

Тонкий клиент - шаг к "мэйнфреймам"?

Когда-то терминалы подключались к мощным серверам и обработка информации происходила на мэйнфреймах. Затем появились достаточно мощные ПЭВМ и информация стала обрабатываться на рабочих местах. Похоже история любит повторяться, но сегодня уже в зеркальном отражении - становится опять экономически выгодно использовать "мэйнфреймы" с подключением легких терминалов - тонких клиентов.

Как показывает практика, чем сложнее становится система, тем больше усилий требуется на поддержание ее в рабочем состоянии. В то же время, становится очевидной специализация тех или иных бизнес-процессов. Можно уже четко определить, какие конкретно ресурсы требуются определенным пользователям для решения их бизнес-задач. Исходя из указанных предпосылок данным пользователям следует выделить необходимый конкретно им инструментарий, но не более того. Кроме этого, установку (а в дальнейшем и обновление) пакетов прикладных программ, хранение электронной документации также рациональнее вести в одном месте - на производительном терминальном сервере. Таким образом, проведя анализ потоков данных и выяснив структуру документооборота, предприятие может попробовать преобразовать структуру рабочих мест.

Из названия следует, что ориентир сделан на создание тонких клиентов, т.е. создание ПЭВМ, которые отображают выполнение программного обеспечения на серверной стороне, при этом, будучи клиентской частью, могут быть совершенно облегченными. Под облегченностью понимается как использование старой техники (класса первых Pentium), так и современных ПЭВМ с интегрированными сетевыми-, видео-контроллерами, но со скромной производительностью, так и специализированных терминалов [1]. С одной стороны мы увеличиваем срок службы старой техники, перенося вычислительные функции на производительный выделенный сервер приложений (я все-таки надеюсь, что у вас он уже функционирует), с другой стороны путем унификации терминалов, можем снизить время простоя отдельно взятого терминала и снизить стоимость обслуживания вычислительной техники - модный за последнее время термин TCO (Total Cost of Ownership) [2]. В свое время TCO-анализ был предложен компанией Gartner Group для оценки прямых [3] и косвенных затрат от приобретения программного и аппаратного обеспечения ЭВМ. В TCO-анализе учитывается не только стоимость обеспечения, но и, например, поддержка в дальнейшем пользователей, и их обучение [4].

Итак, у вас есть старый парк техники, который вы не считаете рациональным выводить из применения, но в то же время отдаете себе отчет, что достичь приемлимых результатов на такой технике уже нельзя или вы организуете рабочие места с чистого листа и решаете для себя вопрос, а как грамотно провести внедрение бизнес-процессов на рабочих местах. Вполне возможно, что после сегодняшнего материала некоторые вопросы для вас исчезнут.

Насколько вы помните в предыдущей публикации [5] мы использовали так называемую barebone-систему на аппаратной платформе VIA EPIA-M. Помимо вполне удовлетворительной производительности на плате интегрированы сетевая и звуковая платы. Данный факт позволяет построить на базе EPIA-M бездисковые терминальные станции, где загрузка операционной системы происходит по сетевому интерфейсу. Одним из довольно любопытных проектов является ThinStation [6]. Его корни уходят в такие проекты как NetStation, PXES. Заимствовав идею сетевой загрузки проект постепенно вырос до сегодняшнего состояния и в определенной степени представляет из себя вполне законченный продукт.

Давайте посмотрим, что же из себя представляет данное творение. Вы уже могли быть с ним знакомы по публикации в журнале за прошлый год [7]. Пакет системных программ busybox (объединяющий в одном флаконе упрощенный вариант bash, awk, cat, chmod, grep и ряда других), серия драйверов для различных сетевых карт, пакет прикладных программ таких как, клиенты rdesktop, telnet, а также серия X-серверов составляют основу "тонкой станции". Дальнейшая сборка позволяет получить образ в виде файлов .nbi (etherboot-загрузка), initrd, .iso, pxelinux, syslinux.

В частности, связка из `initrd` и `pxelinux` позволяет создать сетевые бездисковые терминалы. Помимо указанных пакетов, есть возможность доукомплектовать загрузочный образ такими пакетами как `vncviewer`, `tarantella`, `xnest`, `dillo`, `nx`, `Firefox`, `MPlayer` и рядом других. Дистрибутив `ThinStation` разделяется на часть для конечного пользователя (40 Мб) и на часть, ориентированную, прежде всего, на разработчика (400 Мб). В первой части все программные пакеты уже скомпилированы и задача пользователя заключается в том, чтобы выбрать те пакеты, которые ему потребуются для работы. Вторая часть – это те же самые пакеты, только в исходных текстах. Добавление своей собственной, предварительно скомпилированной, программы в дистрибутив `ThinStation` для конечного пользователя скорее всего закончится неудачно. С чем это связано? Прежде всего, различие в версиях библиотеки `GLIBC`, используемой в дистрибутиве (там версия `GLIBC 2.1.3`) и вашей. Если программа скомпилирована полностью статически, то проблем в принципе не должно возникнуть. Вариант для разработчика предназначен для тех случаев, когда добавление в лоб собственного пакета не срабатывает.

Коммерческие продукты сами не входят в состав дистрибутива, а представлены в виде ссылок. Поэтому, когда вы выбираете клиентов, например, `NoMachine`, `Citrix` или `Tarantella`, то по мере сборки загрузочного образа они будут скачаны `wget`'ом с сайтов производителей.

