

Терминальный сервер на основе Debian Lenny

Содержание

- 1 Первоначальная настройка
- 2 Устанавливаем и настраиваем граф подсистему Xorg и KDE:
- 3 Устанавливаем и настраиваем ПО:
- 4 Устанавливаем последнюю 3-ю версию OpenOffice.org.
- 5 Устанавливаем Wine:
- 6 Устанавливаем 2gis:
- 7 Настройка серверной части терминального сервера:

// [\[править \]](#) Первоначальная настройка

Итак у нас свежеставленная система Debian Lenny с netinst-a. Первый раз заходим под пользователем root. Проверяем настройки сети:

```
nano /etc/network/interfaces
```

Должно быть типа такого:

```
# The loopback network interface auto lo iface lo inet loopback # The primary network interface allow-hotplug eth0 iface eth0 inet static address 192.168.0.83 netmask 255.255.255.0 gateway 192.168.0.37 dns-nameservers 83.246.135.100
```

Проверяем настройки ДНС:

```
nano /etc/resolv.conf
```

Должно быть типа такого:

```
nameserver 83.246.135.100
```

Если вносили изменения то перезапускаем:

```
/etc/init.d/networking restart
```

Если у вас провайдер интелби или интелека то вы можете обновляться с ftp.nix-files.org.ru:
Настраиваем репозитории

```
nano /etc/apt/sources.list
```

удаляем то что там есть, прописываем:

```
deb http://apt.nix-files.org.ru/debian lenny main contrib non-free deb http://apt.nix-files.org.ru/debian-security  
lenny/updates main contrib non-free
```

Если у вас другой провайдер то оставляем как есть. Кирилизуем консоль.

```
apt-get install console-cyrillic cyr
```

Настраиваем sudo

```
apt-get install sudo nano /etc/sudoers
```

здесь в секции User privilege specification добавляем пользователя admin

```
# User privilege specification root ALL=(ALL) ALL admin ALL=(ALL) ALL
```

Выходим из консоли root:

```
exit
```

Логинимся под admin.

[[править](#)] Устанавливаем и настраиваем граф подсистему Xorg и KDE:

Устанавливаем xorg:

```
sudo apt-get install xorg
```

Конфигурируем xorg:

```
sudo X -configure sudo cp xorg.conf.new /etc/X11/xorg.conf
```

Смотрим конфиг файл:

```
sudo nano /etc/X11/xorg.conf
```

В секции Section "Monitor" должны быть прописаны режимы монитора, если их нет то прописываем:

```
HorizSync 30.0-81.0 VertRefresh 56.0-85.0
```

Также в секции Section "Screen" прописываем битность цвета

```
DefaultDepth 16
```

и разрешение по умолчанию:

```
SubSection "Display" Viewport 0 0 Depth 16 Modes "1280x1024@75"
```

Устанавливаем KDE:

```
sudo apt-get install kde-core kdm
```

Устанавливаем русификацию в kde:

```
sudo apt-get install kde-i18n-ru
```

Перезагружаемся

```
sudo reboot
```

При загрузке KDE выбираем русский или после загрузки заходим в меню "K - Settings - Regional & Accessibility - Country/Region&Language", Выбираем Add Language - Russian. Настраиваем переключение раскладки:

В К/Настройка/Региональные и специальные возможности/Раскладка клавиатуры надо оставить на закладке "Раскладка" только ОДНУ используемую раскладку: русскую, но ОБЯЗАТЕЛЬНО выставить check box "Включить латинскую раскладку". Переключение на закладке "Параметры ХКВ" настроить по своему усмотрению, мне удобней всего Ctrl-Shift. После этого KDE по Ctrl-Shift переключает раскладки с русской на английскую. Но в тее флаг при этом не меняется, для исправления делаем: Устанавливаем переключатель клавиатуры:

```
sudo apt-get install kkbswitch
```

