МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРАСНОДОНСЬКИЙ ПРОМИСЛОВО ЕКОНОМІЧНИЙ КОЛЕДЖ

Реферат з предмету

Операційні системи

На тему

Створення віртуальної мережі з віддаленим завантаженням вузлів

Студента групи 1ОКІСМ-06

Петренко Михайла

Краснодон

2009

**Зміст**

Введення

1. Загальні відомості

2. Створення віртуальних комп'ютерів

3. Налаштування клієнта

3.1 Створення образу завантаження

3.2 Запис образу завантаження

4. Налаштування сервера

4.1 Установка операційної системи

4.2 Налаштування адаптера eth0

4.3 Налаштування DHСP

4.4 Налаштування FTP і TFTP

4.5 Настройка DNS

4.6 Створення ядра

4.7 Налаштування NFS

4.8 Налаштування файлової системи для клієнта

Висновок

Література

**Введення**

Завантаження по мережі надає адміністраторам ряд переваг, серед яких:

* легкість установки нового ПЗ;
* централізоване адміністрування;
* можливість відмови від встановлення жорстких дисків на клієнтські комп'ютери, що призводить до підвищення надійності системи і зниженню її вартості.

У даній статті ми розглянемо організацію процесу такої завантаження. Для зручності проведення експериментів будемо працювати не з фізичними комп'ютерами, а з віртуальними, для реалізації яких використовуємо пакет VMware Workstation (використовувалася версія 4.0), встановлений в ОС Windows 2000 SP4.

**1. Загальні відомості**

Ідея мережевий завантаження полягає в наступному [1]. Комп'ютер (клієнт) має в енергонезалежній пам'яті (на жорсткому диску, на дискеті) деякий початковий код завантаження, який дозволяє з'єднатися з іншим комп'ютером (сервером), завантажити з нього ядро, а також використовувати його файлову систему. Таким чином, на клієнті може взагалі бути відсутнім жорсткий диск.

Стандартний процес мережевий завантаження виглядає наступним чином:

1. Після ініціалізації завантажувач визначає необхідні IP-адреси: свій, сервера, шлюзу. Вони запитуються по BOOTP або DHCP.
2. Отримавши відповідь від сервера, він ініціалізує процес завантаження ядра по протоколу TFTP.
3. Ядро підключає файлову систему. Вона може знаходиться на жорсткому диску клієнта, в пам'яті клієнта, або підключатися з сервера по протоколу NFS.

Варіантів реалізації мережевої завантаження може бути багато. Ми розглянемо такий: клієнтський комп'ютер - Бездисковий, для створення завантажувача використовується Etherboot, визначення налаштувань відбувається по DHCP, операційна система клієнта і сервера - Debian (версія ядра - 2.4.18). Підключення кореневої файлової системи здійснимо двома способами: за NFS і з RAM-диска.

**2. Створення віртуальних комп'ютерів**

Запустимо VMware workstation і натиснемо Ctrl + N для створення нового віртуального комп'ютера. Далі, вибираємо:

* Virtual machine configuration: Custom;
* Guest operation system: Linux;
* Virtual machine name: Client;
* Location: каталог для віртуального комп'ютера;
* Memory guest size (MB): 64;
* Network connection: Use host-only networking;
* Disk: Create a new virtual disk;
* Disk size (GB): 0,1;
* Disk file: Client.vmdk.

