

UNIX 是强大的

让聪明人干任何他们想干的事情。

UNIX 的一个特点就是非常高的灵活性，Xwindow 也具有这种灵活性。这种灵活性体现在哪里呢？

UNIX 的程序一般都有很多参数，不管你现在用的着不着，总有人需要某些参数。它们的行为很多都可以用配置文件来改变。比如 GNU bash, 通常缺省的命令行输入方式是 Emacs 方式，但是只要我编辑一个 .inputrc 文件，就可以把它变成 vi 的输入方式，而且我还可以自己绑定键序列到某些操作。我可以用 shopt 来设置它的很多特点，比如是否进行通配符扩展，是否可以把一个变量当作一个目录来 cd，是否可以自动纠正某些明显的目录名打字错误

UNIX 程序设计的思想是提供给用户“机制”，而不限制用户制定“政策”。这是一个重要的尊重用户的作法。

我们再来看看 Xwindow。Xwindow 是一个出色的设计，它把显示服务器和客户程序分开。一个显示上既可以显示本机上的程序，也可以显示别的机器上的 X 程序，而它们都遵守你的窗口管理器的统一指挥，它们之间可以方便的传送剪贴版数据，各种事件 比如有时我的 XFree86 上会出现四个不同机器上的 XTerm，两个不同机器上的 Gvim，..... 它们统一受本机上的 FVWM 指挥。

Xwindow 程序都具有很多很多命令行参数和 resource 参数。你可以随意的在命令行或者 .Xdefaults 文件设置所有的颜色，字体，尺寸..... 而且如果你用 xrdb 把 .Xdefaults 导入到根窗口，那么其它机器上没有经过配置的同样的程序，显示到你的机器上的时候也会遵守同样的外观规定。

Xwindow 的窗口具有 Property，也就是一些可以自己定义的共享数据(原子)。正是因为这些 Property 的存在，使得 Xwindow 具有无比强大的生命力。X 的窗口管理器和其它客户程序之间并没有统一的协议，但是后来出现了 ICCCM(客户程序间通信规范)，这个规范就是通过 property 定义的。现在又有人定义了一套“扩展的窗口协议(EWM Hints)”，使得 Xwindow 可以具有某些 Windows 的特征，比如一个工具条程序可以告诉窗口管理器：“这个屏幕下面被我占据了24个像素的空间，你最大化程序的时候不要越过这个界线。”

一个强大的窗口管理程序比如 FVWM，它收到这样的提示时，可以答应工具条程序的这个要求，也可以不答应。一切选择的权力在于谁？当然是用户了！一切窗口乖乖听话，FVWM 给予用户最大的尊重。

你想想，是不是有些 Windows 程序常常弹出一个窗口要你选择 "Yes or No"？你不点击它它就不下去。你觉不觉得你的程序在侵犯你的尊严？你是一个人，一个智慧的生物，怎能受到一个程序如此的待遇？

还有就是很多 Windows 程序把人当成傻瓜，而它是“智能程序”。比如，有一个程序就是喜欢把你的每句话第一个字母都变成大写，我不说它是谁了，你遇到的时候就知道了。如果连“一句话开头一个字母要大写”这么明显的问题都需要程序帮你纠正的话，人脑还用来干什么？况且如果你故意想要不大写的话，那就更麻烦了，我愣是没有从它那一大堆菜单里找到怎么关闭这个愚蠢的选项。

只有符号才能完全操纵计算机

我们来说说很多初学 Linux 的用户。虽然他们在用 Linux，但是他们打心眼儿里是觉得 Windows 的工作方式好，他们希望 Linux 有一天能“像 Windows 那样”。你说：“我鼠标一点，我菜单一拉，..... 就可以完成我的操作。”但是我要告诉你：“Linux 从来没有摹仿 Windows，将来也不会。Linux 从诞生之日起，它的工作方式就比 Windows 的先进。Linux 属于能勇敢面对符号的人。只有符号才能完全操纵计算机。”

看看优秀的 UNIX 程序，XFree86, FVWM, VIM, Emacs, proftpd, Mutt, wget, tin, ... 没有一个不是用配置文件来设置选项的。为什么这些程序没有方便的菜单可以用来配置？难道它们的设计者就那么低能，连个图形配置界面也写不出来？

