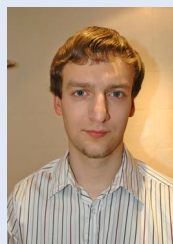


Колонка главного редактора



Теплая пора наконец-то окончательно и бесповоротно настигла российскую столицу. По традиции, начало активного дачного сезона ознаменует заметное движение среди органи-

заторов ИТ-мероприятий. Так, в первой половине апреля Санкт-Петербург стал местом проведения очередных дней технологий Sun, где в фокусе внимания оказались Open Source-проекты компании: OpenSolaris, Java и MySQL. Буквально через несколько дней в Москве прошла профессиональная конференция системных администраторов RootConf 2009, где заметная часть выступлений была посвящена популярным в соответствующей среде программным продуктам (Linux, FreeBSD, OpenVZ и многим другим).

Не менее интересные события нас ожидают и в наступившем месяце. Если говорить о московских, то 14 мая в рамках международной выставки «Связь-Экспокомм» состоится пятый Форум по открытому коду, где можно будет увидеть менеджера Google по программам с открытым кодом, понаблюдать за дискуссией «Светлое будущее открытого кода в России?» и познакомиться с прогнозами специалистов по дальнейшему развитию Open Source. Вскоре после этого, 16 и 17 мая, пройдет вторая отечественная конференция мировой серии YAPC (Yet Another Perl Conference) для Perl-программистов («May Perl – 2»).

До встречи на конференциях?

Главный редактор
Дмитрий Шурупов
(osa@samag.ru)

«Open Source»

электронное приложение к журналу

«Системный администратор»

№43, 4 мая 2009 г.

РЕДАКЦИЯ

Исполнительный директор

Владимир Положевец

Главный редактор

Дмитрий Шурупов

Верстка и оформление

Владимир Лукин

Сайт электронного приложения:

<http://osa.samag.ru>

За содержание статьи ответственность несет автор. Все права на опубликованные материалы защищены.

Новости мира Open Source

Black Duck Software оценила стоимость всего Open Source

Компания Black Duck Software, известная своими продуктами и услугами, способствующими использованию Open Source-компонентов в программных продуктах, провела исследование по оценке совокупной стоимости всего программного обеспечения с открытым исходным кодом. Полученная в ходе исследования сумма превысила 387 миллиардов USD.

В пресс-релизе Black Duck Software заявляется, что компания обладает наиболее всеобъемлющей базой данных продуктов ПО с открытым кодом и связанных с ними метаданных, на основе чего и проводились исследования. По полученным данным, сейчас в мире более 200 тысяч Open Source-проектов, совокупное число строк кода в которых превышает 4,9 миллиардов. Взяв за основу эти данные и применив к ним стандартные для индустрии способы оценки стоимости программных продуктов, в Black Duck пришли к выводу, что сейчас на создание всех Open Source-продуктов с нуля пришлось бы потратить более 387 миллиардов USD (более 2 миллионов человеколет).

Кроме того, исследователи утверждают, что по состоянию на сегодня около 10 процентов времени разработки приложений в индустрии ИТ уходит на создание компонентов, которые можно заимствовать из существующих Open Source-проектов. Если бы американские компании воспользовались этим, то смогли бы сэкономить более 22 миллиардов USD в год (за счет использования готовых Open Source-разработок).

Объявлен список принятых проектов Summer of Code 2009

Как известно, в этом году интернет-гигант Google в очередной раз проводит свою инициативу «Лето кода» (Summer of Code), в рамках которой многие Open Source-проекты получают полезные новшества от студентов, а гранты на их работу обеспечивает Google. Недавно был опубликован список проектов, одобренных в GSoC 2009.

В рамках GSoC 2009 поддержку получили операционные системы FreeBSD (усовершенствования в USB, портирование NetworkManager и puffs, улучшение поддержки ext2fs), NetBSD (порт ZFS), Dragonfly BSD (DevFS, порт системы на amd64), MINIX 3 (файловые системы /proc и /dev; использование виртуальной памяти; реализация разделяемой памяти). Не обошли вниманием и популярные

Linux-дистрибутивы: Gentoo (portage-бэкенд к PackageKit, Gentoo – инсталлятор gem-пакетов, портирование portage и его утилит на NetBSD), Debian (интеграция с Amazon EC2, поддержка GNU/kFreeBSD в Debian Installer, переписывание инфраструктуры авто сборки Debian, независимый от дистрибутива update-manager), openSUSE (портирование системы на ARM и MIPS, образовательный модуль к YaST, синхронизация с мобильными устройствами), Fedora.

Также в GSoC представлены инициативы для таких проектов, как GNU и GCC, языки программирования Perl, Python, PHP и Haskell, CMS-системы Drupal, Joomla, TYPO3, Moodle и Wordpress, графические редакторы Blender и Inkscape, графические среды KDE и GNOME, а также Chromium, MySQL, Mono, Mozilla, Samba, Eclipse, wxWidgets, OpenSSH, GStreamer, Maemo, VideoLAN, WINE и другие. Особого внимания удостоились даже некоторые игры (BZFlag и Battle for Wesnoth).

Корпорация Oracle покупает Sun за 7,4 млрд USD

20 апреля корпорация Oracle и компания Sun Microsystems объявили о достижении договоренности, согласно которой Oracle выкупит акции Sun по цене в 9,5 USD за акцию. Общая сумма сделки составит около 7,4 миллиардов USD. Сообщается, что совет директоров Sun Microsystems единогласно одобрил сделку с Oracle. Завершение процедуры поглощения запланировано на это лето.

Как прокомментировал сделку исполнительный директор Oracle Ларри Эллисон (Larry Ellison), покупка Sun преобразует всю ИТ-индустрию благодаря комбинированию лучших в своем роде вычислительных систем для решения критических задач и программного обеспечения корпоративного уровня. «Oracle станет единственной компанией, которая может создать полностью интегрированную систему, где все компоненты подходят друг другу и совместно работают на благо заказчикам, которым остается лишь пожинать плоды. Наши клиенты выиграют, поскольку затраты на системную интеграцию снизятся, а производительность системы, ее надежность и безопасность – повысятся», – раскрыл свою мысль Эллисон.

Тем временем, все Open Source-сообщество будет с интересом наблюдать за тем, как в Oracle намерены сочетать такие программные решения, как платформы Solaris/OpenSolaris и GNU/Linux,

коммерческие СУБД Oracle и популярную Open Source-систему MySQL.

Вышла Ubuntu Linux 9.04 «Jaunty Jackalope»

23 апреля вышла новая версия одного из популярнейших Linux-дистрибутивов – Ubuntu Linux 9.04 «Jaunty Jackalope».

Компания Canonical, поддерживающая Ubuntu Linux, заблаговременно (20 апреля) сделала анонсы своих трех основных редакций нового релиза дистрибутива: 9.04 Desktop Edition (версия для настольных ПК), 9.04 Server Edition (для серверов), Netbook Remix (система для установки на нетбуки).

В последней версии Ubuntu Linux, 9.04, по сравнению с предыдущим релизом (8.10) отмечаются следующие новшества: переход на Linux-ядро 2.6.28, графическая рабочая среда GNOME 2.26, X-сервер X.Org 1.6, автоматическое подключение планшетов Wacom, офисный пакет OpenOffice.org 3.0, новый стиль уведомлений и их настройки, уменьшение времени запуска системы, поддержка новой файловой системы Ext4 (установка на нее пока происходит только по желанию, переход на режим «по умолчанию» запланирован на следующий релиз Ubuntu), поддержка облачных вычислений в Server Edition (обеспечивается Amazon EC2 API), простая установка почтового сервера (пакет dovecot-postfix).

Образы ISO с Ubuntu Linux 9.04 (Jaunty Jackalope) для архитектур i386 и amd64 уже доступны для свободного скачивания на <http://releases.ubuntu.com/releases/9.04>.

Embedded InnoDB – новая встраиваемая СУБД

На мероприятии MySQL Conference and Expo был представлен новый проект от Innobase, разработчика популярного движка хранилища данных InnoDB для СУБД MySQL, ныне принадлежащего корпорации Oracle. Решение, получившее название Embedded InnoDB, ориентировано на применение во встраиваемых устройствах.

Embedded InnoDB основывается на InnoDB Plugin, улучшенной версии движка InnoDB, но при этом ориентирован на использование без СУБД MySQL. Как сообщается, решение не является ни плагином, ни движком хранилища данных для MySQL – это независимый программный компонент, созданный для использования в приложениях, требующих низкоуровневого решения для управления базами данных без использования SQL.

Embedded InnoDB позиционируется как продукт для разработчиков, производителей устройств и независимых постав-

щиков ПО, являющийся облегченной версией InnoDB с минимальными потреблением памяти и совокупной нагрузкой.

Бинарные сборки Embedded InnoDB уже доступны для платформ GNU/Linux и Windows на <http://www.innodb.com/downloads>. Исходный код опубликован под лицензией GNU GPLv2.

Администрация Амстердама полностью переходит на Firefox, OpenOffice.org

Городской совет Амстердама (Нидерланды) постановил, что Open Source-продукты Firefox и OpenOffice.org станут приложениями по умолчанию на всех 15 тысячах десктопов, используемых в администрации.