Необхідно зробити зауваження щодо пам'яті. За замовчуванням, в VMware діють обмеження на використовувану віртуальними комп'ютерами пам'ять. Так, якщо на комп'ютері встановлено 256 Мб пам'яті, то віртуальні комп'ютери не можуть використовувати більш 192 Мб. Оскільки буде необхідно одночасно запускати клієнтський і серверний комп'ютери, кожному з них не можна виділити більше 80 Мб (по 16 Мб потрібно самій VMware на запуск кожного віртуального комп'ютера). Для трьох віртуальних комп'ютерів це обмеження опускається до 48 Мб. Змінити ліміт за замовчуванням можна в меню Edit пункт Application Settings ..., вкладка Memory. Виконавши цю операцію ще двічі, замінюючи Client на Server і на Work, вибравши розмір жорсткого диска 1,0 і 0,5 відповідно, створюємо віртуальний сервер і робочий віртуальні комп'ютери. Набудуємо обладнання кожного віртуального комп'ютера. Це можна зробити, вибравши в контекстному меню відповідного комп'ютера пункт Edit ..., вкладка Hardware. У клієнта видалимо Hard Disk 1, CD-ROM 1, USB Controller і Audio; у сервера і робочої комп'ютерів - Floppy 1, USB Controller і Audio. Додамо серверу другу мережеву картку, встановивши для неї тип NAT. Тепер необхідно відредагувати налаштування мережі, для чого виберемо пункт Virtual Network Setting ... меню Edit. Перейдемо на закладку DHCP, виберемо адаптер VMnet1 і натиснемо кнопку Remove. Таким чином ми відключаємо службу DHCP від адаптера VMnet1 (до якого за замовчуванням будуть підключені віртуальні комп'ютери), тому що ми будемо налаштовувати її на сервері спеціальним чином. Для подальших налаштувань нам необхідно знати IP адреса адаптера VMnet1. Переглянути його можна, виконавши команду ipconfig у командному рядку ОС Windows. У мене він такий: 192.168.223.1

**3. Налаштування клієнта**

**3.1 Створення образу завантаження**

Для створення образу завантажувача скористаємося вільно поширюваним пакетом Etherboot. Створити образ можна вручну, за допомогою утиліт, що входять в пакет. Однак ми використовуємо сервіс, що надаються сервером ROM-o-matic.net. Тут можна вибрати тип своєї карти за допомогою стандартного Web-інтерфейсу і зробити необхідні налаштування, а сервер автоматично згенерує образ завантажувача й запропонує його завантажити.

При написанні документа користувалася версія 5.2.5 пакету ROM-o-matic. Для створення образу необхідно:

1. вибрати тип мережевої карти (довідку за підтримуваним картками можна отримати тут);
2. вибрати формат завантажувача (залежить від того, куди ви хочете помістити код завантажувача);
3. призвести, якщо потрібно, додаткову настройку завантажувача, натиснувши кнопку Configure;
4. натиснути кнопку Get ROM і скачати замовлений образ.

VMware 4.0 емулює мережеву картку AMD 79c970 PCnet LANCE, тому в першому пункті виберемо pcnet32: lancepci, у другому - Floppy bootable ROM Image, натиснемо кнопку Get ROM і збережемо образ завантажувача в C: \ Temp \ boot.img.

**3.2 Запис образу завантаження**

Образ завантажувача може зчитуватися комп'ютером з жорсткого диска, дискети або з вбудованої FLASH пам'яті мережевої карти. Останній варіант - найбільш зручний, але перш ніж записувати завантажувач під FLASH, необхідно переконається в його працездатності. Тому, для тестування, ми запишемо завантажувач на дискету.

Зробити це можна за допомогою програми rawrite2, яка входить в стандартний дистрибутив Debian (знаходиться на першому диску в каталозі install). Як параметри їй необхідно вказати файл з завантажувачем і пристрій для запису.

Вставимо чисту дискету в дисковод A і дамо команду: *тrawrite2-f C: \ Temp \ boot.img-d a*

Тепер можна спробувати завантажити клієнтський комп'ютер (вибравши його у списку та натиснувши Ctrl + B). Завантаження зупиниться на повідомленні Searching for server (DHCP ).... Після слова lancepci буде відображений MAC адресу мережевої карти, у мене він такий: 00:0 C: 29: E6: 36: BC. Запишемо його, він стане в нагоді при налаштуванні DHCP.

