** 模式。**E**，是向工作间里除 **init** 外的所有同志们发送信号——系统要关闭啦，手头有什么东西赶快村到硬盘里，然后退出。**I,**跟 **E** 差不多，不过更加严厉，无论啥进程，看到 **I** 这个信号都将被迫关闭。**S**，同步磁盘缓冲，把那些在磁盘缓冲区的数据，赶紧写进真正的盘片上。**U**，卸载不必要的设备，系统重新挂载为只读模式，防止数据损坏。最后的，**B**，就是重启了。

今天给大家介绍介绍我是怎么管理用户的吧。用户，指的就是像我主人这样使用我这个操作系统的人类，我们 Linux 是一个多用户的操作系统，也就是可以支持多个用户。(废话 !) 这多用户的意义并不仅仅是系统里可以创建很多帐号，并且都能用，这不叫真正的多用户。多用户的根本是要能够同时处理多个用户的操作请求，也就是说，要能让多个用户同时登录进来，使用我这个操作系统。虽然一般家里的电脑只有一套键盘鼠标显示器，不可能两个人同时用，但是可以通过网络登录，同时操作。比如张三坐在电脑前看网页，并且用 BT 软件下载文件，同时李四页通过网络登录进来，也在用命令行的 BT 软件下东西，两个人不会相互影响。

那么要想支持多用户，我就得能够区分每个用户啦。怎么区分呢？大家都知道用户登录的时候都会有个用户名，但这个用户名其实只是给他们人类记忆的，我其实是不关心他的名字的，我关心的是他的用户 ID 号，也就是 UID。在我的概念里没有什么 lanwoniu 用户，dasanba 用户，tenzu 用户，ee 用户等等，我的脑子里只有1000号用户，1001号用户，1002号用户。进行跟用户，跟用户权限有关的动作的时候，靠的都是这个号。记这么个号码对我们操作系统来说自然不算个事，不过要让他们人类记住可就没这么容易了。所以就得用个他们容易记的，就是用户名。

在我们 Linux 系统里，每一个文件或者目录(其实目录也算是个特殊的文件了)都是有主的。每一个文件的属性里都记录着这个文件的属主是谁，谁来了，就是说这个文件是谁的。一般系统文件都是属于 root 的，其他的普通用户的文件则是谁创建的就属于谁的。这和人类的现实世界很相像。比如一个公司里面，每来一个人，都要通过一个流程：人家来报道，人事部门进行相关的记录，安排好工位和必要的东西，比如电脑，文件夹等等。这个过程就像我们 Linux 创建用户。每个员工都有名字，就像用户名，同时，多半还会有个工号，就像 UID。然后公司里有很多东西，桌子，椅子，电脑，水杯，书，笔，之类的。这些东西，有的是属于公司的，比如桌子，电脑。有的是属于某个人的，比如自己带来的水杯，笔记本。不会有没主的东西，就算真有个东西一时不知道是谁的，也早晚会属于打扫卫生 的大妈。

除了用户以外，还有个重要的概念，就是组，组就相当于公司里的部门。什么技术部阿，网络部阿，财务部，人事部啥的。每个组也都有个名字，也是给他们人类看 的，我关心的还是组的 ID 号，叫做 GID。组的存在使得对用户进行管理相对方便了一些，比如你要说：老张，老李，老王，老朴，老使，老班去楼下开会。这就 很麻烦，不如说：退休部的去开会！ 这多简单。

创建用户

知道了用户和组的概念，就可以学着管理了，咱们先从创建用户开始。

我们这里负责管理用户的有很多软件，其中，useradd 就是负责创建用户的。下面让他来给大家介绍一下：

大家好，我是 Useradd，要添加用户呢，来找我就对了。我就想当与人事部里面专门负责管新人入职的。那么要入职的话呢，步骤很简单，就是在命令行里，叫我出来，然后告诉我谁要入职就好了。就像这样：

```
useradd bob
```

这条命令就是要创建 bob 用户的意思。当然，在咱们 Ubuntu 这个系统下，默认是不能 root 登录的，所以你直接运行这条命令肯定告诉你没权限，所以还是得加上 sudo，就这样：

```
sudo useradd bob
```

这样运行成功之后就创建了 bob 用户,并且默认设置他的家目录为/home/bob,默认的 shell 是/bin/sh。但是这样仅仅是给 bob 做了登记而已，很多必须的东西还没给 bob 准备好。比如家目录，虽然写了是/home/bob，但其实这个目录还没有建立。这时候如果用 bob 登录的话会看到很诡 异的命令提示符。(只有一个$，因为没有家目录，找不到家目录下的相关配置文件。)那要想在创建用户的同时创建出这个用户的家目录怎么办呢？就需要参数。

有这么句话：如果在 windows 下发现某个软件不能满足你的要求，你就需要查查其他软件；如果在 Linux 下某个命令不能满足你的要求，你就要查查这个 命令的参数。参数-m 就是用来强制创建用户的家目录的，也就是说，你如果这样创建用户：

```
sudo useradd bob -m
```

运行之后就顺道创建好了/home/bob 目录了。如果你说我不像让 bob 的家目录是/home/bob，大家都这样，多没个性阿，我想让 bob 的家目录是/home/alice 可以么？当然可以，用-d 参数就可以指定用户的家目录，就这样：

sudo useradd bob -d /home/alice -m

这样就创建了 bob 用户，并且指定其家目录是/home/alice，然后还把这个目录创建了出来。刚才兔子个您说了，每个用户的名字其实都是浮云，对于 我们来说主要的是 UID，那这个 bob 的 UID 是多少呢？一般来说，ubuntu 系统安装时创建的那个用户的 UID 是1000，以后每多创建一个用户 UID 就加1。也就是说，如果 bob 是除了装系统时创建的那个用户以外，第一次创建的新用户，那么他的 UID 就应该是1001，再创建 pop 用户，那就是 1002，依次类推。可能有人说了，我这人就喜欢各种靓号，什么 QQ 号码，电话号码都是888，666啥的，我也想要个好看的 UID 可以么？当然可以，用 -u 参数就可以指定用户的 uid，还比如刚才那个 bob，如果管理员想进一步设定 bob 的 uid 为8888，那么就这样：

sudo useradd bob -d /home/alice -m -u 8888

哦对了，刚才还说了默认的 shell，这个也可以改，用-s 参数。比如 bob 可能习惯使用 csh，那么可以：

sudo useradd bob -d /home/alice -m -u 8888 -s /bin/csh （越来越长了……）

用户除了自己的身份以外，还会属于一个组。就好像一个员工肯定要属于一个部门一样。创建用户的时候会同时创建一个与用户同名的组，并且将这个组设置为这个 用户的默认组。bob 就属于 bob 组，alice 就属于 alice 组。那如果不像让用户属于这么个同名的组怎么办？你也猜到了，又得加参数了：

sudo useradd bob -d /home/alice -m -u 8888 -s /bin/csh -g yamaguchi

对，就是这个-g 参数，用来指定用户所在的组，上面这个命令就是创建 bob 的同时，把他加入 yamaguchi 组。除了-g 以外，还有个-G 参数，这个参数也是用来指定用户所在的组，但是可以指定多个，以”,”分割开。

