

IMPROVING THIN CLIENT NETWORK COMPONENTS

Ya. Shevchenko

Taras Shevchenko National University of Kyiv

A. Moshenskyy

National University of Food Technologies

Key words:

*Diskless workstation
Thin client
LAN
Modification
gPXE
BIOS
LTSP
Virtual machine
Bash-script*

Article history:

Received 10.02.2016
Received in revised form
05.03.2016
Accepted 21.03.2016

Corresponding author:

A. Moshenskyy
E-mail:
ut5uuv@yandex.ua

ABSTRACT

The article describes the technological aspects of setting up hardware and software complex formed on the operating system Linux Debian x86 with installed LTSP terminal server. The recommendations on hardware terminal server are also presented in this study. The ways to improve the diskless computers, equipping them to support the remote preboot environment over the network (gPXE) for simultaneous connection of diskless workstations with the terminal server LTSP, are presented. The article provides recommendations on equipping the thin client with the gPXE environment. Some settings of network infrastructure for the operation of hardware and software system are presented and some improvements are suggested for the bash-script control of virtual machines of a hardware and software system.

УДОСКОНАЛЕННЯ МЕРЕЖЕВИХ КОМПОНЕНТ ТОНКОГО КЛІЄНТА

Я.О. Шевченко

Київський національний університет імені Тараса Шевченка

А.О. Мошенський

Національний університет харчових технологій

У статті розглянуто технологічні аспекти налаштування апаратно-програмного комплексу, сформованого на базі операційної системи Linux Debian x86 із встановленим термінальним сервером LTSP, та наведено рекомендації щодо апаратного забезпечення термінального сервера. Продемонстровано низку способів для вдосконалення бездискових ЕО після обладнання їх підтримкою середовища віддаленого завантаження операційних систем по мережі (gPXE) для одночасного з'єднання бездискових робочих станцій з термінальним сервером LTSP. Сформульовано рекомендації щодо обладнання тонких клієнтів середовищем gPXE, звернено увагу на особливості налаштування мережевої інфраструктури для функціонування апаратно-програм-

ного комплексу та вдосконалено адміністрування віртуальних машин апаратно-програмного комплексу у вигляді *bash-сценаріїв*.

Ключові слова: *бездискова робоча станція, тонкий клієнт, LAN, модифікація, gPXE, BIOS, LTSP, віртуальна машина, bash-сценаріїв.*

Постановка проблеми. Для підвищення ефективності роботи обслуговуючого персоналу підприємства, а також для раціонального використання наявних апаратних, програмних і людських ресурсів у всьому світі використовуються віртуальне програмне забезпечення та віртуальні операційні системи.

При створенні корпоративних інформаційних систем усе більшої популярності набувають рішення на базі технології «Тонкий клієнт» [6]. Зважаючи на це, у статті сформульовано вимоги до апаратного забезпечення сервера віртуалізації, наведено деталізований опис завантаження тонкого клієнта технології LTSP-сервер; наведено графічну схему принципу роботи технології, описано основні способи оснащення PC тонких клієнтів підтримкою протоколу мережевого завантаження gPXE [4,9]; запропоновано систему автоматичного керування віртуальних операційних систем.

Мета статті. Удосконалення бездискових робочих станцій для апаратно-програмного комплексу на базі технології LTSP, дослідження й удосконалення процесу адміністрування тонких клієнтів.

Результати і обговорення. У сегменті обладнання для організації автоматизованих робочих місць персоналу під тонким клієнтом, як правило, мають на увазі будь-який пристрій, який реалізує для користувача роботу за віртуальним робочим столом традиційного персонального комп'ютера або робочої станції, не володіючи повною функціональністю універсального ПК.

Вимоги до апаратного забезпечення сервера залежать від кількості підключених клієнтів і того, чи будуть вони всі працювати одночасно. На основі виконаного аудиту використання PC-клієнтів системний адміністратор установи має змогу обґрунтувати вимоги до апаратного забезпечення сервера віртуалізації.

На основі проведених авторами експериментальних досліджень сформульовано такі рекомендації щодо апаратного забезпечення LTSP-сервера: 256 МБ оперативної пам'яті (RAM) на сам сервер і по 60 МБ RAM на кожного під'єданого клієнта (іноді доцільно виділяти по 100 МБ RAM на X-термінал без урахування виділеної на платформі віртуалізації віртуальної RAM). Оскільки всі програми виконуються на сервері, не завадить мати запас обчислювальної потужності (CPU): оптимальною рекомендацією до CPU є двоядерна або двопроцесорна система. При роботі з великою кількістю клієнтів не зайвим є використання 64-бітного процесора не тільки через додаткову продуктивність CPU, але й тому, що він може працювати з великим об'ємом пам'яті, що часто є найбільш важливим фактором, який обмежує продуктивність. Вимоги до HDD сервера варіюються від інтенсивності використання всього апаратно-програмного комплексу, загальна рекомендація — 100 ГБ HDD на кожного клієнта.

