

Мнение: Как должен вести себя настольный Linux

Шон Оливьеро (Sean Oliviero), Воскресенье, 12 Сентябрь 2004, 20:57

Состояние Desktop Linux

В самом начале был незамысловатый старый Linux с X11 и несколькими уродливыми оконными менеджерами. Используя вечную (не)известную аналогию с машинами, мы можем приравнять это к крутящей ходовой части с мотором, сидением, рулем и коробкой передач. GNOME и KDE дали Linux благоприятную возможность для добавления рамы, ветрового стекла и цельного кузова. Мы больше не глотаем ошибки, но внутренности - это все еще голый металл. И там нет ни стерео, ни коврового покрытия, ни подушек безопасности, ни чего-либо еще. Между тем, мимо проезжает Mac с его роскошным кожаным салоном, спутниковой DVD навигацией и всяческими наворотами. Что не так в этой картине? Да, машина очень хорошо приспособлена именно для вождения, но кто хочет ездить каждый день в недоделаном транспортном средстве?

Недостающие куски в пазле Тукса. Часть I - Оборудование

Linux построен слоями. Каждый слой основывается на другом, начиная от ядра и заканчивая графическим интерфейсом. Но что объединяет их все вместе? Как самый низ связывается с самым верхом и наоборот? Как я могу конфигурировать оборудование из графической панели управления в GNOME? Ответ: "", потому что GNOME еще не связан с этой частью моей системы. Почему я должен указывать Linux, какие диски есть в моей системе? Потому что он не знает! А Windows и Mac OS знают, когда я добавляю к своему винчестеру логический диск.

Как решить проблему оборудования?

На нижнем слое должен быть написан демон, который интеллектуально связывается с оборудованием. Он должен уметь находить новое оборудование, находить соответствующий модуль ядра для него, дать программам сверху знать, когда он не может инициализировать устройство и автоматически конфигурировать файлы `/etc/modules.conf` и `/etc/fstab`. Наверху должен быть написан графический апплет, с помощью которого пользователь может включать/отключать устройство и вручную переключать модули ядра. Диски должны автоматически монтироваться во время их установки, а если они нечитаемы, должна запуститься дисковая утилита для их инициализации. Также должны быть созданы апплеты для конфигурирования специальных возможностей оборудования, таких как FSAA на видео-карте, причем, желательно, самими производителями оборудования.

Недостающие куски в пазле Тукса. Часть II - Программы

Я только что купил новый компьютер с Linux. Я включаю его и хочу сконфигурировать его для работы с Интернет-соединением и домашней сетью. Но подождите, каким способом я это делаю? На системе с Windows, я должен всего лишь нажать на иконку "Сетевые соединения" и указать адреса моих IP/DNS серверов, настроить фаерволл и общие ресурсы для домашней сети. На Linux я должен найти что за файл в директории `/etc` содержит конфигурационные настройки для моей сетевой карты и попытаться представить, как сконфигурировать Samba и

Мнение: Как должен вести себя настольный Linux

<http://www.osrc.info/plugins/content/content.php?content.64>
файрволл таким образом, чтобы я мог сделать общедоступными мои файлы без

необходимости ковыряться в исходниках системы. Кроме того, для дополнительной безопасности я хочу установить антивирус. Однако простого и ясного пути установки этой программы не существует. Я не понимаю скрипты оболочки и этот идиотский менеджер пакетов ругается на отсутствие зависимостей, какими бы они не были. Кстати, где хорошая дисковая утилита чтобы я мог настраивать и проверять свои жесткие диски в GNOME?

Как решить программную проблему

Создать апплет все-в-одном, который конфигурирует все сетевые устройства, настраивает общий доступ к файлам и защищает все ваши документы в сети, вместе с обычными диагностическим утилитами, такими как ping и traceroute. Второе, сделать систему установки, которая будет решать все зависимости и давать возможность безболезненно их поддерживать. Для этого я предлагаю autopackage (<http://www.autopackage.org>) с каким-нибудь лучшим внешним интерфейсом.

Вещи, которые требуют серьезного обновления

Что первым приходит вам в голову, если вы думаете о том, что необходимо серьезно обновить в Linux? Если вы подумали о X11, вы на 100% правы! Но почему? Давайте немного погрузимся в историю. Первая цель в жизни X11 была производить запуск нескольких терминальных окон одновременно и иметь возможность быть запущенным удаленно через сеть. Времена меняются, но X11 изменился мало, и на данный момент его возраст составляет 20 лет. X11 большой, жирный и не поддерживает современные возможности, такие как альфа-сопряжение и переходы. X11 действительно должен быть заменен чем-то более маленьким и современным.

Как исправить X11

Пожалуйста, поддержите Y Window System [<http://www.y-windows.org>]. Уже не осталось возможности исправить X11, если не привлекать супер-гениальных хакеров. Вылизывание кода снова и снова сделают его только еще большим и более раздутым. X11 должен уйти в тень. Y Windows - естественный выбор, так как проект планирует реализовать совместимость с X11 для упрощения миграции. [От редактора OSNews: Несмотря на то, что новые идеи приветствуются, проверенный код не может и не должен выбрасываться. Почитайте JoelOnSoftware для того, чтобы понять почему нужно работать над правкой существующего кода вместо изобретения велосипеда.]