Нам сегодня потребуется дистрибутив `ThinStation` для конечного пользователя [8]. Но перед тем, как начнем сборку загрузочного образа, давайте добьемся, чтобы ПЭВМ могла загружаться по сети.

Добавляем LAN ROM в системный BIOS

В случае, если вы счастливый обладатель старых ПЭВМ и/или сетевых карт без дополнительного слота с микросхемой, содержащей PXE-код, то единственный способ для добавления функции сетевой загрузки – это дополнить системный BIOS материнской платы кодом для сетевой платы. Зачем именно изменять системный BIOS, а не использовать загрузку, например, с `HGMD`? Во-первых, для безопасности. Хотя бездисковые терминалы и не имеют жестких дисков, однако с дискеты возможно запустить более-менее работоспособную версию ОС, чтобы отследить трафик, передаваемый по сети. Возможные последствия? Например, два одинаковых `ip`-адреса в `LBC`. Во-вторых. Подготовка загрузочных дискет, с определенным набором драйверов, не самое лучшее времяпровождение, если есть вариант лучше, а именно, интеграция PXE-кода в системный BIOS. И в третьих, подумайте о людях – если вы сэкономили на нормальной сетевой карте, на более-менее современном терминале, на дисплее, то задумайтесь, а имеете ли вы вообще право требовать с них повышения производительности?

Небольшое отступление – не следует забывать, что обновление BIOS возможно в том случае, если в материнской плате используется Flash-микросхема. Они начали широко применяться в `Pentium`-системах. В более ранних, например на базе 80486 ПЭВМ, использовались как flash-микросхемы, так и однократной записи. В еще более древних (80386) использовались однократные. Подумайте, а будут ли вообще работать такие ПЭВМ с `ThinStation` – ведь существуют вполне определенные минимальные требования для запуска тонкого клиента:

Частота ЦП – x86 100МГц

Размер ОЗУ – 16МБ

Размер видео-памяти – не менее 1МБ

Более подробно, как модифицировать BIOS см. [9]. А сейчас предварительно узнайте, как точно называется ваша сетевая плата, затем на сайте [10] забирайте PXE ROM-код и с помощью утилит `amiflash` [11], `cbrom` производите следующие манипуляции:

```
FLASH879 /SBIOS.BIN
```

Сохранили в файл `BIOS.BIN` содержимое flash-памяти.

В определенных случаях, места в BIOS может не хватить для добавления нового кода. Поэтому давайте узнаем, из каких составных частей состоит ваш BIOS, прежде чем добавить PXE-код.

CBROM.EXE BIOS.BIN /D

CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

***** BIOS.BIN BIOS component *****

No.	Item-Name	Original-Size	Compressed-Size	Original-File-Name
0.	System BIOS	20000h(128.00K)	13E32h(79.55K)	test.BIN
1.	XGROUP CODE	0EE90h(59.64K)	0A362h(40.85K)	awardext.rom
2.	ACPI table	04F6Fh(19.86K)	01A60h(6.59K)	ACPITBL.BIN
3.	EPA LOGO	0168Ch(5.64K)	0030Dh(0.76K)	AwardBmp.bmp
4.	YGROUP ROM	06B50h(26.83K)	04693h(17.64K)	awardeyt.rom
5.	GROUP ROM[0]	06190h(24.39K)	02765h(9.85K)	_EN_CODE.BIN
6.	PCI ROM[A]	0C000h(48.00K)	06431h(25.05K)	NVRAID.ROM
7.	OEM1 CODE	02410h(9.02K)	00DF3h(3.49K)	BSMICODE.ROM
8.	OEM0 CODE	01F26h(7.79K)	01787h(5.88K)	SBF.BIN

Total compress code space = 34000h(208.00K)

Total compressed code size = 2F6A4h(189.66K)

Remain compress code space = 0495Ch(18.34K)

** Micro Code Information **

Update ID	CPUID	Update ID	CPUID	Update ID	CPUID	Update ID	CPUID
-----------	-------	-----------	-------	-----------	-------	-----------	-------

Очевидно, что для работы тонкого клиента код, отвечающий за работу RAID-подсистемы, в принципе не нужен (выделен жирным шрифтом). Предлагаю его удалить, а на освободившееся место поместить нужный нам PXE-код сетевой карты.

CBROM.EXE BIOS.BIN /PCI release

CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

PCI ROM - - - [A] : NVRAID.ROM

[PCI-A] ROM is release

Удалили ненужный компонент.

CBROM.EXE BIOS.BIN /PCI netboot.rom

CBROMV2.15 (C)Award Software 2001 All Rights Reserved.

Adding netboot.rom 68.5%

Добавили PXE-код от сетевой карты.

И еще раз посмотрим на структуру BIOS.