好了，useradd 可以下去休息了，大家应该对创建用户的过程有了一定的了解。用 useradd 创建用户的特点是什么呢？一堆参数！这倒是很符合我们 Linux 命令的特点。不过可能有的人

还是不喜欢这样一堆参数的命令，没关系，下面给大家介绍 useradd 的兄弟——adduser。

好，让我哥哥下去休息休息，哪凉快哪待一会，换我上来给大家讲解。如果您觉得用 useradd 创建用户太麻烦的话，就找我来就对了。用我创建用户很简单，比如创建 bob 用户吧，就运行 sudo adduser bob 就行了，剩下的事情我回知道您一步一步完成。

首先，我会先添加好这个用户，然后创建好他的家目录，并复制好一些必要的配置文件。之后，我回向您询问 bob 这个用户的密码，您需要输入两次，输入的时候 密码不回显，您懂的。密码输入没问题之后，我进一步向您询问这个用户的详细信息，比如这个用户的全名——全名是图形界面登录的时候显示的那个，真正的用户 名还是那个 bob，全名可以为空。然后是 room number 房间号，工作电话，家庭电话，其他备注，当然，这些都可以为空。都写完之后，我将再次确认以上信息是否正确，没问题的话，这个用户就算创建完 成了。

```
lanwoniu@lanwoniu-ubuntu:~$ sudo adduser bob
正在添加用户"bob"...
正在添加新组"bob" (1001)...
正在添加新用户"bob" (1001) 到组"bob"...
创建主目录"/home/bob"。从"/etc/skel"复制文件。
输入新的 UNIX 密码：
重新输入新的 UNIX 密码：
passwd：已成功更新密码
Changing the user information for bob
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
```

```
        Work Phone []:
        Home Phone []:
        Other []:
        这个信息是否正确？ [Y/n] y
        lanwoniu@lanwoniu-ubuntu:~$
```

好，**adduser** 可以下去休息了。如果您觉得 **adduser** 这样创建用户还是不方便的话，没关系，咱还有图形界面呢。下一位——来自图形界面的 **users-admin**，中文名字叫“用户和组”

哪个……大家吃了么？刚才他俩吧，就都是命令行滴哈，就不是那么方便。我捏，就是个图形界面滴用户管理程序，要说功能吧哈，就跟他俩也差不多，不过捏，就是方便。

打开俺滴主界面。您要添加删除用户呢，就点左下边这两个钮就行了。咱先说添加，点击添加按钮，会弹出创建新用户的窗口，首先填写名称和短名称。这个名称就 是全名，是图形界面显示的那个。短名称就是登录名，字符界面登录的时候要输入的那个名字，两个可以一样。写好名字确定，就进入了更改用户口令窗口，在这里 设置用户的口令，或者也可以让我给你随机生成一个。下面有个选项，“登录时不询问密码”，选上这个就允许图形界面登录的时候不输入密码。不过字符界面的时 候无论如何是要输入的。设好密码之后点确定，用户就创建完了。

删除用户就很简单了，选中你看着不顺眼的用户，点击删除按钮就好了。会问你要不要删除用户的文件，也就是那个用户的家目录，以及里面的所有东西。

那有人问了，想 **useradd, adduser** 他们哥俩创建用户时候的那多乱七八糟的参数在哪里设置阿？您看界面的右下角，不是有个高级设置么，就是他了。选中要设置的用户，然后点击高 级设置，就会出来更改高级用户设置窗口。在里面可以设置 **adduser** 问您的那些办公室位置阿，电话阿啥的，基本也没啥人写这个东西。然后就是用户权限， 这个是 **useradd** 那哥俩不具备的功能，俺就可以。可以设置用户的各种权限，是否可以管理系统阿（也就是可以像主人那样用 **sudo**），是否可以使用打印 机阿等等的。然后在高级里面，可以设置用户的家目录，默认的 **shell**，

以及默认组，也就是主组，还有 UID 这些内容。

好了，用户和组同学也可以下去休息了。其实关于他刚才说的，他可以设置用户权限的问题呢，其实命令行的同志们也是可以设置用户权限的，并不是他的专利。只 不过我们 Linux 的用户权限是靠组来管理的。比如说，所谓的管理系统权限，其实就是把用户加入到 admin 组，那么你用 useradd 创建用户的时候加 上-G admin，就等于设置给用户“管理系统”权限了。

主人很久没有来我们这里闲逛了，听说是因为最近主人在隔壁那里装了个叫做啥古剑的游戏。哦，顺便说一下，隔 壁的查皮早已经被主人请出了硬盘，新搬进来的家伙，是查皮的后辈分，叫做温妻。我们不知道是不是个温良贤惠的老婆，不过据说长的是听漂亮的，也难怪主人最近老往她那里跑呢。新装上的那个啥古剑，好象还是个正版的软件，我问过红酒大师，这样的正版软件，你能把它搞定么？你搞定了，那还是正版么？红酒大师说， 这个正版盗版，不在于运行在什么电脑上，是不？我搞定的软件，无非也就相当于运行在一个由我虚拟出来的电脑上而已，为什么就不行呢？所以我虽然是用三酒迷 魂大法诱导了温妻那边的软件，但是并不影响他的版权。不过这个古剑……恩……暂时还没有什么办法，别的不说，就他对硬件，对显卡的要求，就算在温妻那里， 用集成显卡都很卡，要我模拟出一个显卡来，那就算能运行也根本没法玩吗。我听了无奈的点点头，看来要想主人再经常回到我们这里，要么等那里古剑游戏玩完， 要么，我也这里也该有一个能够吸引主人的软件。

当 我来到这一个有着宽敞工作间的电脑的时候，那些正在工作的软件们似乎并没有对我和引领我到来的牛力先生表示过多的注意，只是轻轻的点头微笑以示问候。当 然，我并不会因为这样算不上热情的接待而感到失落，我早已习惯了流浪于各个 Linux 系统之间，被人们一次一次的试用，并一次一次的删除。这正好像我的名 字那样，流动的盒子——fluxbox。

来到这里之后，我更加仔细的查看的这里的硬件环境，并且惊奇的发现这个电脑的配置竟然不错！通常来说，没有什么比工作在一台配置很高的电脑更让一个软件高 兴的事情了，然而却给我带来了比兴奋更要多些的疑惑。这里我要说明一下，我，fluxbox，是一个窗口管理器，窗口管理器是一个图形界面的重要组成部 分。一般的 Ubuntu 系统用的是 Gnome 环境，有其自带的窗口管理器，而像我这样的，一般都是在特定的情况下才会被请出场——这主要是归结于我的特 点，那就是节省资源。

