Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Контроль издалека

Татьяна Казанцева рассказывает, как пресечь «левую» деятельность школьников на уроке, не отходя от кас... учительской кафедры.

Наш эксперт

Татьяна Казанцева в свободное время от корпения над написанием методики скрещивания Scratch и Arduino оттачивает навыки работы со свободным ПО для использования в школе и дома.

«Высоко сижу, далеко гляжу», говорила Маша из небезызвестной сказки, пытаясь показать медведю, что она контролирует процесс доставки пирожков. А можно ли проделать такое в классе и проследить, что происходит на машинах учеников, сидя за учительским компьютером?

Как говорится на ИТ-сленге, «хороший админ – ленивый админ». Давным-давно умные деятели компьютерных наук поняли, что бегать к пользователям по каждому зову – занятие неблагодарное, и придумали способы, позволяющие управлять машинами удаленно. А нельзя ли провернуть такое в компьютерном классе и сделать даже большее – увидеть, чем заняты ученики во время урока: не играют ли в какие-нибудь игры вместо написания задачи на нелюбимом языке программирования и не пытаются ли разукрасить рабочий стол в ультрамодный розовый цвет?

Ответ, как всегда, будет положительный (иначе не было бы этой Школы **LXF**). Вы не только можете увидеть паранормальную активность на рабочих местах, но и непосредственно поучаствовать в этом процессе, перехватить управление и сделать еще больше. И все это – как в текстовом (вдруг среди вас есть ретрограды, которые боятся всего, что отличается от командной строки), так и в графическом режиме.

Пользователи «других форточек» могут сказать, что у них уже давно есть *RAdmin* и прочие примочки для несанкционированного доступа на удаленный рабочий стол; но в том и прелесть свободного ПО, что в Linux этих возможностей в разы больше, и вы даже сможете залезть к «этим форточникам» на их компьютеры и поуправлять ими также.

Давайте по порядку рассмотрим, что же может противопоставить Linux пытливому и одновременно изворотливому уму школьника.

О чем умалчивает ssh

При первом заходе админа в сеть он сразу же попытается сделать две вещи: заблокировать пользователям способы лезть куда не надо и настроить себе доступ по SSH ко всем имеющимся машинам. Про SSH *LinuxFormat* опять же писал неоднократно (см. врезку), и это весьма удобная вещь, позволяющая получить доступ к удаленной командной строке с достаточной безопасностью.

Настроить такой доступ довольно просто. Первым делом вам нужно на машинах учеников (да-да, именно на них) установить необходимые пакеты сервера SSH. Я буду все рассматривать на примере любимого (и одного из лучших, по моему мнению) дистрибутива Edumandriva (многие, конечно, будут шуметь, почему не Ubuntu, но тут дело моего вкуса), для которого нужно установить следующий пакет – *openssh-server*.

На машине учителя, следовательно, должен быть клиент доступа – *openssh-client*.

После установки не забудьте запустить демон *sshd*, дав команду

```
/etc/init.d/sshd start
```

Далее вам нужно настроить доступ по ключам. Можно, конечно, разрешить доступ на машины пользователю root, отредактировав строку в **/etc/ssh/sshd_config** и установив

```
PermitRootLogin yes
```

но это небезопасно. Лучше будет выполнить описанные далее шаги. Первым делом создайте пару файлов ключей с помощью команды

```
ssh-keygen
```

У вас запросят парольную фразу. Оставьте ее пустой. Теперь в каталоге **~/.ssh** есть два файла: **id_rsa** и **id_rsa_pub**. Первый файл – это закрытый ключ, и его нужно хранить в безопасном месте, а второй – открытый, его нужно добавить в файл **~/.ssh/authorized_keys** на удаленном компьютере. Файл можно скопировать на флэшку, вставить ее в удаленный компьютер и добавить ключ с помощью команды

```
cat id rsa.pub >> ~/.ssh/authorized_keys
```

Теперь разрешите аутентификацию по ключу, установив параметр

```
PubkeyAuthentication yes
```

в файле **/etc/ssh/ssh_config** на удаленном компьютере, и перезапустите сервис SSH:

```
/etc/init.d/sshd restart
```

Отключите вход по паролю добавкой в файл **sshd_config** следующих строк:

```
PasswordAuthentication no
```

Если вы все сделали правильно, то команда

```
ssh user@192.168.1.10_
```

позволит вам зайти на машину с IP-адресом 192.168.1.10 от имени пользователя user.