CBROM.EXE BIOS.BIN /D

CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

***** BIOS.BIN BIOS component *****

No.	Item-Name	Original-Size	Compressed-Size	Original-File-Name
0.	System BIOS	20000h(128.00K)	13E32h(79.55K)	test.BIN
1.	XGROUP CODE	0EE90h(59.64K)	0A362h(40.85K)	awardext.rom
2.	ACPI table	04F6Fh(19.86K)	01A60h(6.59K)	ACPITBL.BIN
3.	EPA LOGO	0168Ch(5.64K)	0030Dh(0.76K)	AwardBmp.bmp
4.	YGROUP ROM	06B50h(26.83K)	04693h(17.64K)	awardeyt.rom
5.	GROUP ROM[0]	06190h(24.39K)	02765h(9.85K)	_EN_CODE.BIN
6.	OEM1 CODE	02410h(9.02K)	00DF3h(3.49K)	BSMICODE.ROM
7.	OEM0 CODE	01F26h(7.79K)	01787h(5.88K)	SBF.BIN
8.	PCI ROM[A]	08000h(32.00K)	057E2h(21.97K)	netboot.rom

Total compress code space = 34000h(208.00K)
Total compressed code size = 2EA55h(186.58K)
Remain compress code space = 055ABh(21.42K)

**** Micro Code Information ****

Update ID	CPUID	Update ID	CPUID	Update ID	CPUID	Update ID	CPUID
-----+-----+-----+-----+-----+-----+-----+-----							

Теперь остается записать BIOS во flash-память.

FLASH879 BIOS.ROM /A /B /E /R

Опции, которые используются в примере, обозначают:

/R - произвести перезагрузку, после операции записи во flash-память
/E - очистить CMOS-память во избежание возможных недоразумений после перезаписи
/B - также произвести перезапись секции BootBlock во flash-памяти (опционально)
/A - не запускать GUI-режим, а работать в командной строке

Сохранять/изменять системный BIOS можно также утилитами: awdf flash [12], uniflash[13], phlash[14] или другими. Добавлять код от сетевой платы в системный BIOS нужно следующими утилитами:

cbrom - если используется AwardBIOS
amimm - если используется AMIBIOS
prepare - если используется PhoenixBIOS

Экзотику, такую как SystemSoftBIOS, DELLXBIOS, ACERBIOS, HPBIOS я не рассматриваю, так как по сути такие ПЭВМ встречаются достаточно редко, и утилит для изменения этих BIOS практически не существует для конечных пользователей.

Изменение flash-памяти достаточно непростая процедура, поэтому:

ПРЕДУПРЕЖДЕНИЕ О ПОТЕНЦИАЛЬНОЙ ОПАСНОСТИ РЕКОМЕНДУЕМЫХ ДЕЙСТВИЙ.

ВНИМАНИЕ! ОПИСАННЫЕ В ЖУРНАЛЕ ДЕЙСТВИЯ ПО ИЗМЕНЕНИЮ СОДЕРЖИМОГО FLASH-ПАМЯТИ МАТЕРИНСКОЙ ПЛАТЫ ПОТЕНЦИАЛЬНО ОПАСНЫ. ОТВЕТСТВЕННОСТЬ ЗА КОРРЕКТНОЕ И/ИЛИ НЕКОРРЕКТНОЕ МОДИФИЦИРОВАНИЕ BIOS ЛЕЖИТ ПОЛНОСТЬЮ НА ПЛЕЧАХ ВЛАДЕЛЬЦА ПЭВМ.

Использование BIOS Savior [5] и источника бесперебойного питания позволит избежать многих подводных камней и практически полностью избежать негативных последствий от пропадания питания на вашем рабочем месте по время перезаписи микросхемы flash-памяти.

После добавления сетевого кода в системный BIOS следует записать последний в микросхему Flash-памяти. Следует делать это осторожно, т.к. испортив и/или записав некорректно код вы на выходе получите неработающую материнскую плату. Лично для меня самым удобным вариантом является использование утилиты amiflash. С ее помощью можно и автоматически определить используемый чипсет материнской платы, а соответственно и не ломать голову, как называется микросхема flash-памяти, и проверить насколько правильно была произведена запись в данную микросхему. В целом, интерфейс достаточно симпатичный, а самое главное - утилита небольшая (умещается на системную дискету), для своей работы требует DOS и стабильно работает. Если по каким-либо причинам она не удовлетворяет вашим требованиям, то попробуйте использовать uniflash. Она также работает под DOS, размером несколько меньше amiflash, но по личным ощущениям является несколько перегруженной.

Записали обновленный BIOS в материнскую плату, перезагрузили ПЭВМ - работает. Если не работает - записываем во flash-память предварительно сохраненный, неизменный файл BIOS на программаторе. Хотя, при наличии BIOS Savior, такие проблемы не должны вас особо расстраивать - просто переключите на резервную микросхему и попробуйте еще раз. В BIOS Setup выбираете приоритетность

загрузки - ставите LAN ROM. Если нет пункта с упоминанием сетевой загрузки, то, скорее всего, конфигурация сетевого кода произойдет непосредственно после POST-процедур.

Однако, если вы формируете терминальные клиенты на основе VIA EPIA-M, то PXE-код уже интегрирован в составе системного BIOS. Нужно всего лишь выставить в BIOS Setup приоритетность загрузки ОС - начинать с загрузки по LAN.

Терминальные клиенты на VIA EPIA-M используются, например, в техническом университете Дании. Mike Eriksen использовал именно эти аппаратные решения, когда описал свой опыт [15]. Правда он реализовывал загрузку клиентов с твердотельных накопителей compact-flash.

