



by Guido Socher ([homepage](#))

About the author:

O Guido gosta do Linux não só por ser interessante compreender como é que os sistemas operativos trabalham mas também pelas pessoas envolvidas no seu desenho.

Correndo aplicações remotas com o X11

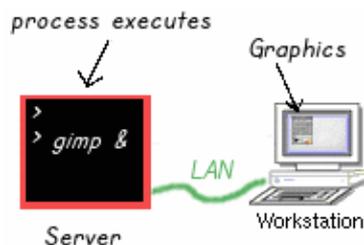


Abstract:

Nos primeiros tempos os utilizadores de Linux pensam que o desktop gráfico sob o Linux é um outro sistema "Windows" onde pode iniciar aplicações e estas aparecem em janelas separadas. Algumas pessoas apercebem-se que podem ter vários desktops e é isto que se parece. O Sistema Linux X Window (X11) é muito mais do que isto! É um sistema de janelas em rede. Veremos quais as novas e poderosas possibilidades que oferece.

O conceito de display

Qualquer aplicação gráfica X Window lê ao iniciar a variável de ambiente DISPLAY para descobrir o ecrã para onde deve enviar os gráficos. Isto, juntamente, com as capacidades de rede do X Window torna possível as aplicações gráficas correrem remotamente. Ou seja, utiliza a capacidade de processamento de uma máquina enquanto tem a aplicação a rodar a partir de outra. Toda a GUI (graphical user interface – interface gráfica do utilizador) aparece na máquina onde está a operar. Nem se dá conta que está a utilizar 2 computadores. A velocidade da rede é, claro, uma questão importante, mas uma ligação normal de 10Mbit/s na sua rede local é mais que suficiente.



Porque é que quereria fazer isto?

Existem imensas aplicações destes "gráficos de rede".

Companhias utilizam-nas para administrarem o equipamento remotamente, que pode estar a milhares de quilómetros de distância e você pode utilizar a mesma aplicação para administrar como se estivesse no site localmente.

Pode ter 2 computadores uma máquina rápida a 1 Ghz e um velho Pentium a 133Mhz. Pode apreciar a velocidade da sua nova máquina mesmo que não esteja sentado á frente dela.

Provavelmente, a sua irmã está sentada à frente da nova máquina e tem uma sessão aberta. Mas isto não importa pois continua a poder beneficiar dela.

Como é que trabalha?

Todas as aplicações X window, podendo-se chamar gimp, xterm, konquerer, netscape, ... são clientes de rede reais que se ligam a um servidor, o servidor X. A tarefa do servidor X é falar com o hardware de gráficos, desenhar as figuras no seu ecrã, ler os dados de entrada do rato e do teclado. Os clientes (os seus programas como o gimp, netscape...) enviam instruções ao servidor como desenhar as frames e botões. Em troca recebem do servidor os eventos de rato e teclado. Obviamente, que precisa de uma espécie de autenticação, caso contrário toda a gente podia-se confundir com o ecrã de toda a gente. Existem dois programas para controlar o acesso:

– xhost: utilizando este programa pode dar permissões a qualquer utilizador numa dada máquina para escrever gráficos para o seu display. Exemplo: está sentado à frente de uma máquina chamada philosophus. Para permitir o acesso a qualquer programa na máquina movietux para o seu display em philosophus você digitaria o comando:

```
xhost +movietux
```

Isto deve ser digitado numa shell em philosophus

– xauth: Este é um cookie à base de autenticação e muito mais sofisticado. Aqui pode, realmente, dar acesso individual às pessoas. É muito mais seguro que o xhost. A autenticação utiliza um cookie armazenado no ficheiro .Xauthority no directório de trabalho dos utilizadores. Se a máquina remota tiver um cookie válido neste ficheiro então o acesso é concedido. Para copiar o cookie da máquina onde está sentado (philosophus) para a máquina onde quer iniciar o programa (movietux) pode utilizar um dos seguintes métodos:

```
xauth extract – philosophus:0.0 | ssh movietux /usr/X11R6/bin/xauth merge
```

ou

```
scp ~/.Xauthority movietux:
```

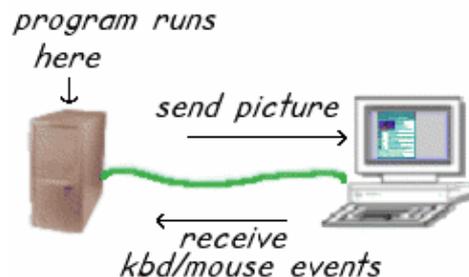
O procedimento que acontece no arranque de um programa (cliente) é o seguinte:

1. O cliente verifica a variável de ambiente DISPLAY para descobrir o servidor, caso contrário liga-se ao servidor da máquina local.
2. O servidor verifica se é permitido enviar "imagens" para o cliente. Se o cliente está autorizado, então o servidor desenhara a imagem no ecrã.