22 апреля городской совет Амстердама единогласно поддержал предложение представителей голландской лейбористской партии и партии GroenLinks (голландских «зеленых») сделать на компьютерах администрации OpenOffice.org офисным пакетом по умолчанию, а Firefox – веб-браузером по умолчанию. Более того, до конца этого года ожидается, что в городском совете будет рассмотрено предложение по переходу на десктопы, полностью оснащенные программным обеспечением с открытым исходным кодом.

Предложение по переходу на OpenOffice.org и Firefox было принято в качестве ответной меры на недавний запрос на очередное лицензионное соглашение с корпорацией Microsoft. По заявлениям администрации, эта сделка с Microsoft является вынужденной, поскольку им необходимо использовать проприетарное виртуализационное программное обеспечение Softnricity.

Европейский союз выделил грант для MINIX 3

Эндрю Таненбаум (Andrew Tanenbaum), автор свободной UNIX-подобной операционной Minix, опубликовал на сайте Амстердамского свободного университета сообщение о том, что на его проект получен грант в размере 2,5 миллионов евро от Европейского совета по научным исследованиям (European Research Council, ERC).

Как заявляется, благодаря гранту Европейского союза будет в течение 5 лет финансироваться работа двух постдоков, нескольких аспирантов и нескольких программистов. Проводимые в рамках гранта работы станут частью проекта MINIX 3. MINIX 3 – это новый проект по созданию компактной и надежной UNIX-подобной операционной системы с открытым исходным кодом. Она во многом похожа на предыдущие версии MINIX (1 и 2), однако имеет и некоторые значительные отличия, бла-

годаря чему фактически стала независимым проектом, над которым сейчас трудится Таненбаум.

По мнению Таненбаума, операционные системы вроде Windows и GNU/Linux из-за использования монолитного ядра менее надежны, чем могли бы быть. Решение этой проблемы в исполнении профессора компьютерных наук – микроядро – используется в его ОС Minix, которую в наши дни больше применяют для исследований в области операционных систем. Ее новая реинкарнация, MINIX 3, призвана достичь еще большего уровня надежности, чему и будет способствовать новый грант.

Состоялись релизы NetBSD 5.0 и OpenBSD 4.5

29 апреля вышла новая версия NetBSD – 5.0, а 1 мая появилось обновление к другой UNIX-подобной BSD-системе – OpenBSD 4.5. Главным новшеством в релизе NetBSD 5.0 стало значительное улучшение производительности и масштабируемости системы на многопроцессорных (SMP) и многоядерных системах. Это стало возможным благодаря переписанной подсистеме поточной обработки (threading subsystem), теперь основанной на модели поточной обработки один-к-одному (1:1), новым примитивам синхронизации ядра, приоритетному прерыванию обслуживания в ядре (kernel preemption), переписанной реализации планировщика, расширениями планирования в реальном времени. В итоге, почти все базовые подсистемы ядра (виртуальная память, распределители памяти, файловые системы и так далее) были проверены и переработаны для использования эффективных параллельных алгоритмов.

В OpenBSD 4.5 представлена поддержка двух новых платформ: gumstix (на базе xscale) и Openmoko (на базе ARM). Из изменений в драйверах устройств можно отметить поддержку WPA-PSK в ath(4), Intel WiFi Link 5100/5300 в iwn(4), контроллеров IBM SATA/SCSI ServeRAID в ips(4), процессоров AMD Family 11h в km(4), SDHC-карт в sdmmc(4). В состав OpenBSD 4.5 вошли две новые утилиты: yp(8) – YP-сервер с LDAP в качестве бэкенда, xcompmgr(1) для хеппаса. Из новых функций в других утилитах можно выделить вывод в ifconfig(8) списка каналов, поддерживаемых устройством IEEE 802.11, поддержку файловых систем FFS в большинстве устройств, функцию аудиосервера в auct(1). Также в систему вошла новая версия OpenSSH – 5.2.

Дмитрий Шурупов,
по материалам www.nixp.ru
(osa@samag.ru)

ТЕА: Краткая экскурсия в чаепитие

Поскольку многие читатели могут не знать, уведомляем: данный обзор ТЕА написан автором и главным разработчиком этого текстового редактора. Петр Семилетов – постоянный автор статей в «Open Source», многие из которых посвящены разработке ПО. И вот наконец-то он написал о своем наиболее популярном Open Source-детисе – кроссплатформенном текстовом редакторе ТЕА, использующим Qt в качестве основы для графического интерфейса. – Прим. ред.

Предисловие

Текстовый редактор – это инструмент, а вернее, даже станок. Запуская его, вы становитесь один на один с текстом, и от многих вещей зависит не только качество вашего текста, но и ход мысли. Если что-то мешает в ее выражении, мысль сбивается, откладывается и переключается на решение новой вставшей перед разумом задачи. Поэтому толковая программа должна вести себя следующим образом:

- ☒ Не мешать пользователю.
- ☒ Давать возможность выполнить задуманное кратчайшим способом.
- ☒ Не подозревать в пользователе неуклюжего медведя, по сто раз переспрашивая, точно ли вы хотите завершить программу.

Всё это принято на вооружение при разработке текстового редактора ТЕА, основное назначение которого – предоставить при работе с текстом все возможные функции, которые могут понадобиться. Наиболее свежая версия ТЕА доступна для скачивания на двух «официальных» сайтах:

- ☒ <http://sites.google.com/site/teaw32> – ресурс для отечественных пользователей, представляет Windows-версию редактора;
- ☒ <http://tea-editor.sourceforge.net> – основной сайт, на английском языке.

ТЕА распространяется по лицензии GPL 3. Основные целевые платформы для ТЕА: GNU/Linux и Windows (включая Windows 98). По большому счету, ТЕА можно собрать и запускать на любой системе, для которой есть библиотека Qt, а это и Mac OS X, и FreeBSD, и так далее. Отличие между Windows и UNIX-версиями

ТЕА заключается в поддержке проверки правописания: в Windows-версии его нет. Зато Windows-версия оснащена удобной программой установки, сделанной с помощью Open Source-системы NSIS (такой же установщик вы можете видеть, например, у Winamp). Версию UNIX можно поставить традиционно, из исходников – всего тремя командами, данными из-под суперпользователя в каталоге с распакованным дистрибутивом ТЕА:

```
qmake
make
make install
```

Разумеется, можно поискать и бинарный пакет ТЕА для вашего дистрибутива: ссылки на такие пакеты приведены на англоязычном сайте ТЕА, однако я осведомлен не обо всех обновлениях таких пакетов, поэтому, возможно, стоит обратиться к хранилищу пакетов вашего дистрибутива. (К сожалению, в репозитории Ubuntu Linux на момент публикации статьи была доступна лишь версия 17.6.1, которая заметно устарела на фоне актуальной 23.4.0. – Прим. ред.)

Особенности интерфейса

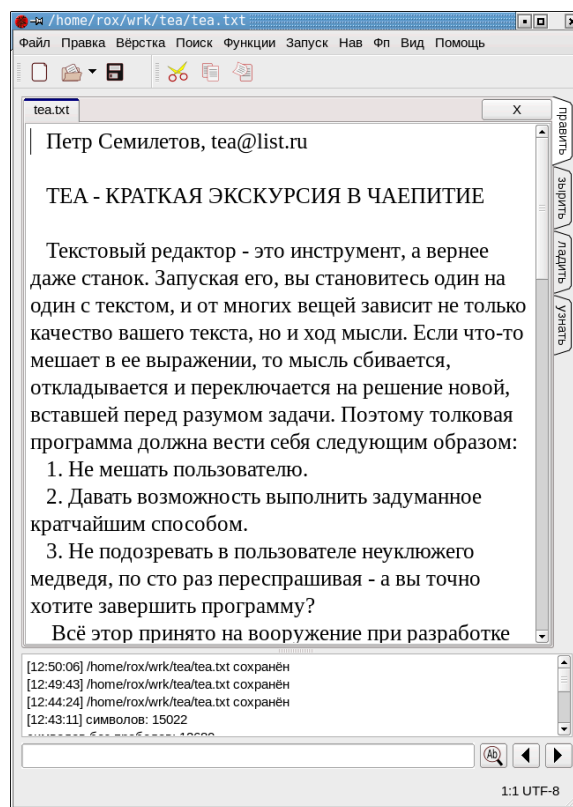
После первого запуска редактора вы обнаружите, что интерфейс не совсем привычен: в первую очередь, это касается открытия и сохранения файлов, а также интерфейса вообще, который состоит из вкладок. Так, традиционный пункт меню «Файл → Открыть» перемещает нас на вкладку «файлового приказчика». (Кстати, везде, где это возможно, в ТЕА используются русские слова вместо заимствованных: я считаю, что русский язык достаточно велик и могуч, чтобы обходиться без помощи разных «менеджеров».) Файловый приказчик в ТЕА представляет собой список каталогов, файлов, кнопок «Открыть» и «Сохранить как». Хотите открыть файл? Двойной щелчок мышью либо нажатие <Enter> на файле в списке. Или же введите имя в поле ввода и нажми-

те кнопку «Открыть». Надо открыть много файлов? Выделите их и нажмите «Открыть». Чтобы сохранить файл под каким-то именем, наберите его в том же поле ввода и нажмите кнопку «Сохранить как». Всё просто. Ниже кнопок – выпадающий список с кодировками. Файл открывается или сохраняется в той кодировке, которая выбрана в этом списке. Пожалуй, всё это – самое очевидное внешнее отличие ТЕА от других редакторов. Подобный интерфейс был присущ в старину у музыкальных программ-трекеров. Впрочем, для любителей традиционные окна «Открыть файл/Сохранить» всё же есть, их можно включить в настройках программы.

