

GDM ile X Terminali NASIL

Yazan:
Nilgün Belma Bugüner

<nilgun (at) belgeler.gen.tr>

24 Nisan 2006

Özet

Bu belgede GDM yardımıyla eski, bir işe yaramaz denilen makinaların hızlı bir sunucudaki XFree86'yı kullandığında nasıl işe yarayacağı yap-işlet modeliyle anlatılmıştır. Daha iyisini yapmak, iynin de iyisi mümkün.

Konu Başlıkları

1. X Terminali Nedir?	3
2. Neler Gerekli	3
3. Sunucu Yapılandırması	3
4. İstemci Yapılandırması	7
5. Sonuç	9

Geçmiş

1.2	24 Nisan 2006	NBB
Lisans değişikliği; Lisans artık GFDL değil, GPL; küçük belgeye boyundan büyük lisans metni eklemenin alemi yok.		
1.1	12 Haziran 2004	NBB
Biraz daha ayrıntı.		
1.0	10 Haziran 2004	NBB
İlk sürüm		

Yasal Açıklamalar

Bu belgenin, *GDM ile X Terminali NASIL* 1.0 sürümünün **tefif hakkı © 2004 Nilgün Belma Bugüner**'e aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan [GNU Genel Kamu Lisansı](#)^(B1)'nin 2. ya da daha sonraki sürümünün koşullarına bağılı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın özgün kopyasını <http://www.gnu.org/copyleft/gpl.html> adresinde bulabilirsiniz.

BU BELGE "ÜCRETSİZ" OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGEYİ "OLDUĞU GİBİ", AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE BELGEYİ DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİLERİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHİ, SORUMLU DEĞİLDİR.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

1. X Terminali Nedir?

Artık üzerinde başka bir iş yapılamayacak kadar zayıf kalmış olan eski PC'ler, eğer güçlü bir sunucu makinası var ise, bir X terminali olarak kullanılabilir. Bu düzenekte, istemci makinanın ekranında, ana sunucu makinasının oturum açma ekranı görünür, kullanıcı ana sunucu makinada sisteme giriş yapar. İstemci makinada hiçbir kullanıcı işlemi yapılmaz, hatta syslog'u bile çalıştırmak gerekmez.

X terminalinden önce terminal kavramı üzerinde biraz duralım. Telnet kullanmayı bilirsiniz en azından. Telnet yapabilmek için önce bir terminal gerekir, buna uçbirim de deniyor; bir de telnet sunucusu olmalı. Her iki uçta da bu işlemi yapacak aletlerin aynı olması gerekir. Bu aletin ismi linuxcada tty. Siz bir tty çalıştırırsınız ve bu tty ile karşı makinadan bir tty istersiniz. X terminali için de böyle, Bir X çalıştırıp, karşı makinadan bir X istiyorsunuz. Burada kafa karıştıran, X'den bahsedilirken her iki makina için de isminin X sunucusu olması. Aslına bakarsanız, ikiside bir diğerine göre sunucu :-). Bu onun doğasında var. Dolayısıyla oturum isteğinde bulunulan sunucu, istemde bulunan istemci oluyor, ama tamamen göreceli. Sizin ne yapmak istediğinize bağlı.

Bu işlem GDM ile oldukça kolaylaştırılmış. Bu belgeyi, eski makinamı bir X terminali olarak nasıl çalıştırım diye bir arayış içine girdiğimde uyguladığım çözümü sizlerle paylaşmak için yazdım. Adım adım anlatacağım, benim için uğraştırıcı oldu ama artık çok kolay :-) (Colomb'un yumurtası)

2. Neler Gerekli

Nelerin gerektiği için benim örneğime bakalım, 2 makina var. Sunucu makinasının özellikleri şöyle: PIII/1GHz işlemci, 512MB bellek, 40GB sabit disk, 10/100 ağ kartı v.s. artı RH 9 kurulu. İstemci makina ise: PI/133MHz işlemci, 64MB bellek, 2GB sabit disk, 10/100 ağ kartı. Vesairesi yok, vesaireyi daha önce yeni makinaya aktarmıştım. İki makina bir ağ kablosu ile birbirine karttan karta bağlı.

