

笨兔兔的故事

作者：懒蜗牛Gentoo



U B U N T U
S T O R Y

笨兔兔的故事官方网站：WWW.UBUNTUSTORY.TK

(1) 开端

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，我是一个操作系统，我是一个linux，我是Ubuntu。

在今年（2008）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，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WindowsXP，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WindowsXP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，呵呵。不过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，去睡觉前，一定要记得把内存里的东西，有用的，都搬回自己的那间硬盘里，以备下次使用。这道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。

(2) 醒来

自从安顿到这个地方，对周围的硬件环境熟悉了不少。我们在Canonical学校的时候就进行了充分的学习，所以这里的東西我基本上都会用的比较顺手。像Realtek的网卡，Intel的北桥，南桥，声卡以及E8400双核CPU，更是应用自如。Intel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 8800 Gt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：给你10秒钟，快说要叫哪个！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，那我就知道了，因为那时我肯定在睡觉，上次说过，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，小子，开工啦！

to be continue...

(3)工作

一听说开工，我很麻利的蹦起来，以迅雷不及掩耳盗铃之势，嗖的一下就越进工作室——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的列表，如 cn99之流。然后，主人决定记录下这些地址，于是，我又叫醒的gedit小弟。gedit个头很小，身体轻盈，一下子蹦进来，开始干活。然后，我又在主人的命令下去叫Rhythmbox，去叫pidgin，去叫.....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，他们当然不知道该做什么，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，那么我就水告诉他俩，一人用一会儿。但是这可不是没人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，可是有不能把他请回硬盘，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，并且把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。

(4)历史

住了一段时间，慢慢的开始对隔壁的邻居感兴趣起来。借主人上网的时间，顺便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fasterfox插件以后。

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——Windows1.0 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Windows2.0。还是那个有点软的公司，还是那张脸，跟他爹长的还真一样。

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，并不被大家看好。而Windows1.0和windows2.0最大的不一样的地方在于——2.0有个好儿子，1.0没有。

Windows3.0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Windows3.0的一个重要分支版本Windows3.2开始认识 Windows的吧。1990年5月22日，Windows3.0正是发布，而第二年，1991年发布了Windows3.0的多国语言版本。而同年，1991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，Windows3.0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Windows3.1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Windows3.2，相信他的样子大家都见过，什么？你没见过？好，贴张照片吧。

Windows 1.0 照片↓



Windows 2.0 照片↓



Windows 3.20 照片↓



(5)也是历史

到了1995年，Windows家的祖坟上终于冒青烟了，windows95一下子把人们从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windows95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，windows95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，windows95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，windows95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，windows98问世了。这个系统是基于 windows95编写的，他修正了windows95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windows95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 98 se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windows95到windows98se这一脉，到这里达到了顶峰，也走到了尽头.....

顺便说一下，windows95一脉还有一个家伙——window 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。随意，基本上可以把他算作windows98的一个不成功改版。

或许你会问，windows98se之后，不是还有windows2000，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.....

(6) 还是历史

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。到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——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，历史课就讲到这里，同学们，洗洗睡吧。

(7)串门

今天接到了任务，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，有人问了，那你能不能读懂呢？谁问的？站起来，好，这个问题，恩，基本上，我可以负责任的告诉你，自从Canonical学校为我们增加了一本ntfs-3g教材以后，ntfs就不在话下了。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。

(8)碎片

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。上次说过了，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，文件就像一个一个，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，或者理解为再拿个箱子也写上“日记”放在原来的箱子边上，可是不管怎样，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。那怎么办的，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”，在折腾到厕所“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。

(9)邻居

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。人们喜欢叫他OO,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!

(10)人才

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？其实很简单，我更倾向于把文件分散的放着，中间有足够的空间可用于扩展。这样就不至于在找东西的时候满屋子乱跑了。

刚刚休息了一下，主人又让我去叫醒一个家伙，他叫作apt-get。

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。征得同意后，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。哪天主人文一句：“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，他并不是只为我打工的。

以前，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 实际什么最宝贵？人才阿！

(11) 来头

这次主人叫apt，是让他去找一个软件，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。然后，在获得主人的同意后，apt出发了.....

“比海更广阔的是天空，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，不知道他是干啥的，有什么本事，不过既然主人要找他，总是有原因的吧。鉴于他名字念这不顺口，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碌，时不时拿出个小本，记录着什么。一直到关机，大家都去睡觉，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，拿着小本本记，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？

(12) 本事

这一天，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。狐狸妹妹被叫醒，一头雾水的看看我，因为每次都是我叫她。我也同样迷惑的看看老p，主人还没有发命令要用firefox阿，怎么就把她叫醒了呢？但既然被吵醒了，狐狸妹妹也就不睡了，迷惑的走进内存，看着老p。老p倒是镇定自若，一点没觉得有什么不对劲，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，所以，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叫进内存，跟firefox站在一起，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，当我看她的时候，她已经在工作状态了。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。没过多久，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，主人用啥他先晓。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。

(13)开源

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。

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，比需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）。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，睡好谁坏，只是不同的理念而已。

(14) 故事

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

说minix，就不得不说说Unix。UNIX也是一个操作系统，而且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奇异电子(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、激光器、太阳能电池、发光二极管、数字交换机、通信卫星、电子数字计算机、蜂窝移动通信设备、长途电视传送、仿真语言、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。麻省理工大学更是历史悠久，技术雄厚。所以，这个MULTICS操作系统的项目在1965年成立，到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。

看似事情就这么结束了，而，其实故事才刚刚开始，因为英雄的出现。

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，系统能够优化的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Dennis 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，当然得有个系统，这个系统，就是Ken Thompson和Dennis Ritchie用汇编语言写出来的，非常简陋的，UNIX的前身。这都是为了玩个游戏阿~

(15)minix

在强大的，玩游戏的欲望的驱使下，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。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。

好，Unix就说到这里，我们的正题是Minix。

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的人才。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

(16) linux

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，linus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。他并不知道即将创造的将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.....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.....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的服务器上让你给大家下载，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。

(17)Friends

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。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可算是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RedHat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）.....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）

(18) 杀毒

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！”

“这软件干什么的呀？”

“本APT有超……”

“算了算了，我不问了，快去吧超级牛力。”

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。心想，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，而且查杀不易。他们有的会伪装成别的软件，查皮叫醒“记事本”去干活，殊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回随着查皮一起起床。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在叫起来。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~” 甚至还能监视着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——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。

(19)免疫

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～

笨兔兔老师第二讲——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）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，只要使用windows系统，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，装防火墙的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，那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。

好了，这节课先上到这里，下次见。

(20) 权利

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，还摆脱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東西。所以，当你用root账户登录进来之前，一定要想清楚，自己身上的责任。

