

Annotation

Это попытка последовательного изложения истории UNIX, Linux и свободных ОС вообще, а также связанных с ними графических интерфейсов. Она разделяется на три части: в первой рассматривается история UNIX-подобных операционных систем, во второй – дистрибутивов Linux, в третьей – их интерфейсов. Основана на печатных и сетевых материалах, воспоминаниях очевидцев, устной традиции и личных впечатлениях.

- [Вопросы истории: UNIX, Linux, BSD и другие](#)
 -
 - ■ [Предисловие](#)
 -
 - [Преамбула](#)
 - [Референсы](#)
 - [Реверансы](#)
 - [Часть I. История систем](#)
 -
 - [Глава первая. Вопросы праистории](#)
 -
 - [Вступление](#)
 - [Истоки и составные части](#)
 - [Computer Science и ARPANET](#)
 - [Зарождение UNIX](#)
 - [«Железные» предпосылки](#)
 - [Компьютеры для народа](#)
 - [Глава вторая. Берклиада](#)
 -
 - [«Похищение Елены Прекрасной»](#)
 - [Начало свободного софта](#)
 - [Возвращение в Беркли](#)
 - [Интермедия](#)
 - [FreeBSD: рождённая свободной](#)
 - [Технологический детектив](#)

- [Цена свободы](#)
- [Глава третья. Возвращаясь к UNIX'ам](#)
 -
 - [Серенада солнечной долины](#)
 - [В оковах проприетаризма](#)
 - [Битва за стандарты](#)
- [Глава четвёртая. Рождение Linux](#)
 -
 - [Предпосылка первая, «железная»](#)
 - [Предпосылка вторая, «денежная»](#)
 - [Предпосылка третья: «студенческая операционка»](#)
 - [Герой выходит на сцену](#)
- [Глава пятая. Берклиада, тур второй](#)
 - [NetBSD и OpenBSD: первый форк в благородном семействе](#)
 - [FreeBSD: десятилетие спокойствия](#)
 - [FreeBSD: переломный год](#)
 - [Оглядываясь вокруг](#)
 - [Предварительный итог...](#)
 - [FreeBSD на Руси](#)
- [Глава шестая. Последний взлёт индивидуального ОСетворчества](#)
 -
 - [Пролог](#)
 - [Чем была AtheOS](#)
 - [Как она получалась...](#)
 - [Как устанавливалась...](#)
 - [И как работала](#)
 - [Эпилог](#)
- [Глава седьмая. «Драгонада» Мэтта Диллона](#)
 -
 - [«Железная» ретроспектива](#)
 - [Софтверная перспектива](#)
 - [У истоков новой системы](#)
- [Глава восьмая. MINIX 3: вторая жизнь](#)
 - [Вступление](#)
 - [О микроядрах](#)

- [Собственно о MINIX 3](#)
 - [Чего нет у_рыб?](#)
 - [Заключение](#)
- [Глава девятая. UNIX: хождения в народ](#)
 -
 - [Зачем нужен десктопный UNIX](#)
 - [Блеск и нищета персональных Workstation'ов](#)
 - [Sun Solaris: глотки свободы](#)
 - [Мир без Солнца, или бессмертна ли мафия?](#)
 - [Дети Солнца](#)
- [Глава десятая. Из истории файловых систем](#)
 -
 - [Вводные слова](#)
 - [Дисковая разметка](#)
 - [Массивы и логические тома](#)
 - [Файловые системы](#)
 - [Из истории систем размещения](#)
 - [Начало истории ZFS](#)
 - [Из истории юриспруденции](#)
 - [ZFS on Linux: технология против крюкотворства](#)
- [Часть II. История дистрибутивов](#)
 - [Глава одиннадцатая. Начало дистрибуции](#)
 -
 - [Был ли дистрибутивом первозданный Linux?](#)
 - [Начало начал](#)
 - [Slackware: первый шаг к Linux'у для всех](#)
 - [Debian: второй шаг к пользователю](#)
 - [Red Hat: совсем для всех?](#)
 - [Глава двенадцатая. Лицом к пользователю](#)
 -
 - [Suse: по стопам Red Hat](#)
 - [Caldera: предыстория грядущего](#)
 - [Магия Linux](#)
 - [Назад, в будущее: Gentoo и другие](#)
 - [На пути к гармонии](#)
 - [Глава тринадцатая. Linux на Руси](#)
 -