Чтобы иметь возможность запуска графических приложений с этой машины, сделайте X-проброс – для этого нужно, чтобы в файле **sshd_config** была установлена опция

```
X11Forwarding yes
```

– и запускайте *ssh* с ключом **-X**.

```
ssh -X user@192.168.1.10
```

Если вас не пускают, не забудьте проверить настройки брандмауэра. У вас должен быть открыт порт 22 (или разрешен доступ по SSH при настройке из графического режима).

Ну и в самом плачевном случае – если вы не понимаете, что тут написано, дайте эту статью способному ученику (или его папе), и пускай он поможет вам с настройкой.

Теперь – само собой разумеющийся вопрос: чего ради эти мучения? Как это поможет контролировать работу учеников? А очень просто – мы получили удаленный доступ к командной строке пользователя, то бишь ученика, и можем делать оттуда все, что захотим.

Давайте попробуем проконтролировать, тем ли он занят на своей машине. Для этого, выполнив вход на удаленную машину, введите следующую команду (мы будем подразумевать, что имя пользователя на всех ученических машинах – user):

```
ps -u user
```

Это выведет нам список всех процессов, запущенных от имени данного пользователя. Вы должны получить что-то типа такого:

```
PID TTY          TIME CMD
5050 ?        00:00:01 lxsession
5115 ?        00:00:00 ssh-agent
5140 ?        00:00:01 gpg-agent
5251 ?        00:00:00 dbus-launch
5252 ?        00:00:00 dbus-daemon
5265 ?        00:00:00 s2u
5271 ?        00:00:27 openbox
5275 ?        00:01:50 lxpanel
5277 ?        00:00:19 pcmanfm
5278 ?        00:00:10 xscreensaver
5280 ?        00:01:33 parcellite
5282 ?        00:00:05 nm-applet
5285 ?        00:00:01 menu-cached
5288 ?        00:00:00 gnome-keyring-d
5295 ?        00:00:00 gvfsd
5302 ?        00:00:02 gconfd-2
5304 ?        00:00:00 volumeicon
5307 ?        00:00:00 gnome-keyring-d
5315 ?        00:03:56 pulseaudio
5329 ?        00:00:00 gvfs-fuse-daemo
5338 ?        00:00:00 gconf-helper
5552 ?        00:00:02 gvfs-gdu-volume
5560 ?        00:00:00 gvfs-gphoto2-vo
7704 ?        00:06:15 chrome
8989 ?        00:02:40 soffice.bin
9985 ?        00:03:50 lxterminal
9986 ?        00:00:00 gnome-pty-helpe
15200 pts/1   00:00:00 bash
15985 ?   00:00:00 gvfsd-trash
18701 ?   00:00:01 sol
19568 pts/1   00:00:00 ps
```

Казалось бы, что можно узнать из перечня непонятных названий? Для достижения понимания введите эту же команду для своего (учителя) пользователя на вашей машине, запустив предварительно только те приложения, которые необходимы в текущем задании. А затем сравните выводы, отметив различающиеся программы. К примеру, в нашем случае учащиеся должны выполнять задания в *LibreOffice*. Из вывода видно, что на компьютере ученика запущен *soffice.bin*, а значит, он работает в чем полагается (для *OpenOffice.org* процесс будет такой же). Но два процесса будут отличаться – это *chrome* и *sol*. Первый, как несложно догадаться, является браузером Chromium, а вот второй процесс – ничем иным, как пасьянсом [solitaire] (скажем, пресловутой Косынкой). Оба эти процесса подлежат уничтожению как не относящиеся к делу.

Названия процессов легко изучить (и записать), запуская программы на своей машине. Последние запущенные процессы будут иметь более высокий PID, и их легко будет отличить от ранее используемых.

Теперь же, распознав, что учащийся занимается «левыми» делами, применим магию командной сроки (вы помните, что мы все еще залогинены через SSH) и уничтожим эти вредоносные для данного урока приложения. Для этого достаточно скомандовать **killall <имя процесса>** –

```
killall sol
```

– и враг повержен!