Elinizin altında bir Linux dağıtımının CD'leri olmalı, eski makinaya bir Linux kurmak lazım, çünkü. Bu işlem için yeni makinadaki CDROM'u eski makinaya taktım. Ben Fedora Core 1 dağıtımının CD'lerinden yararlandım. Kurulum sırasında özel kurulumu seçip, sadece GNOME kurdum. Gereksiz hiçbir şeyi kurmadım. Çünkü gereken herşey ana makinadan kullanılacak. 2GB sabit disk bile fazla geldi. Siz daha itinalı bir seçimle belki 1GB bir sabit disk bile kullanabilirsiniz. Belgeyi önce baştan sona okursanız, neler gerektiği hakkında daha ayrıntılı bir fikir edineceksiniz.

Bütün bunlar sağlandıktan sonra artık sistemi yapılandırma işlemlerine geçebiliriz. Yapılandırma işlemini 2 ana bölüme ayıracağız. Önce sunucuyu yapılandıracağız, sonra da istemciyi.

3. Sunucu Yapılandırması

Sunucumda RH 9 kurulu olmasına rağmen saf bir RH 9 değil. Üzerinde pek çok paketi güncellemişim. Bu nedenle kullandığım paket sürümlerini ayrıca belirteceğim. Bir sorunla karşılaşırsanız en azından bu sürümler çözüm olabilir. Güvenlik ile ilgili basit yapılandırmaları da yapacağız ama, eğer örütbağa bu makina ile bağlanıyorsanız güvenlik duvarınızı yapılandırmayı unutmayın. Çünkü makinanız bir sürü port dinleyecek, bunların örütbağdan değil sadece iç ağınızdan erişilebilir olmasını sağlamalısınız. Güvenlik duvarı yapılandırması bu belge kapsamında değil. Sunucuyu 3 adımda yapılandıracağız:

DHCPD yapılandırması

dhcpcd için bir DNS kurmadım. Siz bu yöntemi büyük bir ağa uygulamak için DNS kullanmak isterseniz sizin bileceğiniz birşey. Benim 2 makinalık bir ağ için DNS'ye ihtiyacım yoktu.

Ağ yapılandırması:

```
~> cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
```

```

IPADDR=10.10.10.1
NETMASK=255.255.255.0
NETWORK=10.10.10.0
BROADCAST=10.10.10.255
ONBOOT=yes
BOOTPROTO=static
~> cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=nilgun.buguner.home
~> cat /etc/sysconfig/static-routes
eth0 net 10.10.10.0 netmask 255.255.255.0 gw 10.10.10.254
~> cat /etc/sysconfig/hosts
127.0.0.1      localhost localdomain
10.10.10.1     nilgun.buguner.home
10.10.10.100   belma.buguner.home
~> cat /etc/hosts.allow
# Bu dosyada yerel ag hizmetlerinden hangi makinalarin
# faydalanabilecegi belirtilir.
#
ALL: .buguner.home
ALL: 127

```

Makina isminin `.home` içermesi sizi şaşırtmasın, yerel ağlar için isim sınırlaması yoktur. :-). Bakalım makinada **dhcpcd** namına ne kurulmuş:

```

~> rpm -qa | grep dhcp
dhcp-3.0pl2-6.16
dhcp-devel-3.0pl2-6.16

```

dhcpcd yapılandırma dosyası `/etc/dhcpcd.conf(1)`:

```

~> cat /etc/dhcpcd.conf
allow booting;
allow bootp;
ddns-update-style none;

# Standart yapılandırma yonergeleri...
option domain-name "buguner.home";

max-lease-time 120;
default-lease-time 120;

subnet 10.10.10.0 netmask 255.255.255.0 {
}
group {
    server-name "nilgun.buguner.home";
    get-lease-hostnames true;
    use-host-decl-names on;
# istemci makina bilgileri; makinanın ismi: belma
    host belma {
# bu adres, istemci makinadaki ag kartinin ethernet adresi
        hardware ethernet 00:80:ad:77:72:7c;
# sabit bir adres veriyoruz. Kimden bahsettigimizi bilelim.
        fixed-address 10.10.10.100;
    }
}

```

Şimdi **dhcpcd**'yi başlatalım ve bakalım çalışıyor mu?

```

~> /sbin/service dhcpd start
dhcpd başlatılıyor: [ TAMAM ]
~> netstat -Nlp | grep dhcp
udp  0      0 0.0.0.0:bootps      0.0.0.0:*          1181/dhcpd
raw  0      0 0.0.0.0:icmp        0.0.0.0:*          7          1181/dhcpd

```

Çalışıyormuş : -) . Çalışma seviyenizde başlatılan hizmetlere **dhcpd**'yi eklemeyi unutmayın.