Клієнтам (X-терміналам) вистачить досить скромних системних ресурсів, оскільки всю роботу виконує LTSP-сервер. Для підтримки X-терміналів оптимальним апаратним забезпеченням є 128 МБ RAM та CPU Pentium II. Основна вимога — комп'ютер повинен мати можливість завантаження по мережі. Якщо ЕОМ клієнта не має апаратної підтримки завантаження по мережі, то потрібно згенерувати образ gPXE [2] і налаштувати клієнтську підтримку завантаження по мережі одним зі способів, описаних нижче. Окрім цього, необхідне гігабітне мережеве з'єднання від сервера до маршрутизатора і 100-мегабітні з'єднання по мережі для клієнтів.

На етапі проектування апаратної частини LTSP-сервера системний адміністратор повинен обрати тип реалізації віртуалізації операційних систем. Розрізняють програмну й апаратну віртуалізацію.

Головним недоліком програмної віртуалізації є використання додаткового програмного забезпечення на сервер віртуалізації, яке виступає як гіпервізор ОС, що дозволяє одночасне/паралельне використання кількох ОС на одному й тому ж сервері. Очевидно, що системному адміністратору необхідно встановити та налаштувати це програмне забезпечення (що потребує додаткових витрат часу), періодично виконувати моніторинг гіпервізора (що теж досить проблематично, оскільки потрібно вибирати такий час для моніторингу гіпервізора, коли X-клієнти не використовують сервер віртуалізації), окрім цього, сам сервер віртуалізації витрачає апаратні ресурси на підтримку працездатності гіпервізора.

Апаратна віртуалізація — віртуалізація з підтримкою спеціальної процесорної архітектури. На відміну від програмної віртуалізації, за допомогою даної реалізації можливе використання ізольованих гостьових систем, які управляються гіпервізором безпосередньо. Гостьова система не залежить від архітектури хостової платформи і реалізації платформи віртуалізації. Розрізняють три технології апаратної віртуалізації [6]: режим віртуального 8086 (застаріла технологія); Intel VT (VT-x, Intel Virtualization Technology for x86); AMD-V. Популярні платформи, що використовують апаратну віртуалізацію [3]: VMware, Hyper-V, KVM.

Головним недоліком другого типу віртуалізації є підбір і придбання CPU з підтримкою апаратної віртуалізації. Апаратна віртуалізація має суттєво більшу кількість переваг над програмною віртуалізацією, окрім цього, в рекомендаціях до апаратного забезпечення LTSP-сервера висунуто досить суттєві вимоги до CPU, оскільки CPU фактично є основою апаратного забезпечення LTSP-сервера. Очевидно, що вибір типу віртуалізації залежить лише від системного адміністратора підприємства, який враховує особливості експлуатації та наявне апаратне забезпечення. В циклі наукових публікацій [1] та [2] описано використання програмної реалізації віртуалізації ОС на основі програми віртуалізації Oracle VM VirtualBox.

Однією з вимог до апаратного забезпечення LTSP-сервера було надання кожному X-клієнту енергонезалежної пам'яті ємністю 100 ГБ. Залежно від кількості тонких клієнтів сумарна кількість пам'яті на HDD LTSP-сервера може варіюватись від 100 ГБ до кількох сотень ТБ. Очевидно, що використання одного HDD не є можливим рішенням даної проблеми, оскільки навіть

після встановлення HDD з таким об'ємом пам'яті даних пристрій може вийти з ладу протягом кількох місяців реальної експлуатації сервера. Зважаючи на це, пропонуються такі варіанти вирішення даної проблеми:

- придбати й обладнати LTSP-сервер системою збереження даних;
- розробити систему збереження даних, використовуючи SCSI-міст [3].

Система складається з 4 HDD різного об'єму пам'яті, SCSI-мосту для залучення їх як апаратної частини LTSP-сервера та дещо спрощеної системи живлення всіх HDD. Використовуючи різні види SCSI-мостів, можна збільшити кількість HDD апаратної частини LTSP-сервера. При цьому спеціальна ніша для встановлення HDD в системному блоці LTSP-сервера залишається вільною, а це означає, що сюди також можна встановлювати HDD.