Загружаем PXE-код

Будем считать, что сетевая плата на клиенте корректно работает и PXE-код посылает DHCP-запросы в окружающий эфир. Следует оправдать ожидания сетевой платы и обработать DHCP-пакеты с помощью службы dhcpd (более подробно, см. /usr/doc/Linux-HOWTOs/DHCP, /usr/doc/dnsmasq-2.8/FAQ). Для этого в загрузочный скрипт /etc/rc.d/rc.local добавляем строчку "/usr/sbin/dhcpd eth0", либо запускаем данный сервис в виде отдельной службы, например, через /etc/rc.d/rc.dhcpd. Или, в случае, когда для старта службы dhcpd используется скрипт /etc/init.d/rc.dhcpd, настроим запуск как "invoke.rc dhcpd start".

Не забудем также раскомментировать (если она, конечно же, закомментирована) в /etc/inetd.conf строчку, упоминающую сервис tftp. В конечном виде, строка выглядит следующим образом

```
tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
-vv
```

Для обладателей RedHat систем, где используется xinetd.conf, добавление выглядит примерно так:

```
service tftp
{
    socket_type          = dgram
    protocol             = udp
    wait                 = yes
    user                  = root
    server                = /usr/sbin/in.tftpd
    server_args           = -l -s /tftpboot
    disable               = no
}
```

Именно по протоколу tftp будет происходить загрузка файлов тонкого клиента ThinStation.

Чтобы суперсервер inetd перечитал свой конфигурационный файл, подадим следующую команду:

```
# killall -HUP inetd
```

Отредактируем файл /etc/dhcpd.conf или создадим его заново, в случае, если он отсутствует. У меня он выглядит примерно так:

```
ddns-update-style      none;

subnet 10.0.0.0 netmask 255.0.0.0 {
    option broadcast-address 10.0.255.255;
    max-lease-time 7200;
    default-lease-time 600;
}

host TS_EPIA_M_01 {
    hardware ethernet 00:40:63:d6:69:d7;
    fixed-address 10.0.0.50;
    filename "pxelinux.0";
}
```

```
}
```

Функционирование сервера dhcp возможно в двух режимах – либо с внесением изменений в DNS-сервер, либо без внесения таких изменений. Естественно, что подразумевается не прямое изменение DNS-записей, а путем отправки пакетов на DNS-сервер. В любом случае, динамическое изменение записей в DNS-сервере нем не требуется, поэтому выставляем глобальный параметр **ddns-update-style** как none.

Чтобы dhcp-сервер смог корректно обслуживать сеть 10.0.0.0 с маской 255.0.0.0 добавляем конструкцию **subnet {}**. Время ожидания (в секундах) запросов от PXE-клиентов выставляется опциями max-lease-time, default-lease-time. В конструкции **host {}** объявляется об узле TS_EPIA_M_01, с MAC-адресом 00:40:63:d6:69:d7, которому будет назначен ip-адрес 10.0.0.50. Для сетевой загрузки этого узла служит файл pxelinux.0, размещенный на tftp-сервере.

После таких приготовлений необходимо подготовить файлы, которые будут загружаться по сети. Итак, переходим непосредственно к приготовлению тонкого клиента – ThinStation.

Генерируем загрузочный образ

Я предлагаю остановить свой выбор на версии 2.1.3. Во-первых, эта версия является самой последней из стабильных. Во-вторых, используется хорошо зарекомендовавшее себя Linux-ядро линейки 2.4. В-третьих, и пожалуй, самая объективная причина – для ThinStation 2.1.3 скомпилировано самое большое количество приложений, среди которых Mozilla Firefox, Sylpheed, MPlayer.

```
$ wget Thinstation-2.1.3.tar.gz
$ tar xzvf Thinstation-2.1.3.tar.gz
$ cd Thinstation-2.1.3
```

Конфигурация для сборки тонкого клиента представлена файлом build.conf. В принципе, для любой x86-системы можно подготовить рабочий вариант. Количество поддерживаемых сетевых карт более 20, включены драйверы для более 15 звуковых карт. Видеоподсистема поддерживается для более 20 разновидностей видео-карт. Названия модулей говорят сами за себя: xorg6-i740, xorg6-i810, xorg6-savage, xorg6-via, xorg6-vesa. В любом случае, даже если специализированный драйвер не подошел у вас остается возможность использовать VGA- или VESA-драйвер.

Список доступных прикладных пакетов указан ниже:

```
rdesktop - RDP-клиент для подключения к терминальным серверам на основе Microsoft Windows
rdesktop_svga - тот же самый клиент, используемый в случае, когда на терминальной машине экстремально мало ОЗУ
vncviewer - VNC-клиент
ica - Citrix-клиент
xnest - XDM-клиент
rxvt - облегченный xterm-клиент
ssh - клиент Secure Shell
tn5250 - терминальный эмулятор 5250
blackbox - оконный менеджер
xtdesk - приложение для обработки пиктограмм на рабочем столе
icewm - еще один легкий оконный менеджер
lp_server - сервис удаленной печати (с поддержкой JetDirect)
tftpd - встроенный tftp-сервер
samba-server - smb-сервер
```

И завершается данный конфигурационный файл списком параметров, включая пароли для vncviewer, пароль для доступа к терминальной станции по сети, а также разрешение для графической подсистемы и некоторые другие.