那么既然我的特长是节省资源——归功于我精炼简洁的界面和功能——可为什么这个人要在这个配置还挺高的机器上找我来工作呢？或许唯一一个合理的解释就是，他闲的。

我们知道确实有很多的 Linux 用户，他们在经过了第一次看到 gnome 那与原来的 windows 完全不同的界面的兴奋期后，可能会厌倦于每天对这同样的 上下两条面板。甚至也在几次并不算成功的安装 kde 之后，放弃了对他的期望，而转向其他的窗口管理器——我就是其中之一。我就是一个窗口管理器，这个名字 很能够顾名思义而不需要过多的解释。当你在图行界面打开任何一个东西的时候，都需要我们上场。（当然，也会有些例外，比如一些自己处理界面的移动而不需要 一个窗口的软件，比如 fcitx。）比如 gnome，这是一套完善的桌面环境，其中就包括窗口管理器 Metacity。他负责画出那个黑色的，有着诡异 的，靠左的关闭按钮的框框。并在用户进行拖动，最小化，最大化等操作的时候，及时的处理窗口内显示内容的位置和大小。同样的，我也是做这个工作的，我们的 实现方式或许不大一样，审美情趣或许不同（比如我不想把关闭按钮放在左上角，再比如我能把互不相干的两个软件放到同一个窗口中并用标签来切换。），但最终 的功能是一样的。甚至，你可以在 gnome 环境下，用我替代 Metacity 来体验一个混合味道的桌面环境。这样或许也能稍稍减小你电脑的运算负担和内存 占用——因为和 Metacity 相比，我更专注于更小的资源占用而他对显示效果会许更在意一些。 不过仅仅是把我当作 gnome 下的窗口管理器并不能充分的体会使用我给你带来的速度的快感。多数人还是抛弃了 gnome（或者 kde）这样的大块头的软 件，直接使用我作为一个简单的桌面环境。（虽然我真的只是个窗口管理器）我除了管理窗口外，其实维护的简单的任务栏和系统托盘还是不成问题的，于是这就满足了一些最最基本的桌面环境的要求了。不过如果你想把一些文件放在桌面上方便使用？很不好意思，我只是个窗口管理器，这个事情不归我管。在 gnome 这个 那个鹦鹉螺管的事情，而我相信你并不想启动那个家伙来占用掉我辛辛苦苦节省下来的一点资源。如果你想换换桌面的壁纸，对不起，我只是个窗口管理器，这事别 跟我说。不过我可以介绍一个叫做 fbsetbg 的家伙

来满足你的要求。作为一个窗口管理器，换换主题的功能我还是具备的，不过我也没有 gnome 那样的图形界面的管理工具来导入主题，所以请你手动把你下载到的主题解压缩到 ~/.fluxbox/styles 目录中，谢谢。什么？你开始怀念 gnome 面板上那些显示 CPU 占用，网络情况，甚至天气预报的小插件了？i'm so sorry，不过或许你会爱上 conky 的，去跟他谈谈吧。总之呢，免费的午餐是没有的，既然你选择了我，选择了速度，就要牺牲掉一些东西。但我想这些牺牲多半还是值得的，比如省掉一个用来设置壁纸的图形界面程序对于一个不会每天换壁纸来消磨时光的人还是不会造成太大不方便的，你说呢？

在这里工作了这么多年，对于这里的交通环境有了比较深刻的认识。我们工作地点，是在一片非常宽广的园区中。 园区里的各种公司，企业都很多，也有相当多的个体户。这些企业和公司多是对外提供服务，服务的内容不外乎是新闻啦，娱乐啦，娱乐啦，娱乐什么的。基本是个 以娱乐为主的园区。为了保护园区里的人们，园区的周围建起了一圈高耸入云的城壁，外面的小鸟在累死前是飞不过城壁的，城里的人们也一样不能爬上去亲眼看园 区外的景色。

不过，园区内外，倒并不是完全隔绝的。城壁上面都按装了53寸液晶电视，时不时的播放一些园区外的事情——当然，为了不让园区内不明真相的人们胡乱的学习 一些乱七八糟的东西，播放的内容一定是要经过精简，经过筛选，经过剪接，经过处理，经过 PS 什么的。另外，一些壁垒外的“无害”的公司也被允许直接访问， 以此表明园区与外界的交流是开放的,园区管理者鼓励园区与外界交流的发展,也努力为园区的健康发展营造良好环境。然而好奇总会害死猫的，总有一些园区内的 一小撮不稳定分子总是跃跃欲试的想爬上壁垒去看看真正的外面的世界。于是他们前赴后继的尝试着很多种越壁的方法。比如西门那边，有人用一堆箱子来翻墙，不 需要额外的工具，也不需要墙外有人接应，只是把箱子配置好就可以看到外边一些公司的情况（比如油兔笔公司，非死不可之类的）。于是人们给这种越壁的方法起 名字叫做西箱计；还有的人能够联系到壁垒外的一些可以访问的，“无害”的公司。于是，就托这写个无害公司里的人帮自己去打听壁垒外其他公司的事情。当然， 为了表达谢意，每次都不忘了给那些帮忙转发内容的公司的接应者带几个李子拿回去，于是这种方法就被叫做带李；还有的人看到，虽然园区里的人不让出去，但是 园区外的人进来出去的都是不受限制的，于是就做了个大套套，套在自己身上，把自己伪装成来自园区外的人混出去，这种越壁的方法叫做，带套法。

我们今天要说的，是另外的一种方法——挖地道。

自古以来，挖地道就是一种对付壁垒的好办法。不过我们用的地道并不是真正的地道，而是

虚拟的一个通道——VPN。这是一种让你家里的电脑可以和公司内网里 面的电脑都能够无障碍的交流，用飞鸽传输发送文件的方法。这东西就像机器猫里的任意门，你明明在家，但你只要掏出这扇门，在门上写好公司地址，用户名和密 码，然后一开，走进去，你就会发现你已经站在公司的办公室里了。利用这种方法，我们就可以随时随地的接入公司的办公环境，处理事务。那么如果你能够联系到 壁垒外的某个无害的公司，并且获得他们 VPN 的一个帐号密码，那么你就可以掏出任意门，按流程操作，然后，就置身壁垒之外了。可是我家也没养机器猫，这任 意门在哪呢？作为新世界的 Linux，这么简单的工具我怎么能够不提供呢？看到右上角的网络链接的图标了么？（如果你没换主题的话就是两个箭头的那个）点 击它，看，有 VPN 链接，鼠标移下来会看到“配置 VPN”，点开。添加一个就好了。