Взаимодействие и вопросы эстетики

У нас есть GNOME и у нас есть KDE. Ни один из красавцев не взаимодействует друг с другом. Каждый имеет свой собственный набор украшений интерфейса, диалогов сохранения/загрузки и свои руководства. Когда вы запускаете приложение KDE из GNOME, вы прекрасно знаете, что вы запускаете не GNOME приложение и наоборот. Я не хочу иметь возможность сказать, что это за приложение, я всего лишь хочу единства.
[newpage]

Как исправить KDE и GNOME

Программы, использующие GTK+ и Qt не должны соперничать друг с другом, они должны приветствовать друг друга. Они должны использовать философию "Когда в Риме..." ("When In Rome...") в отношении взаимодействия друг с другом. Другими словами, когда программа KDE запускается в GNOME, она должна работать и вести себя так, как программа GNOME и наоборот. Это значит, иметь те же самые рюшечки и использовать те же самые диалоги открытия/сохранения и выбора шрифтов. Подумайте над тем, как взаимодействуют между собой приложения Cocoa и Carbon. Я уверен, это может быть реализовано.

Основная философия

И к выгоде, и к ущербу, главная философия написания программ для Linux и UNIX - "Сделано чокнутыми для ". К сожалению, это заклинание не работает удовлетворительно для проектирования программного обеспечения, предназначенного для среднего пользователя. Люди не любят лазить по обширной и комплексной файловой системе и копаться в конфигурационных файлах, чтобы получить то, что они хотят. Домашняя папка пользователя должна быть центром рабочего стола Linux в большей степени, чем корневая файловая система. Папки, имеющие отношение к UNIX в корневой файловой системе должны быть скрыты по умолчанию. Все программы, которые имеют возможность расширения, должны сами предоставлять средства для установки расширений и плагинов. К примеру, обои для рабочего стола пользователя должны храниться в его/ее директории картинок. Нет причины заставлять большинство пользователей шнырять по корневой файловой системе, где он может поудалять системные файлы. Так или иначе, специальные секции для таких вещей, как плагины и данные приложений могут храниться на виду в корневой файловой системе до тех пор, пока они в нужном месте, как внутри папки Library в Mac OS X. Вкратце, больше думайте о прорезинном пластике и меньше о голом металле при проектировании своих программ.

Другие идеи

Так как новых хороших идей для Desktop Linux мало, я предложу еще несколько:

1. Объединенное хранение. (Pooled Storage) Скажем, у вас кончается место на жестком диске. Вместо того, чтобы обновляться до жесткого диска с большим объемом, вы покупаете другой, подключаете его и пространство объединяется в одну гигантскую цистерну свободного места. Добавление места на диске так же просто, как добавление оперативной памяти. Конечно, это не возможно воплотить для маленьких съемных дисков, но это делает расширение места на жестком диске намного проще.
2. Плагины-просмотрщики для Nautilus - Представьте, что вы открываете вашу папку с музыкой и панель просмотра превращается в музыкальный плеер. Если вы хотите просматривать как файлы, вы всего лишь должны отметить пункт "Показывать как " в меню "". То же самое должно быть сделано для почты, контактов, скачанных файлов, картинок и фильмов. Всего лишь сделайте плагин для просмотра, который дает возможность работать с данными.
3. Установка программного обеспечения через web. - Я знаю, что это не совсем новая идея, но давайте я объясню. Мир программного обеспечения меняется. Мы движемся от коробочного ПО к закачке дистрибутивов из Интернета. В конечном счете, сами программы не будут больше находиться на нашем компьютере во многих случаях. Конечно существует ClickNRun Warehouse от Linspire, но это закрытая система. Нам нужна открытая система. Тут я рекомендую систему Zero-Install (<http://zero-install.sourceforge.net>) для web-based программ. Но

Мнение: Как должен вести себя настольный Linux

<http://www.osrc.info/plugins/content/content.php?content.64>

для больших программ, которые измеряются десятками сотен мегабайт мы должны иметь

что-то открытое и интуитивное. Плагин для веб-браузера, который связывает зачатки с Autopackage будет хорошим выбором. Таким образом, когда вы нажимаете на ссылку, чтобы скачать программу, плагин всплывает и обрабатывает зачатку и установку программы.

4. Окно приветствия - Эта программа может быть реализована как окно, которое всплывает при входе в систему или как часть фона рабочего стола, а-ля gDesklets. Это программа проверяет есть ли у вас новая почта, показывает приятелей, находящихся в онлайн, автоматически проверяет обновления программного обеспечения, получает заголовки RSS и информацию о погоде в простом к использованию и настройке пакете. То же самое может быть сделано с gDesklets, но, к сожалению, это не слишком дружелюбная программа.

Заключение

Людам есть что сказать о состоянии Desktop Linux. В большинстве своем, они всего лишь бездельничают и жалуются о его недостатках, как я, но некоторые предлагали детальные решения проблем, включая некоторые новые идеи. Большинство проблем проистекает из-за незаконченного и/или неработоспособного кода. Однако, большой кусок этого имеет отношение к общим основам философии проектирования программного обеспечения с открытыми исходными кодами. Люди любят изобретать велосипед, придумывая свои собственные идеи и делать "пиф-" всем остальным. Такое поведение не может продолжаться более, если мы хотим сильную, конкурентноспособную альтернативу проприетарным операционным системам. Людям необходимо организовать и работать вместе, не против друг друга и не на дистанции. Да, два-три проекта могут соревноваться друг с другом за сердца и разумы пользователей Linux, но их не должно быть десять или двенадцать. Я надеюсь, эта статья сможет побудить разработчиков программного обеспечения для Linux работать вместе и разморозить ледяную поступь Desktop Linux, вместо борьбы друг с другом и повторного изобретения того, что уже было изобретено. Слишком много ценного времени и энергии было выброшено этим путем.

Об авторе:

Я - Шон Оливьеро (Sean Oliviero). Я алчный компьютерный пользователь, игравшийся почти со всеми известными ОС. Мечта моей жизни - увидеть как Linux станет выбором среднестатистического пользователя, не только простая в использовании, но и способная к конкуренции по возможностям и удобству с Windows и Macintosh.