Другое отличие – почти полное отсутствие диалоговых окон. Для ввода слов на поиск, для замены, для ввода разных параметров функций используется ниже поле ввода, которое расположено над строкой состояния. Это поле ввода называется в документации просто – «Знаменитое поле ввода», или ЗПВ. Вводите туда какой-нибудь параметр, применяете функцию – вот как это работает.

Функции, сниппеты, скрипты

Что же за функции можно применять? В основном, они сосредоточены в одноименном меню – Функции. Функции бывают встроенные и созданные пользователем. К последним относятся сниппеты и скрипты. Сниппеты (увы, толкового перевода на русский нет до сих пор) – это кусочки текста, которые вставляются в теку-



Общий вид текстового редактора ТЕА

щий документ по месту курсора. Снимет можно использовать и для «обрамления» выделенного текста: в этом случае в теле снимета надо использовать макрос %s, вместо которого ТЕА подставит выделенный текст во время вставки снимета.

Разберем на примере создание нового снимета. Допустим, мы хотим сделать снимет, который будет «обрамлять» выделенную HTML-ссылку. Делает обычный текстовый файл и помещаем в него такую строку:

```
<a href="%s">%s</a>
```

Сохраняем этот файл как обычно – только в файловом приказчике или диалоге сохранения файла перемещаемся по закладке в особый каталог, где ТЕА хранит сниметы (он так и называется – «сниметы»). Всё: после этого новый снимет будет доступен как пункт меню «Функции → Сниметы». Замечу, что каталог сниметов может включать в себя вложенные каталоги, и эта древовидная структура также найдет свое отображение в меню. Названия сниметов и скриптов не должны повторяться или совпадать с существующими пунктами меню – иначе механизм назначения «горячих клавиш» будет в большом недоумении. Сниметы – удобное средство добавления в ТЕА наборов каких-нибудь тегов разметки – например, LaTeX (хотя основные команды набора LaTeX в ТЕА уже встроены).

И без того широкие возможности редактора расширяются и с помощью скриптов, которые выглядят для пользователя так же, как обычные функции из соответствующего меню («Функции»). Скрипты для ТЕА можно писать на Python, Perl, Ruby и Bash (Sh), а также языке встроеного в ТЕА интерпретатора QtScript. Скрипт получает от ТЕА выделенный текст и параметры из ЗПВ, а обратно отдает выделенный текст. Если же вам надо просто запустить какую-то программу с текущим файлом, то вместо скриптов следует действовать через пользовательское меню Запуск, куда помещаются заданные вами программы. Что до скриптов, то они появляются в меню Скрипты, когда вы помещаете их в служебный для ТЕА каталог scripts. Быстрее всего до него добраться через файловый приказчик или диалоговое окно сохранения файлов – там есть закладка «Скрипты».

Бегло пройдемся по встроенным в ТЕА функциям. Они удобно разделены на рубрики: «Сортировка», «Математика», «Анализ» и так далее. В рубрику «Текст» попали функции общего и частного назначения, которым не нашлось места в дру-

гих подменю. Среди функций, относящихся к математике, есть переводчик чисел из арабских в римские и обратно. Есть также вычислитель выражений – например, вы можете выделить в тексте «2+2», применить функцию «Вычислить» и получить результат. Результаты работы функций, равно как и разные служебные сообщения, выводятся в текстовом поле под панелью редактора.

Очень мощным средством обработки списка строк является функция «Применить к каждой строке». Она позволяет применить к каждой строке некий шаблон. Например, мы хотим в конец каждой строки добавить тег
. Пишем в ЗПВ шаблон %s
 и применяем эту функцию. Как видим, макрос %s означает то место, куда будет помещен текст, взятый из строки. Текст помещается в шаблон, затем обработанный шаблон помещается на место текста строки. Еще пример: заключить каждую строку в круглые скобки – шаблон в этом случае выглядит так: (%s).

Есть также функция нумерации строк (в подменю «Математика»). Опять же, по заданному вами шаблону, она добавляет в текст счетчик. Можно задать шаг его увеличения и количество цифр (т.е. может быть как 1, 2, 3, так и, допустим, 001, 002, 003). Подробности описаны в руководстве к программе. Однако приведу небольшой пример. Дан текст:

```
собака  
кот  
мышь
```

Выделяем его, в ЗПВ вводим шаблон счетчика: 1~3~). Применяем функцию нумерации. Получаем:

```
001) собака  
002) кот  
003) мышь
```

В примере шаблона:

- ✓ 1 – шаг;
- ✓ ~ – символ-разделитель, используемый в шаблоне;
- ✓ 3 – количество цифр в счетчике.

Кроме такой вот шаблонной обработки, ТЕА предоставляет широкий набор средств по фильтрации текста, сортировке, удалению форматирования, частотному анализу. Всё, что можно себе представить – есть даже конвертер азбуки Морзе.

Помимо обработки текста, в редакторе представлены некоторые функции вёрстки на разных языках разметки – в частности, на HTML, XHTML, LaTeX, Lout и Docbook. Режим вёрстки выбирается автоматически по расширению фай-

ла, а также можно переключить его вручную в меню («Вёрстка → Режим вёрстки»). В «Вёрстке» находятся пункты меню для скоростного ввода тегов, причем действуют они в зависимости от текущего (для документа) режима вёрстки. Так, пункт «Жирный» для HTML и LaTeX оформит выделенный текст надлежащим образом. Отмечу, что в ТЕА вы не найдете исчерпывающих списков команд LaTeX или тегов HTML – напротив, схвачены лишь общие черты и воплощены в универсальных пунктах меню с человеческими названиями. А дополнить это дело можно с помощью сниметов.

И всё же, в ТЕА наблюдается небольшой уклон в сторону [X]HTML, поэтому в меню «Вёрстка» вы найдете подменю «Инструменты [X]HTML». В нем, кроме прочего, находятся:

- ✓ Толковый переводчик из простого текста в XHTML или HTML. Он берет обычный текст и создает на его основе правильный с точки зрения стандарта XHTML либо HTML документ, с разметкой параграфов с помощью CSS.
- ✓ Функция «Перевести теги в сущности», которая нужна для правильного отображения тегов на веб-странице. Если вам нужно на веб-странице показать код HTML, или символы вроде амперсанда, то проще всего сделать такое, пропустив код через эту функцию.
- ✓ «Кодировать адрес e-mail супротив спама» – кодирует символы выделенного в тексте адреса в их числовые представления. Браузер отображает их как соответствующие буквы, а вот программы сбора адресов – так называемые harvesters – обычно этот формат не понимают. Пользователь видит адрес, а harvester – нет.
- ✓ «Взвесить документ» – выдает общий размер текущего документа и связанных с ним объектов – картинок и тому подобного.

Другие интересные возможности

Обратимся теперь к меню «Файл», где скрыто немало интересного. Например – функция «Открыть в месте курсора», за которой железно закреплена клавиша <F2>. Если у вас в документе есть имя файла, заключенное в кавычки (например, как в HTML-ссылке на локальный файл или в директиве include), то достаточно поставить на это имя курсор, нажать <F2>, и файл откроется в редакторе. Если файл уже открыт, ТЕА перейдет на его вкладку. Если это ссылка на метку внутри HTML-документа, ТЕА перенесет вас в место метки. Более того, если файл

не текстовый, а является изображением, то редактор покажет его во встроенном просмотрщике изображений. Это справедливо и для файлового приказчика: если в нем установить курсор на файл с картинкой и применить «Открыть в месте курсора», откроется то же средство просмотра изображений. В нем, кстати, действует переход по списку изображений и команды поворота. Подробности читайте в руководстве.

В меню «Файл» доступны подменю «Шаблоны» и «Сессии». Поскольку любому пункту меню в TEA можно назначить сочетание клавиш для его быстрого вызова, быстрота создания нового документа на основе шаблона становится очевидной, как и удобное открытие сессии по нажатию пары клавиш. Ведется список последних закрытых файлов, куда, кроме их названий, вносятся кодировка и место положения курсора. Вариант такого списка, но заполняемого пользователем, представляет собой меню «Закладки».