XFS yapılandırması

xfs, X Pencere Sisteminin yazıtipi sunucusudur (XFS – X Font Server). İstemci makineye X yazıtiplerini kurmayacağız, tabii **xfs**'i de kurmayacağız. İstemci makinenin X Pencere Sistemi sunucu makinedeki **xfs** sunucusunu kullanacak. Bunun için bu makinedeki **xfs** sunucusunu dinlemeye almamız lazım. Hemen yapalım:

/etc/X11/fs/config dosyasının en alt satırına gidin,

```

# don't listen to TCP ports by default for security reasons
no-listen = tcp

```

son satırı iptal edeceğiz ki, **xfs** ağdan gelen istekleri dinlesin:

```

# don't listen to TCP ports by default for security reasons
# no-listen = tcp

```

Eğer **xfs**'i kullanmıyorsanız başlatın, ben kullanıyordum, yeniden başlatıyorum ve dinlemeye geçmiş mi bakıyorum:

```

~> /sbin/service xfs restart
xfs yeniden başlatılıyor:
xfs durduruluyor: [ TAMAM ]
xfs başlatılıyor: [ TAMAM ]
~> netstat -Nlp | grep xfs
tcp  0      0 0.0.0.0:xfs         0.0.0.0:*          LISTEN  1249/xfs
unix  2      [ ACC ] STREAM LISTENING  9433    1249/xfs ↵
/tmp/.font-unix/fs7100

```

Bu da tamamdır. Çalışma seviyenizde başlatılan hizmetlere **xfs**'i eklemeyi unutmayın. Bu arada unutmadan belirtiyim, makinede x-4.3.0 kurulu.

GDM yapılandırması

Geldik için en canalcı kısmına... **gdm**, **xdm**'yi kullanmaz. Bu bakımdan /etc/X11/xdm dizinindeki dosyalarda daha önce **xdm** için bazı değişiklikler yaptıysanız onları öntanımlı durumlarına getirin. Oradaki dosyalara dokunmayacağız.

Kullandığım **gdm** sürümü 2.6.0.2 (güncel sürüm, kaynak paketinden kurdum). **gdm** yapılandırmasını uzun uzadıya anlatmayacağım, merak ediyorsanız, belgelerini okuyun. Sadece /etc/X11/gdm/gdm.conf dosyasının /etc/X11/gdm/factory-gdm.conf dosyasına göre fark dosyasını veriyorum. Bu fark dosyasına bakarak /etc/X11/gdm/gdm.conf dosyanızı düzeltin:

```

/etc/X11/gdm> diff -u factory-gdm.conf gdm.conf > gdm.conf.diff
/etc/X11/gdm> cat gdm.conf.diff
--- factory-gdm.conf      2004-06-07 07:35:12.000000000 +0300
+++ gdm.conf              2004-06-08 17:41:14.000000000 +0300
@@ -46,10 +46,12 @@
# The greeter for local (non-xdmpc) logins. Change gdmlogin to gdmgreeter
# to get the new graphical greeter.
#Greeter=/usr/bin/gdmlogin

