<https://blog.csdn.net/zhangna20151015/article/details/51280578>

linux图形界面基本知识（X、X11、Xfree86、Xorg、GNOME、KDE之间的关系）

LINUX初学者经常分不清楚linux和X之间，X和Xfree86之间，X和KDE，GNOME等之间是什么关系。常常混淆概念，本文以比较易于理解的方式来解释X，X11，XFREE，WM，KDE，GNOME等之间的关系。

首先要明确的是Linux X Window System的基本原理，X是一个开放的协议规范，当前版本为11，俗称X11。X Window System由客户端和服务端组成，服务端X Server负责图形显示，而客户端库X Client根据系统设置的DISPLAY环境变量，将图形显示请求发送给相应的X Server。  
**一、linux本身没有图形界面，linux现在的图形界面的实现只是linux下的应用程序实现的。**

图形界面并不是linux的一部分，linux只是一个基于命令行的操作系统，linux和Xfree的关系就相当于当年的DOS和 WINDOWS3.0一样，windows3.0不是独立的操作系统，它只是DOS的扩充，是DOS下的应用程序级别的系统，不是独立的操作系统，同样 XFree只是linux下的一个应用程序而已。不是系统的一部分，但是X的存在可以方便用户使用电脑。WINDOWS95及以后的版本就不一样了，他们 的图形界面是操作系统的一部分，图形界面在系统内核中就实现了，没有了图形界面windows就不成为windows了，但linux却不一样，没有图形 界面linux还是linux，很多装linux的WEB服务器就根本不装X服务器.这也WINDOWS和linux的重要区别之一。

**二、X是协议，不是具体的某个软件。**

X 是协议，就像HTTP协议，IP协议一样.这个概念很多初学者甚至学习LINUX有一定时间的人都混淆，一个基于X的应用程序需要运行并显示内容时他就联 接到X服务器，开始用X协议和服务器交谈。比如一个X应用程序要在屏幕上输出一个圆那么他就用X协议对X服务器说：喂!我需要在屏幕上画一个圆.X应用程 序只负责告诉X服务器在屏幕的什么地方用什么颜色画一个多大的圆，而具体的"画"的动作，比如这个圆如何生成，用什么显卡的驱动程序去指挥显卡完成等等工 作是由X服务器来完成的。X服务器还负责捕捉键盘和鼠标的动作，假设X服务器捕捉到鼠标的左键被按下了，他就告诉X应用程序：亲爱的应用程序先生，我发现 鼠标被按下了，您有什么指示吗?如果X应用程序被设计成当按下鼠标左健后再在屏幕上画一个正方形的话，X应用程序就对X服务器说：请再画一个正方形，当然 他会告诉服务器在什么地方用什么颜色画多大的正方形，但不关心具体怎么画--那是服务器的事情。

那么协议是需要具体的软件来实现的，这就是下面我要讲的:

**三、X和XFree86的关系.**

有 了协议就需要具体的软件来实现这个协议.就好比我们有了交通法规就需要交警去根据法规维护交通秩序一样.Xfree86就是这样一个去根据法规实现协议的 "交警".他按照X协议的规定来完成X应用程序提交的在屏幕上显示的任务.当然不仅仅是某个特定的交警才可以去维护和实现这个法规，比如还可以由交通协管 员来实现交通法规，必要的时候警察也可以介入，当然前提是他们都要懂得交通法规，也就是要懂得协议.所以实现X协议的软件也并不只有 XFree86，XFree86只是实现X协议的一个免费X服务器软件.商业上常用MOTIF，现在还有XORG，还有很多很小的由爱好者写的小的X服务 器软件.甚至可以在WINDOWS上有X服务器运行，这样你可以在linux系统上运行一个X应用程序然后在另一台windows系统上显示.多么神奇. 你可以用google找到这样的X服务器软件.只不过在LINUX上最常用的是XFree86.(现在的linux发行版都用Xorg了)顺便说一句，苹 果电脑的图形界面用的也是X协议，而且被认为是做的最好的X协议图形界面，并且他对X协议的实施是做在系统内核里的，所以性能明显好很多，这就是为什么很 多大型三维图形设计软件都是在苹果平台上的原因.

为了便于理解拿HTTP协议来比较:

协议是HTTP (hyper text transmission protocol)  
实现这个协议的常用服务器有:apache IIS 等  
请求这些服务器传输文件的客户有:IE ，MOZILLA ，NETSCAPE等.

协议是X

实现这个协议的常用服务器有Xfree86 ，Xorg ，Xnest等

请求这些服务器来完成显示任务的客户:所有的X应用程序.