В качестве примера рассмотрим мой вариант build.conf

```
1 module agpgart
```

```

2 module 3c59x
3 module 8139too
4 module e100
5 module via-rhine

6 module usb-hid
7 module usb-storage

8 module smbfs

9 package xorg6-vesa

10 package keymaps-en_us

11 package rdesktop
12 package vncviewer
13 package rxvt
14 package xterm
15 package ssh

16 package blackbox
17 package xtdesk

18 param rootpasswd      root
19 param xorgvncpasswd    vnc

20 param bootlogo         true
21 param bootresolution 1024x768
22 param bootverbosity    7

```

Строчки 1-5 отвечают (нумерация строк приведена для удобства обсуждения, в build.conf ее естественно быть не должно) за включение в состав формируемого пакета сетевых драйверов для указанных карт. Строка 6 означает, что USB-клавиатура и USB-мышь, будучи подключены к терминалу, смогут функционировать. В строке 7 в формирующийся пакет подключается модуль для работы с usb-брелками. Строка 8 подключает модуль, отвечающий за работу файловой системы SMB.

Для подключения к Citrix-серверам используется загрузка бинарного клиента с сайта Citrix. В файле build.conf он фигурирует в такой форме:

```

package ica
param icaurl \
http://download2.citrix.com/files/en/products/client/ica/current/linuxx86.tar.gz

```

Если вы будете использовать Citrix-клиент, то он будет загружен из сети с указанного сайта.

Чтобы гарантированно запустить графический режим на терминальных клиентах я использовал универсальный VESA-драйвер.

Клавиатурная раскладка используется английская. Есть возможность добавить и нашу национальную раскладку, раскомментировав в вашем файле конфигурации thinstation.conf.network соответствующий пункт с русской раскладкой.

```
KEYBOARD_MAP=ru
```

В строках 11-15 я подключаю необходимые для демонстрационной работы утилиты. Следующие две строки означают, что оконным менеджером будет blackbox, а чтобы на рабочем столе появились ярлыки - отвечает пакет xtdesk.

В строках 18-19 выставляем пароли для доступа к данной терминальной станции и для доступа с терминальной станции на VNC-сервер.

По умолчанию, картинка проекта ThinStation включена во время загрузки - см. строку 20.

Разрешение X-сервера на стороне терминального клиента выставлено в 1024x768, см. строчку 21. И, наконец, в 22 строчке указано, что требуется отображать сообщения, относящиеся к процессу загрузки, инициализации и за начальную настройку сетевых конфигурационных параметров.

Далее запускаем сборку:

```
$ ./build
```

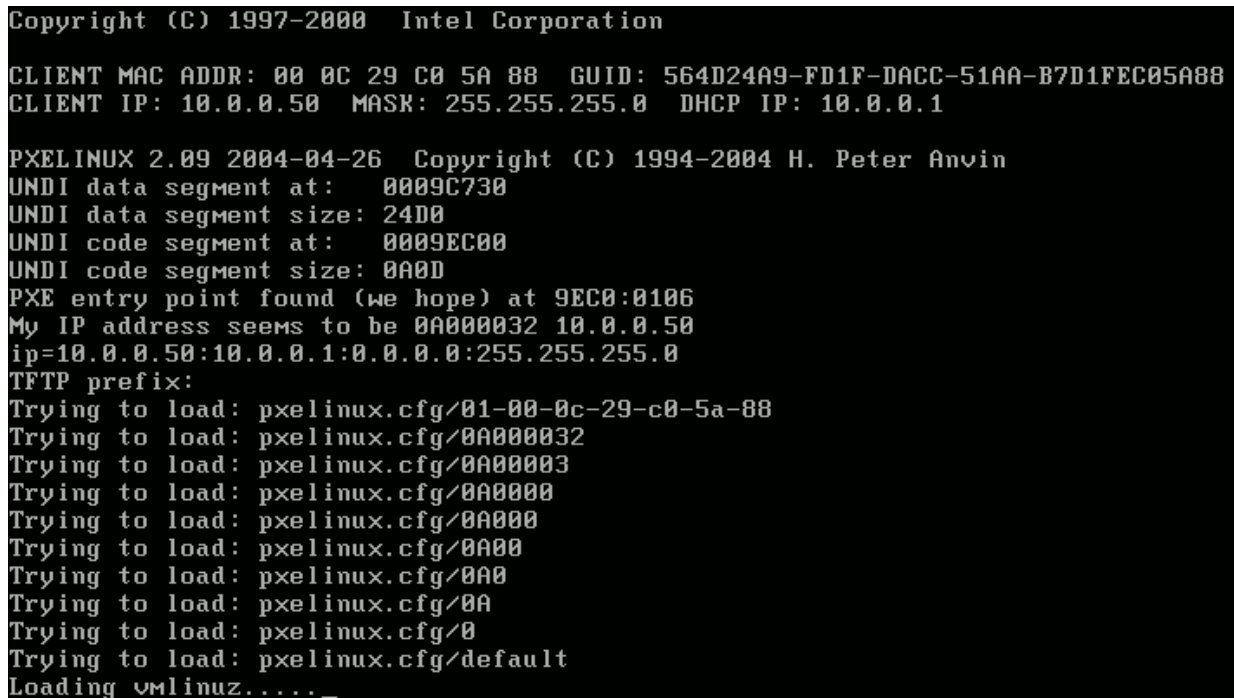
И переписываем приготовленные файлы в каталог tftp-сервера

```
$ su -  
# cp -R ./boot-images/pxe /tftpboot
```

В принципе, на этом этапе можно запускать терминальный клиент и смотреть, как происходит процесс загрузки.

В качестве визуального примера я привожу, как процесс загрузки происходит в виртуальной машине.