Запускаем его из меню К - Службные - KkbSwitch

По умолчанию в KDE в качестве демона, отслеживающий изменения в файловой системе и оповещающий об этом подписавшиеся приложения используется famd. Например можно открыть в Konqueror каталог, параллельно с помощью shell создать или удалить файл в этом каталоге. famd оповестит об изменениях Konqueror, а тот в свою очередь перерисует окно со списком файлов. И всебы ничего но он катастрофически много отжирает процессорного времени особенно если копируешь кучу мелких файлов гденить из сети, причем что удивительно в большинстве дистрибутивов famd уже давно заменили gamin'ом. В списке рассылки рекомендовано заменить famd на gamin. Оба предоставляют библиотеку с одинаковым интерфейсом, но famd является демоном периодически опрашивающим каталоги, а gamin является библиотекой взаимодействующей с подсистемой ядра Linux, называющейся inotify. Эта подсистема позволяет обходиться без отдельного демона и доставляет необходимую информацию всем подписавшимся программам напрямую, без лишних посредников из первых рук.

```
sudo apt-get install gamin
```

Настраиваем звук:

```
sudo apt-get install alsa /usr/sbin/alsaconf
```

[[править](#)] Устанавливаем и настраиваем ПО:

Устанавливаем паблик шрифты от MS:

```
sudo apt-get install msttcorefonts
```

Устанавливаем архиваторы:

```
sudo apt-get install rar sudo apt-get install unrar sudo apt-get install bzip2 sudo apt-get install lha sudo apt-get install zip sudo apt-get install unzip sudo apt-get install arj sudo apt-get install unace sudo apt-get install rpm sudo apt-get install 7z
```

Устанавливаем драйвер для работы с ntfs:

```
sudo apt-get install ntfs-3g
```

Далее ставим файловый менеджер Krusarder

```
sudo apt-get install krusader
```

[[править](#)] Устанавливаем последнюю 3-ю версию OpenOffice.org.

Сначала скачиваем его по ссылке в домашний каталог:

```
sudo wget http://ftp5.gwdg.de/pub/openoffice/extended/localized/ru/3.0.1/OOo\_3.0.1\_LinuxIntel\_install\_ru\_deb.tar.gz
```

Распаковываем архив:

```
tar xzfv OOo_3.0.1_LinuxIntel_install_ru_deb.tar.gz
```

Устанавливаем все пакеты:

```
sudo dpkg -i OOO300_m15_native_packed-1_ru.9379/DEBS/*.deb
```

Устанавливаем пакет интеграции OpenOffice.org с системой, если этого не сделать то в основном меню приложений OpenOffice.org не будет:

```
sudo dpkg -i  
OOO300_m15_native_packed-1_ru.9379/DEBS/desktop-integration/openoffice.org3.0-debian-menus_3.0-9376_all.deb
```

При работе был замечен интересный баг, когда пытаешься открыть локально файлы .doc (пример) все нормально открывается, если же через самбу или с вебстраницы то пиктограмма OpenOffice.org долго прыгает и ничего не открывает, исправление состоит в том чтобы установить правильные права на каталоги OpenOffice.org:

```
chown -R root:root /opt/openoffice.org3 chown -R root:root /usr/bin/openoffice.org3
```

Перезапускаем граф подсистему клавиатурным сочетанием Ctrl + Alt + BackSpace
Устанавливаем программу для снятия скриншотов:

```
sudo apt-get install ksnapshot
```

Устанавливаем и русифицируем icedove:

```
apt-get install icedove apt-get install icedove-locale-ru
```

Устанавливаем rdesktop, вдруг надобудет подключатся к windows терминальному серверу:

```
sudo apt-get install rdesktop [ правиль ] Устанавливаем Wine:
```

Для работы 1С, а также Консультанта необходим WINE@Etersoft версии Network. Вообще у них на сайте есть хороший мануал по настройке <http://www.etersoft.ru/content/view/56/156/> [1], рассмотрим основные действия. После покупки скачиваем по ссылкам пакеты WINE@Etersoft версии Network. Всего пакетов должно быть:

Автор: Administrator

09.05.10 16:59 - Последнее обновление 10.05.10 20:59

```
dkms-aksparlX_2.0-eter4debian_all.deb      haspd-modules_2.0-eter12.2debian_i386.deb
fonts-ttf-liberation_1.04-eter1debian_all.deb  libwine_1.0.9-eter42debian_i386.deb
fonts-ttf-ms_1.0-eter4debian_all.deb        wine_1.0.9-eter42debian_i386.deb
haspd_2.0-eter12.2debian_i386.deb
wine-etersoft-network_1.0.9-eter18debian_i386.deb
```