只 不过HTTP协议的服务器和客户端通常都在两台不同的电脑(服务器和客户机)之间来实现，所以一般人对这个比较容易理解.而X协议的服务器和客户端程序通 常在同一台电脑上，因此很多用户都感到对此很难理解.既然是协议，那么肯定和平台无关的，因此可以让X应用程序显示在任何装有X服务器的远程计算机上.甚 至显示在装有Xnest服务器的windows系统上.只不过在大多数情况下，X应用程序都用本机的X服务器，然后X服务器将显示结果输出到本机的显示 器.这也是很多人不理解X是一种协议的原因.

**四、X和X11R6又是什么关系?**

不知道初学者有没有注意到/usr/X11R6这个目录，这是XFree的默认安装目录  
X11R6 实际上是 X Protocol version 11 Release 6  
(X协议第11版第六次发行)的意思，就是说目前用的X协议是第11版的，然后经过了6次小的修正.不同版本的X协议是不能通信的.就象我们现在IPV4和IPV6不能通信一样，不过不用担心，现在的X服务器软件和X应用程序都遵循X11R6.

另外XFree86 3.3.6 XFree86 4.3.6 等这些版本是实现X协议的软件XFree86的版本号.这是初学者经常高混淆的概念.

协议版本和实现协议的软件的版本--这两个概念的区别你分清楚了吗?

**五、X服务器和WM(window manager 窗口管理器)之间是什么关系.**

平时大家起动图形界面是怎么启动的呢?  
如果你是一开己就进入图形界面那就太遗憾了.应为你错过了了解X服务器起动过程的好时机.不过没关系.你打开一个XTERM输入:

init 3

就可以安全的回到字符界面.

好了，等做完以下实验你就完全明白X和WM(window manager 窗口管理器)之间是什么关系了.

先输入以下命令:  
#startx

起动图形界面，你看到的是一个和平时使用一样的完整的图形界面操作环境.  
你可以最大化，最小化，移动，关闭窗口等.

按ctrl+alt+backspace反回字符界面.

输入:  
#xinit

再次启动图形界面，你看到了什么，你看到一个XTERM.而且不能移动.但是你可以在这个XTERM中输入命令打开X应用程序，如果我输入:  
#mozilla  
打开浏览器，你看到的浏览器和平时有什么不同吗?他在屏幕中间，不能移动，不能最小化，不能最大化，没有边框.

为什么同样一个X应用程序会有这样的不同呢?因为我们用startx起动图形界面的时候同时也启动了一个WM(即窗口管理器)，如果你用KDE就起动了KDE，如果你用GNOME就起动了GNOME.但是你用xinit起动图形界面的时候却没有起动WM.

现在你明白窗口管理器的作用了吗?他的作用就是最大化，最小化，移动，关闭窗口等.而这些不是X服务器来负责完成的.

如果你用xinit起动图形界面并在xterm中输入twm，看看会有什么?  
xterm被加上了一个边框，你可以通过这个边框移动，最大化，最小化这个xterm，twm就是XFree86中自带的窗口管理器，是一个比较简陋的最简单的窗口管理器，但是他具有窗口管理器的全部特征.

如果你不输入twm而输入gnome-session就可以起动GNOME  
或者输入startkde起动KDE.

通过以上的实验你就可以清楚的明白他们之间的关系.

**六、关于KDE和GNOME**

KDE 和GNOME是LINUX里最常用的图形界面操作环境，他们不仅仅是一个窗口管理器那么简单， KDE是K Desktop Environment 的缩写.他不仅是一个窗口管理器，还有很多配套的应用软件和方便使用的桌面环境，比如任务栏，开始菜单，桌面图标等等.

GNOME是GNU Network Object Model Environment 的缩写.和KDE一样，也是一个功能强大的综合环境.

另外在其它UNIX系统中，常常使用CDE作为这样一个环境.

其它的小型窗口管理器有:  
window maker，after step，blackbox，fvwm，fvwm2，等等都是常用的优秀窗口管理器.REDHAT9中有 window maker 但是默认不安装，大家可以装来试试.只要xinit再wmaker&就可以用windowmaker了.

**七、linux图形界面层次关系总结**

linux本身-->X服务器<-[通过X协议交谈]->窗口管理器(综合桌面环境)-->X应用程序.

Xfree86服务器的实现包括两个部分，一部分是和显卡直接打交道的低层，一部分是和X应用程序打交道的上层。上层负责接收应用程序的请求和鼠标 键盘的动作。而和显卡直接打交道的底层负责指挥显卡生成图形，其实就是显卡驱动。上层接收到应用程序的请求后，将请求内容做适当处理，然后交给显卡驱动来 指挥 显卡完成画图的动作.另外，上层的捕捉键盘和鼠标动作的部分会向应用程序提供鼠标和键盘的状态信息，应用程序接收到这些信息后决定是否再有相应的动作.