当然不是。因为图形界面配置方式的能力是极其有限的，而配置文件和程序语言的表达能力却是无限的。用图形界面配置这些程序的话，如果你想达到配置文件的效果，你需要成百上千的菜单，checkbox, radio button, ... 到时候你根本没办法找到你需要修改的地方了！而各个程序的配置文件的语法都有很多相似之处，一般就是一些命令，设置一些变量，参数，..... 一旦用会了一个，其它的也就容易理解了。如果你用惯了 awk, sed, Perl，你会觉得那才是真正的自动化啊。

鼠标虽然是很好的工具，但是它的表达能力是有限的。你不可能光用鼠标就让电脑完全明白你的意思，它毕竟只有3个按钮。看看我的 MetaPost 页你就能体会到鼠标的这一弱点。所以我们虽然很喜欢鼠标，但是却不能完全依赖它。

各个小程序的完美配合

这就是UNIX最重要的特点了，它就是UNIX设计的思想。让每个程序只具有一项专门的能力，然后让它们合作。Xwindow也继承了这种好传统。

这恐怕就是Windows和其它操作系统望尘莫及的地方了。UNIX 程序设计之统一，配合之完美，真使我难以置信！shell, grep, find, awk, sed, make, Perl, Emacs, vi, tin, Mutt, ... 它们是那么的具有一致性！你一旦学会

了 sed 的正则表达式，其它程序基本上都能用了。你一旦学会了 vi 和 VIM，你会发现它的操作是那么的有规律性，似乎vi的设计者在几十年前就已经设计好了 VIM 在今天的完美而统一的操作方式！而且vi的操作还体现在 Mutt, tin 等很多程序中。你甚至可以把 bash 设置为 vi 的输入方式来输入命令行，我就是这么做的。一个程序可以调用另外一个程序来得到数据，可以把数据交给它处理后返回来，可以在自己的窗口里“嵌入”另外一个程序。

在 Windows 和其它非 UNIX 操作系统中，这种合作是非常困难的。我曾经在 Windows 下使用 Perl来进行一些自动工作。但是 Windows 的文件操作，管道是如此的不稳定，程序之间基本不能合作。你别想在 Visual Studio 窗口里面嵌入 UltraEdit 编辑器，你别想用 expect 脚本来控制 telnet 到水木清华 BBS。

Windows 的程序都是大而全，大而杂，所有的电子邮件程序都需要自己提供编辑器，自己发送和收取邮件，自己显示邮件的附件。每一个BBS程序都提供自己的Virtual Terminal，自己的通讯代码。每一个 IDE 都自己提供编辑器，编译器，汇编器，调试器。人们为了使用一种新的程序，需要适应所有这些它提供的界面，而不能使用自己喜欢的编辑器的键绑定，菜单组织..... 不能 DIY！

你要知道，最高级的电脑是定做的，自己想要什么什么CPU，什么主板，多少内存，什么硬盘，键盘，鼠标，显示器都是自己选择的。最高级的滑板，自己想要什么牌子的版面，什么牌子的沙，什么桥，什么轮子，什么轴承，也都是自己选的。最高级的乒乓球拍，木板，胶皮，海绵，胶水都是可以自己选择..... 而用 Windows 程序，你得到的是大杂烩，就像你去买“品牌机”，只有那么几种配置，而且附带很多你不需要的软件和服务；就像你去买组装好的滑板，你想要大一点的轮子和窄一点的板子，但是你没有这种选择余地！Windows 程序就相当于最廉价，最次的滑板。但是它却会花你更多的钱，因为一旦一个部件坏了，或者你不喜欢了，你不能另外找一个好的换掉它，你必需重新买全套配件！

而 UNIX 和 Xwindow 就是高档的“组装货”。比如我用 Mutt 的时候，我可以用 VIM 也可以用 pico 来编辑邮件，我可以用 ImageMagick 也可以用 xv 来显示附件里的图片，我可以用 lynx 把 HTML 附件转成文本嵌入窗口中，我也可以把 HTML 附件交给 Mozilla 图形显示。我可以让 GnuPG 帮我把邮件进行数字签名和加密，我也可以用其它 PGP 程序。我想让 Postfix 而不是 sendmail 帮我发出邮件，我想让 fetchmail 帮我收邮件，转发给 postfix，然后被我自己写的Perl过滤器处理..... 这一切我都可以办到！我可以选择我最喜欢的专门的程序来完成专门的工作，然后把它们结合在一起，我也可以分别得到它们的好处。

[上一页](#)

UNIX 是永恒的

[上一级](#)[起始页](#)[下一页](#)

学 UNIX 绝对不是浪费时间