Есть простая функция печати – с подсветкой синтаксиса. Количество языков, синтаксис которых TEA умеет подсвечивать, пока не очень велико: C, C++, Java, C#, NSIS, Seed7, варианты Verilog, Pascal, BASIC, HTML, XHTML, XML, LaTeX, gettext, Fortran. Подсветкой заведуют модули, выполненные в виде html-файлов с правилами подсветки. Модули эти встроены в TEA как ресурсы. Можно делать свои модули и помещать их в каталог hls-директории, где TEA хранит настройки пользователя. Модули будут подхвачены редактором при перезапуске. Цвета в модулях подсветки не являются жестко прописанными значениями, а лишь ссылаются на цвета-объекты из текущей палитры. В TEA для раскраски интерфейса использует-

ся механизм палитр, которые можно переключать в меню «Вид → Палитры». Таким образом, смена палитры не портит подсветки синтаксиса – цвета остаются подобранными гармонично.

В меню «Вид» есть пункт «Переключить перенос строк», однако по умолчанию в TEA действует перекрывающая эту настройку функция переноса строк согласно опции в модуле подсветки. Например, для обычного текстового файла или для HTML перенос строк включен, а для C или Java – выключен. Это очень удобно при работе с файлами разных типов.

Но несмотря на некоторое сходство TEA со средой разработки, до полноценной IDE программа не дотягивает. Есть переключение с исходника на заголовок, есть переход к строке по номеру, о подсветке уже говорилось, но дальше этого дело не идет. TEA удобно использовать совместно с вызываемым из консоли компилятором.

Невзирая на хорошие возможности расширяемости, TEA – довольно монолитная программа со стремлением нести «всё в себе». Документация отображается во встроенном простом HTML-браузере. Встроенный просмотрщик изображений поддерживает несколько десятков графических форматов – впрочем, благодаря библиотеке Qt, на которой основан TEA.

Из форматов текстовых документов, TEA поддерживает, конечно же, обычный текст, а также – только на чтение – ODT (Open Document), DocX, RTF, FB2 (Fiction Book), Abiword, Scribus. Поскольку многие из этих форматов используют ZIP-сжатие, грех было оставлять это без внимания – так в TEA появились удобные функции создания ZIP-архивов. А наличие в редакторе поддержки графических форма-

тов вылилось в набор функций по массовому конвертированию выделенных файлов из одного формата в другой, с возможностью применения масштабирования. После выполнения такого перекодирования можно автоматически запаковать обработанные файлы в ZIP-архив, чтобы куда-то отправить по почте или загрузить на сайт. Впрочем, количество текстовых функций TEA всё же перевешивает количество посторонних.

Еще одной особенностью редактора является его настраиваемость. Всё, что пользователь может захотеть настроить, доступно для изменения. Во-первых, это тема оформления. О палитрах я уже говорил. «Горячие клавиши» на все пункты меню. Изменение размеров шрифта для интерфейса и самого поля редактирования. Размер иконок на кнопках: благодаря тому, что иконки в TEA – векторные, можете увеличивать их хоть на полэкрана – качество от этого не пострадает. Инструментальные панели можно располагать где угодно по периметру главного окна, вертикально и горизонтально. Подменю можно отрывать и перемещать вне окна.

Заключение

В этой статье я не охватил многих возможностей TEA, а лишь галопом пробежался по самым знаковым. Чтобы разобраться в TEA, надо читать подробное руководство, поставляемое вместе с программой. TEA далеко ушел от «заменителя блокнота» и вообще от традиционного текстового редактора, больше относясь к категории программ, подобных Emacs или vim.

Петр Семилетов
(tea@list.ru)

Двойная загрузка на EeePC-1000HD на примере eeeXandros и SLED11

Предисловие

Для начала стоит объяснить, чем эта статья может быть полезна тем, у кого нет EeePC-1000HD. Ответ прост: в ней освещаются некоторые принципы загрузки системы (с использованием GRUB) и приемы работы в Linux. Теперь – о том, зачем мне это вообще понадобилось. Покупая ноутбук, я был точно уверен, что уда- лось предварительно установленную опе-

рационную систему, заменив ее на недавно вышедший SLED (SUSE Enterprise Linux Desktop) 11, – просто по той причине, что уже много лет пользуюсь SUSE-дистрибутивами и знаю все их плюсы и минусы. Но при этом, основываясь на предыдущем опыте подобной процедуры с ноутбуком MSI Wind U90, я решил, что очень важно оставить возможность возвращения к «заводским настройкам» на случай

гарантийного ремонта (или выяснения, что не так с адаптером Wi-Fi: он перестал работать или это вина неправильно выбранного драйвера).

Сначала я запустил установленную по умолчанию систему – как известно, это eeeXandros. Зачем? Потому что первый этап – выяснить, как организован жесткий диск устройства: есть ли там скрытый раздел, как в случае с MSI Wind, или что-нибудь подобное. И за время знакомства с eeeXandros система показала себя весьма неплохо. Вот что мне понравилось:

- ☑ fastinit обеспечивает очень быструю загрузку;
- ☑ набор программ по умолчанию – достаточный для обычного использования: Firefox, Skype, офис, просмотр медиафайлов, игры;

- ☑ неплохо сделаны обновление системы, управление энергопотреблением аккумулятора, подключение к беспроводным сетям;
- ☑ при желании и умении пользоваться основанными на Debian дистрибутивами можно доставить необходимые пакеты и заполнить настройку «под себя».

И казалось бы: если «родная» система успешно справляется с возложенными на нее задачами, зачем ее менять? Мне гораздо чаще требуется полноценный дистрибутив – под ним я понимаю систему с рабочим столом GNOME/KDE, а не простым оконным менеджером вроде IceWM. Но зачем тогда переходить на SLED, когда можно проинсталлировать нужные программные компоненты в eeeXandros? Здесь причина довольно тривиальна: я, как и большинство пользователей Linux, предпочитаю устанавливать уже полностью готовый дистрибутив, а не собирать его по крохам, как это было во времена появления Slackware. Мне не нравится совершать слишком много действий, чтобы получить то, что уже есть. Кроме того, напомним, что долгое время работаю с SUSE-дистрибутивами, поэтому мне с ними проще.

Однако от идеи полного удаления eeeXandros я отказался и решил организовать двойную загрузку, оставив для себя возможность быстрого (менее минуты) запуска ОС с браузером и Wi-Fi вместо нескольких минут ожиданий запуска «тяжеловеса» SLED. Об этом и пойдет речь в статье.

Шаг 1: Осмотрелись

Как я уже писал, в первую очередь надо понять, как устроен жесткий диск ноутбука. Для этого я загрузил eeeXandros и нашел терминал, запустив «Файловый менеджер» и нажав <Alt> + <T>. Теперь потребуется получить права суперпользователя:

```
sudo bash
```

Результаты вывода команды «mount»:

```
rootfs on / type rootfs (rw)
/dev/sda1 on /.ro type ext2 (ro)
/dev/sda2 on /.rw type ext3 (rw,noatime,data=ordered)
none on / type aufs (rw,xino=/.rw/.aufs.xino,br:/dev/sda2)
...
```

Вывод «fdisk -l»:

```
Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            1         425       3413781    83  Linux
/dev/sda2          426       19455     152858475    83  Linux
/dev/sda3         19456       19456         8032+    c   W95 FAT32 (LBA)
/dev/sda4         19457       19457         8032+    ef  EFI (FAT-12/16/32)
```

Как видно, диск объемом 160 Гб разбит на четыре раздела:

- ☑ **sda1** (3 Гб) – «чистая ОС с заводскими настройками», смонтированная в режиме только для чтения, что защищает ее от «повреждений», которые могли бы сделать необдуманные действия пользователя;
- ☑ **sda2** (150 Гб) – основной пользовательский раздел, на который сохраняются все изменения в настройках и создаваемые/редактируемые файлы;
- ☑ разделы **sda3** и **sda4** – служебные; в них, например, скачиваются обновления BIOS.

«Склеиваются» два первых раздела (sda1 и sda2) с помощью виртуальной файловой системы aufs, позволяющей производить объединённое монтирование. Благодаря ней каталоги, которые

имеют один и тот же путь в разных разделах, будут совместно отображать свое содержимое в объединённом каталоге новой виртуальной файловой системы. Использование aufs в данном случае достаточно удобно как для производителя, так и для пользователей: если что-то перестанет работать, нужно просто стереть соответствующие файлы на sda2 (либо отформатировать его) – это вернет систему к «заводским настройкам», при которых всё гарантировано работает.

Резюме: разделы sda1, sda3 и sda4 надо оставить без изменений, а вот sda2 – можно смело переразбивать.

Шаг 2: Backup

Сделаем резервную копию, чтобы в случае возникновения каких-либо проблем можно было вернуть жесткий диск в исходное состояние. Для этого подключите внешний USB-диск или, например, открытую на доступ по сети директорию, после чего выполните следующие команды:

```
dd if=/dev/sda1 of=/путь_к_примонтированному_каталогу/sda1.img
dd if=/dev/sda3 of=/путь_к_примонтированному_каталогу/sda3.img
dd if=/dev/sda4 of=/путь_к_примонтированному_каталогу/sda4.img
dd if=/dev/sda of=/путь_к_примонтированному_каталогу/_
mbr.img bs=512 count=1
```

Последняя команда сохраняет MBR.