С момента включения VIA EPIA-M до момента определения ip-адреса по dhcp-протоколу прошло 25 секунд.



```
Copyright (C) 1997-2000 Intel Corporation  
  
CLIENT MAC ADDR: 00 0C 29 C0 5A 88 GUID: 564D24A9-FD1F-DACC-51AA-B7D1FEC05A88  
CLIENT IP: 10.0.0.50 MASK: 255.255.255.0 DHCP IP: 10.0.0.1  
  
PXELINUX 2.09 2004-04-26 Copyright (C) 1994-2004 H. Peter Anvin  
UNDI data segment at: 0009C730  
UNDI data segment size: 24D0  
UNDI code segment at: 0009EC00  
UNDI code segment size: 0A0D  
PXE entry point found (we hope) at 9EC0:0106  
My IP address seems to be 0A000032 10.0.0.50  
ip=10.0.0.50:10.0.0.1:0.0.0.0:255.255.255.0  
TFTP prefix:  
Trying to load: pxelinux.cfg/01-00-0c-29-c0-5a-88  
Trying to load: pxelinux.cfg/0A000032  
Trying to load: pxelinux.cfg/0A000003  
Trying to load: pxelinux.cfg/0A000000  
Trying to load: pxelinux.cfg/0A0000  
Trying to load: pxelinux.cfg/0A00  
Trying to load: pxelinux.cfg/0A0  
Trying to load: pxelinux.cfg/0A  
Trying to load: pxelinux.cfg/0  
Trying to load: pxelinux.cfg/default  
Loading vmlinuz.....
```

Рис.1 - IP-адрес получен от DHCP-сервера и происходит загрузка ядра Linux

Еще 25 секунд требуется на распаковку образа в памяти терминального узла, настройку служб и рабочего стола.

Щелкаем на пиктограмму RDP-клиента и вводим адрес Windows Terminal Server:

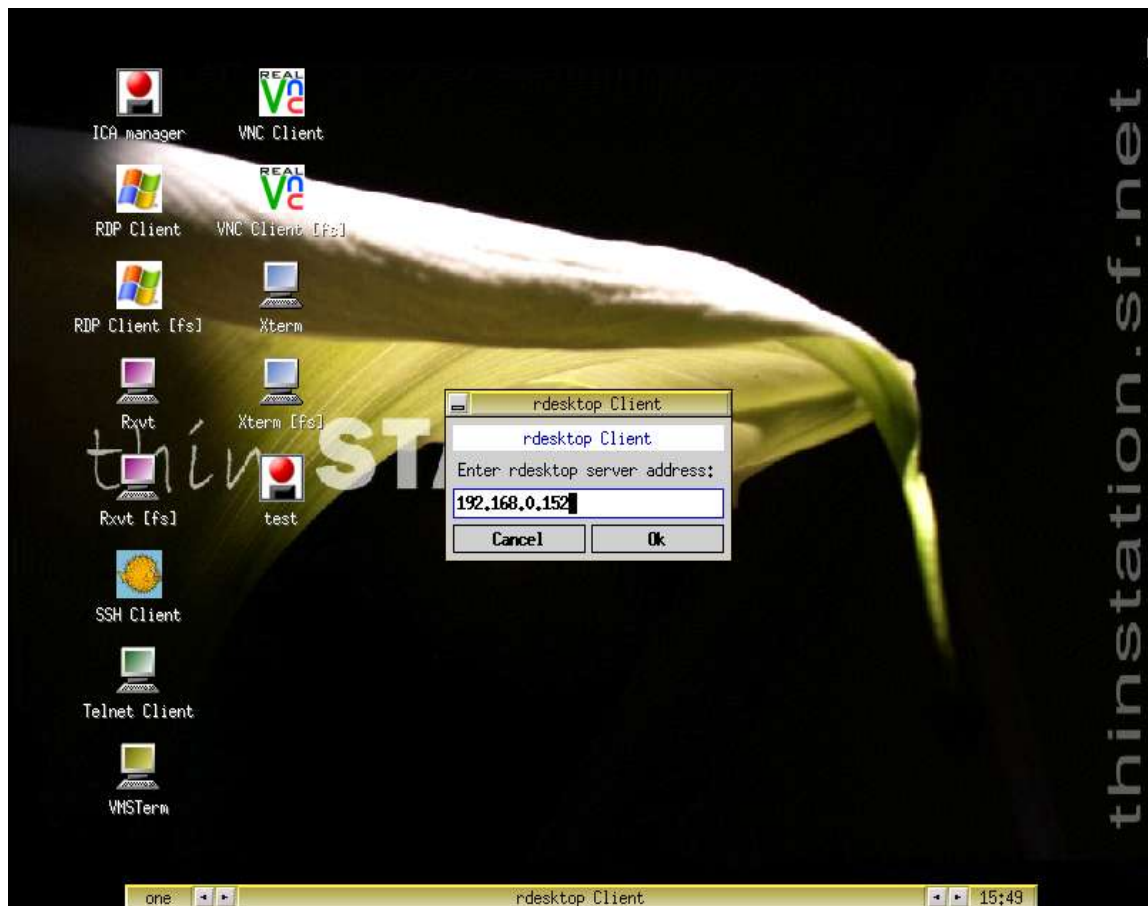


Рис. 2 - Указываем расположение сервера приложений

А также имя пользователя и пароль.