В основу підключення до LTSP-сервера покладено середовище Preboot Execution Environment (PXE) [2]. PXE — середовище для завантаження комп'ютера за допомогою мережевого адаптера без використання локальних носіїв даних (жорсткого диска, USB-накопичувача тощо). Для організації завантаження системи в PXE використовуються протоколи IP, UDP, BOOTP і TFTP. gPXE, будучи розвитком проекту Etherboot, може замінити існуючі ПЗП з PXE-завантажувачем на мережевому адаптері або може бути використаний як завантажувач операційного середовища «по ланцюжку» від уже наявного завантажувача. Кожна з версій gPXE є певною мірою доповненням і розширенням функціоналу попередньої версії. Детальний аналіз версій gPXE виходить за межі даної статті, детально ознайомитись з перевагами та недоліками можна за таким посиланням [4, 9].

gPXE забезпечує початкове завантаження за допомогою одного двійкового файлу в декількох завантажувальних форматах. Виходячи з цього, існує декілька способів для клієнтської експлуатації LTSP-сервера:

1. Створити завантажувальний образ gPXE і записати його на дискету: Floppy bootable image (.dsk); SYSLINUX-based bootable floppy image (.sdsk). Розмір hex-файлу (*.dsk): 255 Кб.

2. Створити ISO-завантажувальний образ gPXE і записати його на зйомний носій інформації: ISO bootable image (.iso); ISO bootable image with legacy floppy emulation (.liso). Розмір hex-файлу (*.iso): 628 Кб.

3. Створити USB-образ gPXE: USB Keychain disk image (.usb). Розмір hex-файлу (*.usb): 271 Кб. Після запису цього програмного hex-коду на зовнішні накопичувачі потрібно налаштувати BIOS X-терміналу таким чином, щоб завантаження відбувалось з цих накопичувачів даних. Це можна виконати за допомогою BIOS-меню Boot Device Priority.

4. Підібрати файл програмного коду для подальшого перепрограмування додаткових мікросхем мережевих плат (прошивки) і перепрограмувати (прошити) додаткові мікросхеми мережевих плат і таким чином апаратно підключати клієнтські EOM до LTSP-сервера (причому клієнтські EOM можуть і не мати HDD, CD-ROM, FDD, оскільки вони стають непотрібними при використанні апаратно-програмного комплексу в такий спосіб): ROM binary (flashable) image (.rom); ROM binary (flashable) for problem PMM BIOS (.hrom). Розмір hex-файлу (*.rom): 5.6 Кб. Після запису цього програмного hex-коду на ПЗП-мікросхему потрібно налаштувати BIOS

X-терміналу таким чином, щоб завантаження відбувалось саме з неї. Це можна виконати за допомогою BIOS-меню Boot From Network.

Перші три способи під'єднання засновані на запису образу gPXE на різні носії інформації, при фізичному з'єднанні носія інформації з ЕОМ клієнта апаратно-програмний комплекс починає працювати з цим клієнтом. Використання четвертого способу під'єднання до LTSP-сервера значно зменшить апаратне навантаження на ЕОМ клієнта (X-термінали). Даний спосіб дозволяє відмовитись від ряду додаткових апаратних пристроїв (CD-ROM, Floppy Disk Drive тощо) клієнтських робочих станцій. Важливою умовою формування коректного файлу-образу для програмування ПЗП-мікросхеми є доповнення цього файлу нульовим байтом '0x00' до точного значення об'єму ПЗП цієї мікросхеми. Наприклад, розмір hex-файлу *.rom у версії gPXE 0.9.8 складає 56320 байт, об'єм ПЗП мікросхеми W27C512 складає 65536 байт, отже hex-файл *.rom потрібно доповнити додатковими 9216 байтами '0x00', що можна виконати скориставшись такою командою:

```
tr '0'\377'</dev/zero|ddbs=1conv=notrunc of=098.ROM seek=56296 count=9216.
```

Вибір способу під'єднання безпосередньо лежить на адміністраторі системи. Спосіб експлуатування LTSP-сервера № 4 є найбільш вдалим, оскільки потребує найменшої кількості апаратних засобів X-терміналів, а отже, полегшення адміністрування клієнтів усього апаратно-програмного комплексу, і зменшує енергоспоживання X-терміналів.