**4. Налаштування сервера**

**4.1 Установка операційної системи**

Для установки ОС необхідний дистрибутив, що складається мінімум з двох дисків. Вставте перший диск дистрибутиву Debian у пристрій для читання дисків та запустіть віртуальний сервер. На запрошення boot: введіть vanilla і натисніть Enter. Це запустить інсталяцію з ядром, що підтримує безліч пристроїв, у тому числі SCSI пристрої. Далі установка проходить у звичайному режимі. Підтримка мережі повинна бути встановлена обов'язково. Якщо Debian знайде мережеву картку, то він автоматично запропонує вам налаштувати мережу. Якщо ж цього не станеться, необхідно вибрати драйвер мережевої картки вручну (pcnet32). Кілька слів про налаштування мережі. Задамо ім'я комп'ютера (hostname): server. Як інтерфейс для встановлення Debian через NFS або HTTP виберемо eth1. На питання про автоматичної конфігурації по DHCP відповімо згодою. Після установки ядра системи і драйверів відбудеться перезавантаження. Інсталятор Debian задасть кілька стандартних запитань з настроювання системи, а потім запропонує встановити додаткові пакети (запустити tasksel і dselect для їх установки). Відмовтеся - це простіше виконати з командного рядка. Поштову систему також налаштовувати не потрібно. Увійдіть в систему як root і дайте наступну команду: *apt-get install mc dhcp ftpd tftpd nfs-kernel-server bind kernel-source-2.4.18 make gcc libc6-dev* Все необхідне програмне забезпечення знаходиться на першому і другому дисках дистрибутива. Детальний опис процесу встановлення знаходиться в [2].

**4.2 Налаштування адаптера eth0**

Побудуємо наступну мережеву конфігурацію: сервер через адаптер eth1 буде пов'язаний із зовнішніми мережами (у тому числі з Internet, якщо комп'ютер підключений до Internet), а через адаптер eth0 - з внутрішньою мережею, що складається з віртуальних комп'ютерів. Коректну налаштування eth1 забезпечить VMware, а інтерфейс eth0 необхідно сконфігурувати вручну.

Для цього відредагуємо файл */ etc / network / interfaces*, додавши в нього наступні рядки:

auto eth0

iface eth0 inet static

address 192.168.223.2

netmask 255.255.255.0

Замість адреси 192.168.223.2 необхідно вказати адресу, наступний за адресою адаптера VMnet1. Після перезавантаження віртуального комп'ютера інтерфейс повинен заробити.

**4.3 Налаштування DHCP**

Сервіс DHCP відповідає за видачу клієнтові його IP адреси, імені, інших параметрів мережі, а також імені файлу з ядром операційної системи. Його настроювання здійснюється з допомогою конфігураційного файлу / etc / dhcpd.conf. Ми використовуємо такий файл:

option domain-name "private.net";

option domain-name-servers server.private.net;

subnet 192.168.223.0 netmask 255.255.255.0 {

group {

filename "net\_boot\_kernel";

host node01 {

hardware ethernet 00:0C:29:E6:36:BC;

fixed-address 192.168.223.3;

option host-name "node01";

}

}

}

Після subnet вказана IP адреса адаптера VMnet1, з заміненої на 0 останньою цифрою. У рядку filename сказано, що ім'я файлу з ядром для мережевої завантаження - net\_boot\_kernel. У рядку hardware ethernet необхідно вказати раніше записаний MAC адресу клієнта, у рядку fixed-adress вказується IP, який буде виданий клієнту. Нам необхідно, щоб DHCP сервер обробляє запити тільки з інтерфейсу eth0 (з внутрішньої мережі), тому відредагуємо файл / etc / default / dhcp, присвоївши змінної INTERFACES значення eth0. Невеликі зміни потрібні для DHCP клієнта. У файл / etc / dhclient.conf додамо наступні два рядки:

supersede domain-name "private.net";

prepend domain-name-servers 127.0.0.1;

Перший рядок вказує домен пошуку, а друга - додає додатковий (до того, який буде отриманий від DHCP сервера) DNS сервер. Ці зміни необхідні для формування правильного файлу resolv.conf. Тепер, при завантаженні клієнта, на екрані повинні відобразитися IP адреси клієнта і сервера, отримані від сервера (службу DHCP попередньо необхідно перезапустити). Більш докладну інформацію з налаштування DHCP можна знайти в [3].