```

```
+Greeter=/usr/bin/gdmgreeter

# The greeter for xdmcp logins, usually you want a less graphically
# intensive greeter here so it's better to leave this with gdmlogin
#RemoteGreeter=/usr/bin/gdmlogin
+RemoteGreeter=/usr/bin/gdmgreeter

# Launch the greeter with an additional list of colon seperated gtk
# modules. This is useful for enabling additional feature support
@@ -111,7 +113,7 @@
# for backwards compatibility reasons with 2.4.4.x
#SessionDesktopDir=/etc/X11/sessions/:/etc/X11/dm/Sessions/:/usr/share/gdm/
#BuiltInSessions/:/usr/share/xsessions/
# This is the default .desktop session. One of the ones in SessionDesktopDir
-#DefaultSession=gnome.desktop
+DefaultSession=kde.desktop
# Better leave this blank and HOME will be used. You can use syntax ~/
# below to indicate home directory of the user. You can also set this to
# something like /tmp if you don't want the authorizations to be in home
# directories.
@@ -135,8 +137,8 @@
#Automatic VT allocation. Right now only works on Linux. This way
# we force X to use specific vts. turn VTAllocation to false if this
# is causing problems.
-#FirstVT=7
-#VTAllocation=true
+FirstVT=7
+VTAllocation=true
# Should double login be treated with a warning (and possibility to change
# vts on linux systems for console logins)
#DoubleLoginWarning=true
@@ -179,7 +181,7 @@
# Note: Anytime we find a -query or -indirect on the command line we do
# not add a "-nolisten tcp", as then the query just wouldn't work, so
# this setting only affects truly local sessions.
-#DisallowTCP=true
+DisallowTCP=false
# By default never place cookies if we "detect" NFS. We detect NFS
# by detecting "root-squashing". It seems bad practice to place
# cookies on things that go over the network by default and thus we
@@ -199,7 +201,7 @@
# allow local access is another alternative but not the safest.
# Firewalling port 177 is the safest if you wish to have xdmcp on.
# Read the manual for more notes on the security of XDMCP.
-Enable=false
+Enable=true
# Honour indirect queries, we run a chooser for these, and then redirect
# the user to the chosen host. Otherwise we just log the user in locally.
#HonorIndirect=true
@@ -313,6 +315,7 @@
#BackgroundImage=
#BackgroundScaleToFit=true
#BackgroundColor=#76848F
+BackgroundColor=#a3bbce
# XDMCP session should only get a color, this is the sanest setting since
# you don't want to take up too much bandwidth
#BackgroundRemoteOnlyColor=true
```

```

@@ -333,6 +336,7 @@
#ShowLastSession=true
# Always use 24 hour clock no matter what the locale.
#Use24Clock=false
+Use24Clock=true
# Use circles in the password field. Looks kind of cool actually,
# but only works with certain fonts.
#UseCirclesInEntry=false
@@ -406,7 +410,7 @@
#
# If you wish to run the XDMCP chooser on the local display use the
# following line
-#0=Chooser
+0=Chooser

## Note:
# is your X server not listening to TCP requests? Perhaps you should look

```

Son satırdaki `0=Chooser` gerekli değil, istemci makinanın **gdm** yapılandırması olmuş mu diye bakmak isterseniz faydalı, o da kendini sunucu zannediyor çünkü. Yapılandırmalar bittikten sonra isterseniz denemek için ona bağlanabilirsiniz :-)

Şimdi sıra `/etc/inittab` dosyasında. Bu dosyayı da düzenledikten sonra sistemi yeniden başlatacaksınız. Dosyada iki satır önemli, bu iki satırı bulup onları aşağıdaki gibi değiştirin:

```

id:5:initdefault:

...

x:5:respawn:/usr/bin/gdm

```

gdm bu makinada artalanda bir sunucu olarak çalışacağından alışılmış olan `-nodaemon` seçeneğini kullanmıyoruz.

Sistemi yeniden başlattığınızda doğrudan X sunucusundan oturum açıyor olacaksınız. Şimdiye dek X sunucusu ağdan gelecek istekleri beklemiyorsa artık bekliyor olacak. **gdm**'de 177/udp'yi dinliyor olacak. **nmap** ile bakarsanız aşağıdakine benzer bir çıktı alacaksınız:

```

~> nmap -sTU 127.0.0.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):
(The 3044 ports scanned but not shown below are in state: closed)
Port      State      Service
67/udp    open       dhcpserver
177/udp    open       xdmcp
6000/tcp   open       X11
7100/tcp   open       font-service

Nmap run completed -- 1 IP address (1 host up) scanned in 9 seconds

```

Burada tekrarlıyorum: **Güvenlik duvarınızı yapılandırmadan bu makina ile örütbağa çıkmayın! Açık portlarınız var.**

4. İstemci Yapılandırması

Başta da belirttiğim gibi istemci makina gerekliliğinin dışında birşey kurmadım, çünkü sistemin 2 GB'lık sabit diske sığması gerekiyordu. Özel kurulumu seçip sadece X ve GNOME kurulumu yapın, ağ yapılandırmasında IP'nin DHCP üzerinden alınmasını sağlayın ve makina ismini verin (Benimki `belma.buguner.home`). Bu makina için güvenlik duvarı olmayacak. Kurulum sırasında dikkat edin. Yoksa sunucu makina bağlanamazsınız.