Деякі версії BIOS не підтримують завантаження по LAN, але за допомогою утиліти FLASHROM [2] можна виконувати і перепрограмування BIOS будь-якої версії. Процес перепрограмування BIOS без використання будь-якого спеціалізованого обладнання наведено нижче. Як приклад було протестовано материнську плату GIGABYTE GA-965P-DQ6 rev2.0. Для перепрограмування BIOS в ОС LINUX слід застосувати програмний hex-код (прошивка) BIOS і програмне забезпечення, яке буде цей BIOS перепрограмувати (програма FLASHROM).

Нижче наведено фрагмент виводу команди *flashrom*:

```
Found chip "Macronix MX25L8005" (1024 KB) at physical address 0xffff00000;  
Found chip "Macronix unknown Macronix SPI chip" (0 KB) at physical address 0x0;  
Multiple flash chips were detected: MX25L8005 unknown Macronix SPI chip.
```

Перед записом нової прошивки в BIOS бажано зберегти стару версію, тобто скопіювати прошивку, якою вже запрограмована BIOS, щоб потім можна було все відновити. Виконати це можна за допомогою такої команди:

```
flashrom -r OldBIOS.bin -c MX25L8005,
```

де *OldBIOS.bin* — це ім'я файлу, в якому збережена стара версія прошивки BIOS, а *MX25L8005* — обраний чіпсет материнської плати.

Після збереження можна перепрограмувати BIOS заздалегідь підготовленим програмним кодом за допомогою такої команди:

```
flashrom -w NewBIOS.bin -c MX25L8005,
```

де *NewBIOS.bin* — це ім'я файлу, в якому збережена нова версія прошивки BIOS.

На цьому процес перепрограмування BIOS є завершеним. Процес програмування ПЗП-мікросхем принципово не відрізняється від процесу перепрограмування BIOS, цей процес детально описано в науковій публікації [2].

Як доповнення до наукової публікації [1] не зайвим є опис принципу роботи (завантаження) LTSP-сервера. На LTSP-сервері в середовищі chroot [1, 2] підготовляється мінімальна операційна система на базі Linux і X Window System [1].

Завантаження мінімального оточення: або з жорсткого диску / USB-диску CD-ROM комп'ютер завантажує ядро Linux, яке ініціалізує обладнання; або, у разі тонкого клієнта, використовується завантаження по мережі з PXE-частиною прошивки мережевої карти, який за протоколом DHCP отримує свою IP-адресу та адресу завантажувального сервера (LTSP-сервер). Завантажувач отримує ядро та initrd [3] за протоколом TFTP[3] з LTSP-сервера. Ядро заново отримує свою IP-адресу та адресу сервера, з якого можна підключити кореневу файлову систему (підготовлене заздалегідь у chroot), після чого монтує його за протоколом Network File System (NFS) або Network Block Device (NBD) [8, 9].

Завантажена система запускає графічну систему X Window System і XDMCP [8]. Починаючи з п'ятої версії LTSP, клієнт спочатку встановлює SSH-тунель до графічного оточення LTSP-сервера, через який локально запускає LDM (LTSP Display Manager) [8]. З цього моменту програми запускаються на LTSP-сервері, а відображаються й управляються й на клієнті.

Графічно процес завантаження LTSP-сервера зображено на рис. 1.