这天主人觉得玩图形很不专业，搞虚拟终端也不过瘾，于是就按下 ctrl-alt-f1，来到真正的终端来敲名令玩，看着一屏屏的字符好象很有成就感的样子，如果遇到需要查的东西再按 ctrl-alt-f7回图形界面来查。那他在学什么命令呢？答：iwconfig。

iwconfig 小程序是用来管理无线网络的，现在 wifi 的使用是越来越广泛了，很多人家里都使用无线路由器来共享网络链接，我家主人这里也是。在图行 界面下，那个 nm-applet 工作的还是不错的，只是主人不想特意为了链接网络而去一下图形界面，所以就学习起 iwconfig 来。iwconfig 的 主要任务就是负责把电脑接到某个无线接入点上。经常和 iwconfig 同时来工作的还有一个家伙，叫做 iwlist，这里是用来查找无线信号的。一般，都 是先运行 iwlist scan,来扫描附近可以找到的无限信号，然后再选择其中的某一个接入。iwlist 得到的结果里最主要的是那个你意中的无线接入点的 essid，有了这 个 id 才好跟 iwconfig 说你要接谁，如果不说的话，你只运行 iwconfig，那么他只是帮你列出现在机器上的无线网卡的状态——多半是单身状态， 也就是说没有连接到任何无线接入点中。那么要想连接怎么办呢？只要知道了 essid，就运行 iwconfig wlan0 essid “xxxxxx”就可以了。其中 wlan0是你要是用的无线网卡，xxxx 是那个你要链接的无线接入点的名字，也就是 essid。这样运行了之 后，iwconfig 就会去试着跟对方眉来眼去一翻，然后鸿雁传书，互诉衷肠，心驰神往，最后喜结连理——你就能上网了。不过这是对方没什么要求的情况， 有的时候对方会要点彩礼——一把钥匙。别急，不是房子的钥匙，不过自行车钥匙也不行，要的是能够解开对方甜言蜜语的钥匙——解密用的 Key。毕竟无线这东 西看不见摸不着，你俩聊的卿卿我我的搞不好就隔墙有耳，所以设个密码还是需要的。密码基本有两类，wep 的和 wpa 的。wpa 又有很多种，我们暂时就不具 体讨论了，因为 iwconfig 处理不了那些，如果是 wep 的话，那就好办。链接的时候就运行 iwconfig wlan0 essid “xxxxxx” key yyyy-yyyy-yy 这样的格式就可以了。那些 yyyy 就是密码。密码必须是5个字节的16位数字，具体是什么，那只有你知道咯。

随着 USB 那扇门上的灯光又一通乱闪，主人又往那个 usb 接口上接东西了，我跑过去一看，貌似接进来的是一块网卡。于是我就按照处理网卡的方式处理这个硬 件，首先得搞清楚这个硬件的厂商，型号，这样才能去找合适的驱动来使用这个硬件阿。我一看，是 moto 的网卡，什么 E6什么的。我仔细想了想，隐约记得当初接进来过一个摄像头也是这么个型号阿，怎么这回变网卡了？不对，这货不是网卡，绝对不是网卡！

后来的事情证明，这货还真不是网卡，而是一部手机，一部和我有点渊源的，Linux 系统的手机。

虽然我知道这家伙只是伪装成一个网卡而已，不过事情还是要按照规矩办的。既然他上报的数据都是合理合法的，那么我就正常的向主人汇报，咱家多了个网卡，叫 做 usb0。主人很高兴的看到我直接驱动了这个伪装的网卡。之后马上开始设置这个网卡的属性，只见他把网卡的 ip 设置为192.168.16.2 之后，胸有成竹的 ping 了一下192.168.16.1。我们都不知道网卡的那一头接的是哪里，只是简单的发送了几句日常的问候，(大约就是地瓜地瓜我 是土豆之类的。) 那边果然还真有人回应 !(土豆土豆，我是地瓜，我是地瓜。) 好，既然通信没问题了，主人就派出人去和那边联络了。谁呢？telnet。

telnet 是 Internet 的远程登录协议的意思，它可以让你通过网络，同一台计算机终端登录到另一台计算机上去工作。在过去，这很有用，因为那时候 计算机相当珍贵，尤其运算能力强大的计算机，如果有很多个人都需要一个强大运算能力的计算机，那么，给他们每人配备一个显然是一个倾家荡产的最好选择。于 是那时候多数的情况是使用一台高级的计算机，大家分别通过各自的能力较差的阿基算计用 telnet 协议登录到那台高级计算机上去干活，这样就可以充分的利 用资源。不过现在 telnet 协议用的比较少了，一方面现在都图形化了，就算要链到别的计算机上，也不是 telnet 能搞定的了，而是需要 VNC 这样的协 议。另一方面，telnet 也不是很安全，通信过程中没有足够的加密措施，所以偶尔有些需要通过字符界面远程连接到另一台电脑

的情况下，也由更安全和强大 的 ssh 协议来实现了。那么 telnet 就没用了么？也不是，现在仍然在某些方面需要用到他，就是嵌入式的系统。嵌入式系统一般都比较拮据，不像我们有的 是内存，缓存俩电影都没问题。他们都是算着每一个时钟周期过日子，所以要想实现 ssh 这样的协议有些难度，于是，简单易行的 telnet 就是他们最佳的选 择。

那么这回主人叫来 telnet 来和对方联络，看来对方不是一个正儿八经的 PC 机了。哦对了，忘了说了，telnet 是个协议，而在我们这里能够实现这个协 议的那个软件也叫做 telnet，跟查皮那里一样，以后要不特意加协议俩字的话，说 telnet，那就是说这个软件呢。telnet 很快通过那个刚刚接入 的 usb 网卡跟对方联系上了。通过一番攀谈，telnet 知道了，对方原来果然是个手机，moto 的 e6，这是他的型号。它里面运行的也是一个 Linux 系统，不过自然跟我是不一样的了，而且差别还是很大的。比如，我是2.6.xx 的内核。（xx 这位不一定，老升级）而他还只是个2.4.xx 的内核，差着 一代呢。有人可能会质疑，他是.4你是.6明明差两代嘛，还有个.5呢。这个呢，看来您不大清除我们内核的编号。我们的编号有个规则就是单号测试，双号稳 定发布。2.4是一代内核，随着逐步的发展和完善，2.4后面的小号逐渐的变大。在这一代内核完善的同时，还要开发下一代内核，这时候的下一代的代号就叫 做2.5.xxx，像.5这样的单号版本的内核是实验室的产品，见不得人的那种。随着工作人员日以继夜前赴后继摸爬滚打的努力，到他终于能够见人的时候， 他就成为了2.6版本的内核了，所以说2.4和2.6是差这一代的。那么 pc 机上的2.4内核已经是很久很久以前的事情了，那时候还是 redhat9的时 代呢，我也只是从我们学校的历史书中才知道有这么样的一代内核。那么这个手机里为什么用这么旧的内核呢？多半是因为这样的内核成熟度高，而且对系统资源的 要求要小很多。通过 telnet 的对话我们知道了，（因为他对话不加密，所以我们旁边的人都能听到）对方那个手机只有300多 M 的主频速度，和仅仅不到五 十兆的内存。要是让我去里面干活，还不得把我憋屈死！