**4.4 Налаштування FTP і TFTP**

Перекачування ядра з сервера на клієнт здійснюється по протоколу TFTP. Для його коректної роботи у файлі / etc / inetd.conf повинна бути присутнім рядок: ***tftp dgram udp wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd /exports***

Останнім параметром зазначений каталог, у якому буде проводитись пошук ядра операційної системи. Всі директорії та файли повинні бути доступні на читання всім. Якщо при спробі завантажити клієнта, ви отримуєте повідомлення TFTP error 1 (File not found), значить TFTP налаштований правильно. Для налаштування ftp відредагуємо файл / etc / ftpusers, видаливши з нього рядок root, що дозволить підключатися до сервера за ftp під ім'ям root з ОС Windows.

**4.5 Настройка DNS**

Доменом нашій віртуальній мережі буде називатися private.net. Сервер буде мати ім'я server.private.net, а клієнти імена nodeXX.private.net, де XX - порядковий номер клієнта.

Основним конфігураційних файлів DNS сервера є / etc / bind / named.conf. В розділ options цього фала додамо рядок:

listen-on { 192.168.223.2; 127.0.0.1; };

Перший IP адреса - адреса інтерфейсу eth0. Тепер DNS сервер буде обробляти запити тільки з цих адрес. У кінець файлу додамо рядки:

zone "223.168.192.in-addr.arpa" {

type master;

file "/etc/bind/server.rev";

};

zone "private.net" {

type master;

file "/etc/bind/server.hosts";

};

Далі необхідно створити файли / etc / bind / server.rev і / etc / bind / server.hosts. Нижче наведено їх листинги.

Файл server.rev:

@ IN SOA server.private.net. root.server.private.net. (

2004120420 ; Serial

8H ; Refresh

4H ; Retry

4W ; Expire

1D ) ; Negative Cache TTL

IN NS server.private.net.

2 IN PTR server.private.net.

3 IN PTR node01.private.net.

Файл server.hosts:

@ IN SOA server.private.net. root.server.private.net. (

2004120420 ; Serial

8H ; Refresh

4H ; Retry

4W ; Expire

1D ) ; Negative Cache TTL

IN NS server.private.net.

localhost IN A 127.0.0.1

server IN A 192.168.223.2

node01 IN A 192.168.223.3

На цьому налаштування DNS-сервера закінчена. За додатковою інформацією звертайтеся до [4].

**4.6 Створення ядра**

Компіляція ядра

Стандартне ядро не підходить для завантаження по мережі, тому необхідно його перекомпілювати. Перед компіляцією ядра необхідно з'ясувати конфігурацію устаткування віртуального сервера. Для цього в Debian можна скористатися командою lspci. Запишемо моделі IDE, SCSI і Ethernet контролерів. У VMware 4.0 це Intel Corp.82371AB PIIX4 IDE, BusLogic BT-946C [MultiMaster 10] і AMD 79c970 [PCnet LANCE] відповідно. Перейдемо в каталог / usr / src і виконаємо команду tar - bzip2-x-f kernel-source-2.4.18.tar.bz2 Тепер зайдемо в каталог / usr/src/kernel-source-2.4.18 і виконаємо команду make mrproper config.

Зараз необхідно вибрати конфігурацію ядра, відповідаючи на питання. Все, що може знадобитися до того, як стануть доступні мережеві ресурси (мережеві карти, підтримка чіпсетів, протоколу TCP, NFS і т.д.) необхідно включити в ядро. Додатково, необхідно включити наступні налаштування:

* CONFIG\_BLK\_DEV\_LOOP
* CONFIG\_BLK\_DEV\_NBD
* CONFIG\_BLK\_DEV\_RAM (тільки при підключенні кореневої файлової системи з RAM диска)
* CONFIG\_BLK\_DEV\_RAM\_SIZE = 30720 (тільки при підключенні кореневої файлової системи з RAM диска)
* CONFIG\_BLK\_DEV\_INITRD (тільки при підключенні кореневої файлової системи з RAM диска)
* CONFIG\_PACKET
* CONFIG\_FILTER
* CONFIG\_IP\_PNP
* CONFIG\_IP\_PNP\_DHCP
* CONFIG\_NFS\_FS
* CONFIG\_NFS\_V3
* CONFIG\_ROOT\_NFS (тільки при підключенні кореневої файлової системи через NFS)

Після завершення налаштувань скомпіліруем ядро командою make bzImage

Передбачається, що версії встановленого і компіляції ядра збігаються. Якщо ж це не так, і при компіляції ядра деякі з опцій були скомпільовані як модулі, необхідно командою make modules призвести перекомпіляцій модулів. Докладніше про компіляції ядра можна дізнатися в [5].