- [Предыстория русского Linux'а](#)
- [IP Labs Linux Team: начало русского Linux'а](#)
- [Новые игроки на Linux-поле](#)
- [Горизонты раздвигаются](#)
- [Единение с миром](#)
- [Глава четырнадцатая. Slackware и первые СБР](#)
 - [Slackware и потомки](#)
 - [Первые СБР](#)
 - [Путь Дзэн](#)
 - [Предназначение](#)
 - [И несколько последних слов](#)
- [Глава пятнадцатая. Debian: история клонирования](#)
- [Глава шестнадцатая. Ubuntu и его клан](#)
 - [Начало истории](#)
 - [Распространение](#)
 - [Появление «разновидностей»](#)
- [Глава семнадцатая. SUSE в истории](#)
 - [Представление семейства](#)
 - [Из предыстории](#)
 - [Начало самостоятельного плавания](#)
 - [К технологическим высотам](#)
 - [Рождение openSUSE](#)
 - [Соглашение с дьяволом](#)
 - [Парадоксы бизнеса](#)
 - [Продажа бессмертной души](#)
 - [Последствие](#)
 - [SUSE на Руси](#)
- [Глава восемнадцатая. Год великого перелома](#)
 - [Этапы Большого пути](#)
 - [Почему 2005-й?](#)
 - [Последствия первые...](#)
 - [... и отдалённые](#)
 - [«Десктопизация» Fedora](#)

- [Не Fedora единой...](#)
- [Конец истории](#)
- [Глава девятнадцатая. Феномен Ubuntu](#)
 - [Немного ретроспективы](#)
 - [Появление героя](#)
 - [Реакция](#)
 - [Итог](#)
- [Глава двадцатая. Linux: история применителей](#)
 - [Очень древнее вступление](#)
 - [Становление сословия пользователей](#)
 - [Прародитель «персонализации» Linux'a](#)
 - [Провал первой «персонализации»: технологические причины](#)
 - [Провал первой «персонализации»: психологические причины](#)
 - [Linux: «персонализация снизу»](#)
 - [Заключение](#)
- [Часть III. История интерфейсов](#)
 - [Глава двадцать первая. История shell'ов](#)
 - [Появление первого шелла](#)
 - [Появление C-Shell](#)
 - [История tcsh](#)
 - [Наконец, Z-Shell](#)
 - [Глава двадцать вторая. Из истории Иксов](#)
 - [Вступление](#)
 - [Доисторический период](#)
 - [Рождение Иксов](#)
 - [XFree86 и другие](#)
 - [XFree86: немного о внутренностях](#)
 - [От XFree86 к Xorg](#)
 - [Хроника поступательного движения](#)
 - [Предпосылки Великого раскола](#)
 - [Повод к Великому расколу](#)

- [Конец XFree86...](#)
 - [... и начало Xorg](#)
 - [Глава двадцать третья. Управители окон: извлечения из истории](#)
 -
 - [Терминологическое введение](#)
 - [Предтечи управителей](#)
 - [У истоков управления окнами: twm...](#)
 - [... и его линия](#)
 - [По стопам Windows 95](#)
 - [По следам легенды](#)
 - [Линия боксов](#)
 - [И всё же ещё минималистичней...](#)
 - [Управители тайлингом](#)
 - [Заключение](#)
 - [Глава двадцать четвёртая](#)
 -
 - [Вступление](#)
 - [Предыстория десктопов](#)
 - [Рождение KDE](#)
 - [KDE – в жизнь](#)
 - [Среда для холериков](#)
 - [Радетели свободы](#)
 - [Великое противостояние](#)
 - [От KDE 4 до GNOME 3](#)
 - [А что же другие?](#)
 - [Послесловие](#)
- [notes](#)
 - [1](#)
-

Вопросы истории: UNIX, Linux, BSD и другие

Краткий курс для времени угара Алексей Федорчук aka alv

Предисловие

Это попытка последовательного изложения истории UNIX, Linux и свободных ОС вообще, а также связанных с ними графических интерфейсов. Она разделяется на три части: в первой рассматривается история UNIX-подобных операционных систем, во второй – дистрибутивов Linux, в третьей – их интерфейсов. Основана на печатных и сетевых материалах, воспоминаниях очевидцев, устной традиции и личных впечатлениях.

Преамбула

Эта книга сочинялась на протяжении многих лет, начиная, наверное, с первых жней III тысячелетия нашей эры. Однако точка в ней была поставлена в момент, который может стать завершающим в истории всех её объектов и персонажей. В момент, когда сами понятия *UNIX* и *UNIX-подобные операционные системы* готовы превратиться в музейные экспонаты. Которые и сами-то представляются ненужными, а уж рассмотрение их истории – и подавно. Ибо, как сказал некогда главный герой наших классиков,

...теперь многие не знают имен героев. Угар НЭПа. Нет того энтузиазма.

Так вот, чтобы напомнить кое-какие имена и события, я и решил собрать воедино все свои прежние записки на исторические темы, причесать их и свести в единую книжку.