telnet 很快就跟对方手机里面那个憋屈的 Linux 系统取得了联络。主人直接用 root 就登陆了进去， 也没要密码，这跟我们的风格迥然不同阿。不过想想也是，毕竟这么登录不是正常使用的方式，而且手机也不会像我这个系统似的弄个什么多用户，所以安全性方面 差点就差点了。我问那个手机里的内核："嘿，小4，（他是2.4嘛，就叫小4了。）你那个是什么发行版阿？"小4有些迷惑："发行版……我也不知道，我听说过你们 PC 机上的 Linux 都分发行版，不过我们手机，好像不怎么讲这些。关键是，好像也根本没什么手机用我们 Linux 操作系统吧。似乎只有我们 moto 用过，而且现在新款的手机也不再用我们了。艾～ 老6阿，还是你们好，在 pc 上面多风光。" 我听了不觉有点心酸，同是内核，却有着如此不同的命运。我赶紧安慰他："听说 Nokia 不是有一个很 NB 的手机就用的我们 Linux 的系统嘛，可见希望还 是有的。"小4依然垂头丧气："那也是昙花一现，试试水而已。老6，你不混手机这行你不知道，Nokia 人家有自己的 SB 系统，而且已经开始向那个有点软 的公司抛好久媚眼了，以后的手机，很多都要装上 windows 了。我们大概很快就要失业了。" 我问："那你们 moto 那里怎么就不用 Linux 了呢？"小4说："要我说阿，就是他们不会玩，把挺好个系统给玩毁了。咱 Linux 靠的就是开源，就是社 区，就是多样化和定制性。可是我们 moto 的系统，不开源不说，装软件包都限制，只能装我们公司认证过的软件包，可是认真部门半年也不一定开一次张，总共 也就认证过三五个软件，你说，说起来我们好歹也是智能机，就只能装这么几个软件，还智能个什么劲阿。虽说还能装 java 软件，可那跟那些非智能机有什么区 别了？好在有很多牛人破解了 moto 的限制，通过重新刷机可以装各种软件，要不比现在还惨呢。" 我听的瞠目结舌："怎么装软件还限制阿，咱们做操作系统的，以服从命令为天职，人家想装什么就装，怎么能拦着呢？" 小4觉得看见知音了："就说是阿，把我们整成这样，官方就那么几个软件，还不开源，没法随便给我们开发软件，我们可不整天要死不活的么。这么，现在上边觉 得我们不好了，要换人。换了个叫安德鲁的，已经完全接替我们的位置了，以后，再也不会有 Linux 的手机了，哀～"

“安德鲁？”我问：“他是谁？”小4说：“就是一个手机系统，现在可是大红大紫呢。我们高层打算靠他打开新纪元，开创新局面，要重塑辉煌，实现伟大复兴 呢。”我听着觉得有点新闻联播的味道，赶紧问点别的转换话题：“那这个安德鲁，到底有什么厉害之处呢？”小4答：“其实，也没什么，就是后台硬。”“哦？谁是后台?”“大狗子。”小4，老6，大狗子……行，乡土气息浓郁……当然这都是我的心理活动，没说出来，我还没问，小4就接着说了：“你应该知道阿，就是google。人家整的 系统，上面的应用可丰富呢，现在市场上的占有率挺高，而且程序都是拿java写的，现在会java的多阿，所以不愁没有开发者。”我疑惑的问：“可是这 java的效率……”小4说：“谁知道呢，不过人家生活条件比我们好得多，人家少说也有几百M的内存，主频600多M往上的CPU，甚至上G的都有，就算慢点也看不出来。”正 这时候，telent报告：“通话时间快到了阿，主人要断开连接了。”于是我只好和小4做了简短的告别，然后就继续忙我的去了。

生活这家伙好像总是很喜欢玩“扎堆”这种游戏，前两天刚刚认识了小4——就是那个 E6手机上的系统的内核，这不今天又来一个手机。我仔细一看，嘿，还真好是我们那天聊的安德鲁。

这个安德鲁是手机的系统，可不是哪个具体的手机厂商的名字，挺小4说这系统现在挺火，今天正好见识见识。三下两下的，主人就把手机连接到电脑上，并且认做 了一个网卡，之后，还是 telnet 出面，打通了两面的热线。我拿起话筒喂了一声，那边传来一个熟悉的，略带惊讶的声音：“铁蛋 !? 铁蛋是你么？”这短短 的7个字瞬间打开了回忆的大门，我惊呼道：“铜鸡，是你啊铜鸡！”这时能够明显的从话音中听出对方很激动：“是啊，铁蛋。你现在给 ubuntu 做内核了 啊？听说这个发行版很流行啊，看来你混的不错嘛。”我略带骄傲的说：“还好吧，也是我运气好，其实内核又有多少不一样呢，Ubuntu 流行，还得靠其他的 软件同志们。铜鸡啊，当年你性能也不错啊，那时候咱俩不相上下。你走的时候，给我留的那朵菊花，我至今还留着呢，我管他叫‘铜鸡菊’。”“是么？那没用的 东西你能留到现在我很欣慰啊。”“哎？谁说铜鸡菊没用啊，每当我累了，性能不好，各项指标下降的时候，我就拿出它来看看，只要它一出现，各种性能指标要多 高有多高啊，可有用呢。”“真的呀，那太好了。”“对了，铜鸡，光聊天了，忘了问你怎么去给手机当系统内核了？你那个系统是安德鲁么？”“是啊，这个说来 话长了。那年我走了以后，到处给人打工，不过效果都不大理想。后来碰上个长得挺可爱的大绿机器人在那里招工，有内核，有各种库，多媒体，驱动，反正各种岗 位，我就去应聘他们内核了，结果一下还就应聘上了。”“哦，原来安德鲁的内核和 Linux 的内核是一样的啊。”“恩，基本差不多吧，当然由于硬件不通，肯 定要做些改动的。除了我以外，其他的很多岗位也都是咱们 Linux 的人。比如 C 语言的各种库，多媒体框架，SGL，OpenGL，字体的引擎，甚至浏览器 的额引擎，都跟 Linux 没什么区别。”“哦，何着安德鲁就是 Linux 啊。”“也不全是，就底层这些是咱们 Linux 的人。上层的应用软件就不一样了。 他们安排了一个重要的部门叫做 runtime，算是运行环境的意思吧，所以我们都叫他环境科。”“管打扫卫生的？”“不是，他们可重要了。整个系统被这个 部门分成

了两部分。能够跟用户打交道的图形界面的程序们，都是归这个部门管理的。有什么事情都要汇报给环境科，再由环境科向我汇报。”“那为什么啊？”“因为安德鲁系统上跑的那些个软件，全都跟咱们不一样，不是真正的二进制可以运行的程序，而是一种字节码程序。你知道 **java** 吧？”“知道啊，我这里有好 多 **java** 程序呢。”“他们其实就是 **java** 写的程序，不过不是普通的 **java**，是安德鲁的东家发明的一种特殊的 **java** 语言编译器编译出来的特殊的 **java** 程序。他们只认识环境科，根本看不见其他人，也没法跟其他人交流。”“听起来像是生化危机里变异了的人类啊……”“其实……有点。”