Устанавливаем WINE@Etersoft версии Network:

```
dpkg -i *
```

Чтобы не устанавливать необходимое ПО под каждым пользователем воспользуемся административной установкой: Для этого добавляем пользователя admin в группу wineadmin (только этот пользователь сможет устанавливать и удалять ПО из wine):

Создаем каталог, который будет играть роль общего win-окружения. Рекомендуется в качестве общего каталога использовать /var/lib/wine/default:

```
wine --admin
```

Создается каталог /var/lib/wine/default где и будет расположено win окружение для всех пользователей. Настраиваем автоподключение к общему win-окружению при первом запуске пользователя, для этого отредактируем /etc/sysconfig/wine:

```
sudo nano /etc/sysconfig/wine
```

Конфиг файл должен быть таким:

```
ATTACH_MODE="yes" # включает режим общего win-окружения WINEADMIN="default" #
путь к общему win-окружению
```

[[править](#)] Устанавливаем 2gis:

Есть два способа: первый это скачать с офиц сайта и пытаться поставить под wine и второй воспользоваться репозиторием. Я сделал по второму. Добавляем в список репозиториев репозиторий 2gis:

```
sudo nano /etc/apt/sources.list
```

Добавляем строчку:

```
deb ftp://10.246.58.76/pub/repository/debian/2gis/ lenny non-free
```

Устанавливаем 2gis, мне больше нравится 2gis версии 2(там рекламы меньше), поэтому буду ставить его:

```
sudo apt-get install 2gis-old-barnaul
```

Вообще список поддерживаемых пакетов из этого репозитория:

ДубльГИС 2gis-old-common — общие файлы, необходимые для работы ДубльГИС
2gis-old-astrakhan — карта Астрахани 2gis-old-barnaul — карта Барнаула
2gis-old-chelyabinsk — карта Челябинска 2gis-old-ekaterinburg — карта Екатеринбурга
2gis-old-irkutsk — карта Иркутска 2gis-old-kemerovo — карта Кемерово
2gis-old-krasnoyarsk — карта Красноярска 2gis-old-kurgan — карта Кургана
2gis-old-nizhnevartovsk — карта Нижневартовска 2gis-old-novokuznetsk — карта
Новокузнецка 2gis-old-nsk — карта Новосибирска 2gis-old-odessa — карта Одессы
2gis-old-omsk — карта Омска 2gis-old-tomsk — карта Томска 2gis-old-tyumen — карта
Тюмени 2gis-old-all — карты всех доступных городов ДубльГИС 3.0 2gis-common
— общие файлы, необходимые для работы ДубльГИС 3.0 2gis-astrakhan — карта
Астрахани 2gis-barnaul — карта Барнаула 2gis-chelyabinsk — карта Челябинска
2gis-ekaterinburg — карта Екатеринбурга 2gis-irkutsk — карта Иркутска 2gis-kemerovo
— карта Кемерово 2gis-krasnoyarsk — карта Красноярска 2gis-kurgan — карта
Кургана 2gis-nizhnevartovsk — карта Нижневартовска 2gis-novokuznetsk — карта
Новокузнецка 2gis-novosibirsk — карта Новосибирска 2gis-odessa — карта Одессы
2gis-omsk — карта Омска 2gis-tomsk — карта Томска 2gis-tyumen — карта Тюмени
2gis-all — карты всех доступных городов

По умолчанию он установится в /usr/lib/2gis-old-barnaul. Теперь необходимо этой папке в wine назначить какойнибудь диск.

```
winecfg
```

В моем случае это диск F. Запускаем и проверяем как работает:

2gis-old-barnaul [\[править \]](#) **Настройка серверной части терминального сервера:**

Основная задумка в том чтобы установить и настроить терминальный сервер с использованием бездисковых терминальных клиентов, т.е клиентский компьютер по dhcp получает свой ип адрес а также путь откуда брать ядро для загрузки а после загрузки ядра автоматически подключается к серверу терминалов через протокол px. Используемые технологии:

DHCP TFTP Thinstation SSH RX@Etersoft (тотже FreeNX только чуть подправленный)

Устанавливаем DHCP:

```
sudo aptitude install dhcp3-server
```

В конфигурационном файле dhcp3-server указываем интерфейс на котором dhcp будет работать:

```
sudo nano /etc/default/dhcp3-server
```