Підготовка ядра для віддаленого завантаження

Щоб ядро можна було завантажувати через мережу за допомогою Etherboot або Netboot, воно має бути спеціальним чином підготовлено. Для цього служить програма mknbi, що входить в пакет mknbi (при створенні цього документа використовувалася версія 1.4.4). Помістити її на сервер можна, підключившись з OC Windows до сервера по ftp.

Для її установки необхідно розпакувати файл з вихідними текстами в каталог / usr/src/mknbi-1.4.4 і виконати команди

make

mkdir / usr / local / lib / mknbi

cp first32elf \* / usr / local / lib / mknbi

Програма має декілька параметрів, з яких ми розглянемо чотири:

* Format = format Вказує формат вихідного файлу.
* Target = target Вказує тип цільового двійкового коду.
* Rootdir = rootdir Зазначає ім'я каталогу, з якого буде подмонтірована коренева файлова система
* Ip = string Дозволяє задати IP адреси клієнта, сервера, шлюзу і маску підмережі. Також можливо вказати dhcp або bootp для автоматичної настройки за допомогою цих протоколів.

Створимо каталог, в якому буде розташовуватися ядро для віддаленого завантаження: mkdir / exports

Подальші дії залежать від того, яким чином ми будемо підключати корову файлову систему. Для підключення через NFS необхідно виконати команду (передбачається, що коренева файлова система буде розташовуватися в каталозі / exports/node01) . / Mknbi - format = elf - target = linux - rootdir = / exports/node01 - ip = dhcp / usr/src/kernel-source-2.4.18/arch/i386/boot/bzImage> / exports / net\_boot\_kernel

Для варіанта з RAM диском команда виглядає трохи інакше: . / Mknbi - format = elf - target = linux - rootdir = / dev/ram0 - ip = dhcp / usr/src/kernel-source-2.4.18/arch/i386/boot/bzImage / initrd / initrd . gz> / exports / net\_boot\_kernel

Про те, як створити файл / initrd / initrd.gz, який містить образ кореневої файлової системи, розповідається в п.4.8.

Після закінчення цього етапу клієнтський комп'ютер вже може завантажувати ядро. За інформацією з mknbi звертайтеся до [6].

**4.7 Налаштування NFS**

Налаштування що надаються сервером NFS в мережеве використання ресурсів здійснюється за допомогою файлу / etc / exports.

У нашому прикладі використовуємо такий файл

/usr node01.private.net(rw,no\_root\_squash)

/exports node01.private.net(rw,no\_root\_squash)

/ 192.168.223.4(rw,no\_root\_squash)

Останній рядок потрібна для доступу робочого віртуального комп'ютера (на якому ми будемо формувати файлову систему клієнта) до сервера і після створення файлової системи клієнта її необхідно видалити.

Додаткова інформація з налаштування NFS доступна в [7].

**4.8 Налаштування файлової системи для клієнта**

Для завершення настроювання мережевої завантаження необхідно підготувати файлову систему клієнта. Для цього на самому початку нами було створено третій віртуальний комп'ютер. Встановимо на нього ОС Debian згідно з тими ж, рекомендаціями, що були дані в п.4.1.