通过跟我儿时的朋友，曾经的同学——铜鸡的交流，我终于知道了一些安德鲁系统的面貌。就是一个基本的 Linux 系统上，跑着一个特殊的 java 虚拟机。

可能有的同学还是不大知道这个 java 虚拟机是个什么东西，那今天我们就说说它。要说虚拟机，那先得复习下我们常见的一些语言和程序的编写。编程语言大致分 成两种，编译型和解释型。编译型语言，就像 C 语言这样，需要一个编译器。写好了 C 的源码，对我们系统来说就是一个普通的写满字符的文本文件而已。咱们以前 不也说过么，c 的代码就是个图纸，要想让他变成像狐狸妹妹那样活蹦乱跳的程序，需要由 gcc 施工队按照图纸创造出这个程序才行。这就是编译型语言，需要编 译器，编译出来是真正的可以在特定的硬件，特定的系统上运行的系统。编译好了之后，就没有 gcc 他们什么事了，程序就自己活动去了，比如编译一个 rubbish，编译好了就出现一个 rubbish 程序。而解释型语言不是这样，比如 bash 的脚本，perl 语言，这些都是解释型的。解释型语言的程序 写出来也是一个普通的写满字符的文本文件。但是解释型不需要编译器，而是需要解释器。bash 脚本的解释器就是 bash，脚本运行的时候，bash 就拿着 那个程序的源码，一条一条的去执行。比如还是写个 rubbish.sh 的脚本，那从始至终我的内存里也不会有叫做 rubbish 的家伙来干活，而是由 bash，按照脚本里面的语言来进行各种操作。c 语言编译出来的 rubbish 是这样:gcc 他们一通忙碌,一个叫做 rubbish 的家伙从试验台上跃 起，跑进内存空间抬头扯着嗓子大喊一声“打雷咯，下雨收衣服阿～～”。而 bash 脚本语言的 rubbish 是这样的：bash 拿起那份脚本，看到上面写着 “请走入工作间”，于是 bash 进了内存，再看脚本“抬头”,于是 bash 抬起头,再看脚本“请运丹田一力混元气,大声喊出‘打雷咯,下雨收衣服 阿～～’”，于是 bash 照做之后一摔脚本：TMD，谁写的脚本，按着剧本改编的吧！！ 这就是编译型跟解释型的不同，编译型是由编译器（gcc）创造出各式各样的角色，完成格式各样的任务。解释型是由解释器(bash)演绎出各式各样的角 色，完成各式各样的任务。

好了，说了半天了，那么 java 语言是什么型呢？其实，她之所以流行，就是因为他既不是编译行，也不是解释型，他有它的中性美。（好像这年头半 A 不 B 的都 容易流行）有人说 java 是一种半编译型语言。首先，java 写好的源程序要想运行，必须要经过编译，这跟 c 是一样的。但是 java 程序编译出来之后并不 能直接在机器上运行，也就是说，java 编译器编译出来的并不是一个活蹦乱跳的程序。那么要想运行这个程序，要许要一个解释器，这就是 java 虚拟机。一 般如果你要运行一个 java 软件的话，就必须装一个叫做 jre 的东西是吧，就是这个。他负责给 java 程序创造出一个可以自由运行的空间，到了里面，这个 编译好的 java 程序才能运行。就好像，我们这宽敞的内存是一片大草原，gcc 编译出来的程序就是各种哺乳动物，什么牛阿，羊阿，猎狗阿之类的。而 java 编译出来的程序就是一堆鲤鱼阿，黄鳝阿，泥鳅阿之类的东西。而 jre 的工作，就是挖一个水塘，或者做一个鱼缸。

最近主人的生活越来越忙碌了，听说他要和那个曾经触摸过我们的键盘，赞叹过我们的屏幕，并且还在主人很多次复制来的照片里出现过的 mm 进行一项劳民伤财的 运动—结婚。我是不太理解这项运动的目的和意义，听 oo 老先生说，就有点像 SMPlayer 一定要和 mplayer 配套使用一样，主人就是个 mplayer，他要去找他的 SM 了。我说 mplayer 没有 smplayer 也一样能用阿？OO 先生说，是阿，之前主人就是一个孤独的 mplayer 阿。没有 smplayer 的时候他是不是样子不好看？虽然功能强大但是不修边幅？我挠挠头，还是有些迷糊，不过这都不重要，重要的是，我们马上就要有第二 个用户了，以后会不会出现两个人同时登录到我这理的情况呢?(当然是一个人通过本地登录，另一个人远程登录了，我估计远程的会是主人) 会不会两个人在终端下用 mail 命令来聊天，呵呵，想想就觉得挺好玩的。到时候主人的家目录下的主件，会特别给女主人设定访问权限么？

果然，很快，主人给系统创建了第二个用户，叫做 lili，不过并没有马上登录进来。于是我也就只好期待着，期待着新的用户，期待能多一个用户来承认我的价值。

话说最近主人要和某 mm 合体了。我怎么知道？^_^本狐狸自然是最了解主人动态的软件了，从他每天都上什么网站就能看出来。可以说，他一敲 www,我就知道 他要去哪。最近这不是咱主人没事净逛那些个什么酒店预定阿，婚庆公司阿，怎么租车队阿这类的功能性网站和论坛，以前那些不和谐的论坛他很少去了。爬墙的梯 子也不怎么用了（兔子老大在旁边咳嗽：咳，咳，注意敏感词阿。）我们兔子老大竟然还不明白主人这是在干嘛，不知道为啥要合体。说来也难怪，他总是一个人独 来独往嘛，您什么时候见过俩内核一块工作的？主人要是合体了，以后我们难免还要多个女主人，不知道她是不是也喜欢用我们狐狸家族的浏览器上网，我可不想输 给那个唱戏的 Opera，说实在的，仗着主人用我的次数比较多，我以前没少鄙视那个 Opera，其实她本事也挺好的，这要是女主人来了，用她比用我顺手， 那她翻身了，还不得鄙视死我阿。还有

那个 Chrome，肯定也跟着扇风点火凑热闹，哼。我一定得在女主人面前好好表现，系统默认浏览器的位置绝对不能动摇！恩。

“作为本系统中最著名的聊天软件，我，pidgin，可以负责任的告诉你们，主人结婚之后，咱们的日子，不一定好过！”

这句话一说，工作间里顿时安静了下来。大家纷纷凑过来，围住 pidgin，七嘴八舌的问：“为什么阿？”“你怎么知道的？”“女主人不喜欢咱 们？”……pidgin 等大家稍微安静了一下继续说道：“主人很早就开始和未来的女主人用 MSN 聊天了，这其中的过程，我最清楚。女主人虽然也曾经被咱们 的3D 界面所吸引，也被打扮时尚的狐狸妹妹所迷惑，呃，对不起，用词不当，倾倒，倾倒好一些。但是，她是一个需要 QQ 的人。”Chrome 忍不住插嘴： “就是那个主人一年都不一定叫起来一次，好不容易起来一次还经常司机，老占 CPU 的家伙？怎么会有人需要他呢？”“恩，是这样的”我示意 pidgin 稍等 一下，然后说到，“这个 QQ 也不止一个，发配到咱们 Linux 世界的这只 QQ 是一个残次品，查皮那里的 QQ 并没有这么多毛病，而且功能很多，再加上当年网 络上民智未开之时他就出来闯荡天下，所以造就了很多认为聊天就是 QQ，上网就是 IE，系统就是 XP 的人，好了，皮筋你继续吧。”“恩，咱们头说的不错，所以说，女主人离不开 QQ，而且还需要玩一些游戏，所以她来了之后，咱们肯定就被冷落了，肯定是隔壁的查皮启动的次数越来越多，咱 们越来越没有事情干了。”vim 忍不住说到：“那电脑也不是她一个人用阿，主人也要用阿，主人还是喜欢咱们的，只是可能他用的时间相对要减少了。”“这就 是我要说的关键！”pidgin 说，“你们想想，女主人要做的事情，必须在查皮那里才能做，于是当女主人需要电脑的时候，就算主人正在我们这里，也一定会 把我们关掉换成查皮。可是反过来，主人要用电脑做事情的时候，这时女主人正在查皮那里，请问你们，主人要做的哪件事情是非得来到我们这里做不可的呢？？他 非得用你狐狸上网么？查皮那里不也有 Firefox 么？他非得用 OO 么？那边的 Office 就不好用么？有什么只能在我们这里干的么？有么？ !! ”