Рис.3 - Для ввода используется русский язык

И присоединяемся к Windows-серверу:

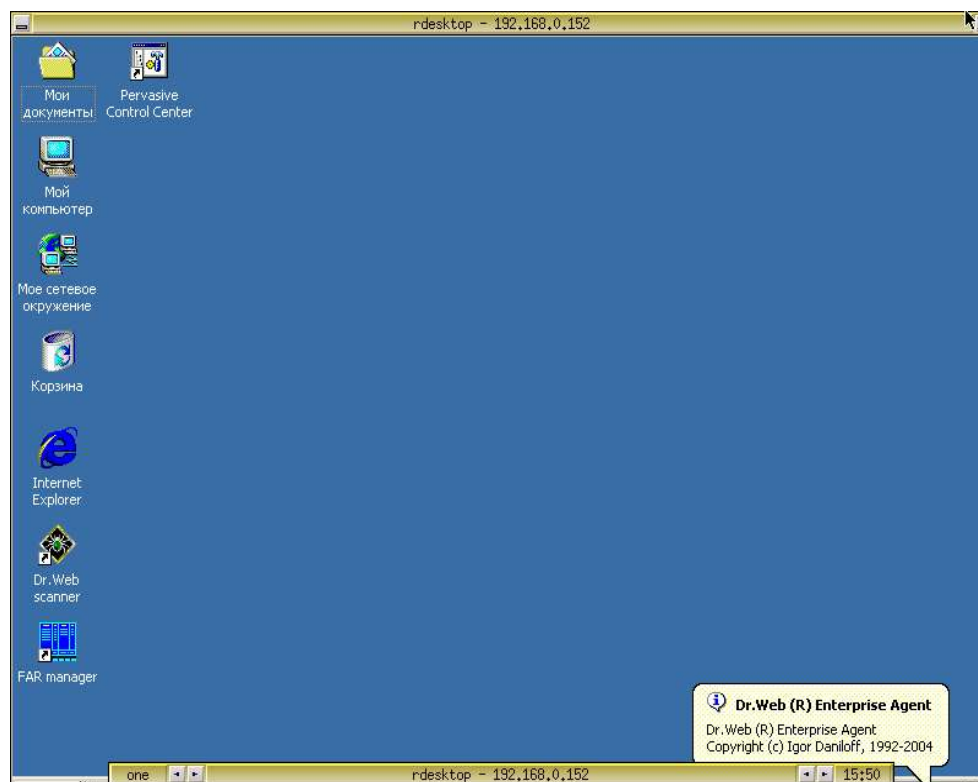


Рис.4 – Экран удаленного сервера

Небольшая особенность работы RDP-клиента в версии ThinStation 2.1.3 – если вы подключите поддержку звука на терминалах, где используется звуковой драйвер i810, то воспроизведение звука будет ускоренным. Проблема, насколько я помню, заключалась в драйвере i810 и решена в новой серии ThinStation 2.2.

Что касается хранимых профилей пользователей, в случае, если надумаете включать в загружаемый образ браузер Mozilla Firefox, то хочу обратить ваше внимание на секцию STORAGE в конфигурационном файле `thinstation.conf.network` (он должен быть расположен на tftp-сервере, как и загружаемый образ). В частности, опции `STORAGE_PATH`, `STORAGE_SERVER` и `STORAGE_PREFIX` отвечают за формирование полного пути к сохраняемому профилю. Профиль может быть сохранен либо по NFS, либо по SMB протоколу.

Также файл `thinstation.conf.network` предназначен для более тонкой настройки каждого конкретного клиента. Например, видеоразрешение для X-сервера задается именно там.

Аппаратные альтернативы

В случае, если у вас за спиной нет обоза старой техники и вы начинаете комплектовать предприятие, что говорится с нуля, то предлагаю ознакомиться с готовыми решениями, которые доступны на рынке на сегодняшний день.

Итак, начнем, пожалуй с тонких клиентов серии e100 от компании Neoware [17]. Данная серия представляет собой младшую линейку тонких клиентов. Загрузка происходит со встроенной карты compact-flash. Размеры варьируются от 32 до 512 Мб. Количество ОЗУ может быть от 128 до 512 Мб. Операционная система на клиентах может Linux, Windows CE, Windows XP. Поддержка протоколов RDP/ICA обязательна для любой модели – в случае, если используется Linux-терминал, применяется `rdesktop` и `ICAClient` под Linux, в случае Windows-терминала – встроенные клиенты `Remote Terminal Connection` и `Citrix Client` под Win32. Максимальное видеоразрешение одинаково для всех моделей данного ряда и составляет 1600x1200.

Название модели	Цена
Neoware e100 (Linux)	от 369 у.е.
Neoware e100 (Windows CE)	от 399 у.е.
Neoware e100 (Windows XPe)	от 599 у.е.

Таблица 1. Цены на тонкие клиенты от Neoware [18].

Если вы внимательно смотрите на названия приведенных моделей, то видите, что последний клиент основан на Windows XPe. Новая операционная система от Microsoft? Вовсе нет – это урезанная версия Windows XP для встраиваемых систем. Кстати говоря, у этой корпорации также есть планы на использование старой техники в качестве тонких клиентов [19].

Симпатично выглядят с эстетической точки зрения и тонкие клиенты от компании Hewlett-Packard. Функциональность у продукции от HP аналогична продукции от Neoware – похоже, что наличие поддержки протоколов RDP/ICA стало стандартом де-факто. Чем же привлекательны тонкие клиенты от HP, неужели ценой?