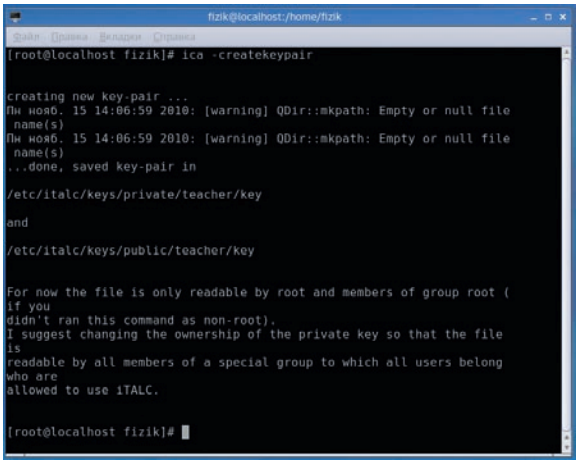
Таким же способом можно выключить удаленно и другие программы.

В порядке побочного эффекта вы получите возможность удаленно выключить или перезагрузить (**poweroff** или **reboot**) машину учащегося или, запустив, к примеру, **mc**, посмотреть текст написанной им программы или запустить свежескомпилированную учебную задачу.

А графически — нельзя?

Но все-таки командная строка для многих преподавателей сродни колдовству, и только посмотрев на экран, они могут понять, что реально происходит на машине. А нельзя ли увидеть экран компьютера ученика? Да, тоже можно. Можно использовать технологии удаленного рабочего стола (к примеру, *VNC*), но лучше применить специализированные программы, которые также позволят транслировать свой рабочий стол на экраны учащихся; перехватывать управление; блокировать все экраны, привлекая внимание; и т. п. Одной из самых известных программ такого рода является *iTALC* (http://italc.sourceforge.net/). Она входит во все известные школьные дистрибутивы, и если вы используете Школьный Линукс, ПСПО или Edumandriva, то ее установка будет очень простой – максимум, вам нужно будет внести пользователя в группу **italc** командой

```
usermod -G italc user
```



Результат работы ica -createkeypair.

SSH — безопасная командная строка

SSH [англ. Secure Shell – «безопасная оболочка»] – сетевой протокол сеансового уровня, позволяющий производить удаленное управление операционной системой. Он шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищенной среде практически любой другой сетевой протокол. Таким образом, можно не только удаленно работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео (например, с веб-камеры). Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удаленного запуска графических приложений. То есть вы можете сделать так называемый X-проброс, запустив, к примеру, *LibreOffice* на школьной допотопной машине со 128 МБ ОЗУ, хотя реально он может выполняться на удаленном 64-ГБ сервере в соседнем здании.

Подробнее об SSH мы уже писали на страницах **LXF**. Вы можете почитать следующие статьи (их можно найти или в PDF-выпусках журнала, или на вики *LinuxFormat* – wiki.linuxforumat.ru):

- *SSH и VNC: Работа издали* (**LXF119**) wiki.linuxformat.ru/index.php/LXF119:ssh
- *Часто задаваемые вопросы. Соединяемся* (**LXF102**) wiki.linuxformat.ru/index.php/LXF102:Ответы
- *Краткая справка. X-проброс* (**LXF106**) wiki.linuxformat.ru/index.php/LXF106:Ответы

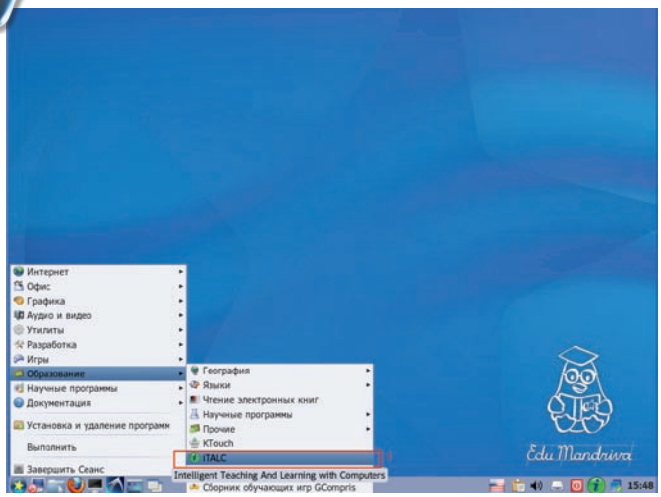
и программу можно будет запускать. Все остальные компоненты, а именно клиенты *italc-client* (ICA) и управляющий *italc-master*, уже будут установлены и настроены. Но давайте рассмотрим установку и настройку, как если бы мы использовали *iTalc* впервые.

