



» Записал  
Евгений Балдин



### История успеха

» Кто

Александр Геннадьевич Fenster, математик-программист, преподаватель информатики.

» Откуда

Механико-математический факультет Новосибирского государственного университета (НГУ).

» Проект

Адаптация курса информатики для выполнения практических занятий в среде GNU/Linux.

» Результат

Технология опробована, студенты освоились,

## GNU/Linux проникает в НГУ

Чтобы удачно внедрить GNU/Linux повсеместно, одобрение руководства, как правило, необходимо. А как быть, если руководству в лучшем случае все равно? Можно ли что-то сделать своими силами? **Александр Геннадьевич Fenster** показывает одну из возможных дорожек в нужном направлении.

**П**реподаванием программирования я заинтересовался, еще будучи студентом старших курсов. Начал вести занятия в Высшем колледже информатики при НГУ и в Летней школе юных программистов. Сейчас в свободное от основной работы время преподаю в самом университете на механико-математическом факультете.

До университета вообще ни разу не работал в системах, отличных от MS-DOS и Windows. На первом курсе ММФ НГУ пошел на факультатив по языку программирования LISP. Преподавателем был сотрудник Института систем информатики СО РАН, который

создал всем учетные записи на своем рабочем компьютере (там был установлен Debian Potato). Как сейчас помню листочек блокнота с выписанными командами *ls*, *ps*, *kill* и чем-то еще таким. Впечатление было, конечно, то еще: как будто пытаешься объяснить на иностранном языке, которым не владеешь. Впрочем, это ощущение быстро прошло. Затем как-то так получилось, что я перенес свою почту на этот сервер и начал читать ее программой *pine*, потом перевез ICQ (сначала использовал *licq*, затем *zicq*, *centericq* и сейчас *centerim*). Приглянулось, что почта и «аська» доступны независимо от того, где я сам физически нахожусь: я просто под-

ключался по SSH к серверу и работал. Где-то в это время уже было несколько машин, на которых у меня были права администратора. Затем я попробовал использовать GNU/Linux на своем рабочем столе – и понравилось. Сейчас с одинаковой частотой использую две системы: Debian Lenny дома и лицензионный Windows XP на рабочем ноутбуке (от корпоративных стандартов никуда не денешься).

Курс программирования состоит из лекций, семинаров и практики. На лекциях студентам читаются общие вещи, касающиеся языков программирования и алгоритмов, без привязки к конкретным компиляторам и операционным системам. На семинарах (в обычных аудиториях, без компьютеров) изучается язык C и решаются задачи по программированию и разработке алгоритмов. На практике в компьютерном классе студенты решают и сдают 15 семестровых заданий. В конце семестра студенты сдают зачет и экзамен.

От семинаров и практики требуется, чтобы студенты по окончании курса знали синтаксис C и могли самостоятельно писать код: например, реализовывать вычислительные методы в курсе вычислительной практики. У меня есть свой набор задач, который сформировался за несколько лет преподавания. Ознакомиться с ним, а также с информацией для студентов, можно на сайте [7113.fenster.name](http://7113.fenster.name) (7113 – это номер студенческой группы).

Раньше на первых практических занятиях я показывал основы работы в Microsoft Visual Studio, и дальше студенты возились с этим огромным монстром, используя от силы 5% его возможностей. С моей точки зрения, учить писать простые (100–200 строк максимум) программы с использованием этой технологии – это как стрелять из пушки по воробьям. Куча времени уходит только на обучение самой среде: как создать проект и как запустить программу. Для начинающих все это совершенно не нужно и только отвлекает от главного – а именно от написания кода. В этом году мне захотелось перейти на более естественный и легкий инструментарий, и среда GNU/Linux выглядела для этого как нельзя более подходящей.

К сожалению, условий для такого рода изменений не было никаких, так как в терминальных классах нашего факультета везде установлена ОС Windows XP, без каких-либо исключений. Однако все оказалось не так страшно, как представлялось. Для организации доступа студентов к среде GNU/Linux нужно было найти сервер и выбрать SSH-клиент. Также перед началом обучения необходимо было определиться с набором базовых программ.

**1 Выбор сервера.** Сначала попробовал пойти официальным путем – зайти к администраторам и попросить создать для студентов учетные записи (или хотя бы одну на всю группу). Т.к. просьба оказалась беспрецедентной для мехмата, то мне сказали, что будут говорить с главным админом и решать. В итоге админ «нарезал» для меня виртуальную машину на базе OpenVZ в сети НГУ (доступна из НГУ, общежитий и местных локальных сетей), где я и «завел» всех студентов. Подозреваю, что если бы не личное знакомство с администратором, то ничего сделать не получилось бы.