Кроме того, если у вас осталось что-нибудь полезное на разделе sda2, можно сохранить и его (либо его части).

Шаг 3: Создание установочного USB-диска

Существует несколько способов установки системы на ноутбук без CD/DVD-привода:

- ☑ подключить внешний USB CD/DVD-привод и запускаться с диска;
- ☑ подготовить компактный (около 10 Мб) загрузочный USB-носитель (образ есть на установочном диске) и установить по Сети;
- ☑ подготовить большой (около 4 Гб) загрузочный USB-носитель и установить с него;
- ☑ извлечь жесткий диск и вставить его в другую Linux-машину, а дальше действовать по аналогии с вариантом большого загрузочного USB-носителя: организовать загрузочный раздел, с которого затем и установить все.

Я воспользовался третьим вариантом и внешним жестким диском (USB):

- ☑ создал на нем отдельный раздел размером в 5 Гб;
- ☑ скопировал на него все содержимое установочного диска SLED 11;
- ☑ сделал его загрузочным, скопировав необходимые файлы для загрузчика GRUB:

```
mkdir /mnt/boot/grub
cp /boot/grub/*stage* /mnt/boot/grub
cp /mnt/boot/i386/loader/{linux,initrd} /mnt/boot
```

Здесь /mnt – каталог, к которому примонтирован раздел с копией установочного диска. Файлы linux и initrd – это ядро и «минимальная ОС», загружаемые по умолчанию загрузчиком ISOLINUX (его конфигурационный файл-меню – /mnt/boot/i386/loader/isolinux.cfg).

Затем требуется установить GRUB: данное действие нужно делать осторожно – чтобы случайно не перезаписать загрузчик в системе вместо загрузчика на загрузочном разделе. В моем случае создаваемый загрузочный раздел виден в Linux как устройство /dev/sdb7: команда mount показывает «/dev/sdb7 on /mnt

type reiserfs (rw)». Для удобства создадим меню для GRUB следующего содержания (файл /mnt/boot/grub/menu.lst):

```
default 0
timeout 80

title INSTALLATION
root (hd0,6)
kernel /boot/linux showopts
initrd /boot/initrd
```

Здесь вместо (hd1,6) указано (hd0,6), потому что во время загрузки этот USB-диск будет уже не вторым устройством, а первым. Теперь установим загрузчик (в моем случае – на /dev/sdb7, что соответствует (hd1,6) в наименовании GRUB):

```
echo "setup (hd1,6) (hd1,6)" | grub --batch
```

В результате должны вывестись сообщения, что все прошло успешно:

```
grub> setup (hd1,6) (hd1,6)
Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Running "install /boot/grub/stage1 (hd1,6) /boot/grub/stage2
p /boot/grub/menu.lst "... succeeded
Done.
```

Последний штрих – пометить раздел как загрузочный. Для этого надо выполнить для своего диска с загрузочным разделом команду:

```
fdisk /dev/sdb
```

Затем в fdisk я сделал следующее:

- ☒ нажал на <A>;
- ☒ на вопрос номера раздела ответил <7>;
- ☒ нажал <P> для просмотра таблицы разделов – чтобы удостоверится, что напротив моего раздела (/dev/sdb7) стоит знак «*» в колонке Boot;
- ☒ вышел с сохранением, нажав на <W>.

Всё – раздел готов.

Шаг 4: Установка SLED11

Теперь можно загрузиться в систему с подготовленного установочного USB-диска (разумеется, если вы выбрали на прошлом шаге этот вариант). Для этого при включении EeePC-1000HD нажимаем <F2>, чтобы попасть в настройки BIOS. Далее в пункте Hard Disk Drives раздела Boot выбираем USB-диск в качестве загрузочного устройства. Теперь выходим из настроек BIOS с сохранением.

Если всё было сделано правильно, появится меню загрузчика GRUB с единственным пунктом – INSTALLATION. После его выбора запустится инсталлятор SLED11. Поскольку он по умолчанию предполагает, что инсталляция производится с DVD, в появившемся псевдографическом меню надо выбрать «установку» и в качестве источника установки выбрать «с жесткого диска». Затем в списке доступных разделов – подготовленный установочный раздел (в моем случае это /dev/sda7), а в нем – каталог «/». Теперь запустится полноценный графический инсталлятор, а дальнейшая установка мало

чем отличается от обычной – кроме двух моментов: разбивки диска и настройки загрузчика.

Шаг 4а: Разбиение диска

- ☒ Удаляем /dev/sda2.
- ☒ Вместо него создаем расширенный раздел.
- ☒ В нем – три раздела (в моем случае – все в ext3):
 - /dev/sda5 (около 10 Гб) – будет использоваться в eeeXandros вместо старого /dev/sda2;
 - /dev/sda6 (около 10 Гб, монтируется как «/») – корневой раздел для SLED11;
 - /dev/sda7 (около 130 Гб, монтируется как «/home») – общий раздел для обеих операционных систем: для хранения документов, видео/музыки и т.д.

Разумеется, каждый волен изменять эти разделы, выбирать другие файловые системы и остальные настройки в зависимости от личных предпочтений. Лично я не стал создавать раздел подкачки: переводить в спящий режим и анализировать дампы памяти не планирую, а количества оперативной памяти (2 Гб) достаточно для обычной работы.

Примечание: не пугайтесь /dev/sdb – это разделы с USB-диска. Их оставляем без изменений.

Шаг 4б: Настройка загрузчика

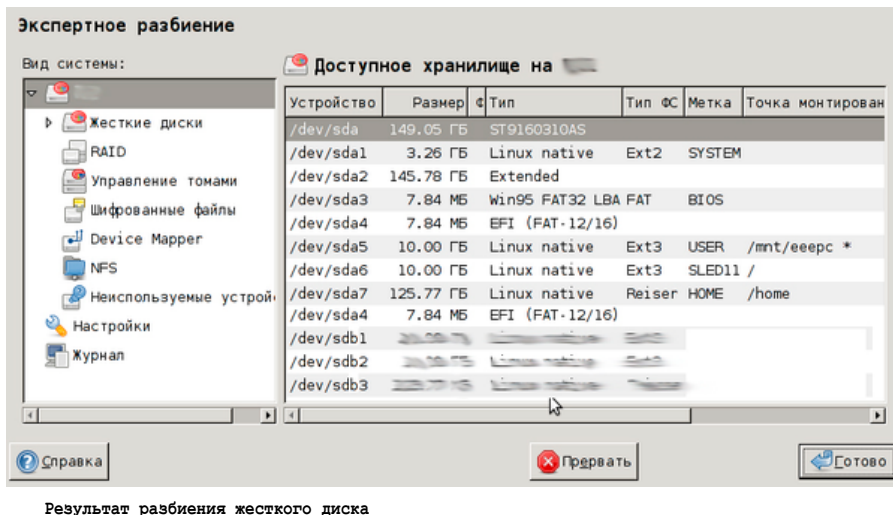
Реализуем следующую схему:

- ☒ В MBR установлен загрузчик от eeeXandros (его позже перенастроим, чтобы он по умолчанию не грузил eeeXandros, а передавал управление загрузчику на /dev/sda6).
- ☒ В загрузочном секторе корневого раздела SLED11 (/dev/sda6) установим загрузчик, который будет грузить SLED (позже в него добавим и eeeXandros).

После того, как оба загрузчика будут настроены, при включении компьютера будет выполняться загрузчик от eeeXandros, который сразу же (без отображения меню) будет загружать загрузчик от SLED, который, в свою очередь, покажет меню с выбором из SLED и eeeXandros.

Соответственно, сейчас мы устанавливаем загрузчик от SLED. Для настройки GRUB надо в инсталляторе на шаге «Обзор установки» перейти в режим «Эксперта». Конфигурация:

- ☒ Местонахождение загрузчика: выбрать только вариант «Загрузить с корневого раздела».
- ☒ Настройки загрузчика: убрать «галки» со всех пунктов (если оставить «Записать общий загрузочный код в MBR», будет изменен загрузчик eeeXandros!).



Результат разбиения жесткого диска

Шаг 4в: Установка

Проверьте все сделанные изменения (особенно если пропустили «Шаг 2: Backup»), прежде чем нажать кнопку «Установить». После нажатия на нее инсталлятор разобьет диск (согласно указанным вами настройкам в «Шаге 4а»), установит пакеты и загрузчик (согласно настройкам «Шага 4б»). Затем инсталлятор перезагрузит ноутбук, чтобы перейти к «постустановочной настройке ОС». Но поскольку загрузчик от eeeXandros, который будет загружаться из MBR, пока еще не настроен запускать загрузчик от SLED, вы увидите лишь сообщение о том, что eeeXandros не нашел раздел /dev/sda2, который мы удалили.

Шаг 4г: GRUB-перезагрузка

Вот здесь мы и воспользуемся одной из возможностей GRUB – загрузка даже тех ОС, которые не прописаны в меню. Для этого перезагружаемся и нажимаем <F9>, чтобы попасть в меню GRUB, а затем – <C>, после чего попадаем в командную строку GRUB. В ней можно сделать не так много, но достаточно для того, чтобы загрузить огромное количество ОС: от различных дистрибутивов GNU/Linux и UNIX-систем до Windows. Нам нужно лишь загрузить другой GRUB, установленный на /dev/sda6 – это делается следующими командами:

```
chainloader (hd0,5)+1
boot
```

Позже эти же строки мы внесем в меню загрузчика eeeXandros, чтобы всё грузилось автоматически.