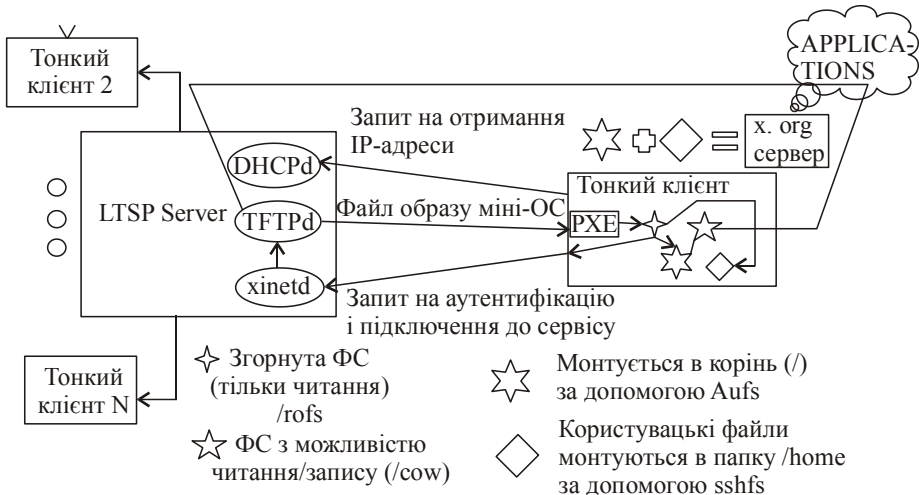


Рис. 1. Графічне зображення принципу роботи LTSP-сервера

Процедура завантаження LTSP-клієнта передбачає успішне виконання двох послідовних обов'язкових DHCP-запитів. Використовуючи значення DHCP-параметра як шлюзу для з'єднання з віддаленими мережами, X-клієнт з'єднується з сервером за протоколом TFTP [8] і завдяки DHCP-параметрам завантажує в оперативну пам'ять файл-образ *tftp://ip-адреса_сервера/ltsp/i386/pxelinux.0*.

Завантаження файлу-образу *pxelinux.0* триває приблизно 5 секунд. Одразу після завершення завантаження цей файл миттєво перетворюється у віртуальний RAM-диск ємністю приблизно 10 Мбайт. На другому етапі завантаження

LTSP-клієнта його мережева плата виконує другий DHCP-запит і реєструється в локальній мережі з тією ж IP-адресою діапазону, що й на першому етапі, але тепер система під'єднується до сервера за протоколом NFS [8] і монтує до своєї файлової системи спільну NFS-папку */opt/ltsp/i386*, вказану в DHCP-параметрі.

Якщо DHCP-сервер налаштований в UNIX/Linux — операційній системі, то параметри служби DHCP повинні бути вказані в конфігураційному файлі DHCP-сервера */etc/dhcp/dhcp.conf* і виглядати таким чином:

```
authoritative;  
subnet 7.7.7.0 netmask 255.255.255.0 {  
  range 7.7.7.2 7.7.7.254;  
  next-server 5.5.5.6;  
  filename "/ltsp/i386/pxelinux.0";  
  option routers 7.7.7.1;  
  option root-path "/opt/ltsp/i386";  
  option subnet-mask 255.255.255.0;  
  option broadcast-address 7.7.7.255;};
```

Експериментально встановлено, що файл-образ *pxelinux.0* підтримується обома сучасними протоколами віддаленого доступу — *Etherboot* та *gPXE* і, навпаки, файл-образ *nbi.img* [1] жодним з перелічених протоколів не підтримується, а отже, умовний оператор *IF*, встановлений за замовчуванням у конфігураційному файлі */etc/dhcp/dhcp.conf* програмою LTSP можна опустити і таким чином істотно спростити налаштування DHCP-сервера в UNIX/Linux операційній системі.

Варто зазначити, що в науковій публікації [1] використовувалась оболонка *xfce* як графічна оболонка LTSP-серверу. Весь створений апаратно-програмний комплекс на базі технології LTSP-сервер і Oracle VirtualBox використовувався для різнопланових робочих місць (тобто не було обрано єдиної ОС для вирішення певного класу задач автоматизованих робочих місць) і вибір графічної оболонки припав саме на оболонку *xfce* через те, що дане середовище робочого столу є одним з «найлегших», «найшвидших» і «найневигодовіших» до ресурсів графічних оболонок, що повністю задовольняє потреби бездискових робочих станцій запускати власні віртуальні ОС. Якщо ж потреби підприємства в LTSP-сервері будуть більш конкретизованими, то системний адміністратор підприємства може встановити вже конкретну графічну оболонку *chroot*-оточення для більшості автоматизованих робочих місць підприємства (наприклад, KDE SC, GNOME, Unity, LXDE, Windows-подібну графічну оболонку *XPde*, тощо [3]). Зробити це можна за допомогою команди: *apt-get install назва_графічної_оболнки*

Якщо ж апаратно-програмний комплекс планується використовувати для різнопланових задач, то пропонується використовувати *bash*-сценарії для полегшення адміністрування віртуальних машин (у *bash*-сценаріях використовувати стандартні команди для роботи з файлами на кшталт *cp*, *mv*, *rm* тощо та/або розробити власну систему електронного аудиту операційних систем із залученням СУБД).

Висновки

Перетворення ЕОМ автоматизованих робочих місць підприємства на Х-термінали дає змогу суттєво скоротити витрати підприємства на оновлення та налаштування ПЗ, на виявлення та знешкодження вірусного, шпигунського і шкідливого ПЗ, оновлення апаратного забезпечення й обслуговування ЕОМ АРМ-ів, аудиту та контролю робочого часу працівників тощо, адже бездисковий тонкий клієнт лише відтворює всі дії користувача, які опрацьовує LTSP-сервер. Окрім цього, перехід на термінальні сервери дозволяє значно спростити й апаратно зекономити на процедурі резервного копіювання і відновлення даних, розв'язати проблему контролюваного використання телекомунікаційних технологій підприємства.