В моем случае получилось так:

```
# Defaults for dhcp initscript # sourced by /etc/init.d/dhcp # installed at
/etc/default/dhcp3-server by the maintainer scripts # # This is a POSIX shell fragment #
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests? # Separate
multiple interfaces with spaces, e.g. "eth0 eth1". INTERFACES="eth0"
```

Настраиваем dhcp сервер: Создаем копию исходного конфига:

```
sudo cp /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf.back
```

Редактируем конфиг:

```
sudo nano /etc/dhcp3/dhcpd.conf
```

В конфигурационном файле можно указать пул адресов который будет выдаваться клиентам, жесткую привязку ип адреса по мак адресу, днс сервера и шлюзы клиентов. У меня нету своего днс сервера, также динамическая выдача адресов мне тоже никчему, поэтом добавляю (пример для одного клиентского компьютера):

```
subnet 192.168.0.0 netmask 255.255.255.0 { } group { # PXE-specific configuration
directives... next-server 192.168.0.7; filename "/srv/tftp/thinstation.nbi"; # You
need an entry like this for every host # unless you're using dynamic addresses host
anisimova { hardware ethernet 00:20:ED:B7:E8:CF; fixed-address
192.168.0.34; } }
```

где

next-server 192.168.0.7; - указывает что dhcp сервер находится на ип адресе 192.168.0.7 (ип адрес сервера) filename "/srv/tftp/thinstation.nbi"; - указывает откуда и какое ядро будет грузиться на клиентском компьютере. host anisimova - имя клиентского компьютера hardware ethernet 00:20:ED:B7:E8:CF; - мак адрес сетевой карты клиентского компьютера fixed-address 192.168.0.34; - ип адрес который выдастся на основе мак адреса сетевой карте клиентского компьютера.

Запускаем dhcp:

```
sudo /etc/init.d/dhcp3-server start
```

Устанавливаем TFTP.


```
sudo apt-get install tftpd
```

Смотрим inetd.conf (конфигурационный файл супер сервера inetd):

```
sudo nano /etc/inetd.conf
```

Должно быть типа такого:

```
tftp      dgram  udp    wait  nobody /usr/sbin/tcpd  /usr/sbin/in.tftpd /srv/tftp
```

Если в inetd.conf внесли изменения то перезапускаем супер-сервер inetd:

```
sudo invoke-rc.d openbsd-inetd restart
```

Проверяем прослушивает ли inetd порт 69 (порт TFTP):

```
sudo netstat -lnp | grep :69
```

Должно быть так:

```
udp      0      0 0.0.0.0:69          0.0.0.0:*        7331/inetd
```

Из inetd.conf видно что TFTP сервер будет располагаться в /srv/tftp. Создаем необходимую папку:

```
sudo mkdir /srv/tftp
```

Устанавливаем владельцем папки пользователя nobody, т.к. демонт tftpd запускается от пользователя nobody

```
sudo chown nobody /srv/tftp
```

Устанавливаем и настраиваем SSH.

```
sudo apt-get install ssh
```

Редактируем конфиг файл демона sshd:

```
sudo nano /etc/ssh/sshd_config
```

В нем проверяем значения, если они отличаются то заменяем их:

```
PubkeyAuthentication yes  AllowUsers nx
```

Если в конфиг файле внесли изменения то перезапускаем сервис:

```
sudo service sshd restart
```

Устанавливаем RX@Etersoft.

Скачиваем все пакет RX@Etersoft:

```
sudo wget ftp://updates.etersoft.ru/pub/Etersoft/RX@Etersoft/3.3.0-14/Debian/5.0/freenx-server\_0.7.4-eter14debian\_i386.deb sudo wget ftp://updates.etersoft.ru/pub/Etersoft/RX@Etersoft/3.3.0-14/Debian/5.0/nx\_3.3.0-eter5debian\_i386.deb
sudo wget ftp://updates.etersoft.ru/pub/Etersoft/RX@Etersoft/3.3.0-14/Debian/5.0/nxclient\_3.3.0.6-eter6debian\_i386.deb
sudo wget ftp://updates.etersoft.ru/pub/Etersoft/RX@Etersoft/3.3.0-14/Debian/5.0/nxsadmin\_0.2.1-eter3debian\_i386.deb
```

Устанавливаем:

```
sudo apt-get install libcupsys2 sudo apt-get install expect sudo dpkg -i *.deb sudo nxsetup --install --setup-nomachine-key
```