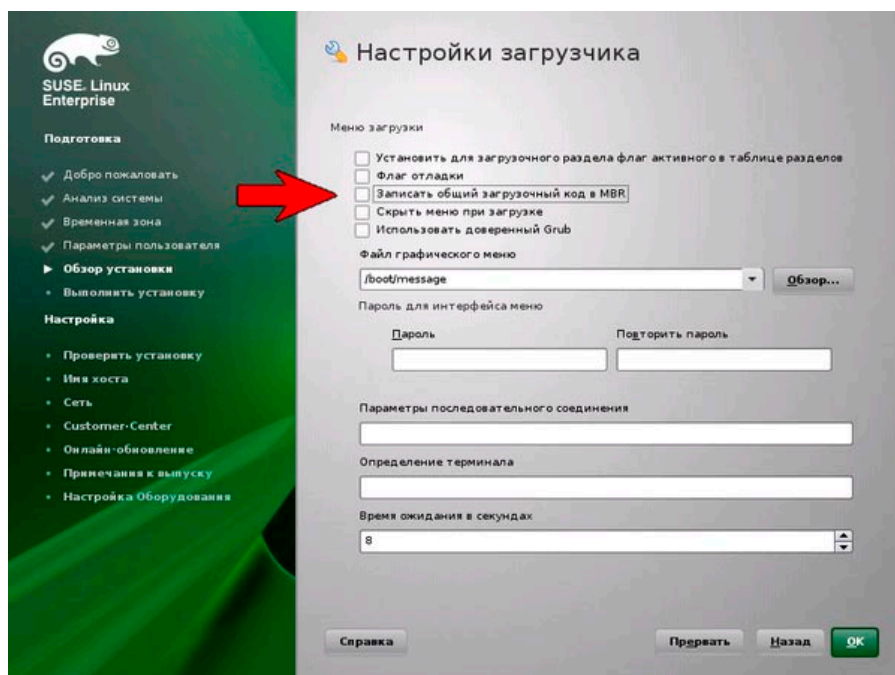
Шаг 4д: Завершение установки

Теперь следует обычное окончание установки: настройка различного оборудования сетевой карты, монитора и т.д. После нее вы попадете в установленную систему.

Шаг 5: Последние штрихи в eeeXandros

Нам нужно добавить запись в меню загрузчика от eeeXandros, а также заменить в настройках все упоминания про /dev/sda2 на /dev/sda5. Для этого смонтируем корневой раздел eeeXandros:

```
mount /dev/sda1 /mnt
```



Настройки загрузчика GRUB в инсталляторе SLED

Шаг 5а: Меняем sda2 на sda5 – eeeXandros

Я провел поиск по наличию слова sda2 в директориях с корневого раздела eeeXandros:

```
grep -rls sda2 /mnt/{etc,bin,sbin}
```

Он показал на присутствие его в двух скриптах: /mnt/sbin/scanuser.sh и /mnt/sbin/formatuser.sh. Первый проверяет файловую систему на /dev/sda2, а второй – форматирует /dev/sda2. Заменим sda2 на sda5 в обоих файлах:

```
sed -i "s/sda2/sda5/g" /mnt/sbin/scanuser.sh
sed -i "s/sda2/sda5/g" /mnt/sbin/formatuser.sh
```

Но упоминания в двух скриптах мало для того, чтобы этот раздел монтировался по умолчанию, как это показывал вывод команды mount (в «Шаге 1») – пора обратить внимание на initrd.

Шаг 5б: initrd для eeeXandros

Если посмотреть конфигурационный файл (/mnt/boot/grub/menu.lst) загрузчика eeeXandros, можно увидеть следующие строки:

```
initrd /boot/initramfs-eeepc.img
```

А что такое initrd? Initrd – это «небольшой Linux», который выполняет различные подготовительные действия до и для запуска «большого Linux». Разберемся с устройством /boot/initramfs-eeepc.img:

```
mkdir /tmp/INRD; cd /tmp/INRD
zcat /mnt/boot/initramfs-eeepc.img | cpio -i; ls -F
```

Увидим следующее содержимое:

```
bin/ dev/ etc/ init* mnt/ mnt-system/ mnt-user/ modules/ proc/ sys/
```

Поискем по наличию слова sda2 и снова изменим sda2 на sda5:

```
grep -rls sda2 etc/ bin/ init
sed -i "s/sda2/sda5/g" init
```

Кстати, скрипт init (в некоторых дистрибутивах – linuxrc) – это тот самый скрипт, в котором и прописываются подготовительные действия для запуска «большого Linux». Если посмотреть его содержимое и сопоставить с настройками загрузчика в следующем шаге – видно, что при выборе пункта Partition Scan выполняется /sbin/scanuser.sh (т.е. проверяется файловая система), а при выборе Restore Factory Settings – /sbin/formatuser.sh (т.е. форматирование раздела /dev/sda2, а теперь – /dev/sda5). Так как все необходимые изменения внесены, создадим новый исправленный initrd, соблюдая формат архива и степень сжатия, как у исходного:

```
cd /tmp/INRD
find . -print | cpio -H newc -o | .J
gzip -9 -c > /mnt/boot/.J
initramfs-sda5.img
```

Шаг 5в: Загрузчик eeeXandros

Теперь исправим записи initrd на свои в конфигурационном файле загрузчика

eeeXandros (/mnt/boot/grub/menu.lst), также добавив в него запись о загрузке загрузчика SLED:

```
hiddenmenu
default=0
timeout=0
fallback=2

title GRUB (SLED)
    rootnoverify (hd0,5)
    chainloader +1

title Normal Boot
    root (0x80,0)
    kernel /boot/vmlinuz-2.6.21.4-eeepc quiet rw vga=785
    irqpoll root=/dev/sda1
    initrd /boot/initramfs-sda5.img

title Perform Disk Scan
    root (0x80,0)
    kernel /boot/vmlinuz-2.6.21.4-eeepc quiet rw vga=785
    irqpoll root=/dev/sda1 XANDROSSCAN=y
    initrd /boot/initramfs-sda5.img

title Restore Factory Settings
    root (0x80,0)
    kernel /boot/vmlinuz-2.6.21.4-eeepc quiet rw vga=normal
    nosplash=y irqpoll root=/dev/sda1 XANDROSRESTORE=y
    initrd /boot/initramfs-sda5.img
```

Рекомендую перед внесением сделать его копию на случай восстановления (cp /mnt/boot/grub/menu.lst /mnt/boot/grub/menu.lst-old). Изменения же следующие: параметр fallback, добавлен новый пункт меню GRUB (SLED) и во всех пунктах заменен initrd на новый.

Теперь подробнее по пунктам:

- ☑ **hiddenmenu** – говорит, что меню загрузчика показываться не будет.
- ☑ **default=0** – указывает на пункт меню, который будет загружаться по умолчанию. Напомню, что в GRUB всё нумеруется с 0, и пункты меню тоже. То есть, default=0 означает, что будет загружаться первый пункт меню – GRUB (SLED).
- ☑ **timeout=0** – указывает время, через которое будет загружаться пункт по умолчанию (в нашем случае это произойдет сразу).
- ☑ **fallback=2** – указывает на пункт меню, который будет загружаться в случае, если не загружается пункт по умолчанию; я выбрал Perform Disk Scan – в этом случае проверяется файловая система на разделе eeeXandros (исначально на /dev/sda2).
- ☑ **rootnoverify (hd0,5)** – указание, с какого раздела грузиться: «(hd0,5)» (он же /dev/sda6) – корневой раздел SLED, в загрузочном секторе которого находится его загрузчик.
- ☑ **chainloader +1** – указание, чтобы грузиться с первого сектора.

