

HOW-TO: Загрузка по сети

0. Вступление

Загрузка по сети - это очень удобная, и зачастую даже просто незаменимая вещь. Не нужно вскрывать корпус компьютера(порой это совсем не так просто, как с обычными PC-блоками), не нужен cdrom, не нужен свободный ide-шлейф, не нужен дисковод, не нужно USB.

Система для загрузки проста для изменения - её не нужно куда "залить" и записывать. Это просто каталог.

Почему-то эту тему боятся и избегают многие. Это кажется чем-то сложным, проблематичным и труднореализуемым. На самом деле всё, как и остальное гениальное, просто.

1. Что нужно, чтобы загрузить Linux на машину по сети?

- В машинке должна быть сетевая карточка с поддержкой [Intel's PXE](#) . Есть ещё несколько реализаций протоколов загрузки по сети, но PXE наиболее распространён и является, практически, де-факто подразумеваемым протоколом загрузки по сети.

- Настроенная Linux-система(назовем её машина-хост), которая будет выдавать IP-адрес, и содержать загружаемое ядро и сам образ системы. Это, вообще-то, не обязательно может быть Linux-система, но в данном how-to это будет подразумеваться.

- Сеть Ethernet. Машину-хост и машинкой-жертвой можно соединить простым кросс-кабелем, а можно и обычными методами. :)

- Знание MAC-адреса машинки-жертвы. Его можно посмотреть либо по логам DHCP-сервера, либо переписать прямо с экрана в момент начала загрузки по PXE, ну а можно и вовсе обойтись, и настроить DHCP-сервер так, чтобы он выдавал нужный IP и образ всем без разбору. Но это совсем другая история(хорошо описанная в гугле) и она выходит за рамки данного how-to.

2. Как происходит процесс загрузки по сети?

Быстрый ответ: автоконфигурируется сеть по dhcp, загружается загрузчик по tftp, который загружает ядро по tftp, которое загружает всю остальную рут-систему по nfs.

Подробный ответ: Первое, что делает при загрузке машинка, в которой установлена загрузка через PXE - посылает широковещательный DHCP-запрос в сеть в поисках сервера. Машина-хост, на которой мирно посапывая дремлет DHCP-демон, выдаёт

адрес и путь к файлу для загрузчика. PXE Boot ROM на основе полученной информации, конфигурирует сетевой адаптер. Если всё проходит успешно, загрузчик загружается по [TFTP протоколу](#) и берёт на себя дальнейший ход событий. В общей схеме события развиваются дальше так - загружается по тому же TFTP-протоколу специально подготовленное для загрузки по сети Linux-ядро. Ядру при этом загрузчик передаёт нужные параметры для загрузки по [NFS-протоколу](#). Ядро, загрузившись, монтирует nfs-раздел на машине-хосте и загружает систему уже оттуда.

3. Как это всё настроить и поднять с нуля?

Актуальный вопрос :)

Ведь нам нужны:

- а) установленные, настроенные и запущенные dhcp-, tftp- и nfs-сервера
- б) загрузчик и готовый образ корневой системы, в который можно chroot-иться и настраивать по надобностям, устанавливая/добавлять пакеты и тп.
- в) ядро, подготовленное для загрузки по сети.

Итак, по пунктам, подразумевая, что система у нас Debian(и радуясь, что когда-то сделали правильный выбор дистрибутива), и что все относящиеся к делу файлы мы поместим в каталог /tftpboot:

0. Создание каталогов и установка загрузчика pxelinux.

Для начала создадим и определим каталоги.

Нашим "главным" каталогом пусть будет /tftpboot.

В нем будет два подкаталога: boot/ (с корневой системой) и pxelinux.cfg/ (с настройками загрузчика)

```
# mkdir /tftpboot # mkdir /tftpboot/boot # mkdir /tftpboot/pxelinux.cfg
```

Далее, в качестве загрузчика мы будем использовать [pxelinux](#) .

Установим пакет syslinux и перепишем оттуда один-единственный файл - pxelinux.0 - в каталог /tftpboot:

```
# apt-get install syslinux # cp /usr/lib/syslinux/pxelinux.0 /tftpboot/ #
```

Далее - в каталоге /tftpboot/pxelinux.cfg/ создаем файл default с примерно таким текстом:

```
serial 0 prompt 1 timeout 99 default pxeboot label pxeboot kernel bzImage append ip=dhcp nfsroot=192.168.150.126:/tftpboot/boot root=/dev/nfs init=/sbin/init
```

Его синтаксис похож на синтаксис lilo.conf. Комментарии, вроде бы тоже не нужны.

Если нужно создать различные конфигурации для разных машин, то можно создавать вместо default файлы для каждого MAC-адреса отдельно. Подробнее про это можно прочитать на [странице pxelinux](#) .

1. Создание корневой файловой системы

Это самый интересный вопрос. Здесь стоит сформулировать чётко постановку задачи - для чего нужна будет система? Какой софт там должен быть, на каком железе она будет запускаться и тп.

В принципе, для простейшей загрузки достаточно минимальной системы в пару мегабайт. Образов и руководств по созданию оных в гугле море.

Это может быть rescue-система, может быть инсталлятор чего-нибудь, а может быть и полноценная desktop-система.