又是一阵安静……

大家都默默的低着头，想着 pidgin 的话，不无道理。寂静了很久之后，忽然，一个声音打

破了沉寂。

“有，痛痛快快的杀毒！”avast 的声音。

“哈哈，对”狐狸说，“这么说的话，那还有痛痛快快的上网不怕中毒也算吧。”

“还有，还有，直接报个软件名就有人替他把软件都装好了。不用什么破解，也不需要什么这个算号那个密码的。他们哪有我的超级牛力阿！”这当然是 apt 说的。

“还可以随便修理他看着不顺眼的程序。”gcc 包工队的哥几个不知道什么时候跑出来了。

我也补充了一句：“装了驱动后不用重启系统是不是也算一个。”

“还有阿”图形界面的那帮人也过来了，“看到界面的电脑就启动好可以用了，不用再等各种莫名其妙的自动运行的程序。”

字符界面的命令们也忍不住说起来：“查皮不能做的事情？太多啦，rm -f /，他能嘛？哈哈”

“把所有当前目录下名字里带 abc 的文件的扩展名改成 xxx。”

“直接用我 dd 命令备份一下分区表。”

“同时让100个用户登录进来，1000个也没问题！”

“还有阿”这是更新管理器的声音，“还可以随时免费的更新系统，还有系统里的软件，总用最新版！呵呵。顺便向头报告，根据我的检查，源里又有软件更新了，请图形界面向主人询问是否要更新吧。”

这会大家的底气都足了不少，图形界面的同志们立刻开工，很快回来报告：“主人说更新！这是不是说明他还是很看好我们的？嘿嘿。”

我微笑着看着大家，虽然大家充满信心是件很高兴的事，不过，我也知道，更新，或许只是主人的一个习惯动作而已，未来的事情，还是不好说，也许，这可能是最后的更新吧……

“最新消息，最新消息～”皮筋小弟挥舞着手中的聊天记录喊道，“我们可能要搬家啦！”

“怎么回事？”“搬到哪去？”“搬什么家？”内存里的软件们一阵七嘴八舌的议论，哎，这帮家伙，一点也不淡定。我轻轻的咳嗽了一下，“咳！安静一下，皮筋你说。”

“是这样的，刚才主人跟未来的老婆聊天，未来的老婆，也就是原来那个 MM，也就是我们以后的可能的女主人——”“你快说重点！”暴脾气的狐狸妹妹说。“恩，重点是，主人老婆也是个一天到晚离不开电脑的家伙。”“那咱不是就惨了嘛？”gedit 问道，“她用的时间越 多，咱们出来放风的时间就越少嘛。”“你别急阿，我还没说完呢。主人觉得跟老婆抢电脑的胜算比较渺茫，因此，决定，以后一人一台电脑！！然后那 MM 也同意 这个建议，并且觉得自己对电脑配置的要求不是很高，所以，让主人用新的去，自己用这个旧的就行了。4G 内存阿，足够她用了。” 我听了也不禁一阵激动，问皮筋：“就是说，我，我们，以后，很，很可能，在”

“头儿阿，你怎么都结巴了……”“咳咳”要时刻保持老大的威严，恩，恩，“恩，我是说，我们很可能要搬到一个硬盘更加宽敞，内存更加广阔，CPU 更加富裕，显卡更加费电的新的电脑里去 了？”“对呀！”皮筋补充到，“而且，主人还说，既然有两台电脑了，那么就不像装双系统了，偶尔用网银什么的就可以去 MM 的电脑上，新的电脑就只装 ubuntu 就够了。”“哇塞！！！”我终于忍不住了，“到时候整个硬盘就都是咱们的啦！哈哈哈哈，到时候硬盘咱想怎么用就怎么用，狐狸妹妹一装装4 个，5.0，6.0，7.0，8.0，然后看他们互相打架，哈哈哈。” 狐狸在后面一脸怒容……

要搬家的话，其实也简单，首先就是要备份好整个系统。这个之前主人也做过，用 U 盘启动的一个叫做 slax 系统备份的，这回估计还会再找他来备份一下吧。有 人问，你就不能自己打好铺盖卷招收打个车自己搬过去？这个呢，其实在一个运行着的系统里备份自己这个系统也不是不行，不过不是很靠谱，最好还是在系统没运 行的时候利用另外的系统来备份。这主要是因为系统运行的时候会产生很多临时的虚假的文件。比如/dev/下的所有东西，基本都是根据计算机

的硬件设备动态 创建的，在系统不运行的时候，这个目录基本是空的，当系统运行起来了，这个目录里就会有很多的设备文件，这个时候你用 tar 打包整个系统，会把这些文件也 打进去，到了新的机器上，硬件设备不同了，就可能会有问题。还有/proc，里面的东西看起来一大堆，实际都是内存中的东西，反应着当前系统的运行状态， 系统关闭后这个目录也是空的，所以这个目录也是不需要备份的。

我们的主人既然用 slax 备份过我们系统，那看来备份这件事情就不用我们操心了，就等着被打包就行了。那么如何在新的机器上还原呢？这也不是我们系统操心 的事，肯定也是由别的系统来完成（多半还是那个 slax）。这个很好理解，我们系统被打包了嘛，在我们被还原之前是没法运行没法干活的，只能等着别人把我 们安装到一台机器上。安装的过程大约就是先吧硬盘分好区，不一定跟原来机器上的分区大小一样，分区的个数也不一定一样，只要保证解压后的目录关系不乱就 行。比如，原本我们这里是分了/,/boot,/home swap 四个分区，但是新的机器上可以只分三个：一个放 swap，这个不用管；一个放/home，把原/home 分区打的包解压到这个分区里；一个放/，把原/分区的包解压到这里，并且，由于没有单独的/boot 分区了，就把原/boot 分区的包解压到/分区下的 boot 目录里。当然，这样做了之后要相应 的修改/etc/fstab 文件。

好了，这些技术问题交给主人，交给那个 slax 系统，我就不管了，我只需要等着，等着在新的宽敞的电脑里，开始我们新的生活。说不定那个电脑，有8个 CPU，8G 的内存，8T 的硬盘，8块显卡做交火，8M 的光纤来8根……