Шаг 5г: Загрузчик SLED

Оформим и меню загрузчика SLED (/boot/grub/menu.lst):

```
default 0
timeout 8
gfxmenu (hd0,5)/boot/message
```

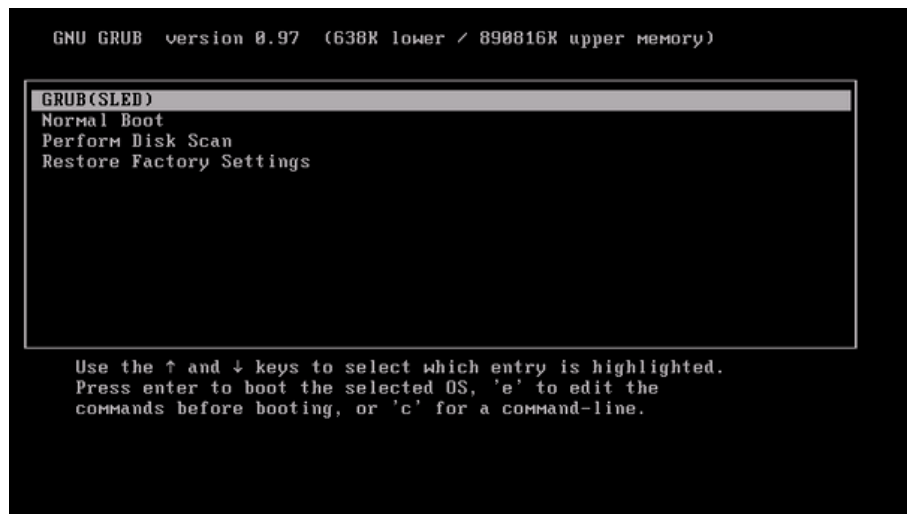
```
title SLED 11
    root (hd0,5)
    kernel /boot/vmlinuz-2.6.27.19-5-pae
    root=/dev/disk/by-id/ata-ST9160310AS_5SV0NGJN-part6
    resume=/dev/disk/by-id/usb-IC25N060_ATMR04-0-0:0-part2
    splash=silent showopts
    initrd /boot/initrd-2.6.27.19-5-pae

title eeeXandros
    root (0x80,0)
    kernel /boot/vmlinuz-2.6.21.4-eeepc quiet rw vga=785
    irqpoll root=/dev/sda1
    initrd /boot/initramfs-sda5.img

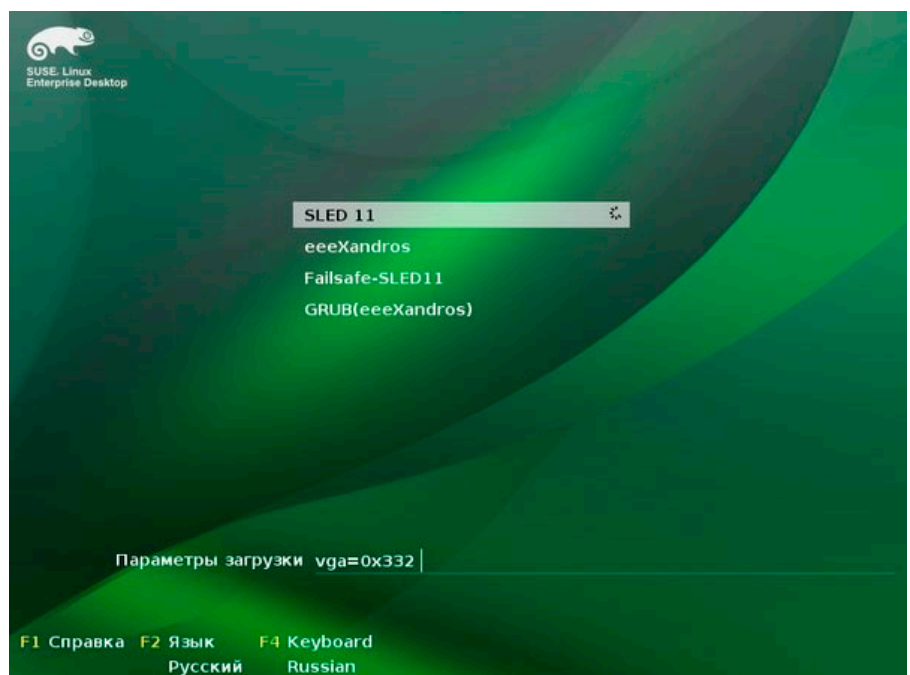
title Failsafe-SLED11
    root (hd0,5)
    kernel /boot/vmlinuz-2.6.27.19-5-pae
    root=/dev/disk/by-id/ata-ST9160310AS_5SV0NGJN-part6
    showopts ide=nodma apm=off noresume nosmp maxcpus=0
    edd=off powersaved=off nohz=off highres=off
    processor.max_cstate=1 xllfailsafe
    initrd /boot/initrd-2.6.27.19-5-pae

title GRUB (eeeXandros)
    rootnoverify (hd0)
    chainloader +1
```

Изменения следующие: добавлены пункты eeeXandros и GRUB (eeeXandros) – по аналогии с настройками меню загрузчика eeeXandros. В итоге, получаем меню с четырьмя пунктами:



Получившийся загрузчик eeeXandros



Получившийся загрузчик SLED

- ☑ SLED 11;
- ☑ eeeXandros;
- ☑ «Безопасный режим для SLED11»;
- ☑ «Переход обратно к загрузчику eeeXandros»

По умолчанию запускается SLED 11 (через 8 секунд ожидания).

Примечание: в полученном SLED нужно заменить источник установки с «раздел USB-диска», например, на ISO-образ, предварительно скопировав его на жесткий диск.

Общий итог

Статья демонстрирует, как можно разобратся в работе ОС и выполнить необходимые изменения в настройках нетбука с Linux.

Немного об установленном SLED на eeePC-1000HD: без дополнительных действий заработали Wi-Fi, видео-

карта (с эффектами Compiz), веб-камера, звук. Не разобрался в медийных клавишах (вроде требуется установка пакета под ееерс) и в работе Bluetooth (за ненадобностью). В общем, то, что было нужно. В результате, у меня большую часть времени на нетбуке запущен SLED и лишь совсем в исключительных случаях – eeeXandros.

И напоследок – о гарантии...

Во-первых, если установить в загрузчике SLED «timeout=0» и «default=1», т.е. сразу же будет запускаться eeeXandros, то внешне система никак не изменилась. С учетом опыта общения с гарантией на другой нетбук могу сказать, что пока техническая поддержка у производителей Linux-нетбуков знаниями «матчасти» не сильно блещет. Впрочем, им это и не очень надо: скинули к заводским настройкам, поставили обновление (если проблема уже извест-

на и исправлена) и всё. Кстати, замечу, что функцию сброса eeeXandros к заводским настройкам мы оставили: это делается при включении компьютера и нажатии на клавишу <F9> до того, как появится меню загрузчика SLED. Причем сброс к заводским настройкам не навредит установленному SLED.

Во-вторых, те, кто сделали «Шаг 2: Ваксир», могут всё вернуть к первоначальному состоянию, переразбив заводский диск (исправляя изменения «Шага 4а», т.е. удалив созданные разделы и вернув один большой /dev/sda2) и восстановив настройки загрузчика eeeXandros, сделанные в «Шаге 5в». В этом случае все полностью вернется к первоначальному состоянию.

Николай Кузнецов
(nick@academy.ru)

Беседы о Qt. Часть 3: Списки

Списки строк

Завершая тему строк, поднятую во второй части статьи (см. «Open Source» 042 от 13.04.2009), рассмотрим их списки. Такими списками в Qt заведует класс QStringList – частный случай QList. Чтобы преобразовать в QStringList экземпляр QString, хранящий массив строк, разделенных переводами строки, достаточно вызвать функцию split(). Например, вот как можно превратить содержимое виджета QPlainTextEdit или QTextEdit из QString в QStringList:

```
QStringList list = textEdit.toPlainText().split("\n");
```

Замечу, что для превращения не всего текста, а только выделенного, функцию split() надо вызывать в параметром QChar::ParagraphSeparator:

```
QStringList list = textEdit.textCursor().selectedText().split(
    QChar::ParagraphSeparator);
```

Для обратного действия – слияния списка строк в одну строку – используем функцию QStringList::join():

```
QString s = list.join("\n").trimmed();
```

Функция trimmed(), вызванная после join(), обрезает начальные и конечные пробелы: ведь при join() в конец строки добавился лишний перевод строки, и от него нужно избавиться. К слову, для этого есть много способов, и trimmed() – один из них, хотя он и удалит пробелы в самом начале. Если они вам нужны – ищите другой способ: например, вызывайте функцию QString::truncate(), чтобы обрезать строку на символе по номеру, равному длине строки. Пример:

```
QString s = list.join("\n");
s.truncate(s.size() - 1);
```

Увы, truncate() объявлена как возвращающая void, поэтому сходу вызвать ее для присвоения чему-либо не получится. Одно из наиболее частых действий, производимых над списком, – это прохождение в цикле по всем его элементам. Qt дополняет язык C++ ключевым словом foreach, удобным в использовании со списками и хэш-таблицами. Пример:

```
QStringList list;
list << "one" << "two" << "three";
foreach (QString s, list)
    qDebug() << s;
```

В каждой итерации цикла переменная s будет содержать значение очередного элемента списка list. Несмотря на очевидную простоту, минусы такого подхода всё же есть. Первый – данные доступны только на чтение. Второй (он же причина первого) – при входе в цикл foreach, контейнер копируется в новый, временный, и цикл работает уже с этой копией.

Если мы хотим изменять значение элементов списка строк, есть два выхода. Например, обращение к элементам с помощью оператора индексирования:

```
for (int i = 0; i < list.size(); ++i)
{
    list[i] = QString::number(i);
}
```

Здесь мы заполняем каждый элемент списка строк числом, соответствующим значению счетчика цикла. А вот еще один способ – прибегнем к помощи итератора QListIterator:

```
QMutableListIterator<QString> i(list);
while (i.hasNext())
{
    QString s = i.next();
    i.setValue("hello");
}
```

Итак, внутри цикла мы можем получить значение текущего элемента, вызвав i.next(), и установить для элемента значение, вызвав i.setValue(). Какой из способов выбирать – решать вам.