**2 SSH-клиент.** В качестве клиента SSH я выбрал PuTTY, доступный по адресу [www.chiark.greenend.org.uk/sgtatham/putty/](http://www.chiark.greenend.org.uk/sgtatham/putty/) [и, кстати, являющийся свободным ПО, – прим. ред.]. На мехмате НГУ Windows переставляют из образов чуть ли не каждую неделю, а официально внести PuTTY в список нужного для установки ПО я на начало семестра не успел. В итоге придумал простой вариант: на своем сайте ([7113.fenster.name](http://7113.fenster.name)) разместил ссылку на .exe-файл PuTTY, лежащий в НГУ'шной сети. Очень удобно, что инсталляции он не требует. Процесс работы такой: студенты идут на сайт (они в любом случае идут туда читать задачи), щелкают по ссылке, скачивают программу в 400 КБ, запускают, затем вводят адрес сервера, имя пользователя и пароль – вуаля! Быстрее, чем *Studio* запустить.

**3 Выбор программ.** Подумав, решил остановиться на варианте *bash* + *vim* + *make*. Почему, скажем, не *emacs*? В основном потому, что сам использую *vim* и хочу учить тому, что сам знаю достаточно хорошо. Решил не навешивать никуда никаких псевдонимов,

а заставить писать команды полностью: думаю, так полезнее. Программы *mc* и *mcedit* тоже показал, но намного больше внимания уделил *vim*.

Код компилируется при помощи GCC. Некоторые студенты из особо продвинутых разобрались с отладчиком *gdb*.

**4 Документация.** Написал и выложил краткую инструкцию по командам на русском языке: [7113.fenster.name/commands.pdf](http://7113.fenster.name/commands.pdf).

Адаптация студентов к незнакомой для них среде прошла лучше, чем я ожидал, хотя до начала семестра с GNU/Linux никто из них не сталкивался (я специально это уточнил).

На первом занятии я потратил 40 минут на объяснение нескольких базовых вещей: перемещение по каталогам, использование текстового редактора и команды *make* для компиляции программ. Добился, чтобы каждый студент написал, откомпилировал и запустил «Hello, world!». К моему удивлению, этого оказалось вполне достаточно, чтобы студенты начали работать самостоятельно.

Судя по выводу команды *last*, студенты активно работали и из дома (из общежитий). Был очень приятно удивлен, когда некоторые студенты начали сдавать задачи на своих личных ноутбуках, на которых был установлен GNU/Linux. Здесь, конечно, сыграло свою роль наличие расположенного рядом, в пределах кампуса, зеркала Debian: [debian.nsu.ru](http://debian.nsu.ru).

Сам курс практически не изменился. Я считаю, что начинать изучение программирования всегда следует с простых программ (консольных приложений), и нет разницы, в какой операционной системе это делать. Так что везде, где есть возможность дать студентам входы на Linux-машину, можно без особых сложностей отказаться от использования громоздких IDE, установленных локально, и перейти на удаленный вариант работы.

По опыту этого семестра можно отметить, что исчезли забытые дома «флэшки» с программами (ведь сейчас все делается на сервере), а также проблемы с созданием, сохранением и открытием проектов в *Visual Studio*. Также появилась возможность заменить некоторые задачи более приближенными к реальности: например, вместо какого-нибудь стандартного задания на обращение с файлами и строками можно попросить разобрать формат */etc/passwd*. В то же время, работы с моей стороны немного прибавилось: необходимо было написать инструкции, настроить виртуальный сервер и завести учетные записи.

Другие преподаватели к моим начинаниям относятся с интересом, и я надеюсь к следующему году расширить эксперимент на несколько групп. У меня есть лето, чтобы придумать, как перевести следующий курс, где изучается создание графических интерфейсов с использованием *Borland Developer Studio*, на свободные продукты. Если получится придумать адекватную по сложности замену, то это будет просто здорово.

По окончании семестра можно с уверенностью сказать, что эксперимент прошел удачно. Собственно говоря, в этом нет ничего удивительного, так как GNU/Linux давно пригоден для работы.

Основная же, с моей точки зрения, проблема для такого рода переходов, как и десять лет назад, такая: чтобы нормально разбираться с системой, нужно иметь под рукой «гуру», или же достаточно количество терпения для чтения документации, причем чаще всего на английском языке. Если вы готовы тратить часы на чтение «манов» – вы сделаете все, что вам нужно. Но пользователь хочет иметь систему, работающую «из коробки», и хотя каждый новый выпуск того же Debian'a в этом смысле становится все лучше и лучше, до идеала, мне кажется, еще далеко.

В принципе, роль того самого «гуру» может эффективно играть сетевое сообщество. Например, перед самым началом эксперимента я задал вопрос на местном Linux-форуме ([www.nsu.ru/phpBB/viewforum.php?f=9](http://www.nsu.ru/phpBB/viewforum.php?f=9)) и получил несколько достаточно дельных советов.

Не надо бояться экспериментировать. Многие вещи, которые в вузах традиционно выполняются в других ОС, можно делать при помощи GNU/Linux, а если не видно разницы – ... **LXF**