В моем случае, мне нужно, чтобы система загружалась практически на любом железе и стартовала мой собственный инсталлятор. Поскольку места мне не жалко, то я решил просто поставить дефолтную минимальную debian-систему. Радуюсь тому, что когда-то выбрал правильный дистрибутив, это оказалось очень просто сделать:

```
# apt-get install debootstrap ... # debootstrap sarge /tftpboot/boot
```

При завершении, в каталоге /tftpboot/boot будет полностью рабочая, функциональная и загружаемая система, примерно 140Мб весом. В этот каталог можно за-chroot'иться, инсталлировать и удалять пакеты, изменять rc-скрипты, и вообще творить всё что угодно. Такой себе линукс в линуксе.

2. Создание ядра.

Для того, чтобы ядро можно было загружать по сети нужно несколько условий при его сборке:

а) включить опцию "Network Options -> IP: autoconfiguration -> dhcp" (для универсальности, лучше все варианты тоже включить)

б) включить поддержку NFS Filesystem

в) включить "FS->Network Filesystems->Root Over NFS".

Это обязательная часть спецификации. Остальное же - на ваше усмотрение. Что нужно в этом ядре, что не нужно. Лучше не жалеть на размере, и вкомпилировать в него больше драйверов, отключив модули совсем, чтобы избавиться от необходимости двухступенчатой загрузки и initrd-файла.

Далее просто копируем bzImage в каталог /tftpboot. Загрузчик у нас уже настроен именно на этот путь и имя файла.

3. Настройка DHCP.

Инсталлим сервер:

```
# apt-get install dhcpd
```

Автор: adm

22.12.10 18:32 - Последнее обновление 07.01.12 14:15

Редактируем его конфигурационный файл(/etc/dhcpd.conf) под свою сеть и добавляем в него запись для нашей машинки-жертвы:

```
host pxeboot { hardware ethernet 08:00:0e:aa:bb:cc; fixed-address 192.168.150.127;
filename "/tftpboot/pxelinux.0"; }
```

Комментарии, думаю, излишни.

Запускаем:

```
# /etc/init.d/dhcp start Starting DHCP server: dhcpd. #
```

Можно использовать и bootp-сервер, но dhcp предпочтительнее.

4. Настройка TFTP

Загрузчик pxelinux требует для загрузки от tftp-сервера поддержки опции "tsize" (RFC 1784/RFC 2349). Поэтому стандартный простейший TFTP-сервер tftpd не подходит - вместо него нужно использовать либо atftpd, либо предпочитаемый и используемый мною сервер tftpd-hpa.

Инсталляция так-же проста:

```
# apt-get install tftpd-hpa
```

Для уверенности, указываем опции запуска сервера в файле /etc/default/tftpd-hpa:

```
OPTIONS="-l -s /"
```

Запускаем:

```
# /etc/init.d/tftpd-hpa start Starting HPA's tftpd: in.tftpd. #
```

5. Настройка NFS-сервера.

Для наших целей будем использовать nfs-user-server.

```
# apt-get install nfs-user-server portmap
```

Все конфигурирование заключается в указании каталогов и доступа в файле /etc/exports, в который мы добавляем одну-единственную строчку:

```
/tftpboot/boot 192.168.150.0/255.255.255.0 (rw,no_root_squash,insecure)
```

Автор: adm

22.12.10 18:32 - Последнее обновление 07.01.12 14:15

Подразумеваю, что в каталоге /ftpboot/boot будет лежать наша система и что машина-жертва находится с нами в сети 192.168.150.0.

Запускаем:

```
# /etc/init.d/portmap start Starting portmap daemon: portmap. # /etc/init.d/nfs-user-server start Starting NFS servers: nfsd mountd. #
```

6. Проверка перед полётом

Можем убедиться в том, что все нужные нам сервисы работают командой netstat -tlnu.

Должно быть что-то вроде этого:

```
# netstat -tlnu Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address          Foreign Address    State       PID/Program name
tcp        0      0 0.0.0.0:*           LISTEN          14464/rpc.nfsd
tcp        0      0 0.0.0.0:*           LISTEN          14466/rpc.mountd
tcp        0      0 0.0.0.0:21         LISTEN          3383/ftpd:acceptin
udp        0      0 0.0.0.0:2049       0.0.0.0:*       14464/rpc.nfsd
udp        0      0 0.0.0.0:651        0.0.0.0:*       14466/rpc.mountd
udp        0      0 0.0.0.0:67         0.0.0.0:*       14416/dhcpd
udp        0      0 0.0.0.0:*           14427/in.tftpd
udp        0      0 0.0.0.0:*           14449/portmap #
```

6. Запуск

Проверяем соединения, включаем машину-жертву, настраиваем её в биосе на загрузку по сети и наблюдаем за процессом. Если всё настроено правильно и никаких проблем не возникает, то мы будем наблюдать процесс загрузки линукса, с приглашением login: в итоге.

Если нет - то первым делом нужно смотреть в лог /var/log/syslog и находить решения проблем.

4. Заключение

Собственно, всё.

Я и раньше загружался по сети, но использовал для этого чужие образы, чужие конфиги и, вообще, не совсем понимал как и что происходит. А сейчас, в данной системе, я полностью контролирую весь процесс и - главное - система так доступна для изменения, что позволяет сделать в два движения просто страшные вещи. :) Особенно в паре с autologin. :)

Надеюсь, кому это пригодится, хотя это запись для себя :)

Автор: adm

22.12.10 18:32 - Последнее обновление 07.01.12 14:15

ИСТОЧНИК <http://eth0-blog.livejournal.com/39843.html>