A variável de ambiente DISPLAY tem a seguinte sintaxe:

```
bash:export DISPLAY=hostname:displaynumber.screennumber
```

```
tcsh:setenv DISPLAY hostname:displaynumber.screennumber
```



Não falarei aqui, acerca do `displaynumber` e do `screennumber`. Normalmente é somente 0:0. Um exemplo para a bash seria:

```
export DISPLAY=philosophus:0.0
```

Iniciando uma aplicação remota

Após toda a teoria, um exemplo prático. Tente-o somente. Estamos novamente sentados no computador chamado `philosophus` e queremos iniciar algo remotamente a partir de `movietux`.

1. Diga ao servidor X que os clientes de `movietux` têm permissões para desenhar imagens no seu ecrã:
`xhost +movietux`
2. Faça login na máquina remota `movietux`:
`slogin -l username movietux`
3. Agora está ligado a `movietux` e todos os comandos que corre são executados no `movietux`.
4. `export DISPLAY=philosophus:0.0`
5. inicie o programa. Por exemplo:
`netscape &`

Se o `movietux` for uma máquina poderosa então notará que a renderização das páginas html no `netscape` é muito mais rápida de quando o corre localmente. Ao mesmo tempo não se apercebe, do ponto de vista de utilização, o `netscape` não é inicializado localmente visto que interage como ele do mesmo modo, com o rato e teclado.

Fascinante, não é?

Levando o display comigo

Apesar de ser fácil correr uma aplicação remotamente existe ainda alguma digitação extra, necessária para definir o `DISPLAY`. É possível automatizar isto:

- Se está a utilizar o `ssh` para se ligar à máquina remota então o `DISPLAY` é definido automaticamente. Existem, também outros programas de login remotos que suportam o `DISPLAY` mas o `ssh` é muito comum.
- Se está a utilizar o `slogin` ou outros comandos então pode instalar as seguintes scripts na máquina remota.
Para a `tsh`:

```
# take your display with you at remote login:
# Put it into your ~/.login file
set whoami=`who -ml`
set remhost=`expr "$whoami" : '.*(\(.*\))`
if ( "$remhost" != "" ) then
setenv DISPLAY "$remhost":0.0
endif
```

A script trabalha obtendo o nome da máquina remota a partir do comando `"who -ml"`. Este comando devia retornar algo do tipo

```
>who -ml  
movietux!guido pts/3 Oct 26 21:55 (philosophus.tux.org)
```

Se estiver a utilizar a bash então precisa da seguinte script:

```
# take your display with you at remote login:  
# Put it into your ~/.bash_profile  
whoami=`who -ml`  
remhost=`expr "$whoami" : '.*(\.*)"`  
if [ -n "$remhost" ]; then  
DISPLAY="$remhost":0.0  
export DISPLAY  
fi
```

OpenGL

Enquanto que as possibilidades de rede do Sistema X window são muito boas os gráficos são um pouco mais lentos devido ao facto de enviar os dados por cima de um protocolo de rede. Normalmente, não notará nenhuma diferença mas existem aplicações bastante rápidas e jogos com gráficos intensivos onde a velocidade é importante.

Estes jogos, no geral, são baseados nas bibliotecas equivalentes OpenGL (Open Graphics Library) e GLX (OpenGL Extension para o sistema X Windows). Estas bibliotecas fornecem interfaces de programação independentes que fornecem acesso directo ao hardware de aceleração 3D das placas gráficas. Ou seja, a aplicação envia a descrição de um objecto na forma de pontos, linhas e polígonos à placa gráfica e toda a renderização é feita dentro do hardware gráfico. Isto fornece gráficos bastante rápidos.

Correntemente a maioria dos drivers das placas gráficas (servidores X) não suportam a aceleração por hardware GLX/OpenGL para aplicações remotas. Suportam a aceleração por hardware para aplicações locais. O efeito é que remotamente as aplicações OpenGL iniciadas muito dificilmente arrancam e são muito lentas. Uma excepção são os drivers NVidia fechados. Eles têm uma interface de renderização directa que suporta renderização indirecta para aplicações remotas.

Conclusão

Utilizando o poder de computação a sua rede com o X11 é muito fácil. Pode trabalhar com aplicações remotas do mesmo modo que localmente. A única diferença que encontrará é que verá os ficheiros e directórios de trabalho da máquina remota. Contudo com o NFS e o NIS instalado pode até esconder esta pequena diferença e utilizar todo o poder de processamento das máquinas mais rápidas nas suas redes sem pensar nisso.

Ligações

[X11 system. x.org](http://X11.system.x.org)

xfree86.org o sistema X11 utilizado no Linux

Webpages maintained by the LinuxFocus Editor team

© Guido Socher

"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

Translation information:

en --> -- : Guido Socher ([homepage](#))

en --> pt: Bruno Sousa <bruno/at/linuxfocus.org>

2005-01-10, generated by lfparsr_pdf version 2.51