Название модели	Цена
HP Thin Client t5710	от 509 у.е.
HP Thin Client t5525	от 379 у.е.
HP Thin Client t5520	от 359 у.е.
HP Thin Client t5515	от 330 у.е.
HP Thin Client t5510	от 359 у.е.
HP Thin Client t5125	от 289 у.е.

Таблица 2. Цены на тонкие клиенты от HP [20].

Было отрадно также увидеть действительно сверхкомпактные терминалы с аппаратной начинкой, аналогичной что и у "старших" братьев [1].

На рынке тонких клиентов работают такие производители как Wyse Technology, Boca Research, Netier Technologies, Network Computer Devices, TeleVideo, IBM, Compaq, Dell, Fujitsu-Siemens, Unisys и другие.

Однако не стоит думать, что тонкие клиенты выпускаются только западными компаниями. Это не так. У нас тоже умеют собирать тонкие клиенты. Среди таких компаний фигурируют следующие имена – K-Systems, Nienshanz, AK-Systems, Kraftway. Цены на тонкие клиенты чуть-чуть ниже, нежели у зарубежных коллег.

Выводы

Ознакомившись с показанным решением я конечно же не агитирую вас пересматривать ситуацию, которая, как иногда утверждают старожилы "исторически сложилась" на рабочих местах в ту или иную сторону – оставить всё как есть или обратить внимание на новые возможности. В любом случае решение всегда остается за вами. Следует учесть, что ThinStation – это конструктор, у которого есть, естественно, и плюсы и минусы. Плюс – можно сконструировать практически любую конфигурацию для рабочей станции (в рамках терминального варианта). Минус, и достаточно жирный, заключается в том, что добавление собственного приложения в ThinStation требует, чтобы у разработчика было не менее 3Гб для распакованного архива и еще не менее 3Гб для собираемых пакетов. В общем – это отдельная история. Если вам требуется всего лишь посмотреть на ThinStation, то позволю себе смелость отослать вас на сайт [21], где вы можете выбрать на web-форме необходимые вам пакеты и получить уже скомпонованный загружаемый образ.

Впрочем, если ваше добавляемое приложение не использует видео-акселерацию, (а под ней я понимаю прямой доступ к видео-карте, например, с помощью SDL-

библиотеки), то разумнее разместить его на сервере, и задача сводится к банальной правильной настройке ThinStation, которую мы с вами уже провели. В противном случае, придется долго тянуть за ниточки и добавлять все необходимые пакеты. Впрочем, если ваше приложение того стоит, то почему бы и нет?

Отдельной графой необходимо упомянуть о терминальном сервере. Это одна из ключевых фигур на предприятии, на которую ложится основная нагрузка. Как и всё в природе не убывает и не прирастает на пустом месте, так и в этом случае надо помнить, что уменьшив производительность на терминалах, следует нарастить функционал на терминальном сервере [22]. Это большая отдельная тема. Для оценки суммарного количества ОЗУ, которое будет использоваться на сервере, применяйте, например, следующие цифры (в каждом конкретном случае они будут, естественно, разные) – минимальный объем для каждого клиента начинается со 100-150 Мб памяти. Умножаем на количество подключаемых пользователей и получаем ОЗУ терминального сервера.

Благодарности

Спасибо Mike Eriksen за любезное предоставление идеи по использованию VIA EPIA-M в качестве легковесного терминального решения.

- [1] <http://www.windowsfordevices.com/news/NS3139003780.html>
- [2] http://en.wikipedia.org/wiki/Total_cost_of_ownership
- [3] <http://www.darwinmag.com/read/110103/question74.html>
- [4] <http://www.ibusiness.ru/marcet/CIO/19102/>
- [5] Антон Борисов - "Используем LinuxBIOS на системах VIA EPIA-M". "Системный администратор" 08'2005
- [6] <http://thinstation.sf.net>
- [7] Андрей Маркелов - "Использование бездисковых Linux-станций с загрузкой по сети". "Системный администратор" 11'2004
- [8] <http://peterhost.dl.sourceforge.net/sourceforge/thinstation/Thinstation-2.1.3.tar.gz>
- [9] Крис Касперски - "Модифицируем BIOS". "Системный администратор" 06'2005
- [10] <http://www.rom-o-matic.net>
- [11] <http://www.ami.com/support/downloads/amiflash.zip>
- [12] <ftp://ftp.support.acer-euro.com/utilities/awdf flash/awdf flash826b.zip>
- [13] <http://www.uniflash.org/download/uf140.zip>
- [14] <ftp://ftp.support.acer-euro.com/utilities/winphlash/winphlash-utility.zip>
- [15] <http://struktur.kemi.dtu.dk/thinstation/> - "Thinstation and VIA Epia - a perfect match"
- [16] <http://www.xbitlabs.com/articles/mainboards/display/ami-bios-update.html>
- [17] <http://www.neoware.com>
- [18] <http://www.neoware.com/thin-clients/e100.html>
- [19] <http://www.thinplanet.com/trends/mseiger0505.asp> - Microsoft Preps Thin Client for XP
- [20] <http://h10010.www1.hp.com/wwpc/us/en/sm/WF04a/12454-321959-89307-338927-89307.html>
- [21] <http://struktur.kemi.dtu.dk/thinstation/TS/>
- [22] <http://www.entry.kiev.ua/systems/citrix/index.html>