

关于 Linux 兼容内核的知识产权问题(一)

毛德操

自从笔者提出 Linux 兼容内核的设想和技术路线、把兼容内核的主体归结为“一个框架，两个界面”以来，听到不少朋友表示了对于知识产权问题的关切。“你在 Linux 内核中实现微软的 Windows 系统调用界面；还拿着 Windows DDK 按图索骥，实现所列举的内核导出函数；又要实现它的设备驱动框架，将来微软会不会告你侵犯它的知识产权？那可是不得了的事！”。当然，我们必须重视和正视知识产权问题，不能乱来。这意味着我们必须小心谨慎，不要越出法律所规定的界线；但是这同样也意味着我们应该具体问题具体分析，而不要草木皆兵，自己把自己吓住，连法律允许做的事情也不敢去做了。这里，笔者要对 Linux 兼容内核的实现是否会侵犯微软的知识产权作一番分析和探讨，意在引起冷静而科学的讨论。

笔者并非法律专家，也没有深入研究过中、美两国知识产权方面的法律条文。笔者在知识产权方面的知识主要来源于两个方面。一方面来源于本人在中国和美国申请专利的经验。另一方面，更重要的是，来源于“Patent It Yourself”一书。这本书在美国的知识产权领域可谓尽人皆知。书的作者 David Pressman 是法学博士，又是个有着 40 多年经验的专利律师，还在美国专利与商标局(US PTO)当过审查员。该书从 1985 年初版以来，到 2004 年已是第十版了。笔者手头所有的正是这本书的第十版，今年春节时才买的。原来有个老的版本，搬家时不胜重负而割爱了，后来因为考虑 Linux 兼容内核的知识产权问题而又想到了这本书，只好再买一本新的，真是早知今日何必当初。

“Patent It Yourself”书中的叙述自然代替不了正式的法律条文，何况书中所讲是美国的法律，阅读理解还可能有偏差，所以笔者希望国内法律界人士不吝赐教，更希望他们能对开源软件施以援手。那怕与笔者所见相反，说出来让大家讨论也是件好事。不过，从另一个角度讲，则笔者所言毕竟还有些根据，并非自说自话。况且这本书在二十年中出了十版，即使早期版本中有错误也应该早已纠正了。

据“Patent It Yourself”说，知识产权分 4 种，即：

- 专利 (Patent)
- 商标 (Trademark)
- 版权 (Copyright)
- 商业机密 (Trade Secrets)

此外，还有“不正当竞争(Unfair Competition)”也与知识产权有关，有些行为够不上对 4 种知识产权保护中的任何一种的侵犯，却还是可以按“不正当竞争”提起诉讼。

专利是受保护力度最强的，特别是如果受保护的是一种“方法”，那就比较宽泛，比较难以绕过。但是，专利受到的限制也很大。首先是必须充分公开实施的过程与方法。所谓充分公开是指：具有一定基础的同行只要发挥合理范围内的努力，按照所公开的过程与方法去

做，就可以实现所述的结果。另一方面，专利受到保护的时间限于 18 年，此后就不受保护了。还有，如果一项技术出现在商品中，那么必须在商品上市后的一年之内提出申请，过了一年以后就不再受理了。现在我们可以来对照一下，看看“一个框架，两个界面”是否有可能触及微软的专利。先看 Windows 的系统调用界面，这个界面本身就是不公开的，更别说界面的实现了。人们现在之所以知道这个界面以及它的一些细节，是因为有人通过“逆向工程”作了分析、实验，并写成书出版了。显然，这个界面不可能受专利保护，否则就违反了公开原则。不过，这并不排除个别系统调用的实现方法是受专利保护的，所以当我们涉及具体系统调用的实现时还是得要小心。但是有一点是明确的，那就是在这一方面至今还没有专利的东西就永远不可能有专利了，因为这个界面已经存在多年。还有，今年是 2005 年，减去 18 年就是 1987 年，凡是 1987 年以前的专利(即使有的话)都已经过期了。再看 DDK 所列举的内核导出(export)函数，这些函数构成了一个设备驱动界面。这个界面倒是公开的，但是这些函数的实现方法却都不是公开的，所以也不可能受专利的保护。至于 Windows 的设备驱动框架，人们对其实现确实知道得比较多一些，那是得益于微软出版社出版的几本书(见“若干重要的 Windows 参考资料”)。微软既然要让人们为 Windows 开发各种第三方设备驱动软件，就得把设备驱动的整体框架从宏观上作些介绍。但是，即便如此，公开的程度也远远没有达到专利申请所要求的程度。更何况，Windows 的设备驱动框架和一些机制在很大程度上来源于 VMS 操作系统，而 VMS 是早就存在了的东西。此外，专利保护并不是全球性的。假定某个方法在美国得倒了专利保护，但是并没有在中国申请专利，那么在中国市场上就不存在侵犯该项专利的问题。可想而知，微软的有些技术即使在美国受专利保护，在中国却未必有效。

由此可见，从总体上说，“一个框架，两个界面”都不可能触犯微软的任何专利。

第二种知识产权是商标，显然兼容内核跟微软的商标没有任何瓜葛。

再看版权。版权的取得比专利容易得多，只要登记就可以了，甚至即使不登记也可受到版权保护。版权受保护的年限也比专利长得多(发表后 95 年，或作者死后 70 年)。可是，版权所保护的范围却很窄。“Patent It Yourself”书上说：“While a patent can effectively provide offensive rights on an idea per se,a copyright covers only the author’s or artist’s particular way of expressing an idea”。就是说，专利可以保护一个主意(必须可以实现)的本身，而版权却只能保护对一个主意的特定的表达。David 进而举了一个例子加以说明：“Put otherwise, you are free to publish any of the ideas, concepts, and information in this (or any) book, provided that you write it in your own words”。就是说，如果把“Patent It Yourself”改写一遍，那么即使你另外出版也不算侵犯版权(原书中没有提到翻译是否算是改写)。但是，书中也提到，1998 年的“Digital Millennium Copyright Act”法案有个补充，就是如果有谁提供一种技术来破除版权保护，那么这是一种刑事犯罪。笔者对此的理解是：例如，要是有人写一个程序，自动