Кожен Х-термінал має бути оснащеним gPXE-клієнтом. Якщо материнська плата Х-клієнта не підтримує завантаження по мережі, то ЕОМ тонкого користувача потрібно обладнати середовищем gPXE одним із способів, вказаним у цій публікації. Вибір способу залежить від системного адміністратора підприємства, який враховує особливості автоматизованих робочих місць підприємства. В статті наведено низку способів оснащення Х-терміналів середовищем gPXE.

Віртуальні ОС зберігаються у вигляді файлів, з якими досить зручно проводити стандартні маніпуляції (копіювання/видалення). За допомогою операції копіювання можна різко скоротити час на розповсюдження й адміністрування клієнтських ОС, при виведенні з ладу певної ОС системи проблема вирішується шляхом видалення файлу та заміни його новим файлом з ОС.

Література

1. *Загороднюк С.П.* Організація робочих місць в режимі бездискових робочих станцій на основі програми LTSP. Налаштування сервера, тестування сумісності і швидкодії / С.П. Загороднюк, Я.О. Шевченко, О.С. Баужа, А.Г. Донець // Науковий журнал «Інженерія програмного забезпечення» (Software Engineering). — Київ, 2015. — № 2, (22). — ISSN 2306-6512.
2. *Загороднюк С.П.* Організація робочих місць в режимі бездискових робочих станцій. Налаштування робочих станцій на підтримку протоколу мережевого завантаження gPXE / С.П. Загороднюк, О.С. Баужа, Я.О. Шевченко, А.Г. Донець // Науковий журнал «Інженерія програмного забезпечення» (Software Engineering). — Київ, 2015. — № 3, (23). — ISSN 2306-6512.
3. *Стахнов А.А.* Сетевое администрирование Linux / А.А. Стахнов. — Санкт-Петербург: BHV-Петербург, 2004. — 480 с.
4. *Глобальна веб-програма для генерації файлів-образів з програмним кодом протоколу PXE* [Електронний ресурс]. — Режим доступу: <http://rom-o-matic.net/>.
5. *Michael M.* System V Application Binary Interface AMD64 Architecture Processor Supplement / [Електронний ресурс] / М. Michael, J. Hubicka, A. Jaeger, M. Mitchell. — Режим доступу: http://www.x86-64.org/documentation_folder/abi.pdf.
6. *Маркелов А.* Использование бездисковых Linux-станций с загрузкой по сети [Електронний ресурс] / А. Маркелов // Проект «Линукс-Центр». — Режим доступу: <http://www.linuxcenter.ru/lib/articles/networking/thinstation.phtml>.
7. *Surhone L.M.* GPXE / L.M. Surhone, M.T. Tennoe, S.F. Henssonow. — Betascript Publishing, 2010. — 68 p.
8. *Russell J.* OpenThinClient / J. Russell, R. Cohn. — Bookvika Publishing, 2012. — 100 p.

СОВЕРШЕНСТВОВАНИЕ СЕТЕВЫХ КОМПОНЕНТОВ ТОНКОГО КЛИЕНТА

Я.А. Шевченко

Киевский национальный университет имени Тараса Шевченко

А.А. Мошенский

Национальный университет пищевых технологий

В статье рассмотрены технологические аспекты настройки аппаратно-программного комплекса, сформированного на базе операционной системы Linux Debian x86 с установленным терминальным сервером LTSP, и приведены рекомендации по аппаратному обеспечению терминального сервера. Продемонстрирован ряд способов для совершенствования бездисковых ЭО после оборудования их поддержкой среды удаленной загрузки операционных систем по сети (gPXЕ) для одновременного соединения бездисковых рабочих станций с терминальным сервером LTSP. Сформулированы рекомендации по оборудованию тонких клиентов средой gPXЕ, обращено внимание на особенности настройки сетевой инфраструктуры для функционирования аппаратно-программного комплекса и усовершенствовано администрирование виртуальных машин аппаратно-программного комплекса в виде bash-сценариев.

Ключевые слова: *бездисковая рабочая станция, тонкий клиент, LAN, модификация, gPXЕ, BIOS, LTSP, виртуальная машина, bash-сценарии.*