平时说的VESA，VGA ，fbdev等其实就是针对不同模式显卡的驱动程序.  
VESA(Video Electronics Standards Association)  
VGA (Video Graphics Array)  
fbdev (FrameBuffer Device)  
等  
都是不同的显卡标准，不过这些标准都已经很老了.现在的显卡都兼容这几种模式.

framebuffer模式的显卡本身不具有任何运算数据的能力，他好比是一个暂时存放水的水池.CPU将运算后的结果放到这个水池，水池再将结果 流到显示器.中间不会对数据做处理.在这中情况下，所有显示任务都有CPU完成，CPU负担很重.从frambuffer这个名字我们就能猜测到他的功能 了(数据侦缓冲).

在linux内核中有有framebuffer机制，摹仿framebuffer显卡的这种功能.他的好处是把显卡的硬件结构抽象掉，把所有的显卡都当做一个"水池"来用.应用程序也可以直接读取这个水池的内容.framebuffer的设备名是 /dev/fb  
可以用命令:  
#dd if=/dev/zero of=/dev/fb  
清空屏幕.如果你的X用的是framebuffer的驱动，屏幕将全变黑.通过移动鼠标，切换窗口等动作可以让原先的内容重新显示.

如果你不用framebuffer驱动，可以在字符模式下做这个试验.屏幕将全部被清空.  
以下的命令:  
#dd if=/dev/fb of=fbfile  
可以将fb中的内容保存下来，以后可以重新写回屏幕:  
#dd if=fbfile of=/dev/fb

作用类似于屏幕截图，但是截下的图不能用普通图片浏览器看.

现在知道linux中framebuffer设备的神奇了吧.  
linux字符界面在高分辨率下，启动时会有一个小企鹅logo，这个logo就是用framebuffer功能写上屏幕的.

VESA和VGA比FB要好很多，但是他的显示能力也很有限.CPU仍旧要处理大部分的显示任务.

现在的显卡就不一样了，CPU几乎不参数显示任务，而专注于运算工作.我们平时在linux下装显卡驱动，其实就是替换掉XFree86中底层的那 一部分. 使他有更强的显示能力.比如你的显卡是nvidia的，XFree默认的驱动是"nv"，他是一个nvidia显卡的通用驱动，但是他不能发挥 nvidia显卡的性能.这个"nv"驱动模块可以在:

/usr/X11R6/lib/modules/drivers

目录下找到，文件名是:nv\_drv.o

这个目录下还有很多显卡驱动模块.  
apm\_drv.o dummy\_drv.o nsc\_drv.o sis\_drv.o  
ark\_drv.o fbdev\_drv.o nv\_drv.o tdfx\_drv.o  
ati\_drv.o glint\_drv.o r128\_drv.o tga\_drv.o  
atimisc\_drv.o i128\_drv.o radeon\_drv.o trident\_drv.o  
chips\_drv.o i740\_drv.o rendition\_drv.o tseng\_drv.o  
cirrus\_alpine.o i810\_drv.o s3\_drv.o vesa\_drv.o  
cirrus\_drv.o linux s3virge\_drv.o vga\_drv.o  
cirrus\_laguna.o mga\_drv.o savage\_drv.o  
cyrix\_drv.o neomagic\_drv.o siliconmotion\_drv.o

fbdev vga evsa 等都在其中.

只要在XF86Conf文件中的Section module段中指定了什么驱动，X就用哪个驱动.  
代码:

Section "Module"  
Load "dbe"  
Load "extmod"  
Load "fbdevhw" #这里用的是fbdev  
Load "glx"  
Load "record"  
Load "freetype"  
Load "type1"  
Load "dri"  
EndSection

这是XFree86 4.0以上版本的方法.即X主程序和驱动模块是分开的.

主程序是/usr/X11R6/bin/XFree86 (即上层部分)  
然后主程序启动的时候根据配置文件决定用哪一个驱动.

但是，在XFree86以前的版本(3.3.6)中， X主程序和驱动模块是合在一起的，因此，以前的Xfree86 3.3.6 有很多针对不同显卡的版本.

另外其他的X服务器如KDrive 也是驱动也主程序整合的，因此同样是KDrive X服务器会有 VESA ，SVGA ，和 fbdev的版本之分.因为这些都是轻量级的X服务器，为了简化结构他做成这样.

Xfree86 4 做成驱动和主程序分离的好处是用户可以自己添加升级驱动.