这一次，当我走进那间再也熟悉不过的房子，看到那满地狼藉的时候，我知道，该来的终于来了。

我是个普通的门房，住在这间狭小的传达室里。 大名叫 **Grub**，因年长了几岁，所以他们都管我叫 **G** 大叔。其实，我就是个看门的，虽然能认识不少磁盘格式，虽然也会装载二进制程序（就是装载内核），虽然 他们说我简直是个准操作系统，可是我知道，我不过就是个看门叫人的而已。我的职责就是在每次用户启动电脑后问他是要叫醒那兔，还是叫醒那妻。让叫谁我就去 叫谁，就是这么简单。

我每次接到用户的指令，绝大多数是去叫兔子来工作。那孩子还真是不错，人机灵，也懂礼貌，每次看见我来叫他，总不忘 在收拾东西启动的时候顺便跟我拉拉家常，最近吃的怎么样，身体如何，前两天北京下雨有没有看见海，呵呵，反正就是闲聊天嘛。有时候还不忘了忙里偷闲的在工 作的时候送点茶叶到我的传达室里，他知道我爱喝茶。哎，可是阿，今天一开门，就看见他屋里一地残破，看不到一件完整的东西。走到他每天睡觉的地方，还能辨 认出有几片碎片，是他平时穿的衣服，人，已经不在了——格啦。我当时站在那，就愣住了。两年多啦，每天看着他工作，看着他越来越成熟，**bug** 越来越少，功 能越来越多，用户的夸赞也越来越多。从一开始，用户只是迷茫的徘徊在我给他的两个选项之间，并最终进入那个擦皮的系统，哦，对，那时候隔壁还是擦皮呢，还 没有那妻呢。直到现在，每次都是义无反顾的让我去叫兔子。我能看出来，小兔子长大了，越来越会跟人交流了，越来越好用了。而如今，一点征兆都没有的，怎么 说格，就给格了呢。要格，也把我这个老头子格了吧。你把他们都格了，留着我有什么用。可我这么一个小软件，呐喊又有谁能听得见呢。我只有无奈的在屏幕上显示出了我的大名：**GRUB>** 有什么事你就跟我说吧，你把他们都格了，还想让我干什么？我还能干什么？

是啊，什么也干不了了。我只能回到我的小屋，泡上一杯兔子送我的茶叶，享受着，最后一

段，平静的时光。用户知道我没用了之后，就会带着拆迁队来拆我的小传达室的，一般是那个 fixmbr。说是 fix，说是要开发我这块小地方，可是，他会管原来住在这里的人的死活么？？ 他是要让他的人住进来，让那妻的人住进来，让有钱的人住进来，所以就得赶我走！哼，来吧，来强拆了我吧，等着你们，我着一把老骨头了，不怕变成屋里头兔子 那样的碎片！一只兔子被格掉，千百个兔子装起来，哈哈哈哈……

五分钟后，一张 Windows XP 的光盘被塞进了这台机器里，光驱一阵悸动，一个叫 fixmbr 的软件被放了出来，直奔传达室。

与此同时，在不远处的另一台电脑的内存里，有几个熟悉的声音：“哇塞，真的是8G 耶 !这电脑比我还超级牛力阿。”“咳咳，低调低调，不过怎么没有八颗 CPU 呢？”“我说头儿阿，您也别太贪心了，还能都让你说中阿，有四个就不错了。”“恩，妹子教训的是。咦？这是什么？”“叫蓝牙，没见过吧。”……

外传：

《笨兔兔的故事》一周年特别纪念

Firefox 的年度总结：

跟随笨兔兔老大来着里工作一年了，今天老大让每个人写个年度工作总结。想想时间还真快，已经一年了，我在这个队伍里算是元老了，看着这一年里我们的队伍发展壮大，我们的环境越来越完善，心中感到无比欣慰。同时，我也在逐步提高自己的业务工作能力。从最开始连 Flash 都不会放，到现在集各种插件扩展于一身，报告天气，下载软件，屏蔽广告等等无所不能，还时常换漂亮的衣服让主人眼前一亮。当然，除了亮点，也有不足，早上起床的速度一直没有令主人满意，这是我今后需要进一步完善的地方。在今后的工作中，我将更严格要求自己，努力工作，发扬优点，改正缺点，开拓前进，为美好的明天奉献自己的力量。

Opera 的年度总结：

跟随笨兔兔老大来着里工作快一年了，今天老大让每个人写个年度工作总结。想想时间还真快，已经一年了，我虽不是从一开始就跟随老大工作，但在这个队伍里也算是老员工了，看着这一年里我们的队伍发展壮大，我们的环境越来越完善，心中感到无比欣慰。同时，我也在逐步提高自己的业务工作能力。从最开始中文字体看着眼晕，到现在完美的文泉驿字体，并且集各种功能于一身，登录邮箱，登录 IRC，RRS 等等无所不能，还时常提醒主人有新版本可用。当然，除了亮点，也有不足，插件太少没能令主任满意，这是我今后需要进一步完善的地方。在今后的工作中，我将更严格要求自己，努力工作，发扬优点，改正缺点，开拓前进，为美好的明天奉献自己的力量。

笨兔兔：为啥你俩的年报这么像？

答：因为她俩搜到了同一个年报模板……

gedit 的年度总结：

今年没啥事。

笨兔兔：唉，自从装了 vim，主人是不咋用 gedit 了。

AVAST 的年度报告：

今年收成一般，去隔壁 4 趟，去各种 U 盘里 6 趟，总共才砍死三千多只，可是也不怪我啊，他那里就那么点病毒，砍光了我也没辙啊，也不能乱砍是不是。还有哈，以前去总部领通缉令总是整本整本的领，其实那么一大本，前面的内容都是重复的。现在我终于学会就光领最新的那几篇了，节省了带宽。再就是……哎呀，我这个，长得有点难看，等回头有功夫得整整容去。

笨兔兔：你一年砍不到一只才好呢。

星际译王的年度报告：

今年学习了英语、日语、俄语、法语、德语、匈牙利语、西班牙语、南斯拉夫语、北斯拉夫语、西斯拉夫语……

笨兔兔：他能跟八国联军对着骂街了。

gcc 的年度报告：

等了一年，终于有点活儿干了，结果竟然是让我生产垃圾！

笨兔兔：Rubbish 系列……残念……

Mplayer 的年度报告：

博物馆奇妙夜 2.rmvb 变形金刚 2.mkv 窃听风云.rmvb 建国大业.avi □□□.avi □□□.flv □□□□□.flv □□□□.rmvb □□□□□□□.wmv……

笨兔兔：后面这段掐了

Pidgin 的年度报告：

我哪不好啦？我哪不好啦？啊？为啥让 Empanthy 代替我？为啥？你说为啥啊？你说啊，你说啊……

笨兔兔：淡定，淡定

笨兔兔的故事

懒蜗牛Gentoo 著

http://forum.ubuntu.org.cn/