把“Patent It Yourself”变换成另一本书、即另一种表达，那么这个人是要受到惩罚的。具体到计算机软件，该书又说：“...copyright covers only the particular way the program is written, not what it does”。这就是说，如果两段程序实现的是相同的功能，但是编写的方式不同，那就不存在侵犯版权的问题。

当然，卖盗版盘是侵犯版权，因为卖的正是别人的“特定的表达”；把好几张盘片的内容压缩到一张盘片上然后加以生产可以是刑事犯罪，因为这正是在提供一种技术来破除版权保护。

可是，联系到兼容内核的实现，我们根本就看不到微软的代码，看不到他们“特定的表达”，又怎么能因为我们自己的代码而侵犯他们的版权呢？退一步说，即使我们看到了他们的代码，只要不是成段照抄，也还是谈不上侵犯版权。不过这里倒有个问题需要注意：在设备驱动框架中有不少宏操作(Macro)是在 DDK 的一些.h 文件中定义的，我们不能把这些.h 文件照搬和拿来就用，而需要加以改写，以免授人以柄。

还有一种知识产权是商业机密。商业机密的保护范围就更小了，基本上只是用来防止雇员泄密和偷盗。但是，如果明知某人手里有别人的商业机密而购买之，或者指使或怂恿别人偷盗商业机密或泄密，那也是违法的。除此之外，就没有什么限制了。“Patent It Yourself”中有一段话很说明问题：“...they can't be maintained when the public is able to discover the information by inspecting, dissecting, or analyzing the product (called “reverse engineering”)”。这里的“they”指商业机密，只要公众能通过仔细观察、解剖、分析得知个中奥妙，商业机密就维持不下去了。值得注意的是这里提到了“逆向工程”，显然“逆向工程”并不违法。有些软件产品在安装时有要求使用者不得进行逆向工程的条款，但那只是使用者与提供者之间的协议，而与法律无关。

具体到我们的兼容内核，这就是说，只要我们不从微软偷盗源代码，也不使用别人从微软偷盗出来的源代码，那么即使通过反汇编等手段进行逆向工程，那也没有违法，更何况我们根本没有这个需要。Linux 内核中有那么丰富的素材，有那么优秀的源代码，我们何用劳心费力去做 Windows 的逆向工程？

如果说上面 4 种知识产权都有“硬杠子”、有具体的条文可循，那么“不正当竞争”可以说是个灰色地带。书中说：“The scope of unfair competition law is nebulous in the first place and is regularly being changed by judges who make new and often contradictory rulings”。模糊(nebulous)自不必说，这里说“老是在被法官们变化”是因为美国很重判例，这个法官这次那样判，就好比为不正当竞争增加了一条司法解释。总而言之，对于“不正当竞争”诉讼，法官们的发挥空间比较大。那么什么样的行为可以看作是不正当竞争呢？书中举了几个例子。例如借用别人的广告词；又如故意使自己的商标与别人的商标非常接近；再如有的店铺如麦当劳的房屋外形和室内布局都有统一的格局和特色，如果别人新开一家快餐店，却把店

铺的格局搞得和麦当劳一样；还有些产品的外包装很有特色，久为人们所熟悉，如果别人新出一种产品，也采用同样或很相似的包装；等等。总之，是使用户一不小心就误以为你的产品是另一家的产品。在这一方面要注意的是：将来，当兼容内核直接支持 Windows 应用时，作为人机界面的“桌面”布局或图案要与 Windows 有所不同，使人们一看就知道这实际上并不是 Windows。可是，相似、很相似、这些都是主观感受，很难有个确切的标准，这就是之所以模糊的原因。

美国公司 Linspire 与微软之间的诉讼是个很有意思的实例。Linspire 原名 Lindows，其产品是安装了 Wine 的 Linux 桌面系统，因而可以在 Wine 上面直接运行一些常用的 Windows 应用，例如 Office 等等(所以才叫 Lindows，笔者猜想)。Wine 在内核外面(用户空间)实现了 Windows 的 Win32 API，本质上可以跟在内核里面实现设备驱动界面相类比。后来，微软告了 Lindows。但是微软告的并不是 Lindows 通过 Wine 侵犯了 Win32 API 的版权(根据前面的叙述，我们知道那是不成为诉讼理由的)，而是说“Lindows”与“Windows”只是一字之差，太相似了(我们知道那有可能被认为是不正当竞争)。可是，“Lindows”与“Windows”到底有多相似呢？确实只有一个字母的差别，但是外形却不像(如果是 Vindows 那倒还像一些)，这就公说公有理、婆说婆有理了。这个案子最后以庭外和解告终，Lindows 把名字改成 Linspire，而微软则为此付了 Linspire 一笔钱(是微软付钱给 Linspire，不是 Lindows 付给微软)，以补偿 Lindows 因改名而造成的损失。前面讲过，美国很重判例。有这个案子在先，微软实际上已经不可能再就 Wine 实现了 Win32 API 而提起诉讼。那么，兼容内核实现系统调用界面和设备驱动界面与此并无实质的不同。

由此可见，我们完全不必草木皆兵，以为一开发兼容内核就会触犯微软的知识产权。当然，小心谨慎还是应该的。

本文所述只是针对兼容内核本身的考虑，而没有涉及用户空间的一些问题，那将是另一篇短文的内容。