Eğer sistem X sunucusu ile açılmaya ayarlanmışsa ve yazıtlarını de yüklemeyerseniz açılmayacaktır. X açılmayacak, hata verecek ve konsola düşeceksiniz.

Burada ağ yapılandırmasını, sunucu için yaptığım gibi anlatmayacağım. Kurulum bitip sistem açıldığında IP adresi olarak sunucudan `10.10.10.100` adresini alacak. **ifconfig** komutu ile ağ yapılandırmanızı görebilirsiniz. Gerekli değil ama, **nmap 10.10.10.1** ile de sunucu makina bir port taraması yapıp, iki makinanın birbirini görüp görmediğine bakabilirsiniz.

Bir DNS sunucumuz olmadığından sunucu makinanın ismini istemci makina öğretmemiz lazım. Bunun için, `/etc/hosts` dosyasını düzenlememiz gerekli.

```
~> cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1        localhost.localdomain localhost
10.10.10.1       nilgun.buguner.home
10.10.10.100     belma.buguner.home
```

XF86Config dosyası kurulum sırasında yapılandırılmış olmalı. Dosya üzerinde tek bir satırda değişiklik yapacağız ve X sunucusu **xfs** sunucusunu bulabilecek. `/etc/X11/XF86Config` dosyasını bir metin düzenleyici ile açın ve **FontPath** satırını bulup, satırı aşağıdaki gibi düzenleyin:

```
FontPath      "tcp/10.10.10.1:7100"
```

Burada `10.10.10.1` sunucu makinanın adresi. Sizinki farklıysa onu yazın. Artık X sunucunuz açılacaktır. Diğer işlemlere geçmeden önce deneyin. **startx** ile X'i başlatabilirsiniz. GNOME'un açılma hızını belleğinize kaydedin, yapılandırmanızı bitip, sunucu makina bağlandığınızda KDE açılacak ve buna rağmen hız farkını farkedeceksiniz :-) (Dikkat, istemci makina KDE kurmadım; KDE, sunucu makina kurulu. Böylece ne yaptığımızdan %100 emin oluyoruz.)

X oturumunun çalıştığını gördükten sonra artık **gdm**'yi yapılandırabiliriz. Yöntem aynı; bir fark dosyası veriyorum, siz bu dosyaya bakarak kendi dosyanızı düzenliyorsunuz.

```
/etc/X11/gdm> diff -u factory-gdm.conf gdm.conf > gdm.conf.diff
/etc/X11/gdm> cat gdm.conf.diff
--- factory-gdm.conf      2004-06-07 07:07:44.000000000 +0300
+++ gdm.conf              2004-06-10 01:08:33.000000000 +0300
@@ -46,10 +46,12 @@
# The greeter for local (non-xdmp) logins.  Change gdmlogin to gdmgreeter to
# get the new graphical greeter.
#Greeter=/usr/bin/gdmlogin
+Greeter=/usr/bin/gdmgreeter

# The greeter for xdmp logins, usually you want a less graphically intensive
# greeter here so it's better to leave this with gdmlogin
#RemoteGreeter=/usr/bin/gdmlogin
+RemoteGreeter=/usr/bin/gdmgreeter

# Launch the greeter with an additional list of colon seperated gtk
# modules. This is useful for enabling additional feature support
@@ -199,7 +201,7 @@
```



```
# allow local access is another alternative but not the safest.
# Firewalling port 177 is the safest if you wish to have xdmcp on.
# Read the manual for more notes on the security of XDMCP.
-Enable=false
+Enable=true
# Honour indirect queries, we run a chooser for these, and then redirect
# the user to the chosen host. Otherwise we just log the user in locally.
#HonorIndirect=true
@@ -333,6 +335,7 @@
#ShowLastSession=true
# Always use 24 hour clock no matter what the locale.
#Use24Clock=false
+Use24Clock=true
# Use circles in the password field. Looks kind of cool actually,
# but only works with certain fonts.
#UseCirclesInEntry=false
@@ -389,7 +392,7 @@
# and they will always be started. Each line must start with a unique
# number and that will be the display number of that server. Usually just
# the 0 server is used.
-0=Standard
+#0=Standard
#1=Standard
# Note the VTAllocation and FirstVT keys on linux. Don't add any vt<number>
# arguments if VTAllocation is on, and set FirstVT to be the first vt
@@ -400,9 +403,9 @@
# send patches. X servers will just not get any extra arguments then.
#
# If you want to run an X terminal you could add an X server such as this
-#0=Terminal -query serverhostname
+#0=Terminal -query nilgun.buguner.home
# or for a chooser (optionally serverhostname could be localhost)
-#0=Terminal -indirect serverhostname
+#0=Terminal -indirect nilgun.buguner.home
#
# If you wish to run the XDMCP chooser on the local display use the following
# line
```