Основной конфигурационный файл располагается /etc/nxserver/node.conf. К сожалению конфиг на русском языке я не нашел, пришлось перевести самому (коряво конечно получилось, но суть понять можно): [Медиа:Node.conf_ru](#)

После установки инициализируйте окружение системного пользователя nx с помощью команды:

```
sudo /etc/init.d/freenx-server setup
```

После успешного завершения, запускаем сервис:

```
sudo /etc/init.d/freenx-server start
```

Добавляем пользователей и указываем их пароли для NX-сервера:

```
sudo nxserver --adduser пользователь sudo nxserver --passwd пользователь
```

Установка Thinstation 2.2 + PXE

Thinstation представляет собой среду для компоновки (возможна и компиляция) загрузочного линукс ядра. Благодаря Thinstation можно скомпоновать из готовых пакетов единое ядро с поддержкой именно того железа которое стоит на стороне клиента и именно тех технологий и протоколов которые необходимы именно для конкретного клиента. Если в списке доступных пакетов нет необходимого, но в тоже время есть исходники этого драйвера, то можно скачать с офиц. сайта исходники системы, развернуть их, добавить в них исходники нежостающего драйвера, скомпилировать его, а потом как обычно скомпоновать готовое ядро.

Скачиваем с официального сайта(размер около 60 мегабайт):

```
sudo wget http://garr.dl.sourceforge.net/sourceforge/thinstation/Thinstation-2.2.2.tar.gz
```

Распаковываем:

```
sudo tar -zxvf Thinstation-2.2.2.tar.gz
```

Получаем такую структуру:

```
/Thinstation-2.2.2/boot-images - каталог куда помещаются скомпилированные образы загрузки  
/Thinstation-2.2.2/conf /Thinstation-2.2.2/Documentation - документация  
/Thinstation-2.2.2/kernel - ядро /Thinstation-2.2.2/packages - пакеты для сборки образа  
/Thinstation-2.2.2/src /Thinstation-2.2.2/utils /Thinstation-2.2.2/build  
/Thinstation-2.2.2/build.conf - конфигурационный файл где указываются все настройки сборки образа  
/Thinstation-2.2.2/build.conf.example - примерный конфигурационный файл где указываются все настройки сборки образа  
/Thinstation-2.2.2/ChangeLog  
/Thinstation-2.2.2/thinstation.conf.buildtime /Thinstation-2.2.2/thinstation.conf.buildtime.sample  
/Thinstation-2.2.2/thinstation.hosts.example
```

К сожалению конфиг файла сборки ядра на русском языке я не нашел, пришлось перевести самому (коряво конечно получилось, но суть понять можно): [Медиа:Build.conf.example_ru](#)

Редактируем build.conf:

```
sudo nano ~/Thinstation-2.2.2/build.conf
```

Чтобы при подключении клиента NX к серверу каждый раз не появлялось окошко с предупреждением о незнакомом хосте, создадим в корне Thinstation файл known_hosts:

```
ssh-keyscan -t rsa nxserver_ip>>~/thinstation/known_hosts
```

В качестве «nxserver_ip» надо указать IP-адрес NX-сервера. Таким образом клиент будет знать о цифровом отпечатке rsa-ключа NX-сервера при аутентификации.

Создаем образ запустив скрипт ~/thinstation-2.2.2/build. После компиляции в Thinstation-2.2.2 создается файл thinstation.conf.sample со всеми настройками самого ядра и всех интегрированных пакетов, к сожалению конфиг на русском языке я не нашёл, пришлось перевести самому (коряво конечно получилось, но суть понять можно):

Готовый файл образа находится по адресу
~/thinstation/boot-images/etherboot/thinstation.nbi. Копируем его на TFTP-сервер:

```
sudo cp ~/Thinstation-2.2.2/boot-images/etherboot/thinstation.nbi /srv/tftp sudo chown nobody /srv/tftp/thinstation.nbi sudo cp ~/Thinstation-2.2.2/thinstation.conf.buildtime /srv/tftp/thinstation.conf.network sudo chown nobody /srv/tftp/thinstation.conf.network
```

ИсточНИК — «
http://wiki.altlug.ru/%D0%A2%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%B%D1%8C%D0%BD%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%BD%D0%B0_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%B5_Debian_Lenny
»

оригинал:

[тут](#)

//