И напоследок – сортировка списка строк без учета регистра. В классе QStringList есть функция sort(), но она сортирует, учиты-

вая регистр. Что делать? Прибегнем к помощи класса QMap:

```
QStringList l;  
//Предположим, что l у нас заполнен строками  
  
//Объявим экземпляр QMap:  
QMap<QString, QString> map;  
  
//Пройдемся в цикле по всем элементам l, добавляя их  
//в «карту» следующим образом: в качестве ключа даем  
//переведенную в нижний регистр строку, а в качестве значения –  
//строка как есть, подлинник:  
foreach (QString value, l)  
    map.insert (value.toLower(), value);  
  
//Объявим список для записи результата:  
QStringList sl;  
  
//Просто заполняем его элементами из «карты», которые уже  
//расположены в порядке сортировки  
foreach (QString value, map)  
    sl.append (value);
```

В результате мы получим в sl отсортированный список строк. Причем в список будут помещены строки не в нижнем регистре, а в исходном – поскольку в пару для каждого ключа «карты» мы добавляли именно неизменённую строку.

Отображение списков

Мы наконец вплотную подошли к графическому интерфейсу. Но начну я с вещи, которая поистине является камнем преткновения новичков в программировании не только на Qt, но и GTK+ – особенно если человек раньше видел только Delphi. Речь идет про отображение списков и деревьев. В GTK+ для этого дела вообще есть только API самого низкого уровня, и для вывода простого списка строк надо либо писать много кода, либо сделать для удобства свою библиотеку-оболочку. В Qt – как бы два API: упрощенное (высокого уровня) и сложное (нижнего уровня). Для простейшего случая – вывода строковых значений, по одному в каждом ряду – используется QListWidget:

```
QListWidget *lw = new QListWidget (this);  
lw->addItem ("one");  
lw->addItem ("two");  
lw->addItem ("three");
```

Или же, для добавления заранее подготовленного списка строк, можно добавлять целый список в виде экземпляра QStringList:

```
QListWidget *lw = new QListWidget (this);  
QStringList l;  
l.append ("one");  
l.append ("two");  
l.append ("three");  
lw->addItems (l);
```

Итак, отображение простейшего списка – дело нехитрое, но кроме этого ведь нужно уметь и другие вещи. Например, получать текущий элемент и устанавливать его. Номер текущего элемента возвращает функция QListWidget::currentRow(), а задать такой элемент можно функцией setCurrentRow(). Чтобы получить

текст текущего элемента, прямой функции нет. Надо сначала получить по номеру экземпляр элемента. Делается это так:

```
QListWidget *lw = new QListWidget (this);  
[...]  
int index = lw->currentRow();  
if (index == -1)  
    return;  
QListWidgetItem *item = lw->item(index);  
qDebug() << item->text();
```

Так мы получаем номер текущего элемента: если он равен -1, текущего элемента нет. Обычно так бывает, когда список не получил фокус. Первый же элемент списка идет под нулевым номером. По номеру получаем от QListWidget элемент с помощью функции QListWidget::item(int index). Она возвращает нам объект QListWidgetItem, у которого есть функция text(). Вот она-то и дает текст элемента. Очевидно, что таким образом можно получить не только один элемент, но и пройти по списку в цикле.

У QListWidgetItem есть еще две полезные функции: findItems() и selectedItems(). Первая возвращает список элементов, содержащих подстроку, а вторая – список выделенных элементов. Приведу пример для findItems (в случае selectedItems все делается по аналогии). Итак, заполним список строками и получим все элементы, которые начинаются на dog. Получив их, выведем названия на консоль:

```
QListWidget *lw = new QListWidget (this);  
lw->addItem ("dog");  
lw->addItem ("cat");  
lw->addItem ("doggy");  
lw->addItem ("kitty");  
  
QList<QListWidgetItem *> list = lw->findItems ("dog", Qt::MatchStartsWith);  
foreach (QListWidgetItem *item, list)  
    qDebug() << item->text();
```

Другие важные функции QListWidget:

- ☑ **scrollToItem()** – прокручивает список к указанному элементу (полезно вызывать после setCurrentRow);
- ☑ **removeItemWidget()** – удаляет элемент;
- ☑ **editItem()** – включает встроенный редактор.

Существуют также сигналы для обработки таких событий, как смена текущего элемента и так далее.

В следующей части

Всё это очень просто и удобно, но ведь речь идет о простых «одномерных» списках строк. А что, если нужно несколько колонок? Вывод каких-то иконок? Деревообразные списки? Вот тут оказывается, что придется знакомиться с парадигмой «Модель → Вид» («Model → View») вплотную. Об этом – в следующем выпуске «Open Source».

Петр Семилетов
(tea@list.ru)

«Open Source» приглашает к сотрудничеству!

Электронное приложение «Open Source» всегда открыто для сотрудничества с новыми авторами, с читателями и их конструктивными предложениями по улучшению издания, обоснованной критикой и любыми отзывами, с компаниями, занимающимися разработкой и продвижением программного обеспечения с открытым кодом. Приветствуются все энтузиасты, желающие опубликовать у нас свои статьи. Тематика нужных материалов очевидна из предназначения приложения,

то есть FOSS (Free and Open Source Software): теория и практическое применение; исторические сведения, анализ сегодняшнего положения, прогнозы на будущее и другие аспекты, связанные с открытым ПО.

Среди наиболее интересных на данный момент общих тем можно выделить:

- ✓ общие обзоры новых и/или интересных проектов Open Source и конкретных приложений, свежих версий дистрибутивов Linux, *BSD и других систем;

- ✓ советы и рекомендации новичкам в GNU;
- ✓ истории успеха применения/распространения ПО с открытым кодом;
- ✓ философия и идеология Free Software;
- ✓ разработка приложений с применением средств Open Source.

Желательный объем статей: 6000 или 12000 символов (с пробелами). Примеры актуальных сейчас тем для статей публикуются на <http://osa.samag.ru/todo>. Но не стоит строго ограничиваться приведенными выше рамками!

Публичное обсуждение «Open Source» проводится на форуме сайта журнала «Системный администратор» по адресу: <http://osa.samag.ru/forum>. Связаться с редакцией можно по электронной почте osa@samag.ru.

P.S. За статьи мы платим.

Подписные индексы:

20780*

+ диск с архивом статей
2008 года

81655**

без диска

по каталогу агентства
«Роспечать»

88099*

+ диск с архивом статей
2008 года

87836**

без диска

по каталогу агентства
«Пресса России»

* Годовой
** Полугодовой
*** Диск вкладывается
в февральский
номер журнала,
распространяется только
на территории России

Подписка на журнал «Системный администратор»

Российская Федерация

- ✓ Подписной индекс: годовой – **20780**, полугодовой – **81655**
Каталог агентства «Роспечать»
- ✓ Подписной индекс: годовой – **88099**, полугодовой – **87836**
Объединенный каталог «Пресса России»
Адресный каталог «Подписка за рабочим столом»
Адресный каталог «Библиотечный каталог»
- ✓ Альтернативные подписные агентства:
агентство «Интер-Почта»
(495) 500-00-60, курьерская доставка по Москве
агентство «Вся Пресса»
(495) 787-34-47
агентство «Курьер-Пресссервис»
агентство «ООО Урал-Пресс»
(343) 375-62-74
- ✓ Подписка On-line
<http://www.arzi.ru>
<http://www.gazety.ru>
<http://www.presscafe.ru>

СНГ

В странах СНГ подписка принимается в почтовых отделениях по национальным каталогам или по списку номенклатуры «АРЗИ»:

- ✓ **Азербайджан** – по объединенному каталогу российских изданий через предприятие по распространению печати «Гасид» (370102, г. Баку, ул. Джавадхана, 21)

- ✓ **Казахстан** – по каталогу «Российская пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»
- ✓ **Беларусь** – по каталогу изданий стран СНГ через РГО «Белпочта» (220050, г. Минск, пр-т Ф. Скорины, 10)
- ✓ **Узбекистан** – по каталогу «Davriy nashrlar», российские издания через агентство по распространению печати «Davriy nashrlar» (7000029, г. Ташкент, пл. Мустакиллик, 5/3, офис 33)
- ✓ **Армения** – по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г. Ереван, пл. Сасунци Давида, д. 2) и ЗАО «Контакт-Мамул» (375002, г. Ереван, ул. Сарьяна, 22)
- ✓ **Грузия** – по списку номенклатуры «АРЗИ» через АО «Сакпресса» (380019, г. Тбилиси, ул. Хошараульская, 29) и АО «Мацне» (380060, г. Тбилиси, пр-т Гамсахурдия, 42)
- ✓ **Молдавия** – по каталогу через ГП «Пошта Молдовей» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
по списку через ГУП «Почта Приднестровья» (МД-3300, г. Тирасполь, ул. Ленина, 17)
по прайс-листу через ООО агентство «Editil Periodice» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
- ✓ Подписка для **Украины**:
Киевский главпочтамт
Подписное агентство «KSS»
Телефон/факс (044)464-0220