iTALC был разработан именно для использования в школе и предлагает много возможностей для учителей, таких как

- просмотр, что происходит в компьютерной лаборатории, в режиме Обзор, с возможностью сделать снимки;
- дистанционное управление компьютерами, для поддержки и помощи другим людям;
- показ демо (в полноэкранном режиме либо в окне) – вывод экрана учителя на компьютеры всех учеников в режиме реального времени;
- блокировка рабочих станций для перемещения пристального внимания на учителя;
- отправка текстовых сообщений для учеников;
- удаленное включение, выключение и перезагрузка компьютера ученика;
- удаленный вход в систему и выход из системы и удаленное выполнение произвольных команд/скриптов;
- домашнее обучение – сетевая технология *iTALC* не ограничена подсетью, так что ученик может присоединиться к уроку и дома, через VPN-подключение, всего лишь установив клиент *iTALC*.

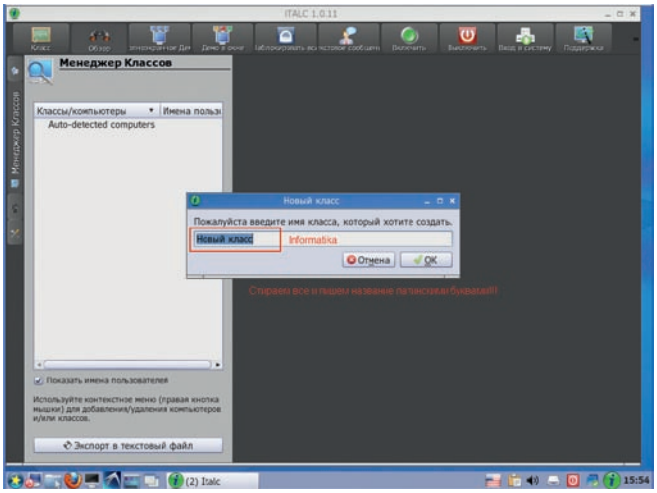
Итак, мы имеем в наличии компьютерный класс, где *n* ученических компьютеров и 1 компьютер преподавателя объединены в единую локальную сеть. ПО *iTALC* состоит из 2-х частей: клиента (устанавливается на ученическую машину и на компьютер учи-

Шаг за шагом: Настраиваем iTALC



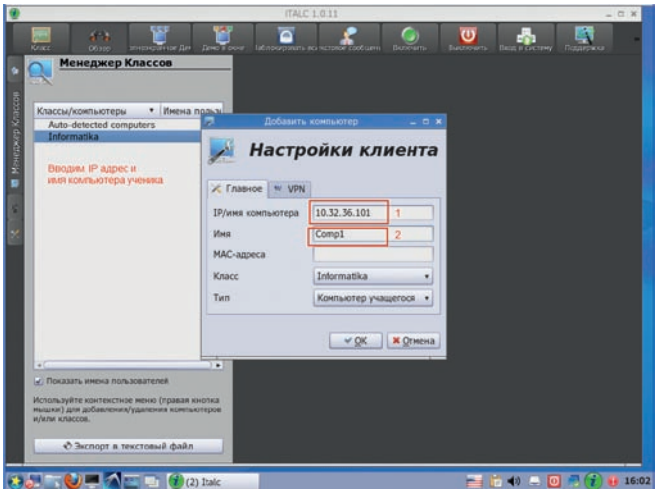
1 Главное — начать

На учительском компьютере запускаем административную часть программы.



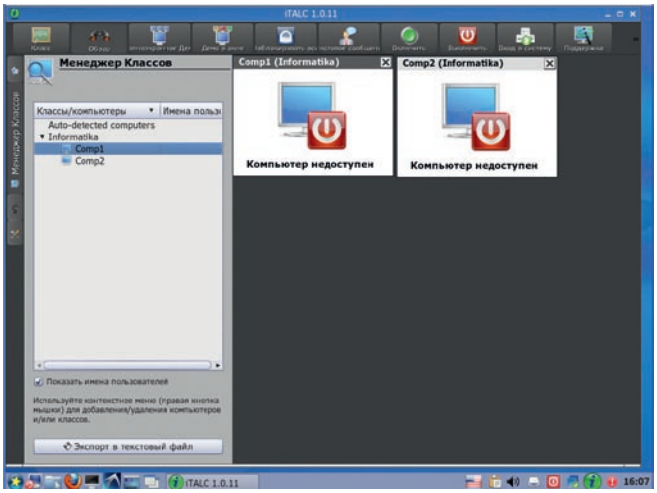
2 Создаем новый класс

Хитрость в том, что вы можете управлять более чем одним классом, и даже под управлением Windows!