Şimdi sıra `/etc/inittab` dosyasında. Bu dosyayı da düzenledikten sonra sistemi yeniden başlatacaksınız. Dosyada iki satır önemli, bu iki satırı bulup onları aşağıdaki gibi düzenleyin:

```
id:5:initdefault:

...

x:5:respawn:/usr/bin/gdm -nodaemon
```

veya:

```
id:5:initdefault:

...

x:5:respawn:/usr/X11R6/bin/X -query nilgun.buguner.home
```

Artık hazırsınız. Sistemi yeniden başlatın. Doğrudan, hiçbir ek işlem yapmadan sunucu makineye bağlanacaksınız. Sunucu makinada oturum açarken sunucu makinada geçerli bir kullanıcı adı ve parolası belirtmeyi unutmayın. İstemci makinada geçerli kullanıcı ismini ve parolasını orada kullanamazsınız.

5. Sonuç

Bu belge hemencecik birşeyler yapıp bir X terminali çalıştırmak için yeterlidir. (Belge böyle bir çalışmanın ürünüdür.) İhtiyaçlarınıza uygun bir yapılandırma için farklı çözümler geliştirilebilir. Bu ihtiyaçların bir kısmına çözüm olarak geliştirilmiş projeler de var. Geliştirilecek başka projeler de olabilir, böyle bir projeyi neden siz başlatmaysınız... Çeşitli ihtiyaçlar ve bir kaç örnek:

- Diskli bir makinayı X terminali haline getirmek. Bunu anlattım.
- Disksiz bir makinayı, yerel işlem de yapabilir şekilde kullanabilmek. Bunun bir projesi var: [LTSP^{\(B3\)}](#).
- Disksiz bir makinayı yerel işlem yapamaz şekilde kullanmak. Bunun bir projesi var: [PXES^{\(B4\)}](#).
- Disk yerine yerel depo olarak başka şeyler kullanmak, flash vs. gibi. Hazır terminaller örneğin.
- Mevcut bir ağı, örneğin bir windows ağının sadece sunucusunu Linux (ve samba) ile değiştirip, ağın tamamını, "gerektiği anda" sunucuda bir uygulama çalıştırıp X terminalleri ağına dönüştürmek. Böyle birşey okullara lazım :-)

Hatırlatma: Dosyalarınızı yapılandırırken makina isimleri olarak `nilgun.buguner.home` ve `belma.buguner.home` isimleri yerine kendinize özgü isimler kullanmayı unutmayın.

Hayırlı olsun.

Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

^(B1) [../howto/gpl.pdf](#)

- ⁽¹⁾ Bir ağda, sadece X terminallerinin, mac adreslerine göre sabit adres değil, rasgele adres almasını sağlayabilirsiniz (mac adresi toplamak zor iştir). Bunun için DHCP sunucusu olarak çalışan makinanın `/etc/dhcp.conf` dosyasına aşağıdaki gibi bir kayıt girebilir,

```
class "X-Terminal" {
    match if option dhcp-client-identifier = "x-terminal";
}
```

hatta bu sınıfın üyelerine belli bir adres bloğunu ayırabilir,

```
option vendor-class-identifier

pool {
    allow members of "X-Terminal";
    range 10.10.10.128 10.10.10.253;
}
```

ve istemci makinanın `/etc/dhclient.conf` dosyasına ise aşağıdakini,

```
interface "eth0" {
    send dhcp-client-identifier "x-terminal";
}
```

yazabilirsiniz. "x-terminal" yerine her iki dosyada aynı olmak üzere sizce uygun başka bir şey de yazabilirsiniz. Bu çözümü gönderen Deniz Akkuş'a teşekkürler.

^(B3) <http://www.ltsp.org>

^(B4) <http://pxes.sourceforge.net>

Bu dosya (gdm-terminal.pdf), belgenin XML biçiminin T_EXLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

27 Ocak 2007