Є один нюанс: оскільки ми роздаємо IP адреси статично, перш, ніж новий комп'ютер зможе отримати конфігурацію від DHCP сервера, необхідно підправити файл / etc / dhcpd.conf. Коли установник Debian задасть питання про конфігурацію по DHCP, переключимося на другий термінал (натиснувши Alt + F2), виконаємо команду ifconfig-a і запишемо MAC-адреса адаптера. Після цього переключимося у віртуальний комп'ютер сервера, і відредагуємо файл / etc / dhcpd.conf, додавши в розділ group рядки:

host work {

hardware ethernet 00:0C:29:03:AB:CA;

fixed-address 192.168.223.4;

}

У другому рядку необхідно вказати записаний MAC адресу. Щоб зміни вступили в силу, необхідно перезапустити DHCP сервер командою / Etc / init.d / dhcp restart

Можна продовжувати установку. Команда для встановлення пакунків з програмним забезпеченням тепер буде виглядати наступним чином: apt-get install mc portmap

Для варіанту з підключенням кореневої файлової системи по NFS будемо користуватися каталогом / tmp. Для варіанта з RAM-диском необхідно заздалегідь створити образ RAM-диска. Зробити це можна різними способами. Наведемо один з них:

dd if = / dev / zero of = / initrd / initrd bs = 1k count = 30720

mke2fs-m0-F / initrd / initrd

mount-t ext2-o loop / initrd / initrd / mnt

Тут ми створюємо порожній файл об'ємом 30 Мб, потім формуємо у ньому файлову систему ext2 і монтуємо його як loop пристрій у каталог / mnt, в якому і будемо створювати файлову систему.

Процес створення файлової системи однаковий для обох варіантів і полягає в наступному:

1. Скопіюйте каталоги / bin, / dev, / etc, / lib, / sbin цілком;
2. Створити каталоги home, root, proc, tmp;
3. Створити каталоги usr, usr / bin; скопіювати каталог / usr / sbin, скопіювати файл expr в каталог usr / bin;
4. Створити каталоги var, var / lock, var / log, var / log / news, var / run, var / tmp, var / lib; скопіювати каталог var / lib / dhcp.

Підправимо файл / mnt / etc / fstab. У самому простому вигляді він може виглядати так:

/dev/ram0 / ext2 errors=remount-ro 0 1

proc /proc proc defaults 0 0

server.private.net:/usr /usr nfs

Для підключення кореневої файлової системи по NFS перший рядок необхідно замінити на

server.private.net:/exports/node01 / nfs

Тепер необхідно створити файл / etc / dhclient-enter-hooks, внести до нього наступні два рядки, щоб кожен клієнт встановив своє ім'я, отримане від DHCP сервера

echo "$new\_host\_name" > /etc/hostname

hostname -F /etc/hostname

і зробити цей файл виконуваним: chmod a + x / etc / dhclient-enter-hooks

Якщо ви скомпілював ядро більш нової версії, то необхідно в каталог / lib / modules скопіювати каталог з перекомпілювати модулями, а старий каталог видалити.

Якщо змонтований файл з RAM диском, його необхідно відключити командою umount / mnt, а потім виконати команду mount-t nfs server.private.net: / / mnt

для підключення файлової системи сервера в каталог / mnt. Далі, для першого варіанта необхідно скопіювати вміст / tmp в каталог / mnt/exports/node01, а для другого - виконати команди

gzip -9 / initrd / initrd

cp / initrd / initrd.gz / mnt / initrd

після чого файлову систему сервера необхідно демонтувати, виконавши umount / mnt

Тепер необхідно приєднати отриманий образ до ядра, як описано в кінці п.4.6.

**Висновок**

Отже, ми докладно розглянули організацію процесу мережевий завантаження. У прикладі ми створили тільки одного клієнта, з назвою node01. Для підключення наступних, на сервері в файли dhcpd.conf, server.rev, server.hosts, exportfs потрібно додати відповідні записи. При необхідності можна створити і нову файлову систему, помістивши її в каталог / exports / nodeXX, або створивши нове ядро для віддаленого завантаження. Всі вищеописані дії без будь-яких змін можуть бути застосовані для створення реальної мережі з реальних машин.

**Література**

1. Введення в завантаження по мережі і Etherboot
2. Installing Debian GNU / Linux 3.0 For Intel x86
3. DHCP mini-HOWTO
4. DNS HOWTO
5. Ставимо ядро 2.6, або Ядерна фізика для домогосподарки
6. Документація по mknbi
7. NFS HOWTO
8. Завантаження машин по мережі
9. Віддалене завантаження і Linux: Налаштування віддалено завантажуються робочих станцій з Linux, DOS, Windows 95/98 і Windows NT