3 Добавляем компьютер

учащегося. Перед этим желательно составить схему сети класса с указанием IP-адресов машин.



4 Повторяем добавление

компьютеров для каждого ученического компьютера. Не забудьте экспортировать список в текстовый файл, чтобы не проделывать эту работу каждый раз.

Дважды щелкая по имени компьютера, получаем мини-копию экрана удаленного компьютера (они появляются в левом верхнем углу рабочего поля программы, поэтому их нужно сдвигать на свободное место). Дальше остается изучить основные функции программы. Это можно сделать самостоятельно.

теля) и мастера (устанавливается только на компьютер учителя). Клиент позволяет подключаться мастеру и управлять работой удаленной машины. Мастер содержит интерфейс для управления удаленными компьютерами (компьютерами учеников).

Первым делом добавим пользователя, от имени которого будем работать, в группу **italc**. Потом откроем в брандмауэре порты TCP и UDP 5800–5900 для работы (если у вас разрешены все подключения, этот шаг можно пропустить).

(В Edumandriva эти настройки уже сделаны по умолчанию, поэтому данный шаг пропускаем.)

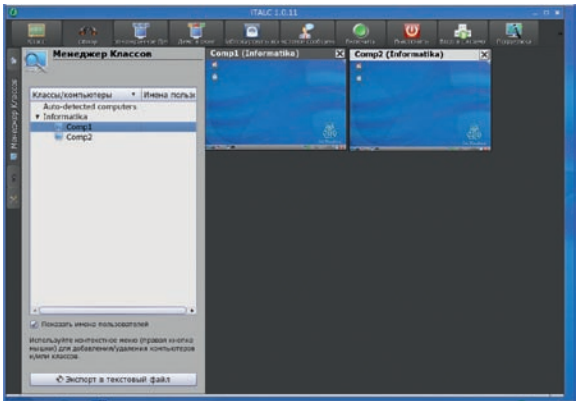
Создадим ключи: публичный [public] – для компьютеров учеников и приватный [private] – для компьютера учителя. Для этого войдем в консоль администратора (в любом эмуляторе терминала переключимся на root) и выполним команду:

```
ica -createkeypair
```

которая сгенерирует необходимые ключи и расположит их в нужных папках.

На рисунке виден отчет генерации и пути к ключам:
/etc/italc/keys/private/teacher/key – приватный ключ (недоступен для пользователя)
/etc/italc/keys/public/teacher/key – публичный ключ (доступен для файловых операций)

Для работы клиента, как и в случае с SSH, нужно скопировать публичный ключ с учительского компьютера на все компьютеры учеников в соответствующую папку. Напомню, он лежит в папке **/etc/italc/keys/public/teacher/key** и не имеет расширения. Этот файл необходимо скопировать на все ученические компьютеры класса в эту же директорию (**/etc/italc/keys/public/teacher/key**). Если в ней имеется ключ, то его необходимо перезаписать новым. Чтобы скопировать ключ на ученический компьютер, нужно войти как root, в противном случае система не даст перезаписать файл ключа. Эти действия нужно повторить на каждом компьютере ученика. Затем настраиваем *iTALC*.



› В результате вы должны получить картинку типа вот этой.

После этого вы спокойно можете контролировать работу учащихся, выдавать на их компьютеры изображения со своего («полноэкранное демо»), делать картинку-в-картинке, чтобы учащийся мог повторять за вами действия («демо в окне»). Щелкнув два раза на окне учащегося, вы сможете подробнее рассмотреть, что он делает, и в случае необходимости даже перехватить управление. Попробуйте, это очень просто!

Напишите нам, если вы чего-то не поняли или хотите, чтобы мы рассмотрели какие-то моменты поподробнее. Мы ждем ваших откликов. LXF