Описать историю всех свободных ОС и хотя бы основных их дистрибутивов (не говоря уже о графических системах и приложениях) оказалось задачей, для меня непосильной. Поэтому я ограничил свою задачу в основном теми ОСями, дистрибутивами и графическими системами, с которыми практически работал и за развитием которых следил. В результате в моём повествовании образовался ряд лакун, о наиболее существенных из которых будет сказано в заключении.

Вполне возможно, что я упустил и какие-то другие важные линии развития UNIX, их «подобий» и FOSS вообще. Хотелось бы надеяться,

что все эти лакуны будут когда-нибудь восполнены другими авторами, более информированным в этих кругах вопросов, чем я.

Наконец, я полностью и сознательно исключил из своего сочинения историю прикладных программ. Хотя в развитии текстовых редакторов, офисных пакетов и ряда других классов приложений можно найти немало увлекательных сюжетов. Но тут могу оправдаться стопроцентно – словами Козьмы Пруtkова, лейтмотивом сочинений которого было:

Нельзя объять необъятное!

Книга не представляет собой описания событий в хронологическом порядке – по причинам, о которых говорится во вступлении к главе первой, это невозможно. Поэтому она разделена на три части, сюжеты которых развиваются параллельно. Первая часть посвящена истории операционных систем семейства UNIX и им подобных – от предпосылок их возникновения до создания Linux, с одной стороны, и развития BSD систем вплоть до нашего времени – с другой.

Сюжет второй части ответвляется от первой с момента оформления Linux как общедоступной операционной системы. И посвящён он истории её дистрибутивов – от появления первых систем распространения этой ОС до почти настоящего времени, когда вопрос этот переходит из сферы истории в область политики.

Наконец, в третьей части рассмотрена история графических интерфейсов UNIX-подобных операционок – оконной системы X, менеджеров окон и интегрированных рабочих сред. А в качестве вводного материала затронута история текстовой консоли и интерфейса командной строки, то есть командных оболочек (shell).

Ранее фрагменты их этой книги размещались на разных моих ресурсах, в том числе и на Блогосайте, а также печатались в журнале Linuxformat с июня 2011 по декабрь 2013 года в виде исторического цикла статей. И все они включены в неё в расширенном, взаимоувязанном и так или иначе переработанном виде. Так что как системная целостность содержание этой книги публикуется впервые.

Референсы

Эта книга предназначена в первую очередь для оффлайн-чтения с ебуков, планшетов и тому подобных смартфонов. Поэтому в её тексте не будет активных ссылок, задумчиво называемых пруфлинками (proof link). Да и смысла в них я не вижу – на самом деле эти самые «пруфы», как правило, ничего не доказывают, а, в лучшем случае, только иллюстрируют сказанное автором. А в некоторых случаях и просто не имеют к нему отношения.

Тем не менее, в основе этого сочинения лежат, разумеется, некоторые источники – материалы официальных сайтов проектов, списки рассылки, актуальные на момент выхода данной версии системы или дистрибутива обзоры, тексты старых книг по UNIX, старые журналы и тому подобные «бумажные» и сетевые материалы. Многие из которых или отыскиваются с трудом, или вовсе недоступны.

Поэтому одним из существенных источников книги стали личные воспоминания и впечатления автора – начиная с 1996 года, он был свидетелем большинства описанных в ней событий, а с 1998 года – и участником некоторых из них. Причём, большая их часть документировалась им «по горячим следам», что нашло отражение в моих журнальных и сетевых публикациях, приводить ссылки на которые было бы скучно. В случае расхождения своих воспоминаний с более иными «первоисточниками», типа Википедии, я, по понятным причинам, отдавал предпочтение первым.

Кроме того, многие из приводимых здесь данных получены в ходе общения, реального и виртуального, с участниками описываемых событий. Так что эта книга – своего рода и летопись устной традиции, сохранявшейся старыми юниксоидами и линуксоидами. Ибо, как показывает опыт истории, рукописи горят (да ещё и ярким пламенем), сайты исчезают также легко, как и возникают, и только память человеческая хранит воспоминания о событиях. По крайней мере, до тех пор, пока они кому-то ещё интересны.

Реверансы

Как я только что сказал, эта книга в значительной мере основана на сведениях, полученных от моих личных знакомых, коллег и

товарищей – носителей устной и сетевой традиции мира FOSS. И потому я хотел бы выразить признательность:

- Алексею Костареву – создателю одного из первых в стране Linux-порталов на сайте **Невод** и автору первой русскоязычной книги, написанной о свободном софте – и исключительно его средствами;

- Алексею Новодворскому и Алексею Смирнову – основателям IPLabs Linux Team (ныне – фирма Altlinux), стоявшей у истоков Русского Linux'a;

- Виктору Костромину – создателю Виртуальной энциклопедии «Linux по-русски», хранящей память о всех русскоязычных сетевых материалах, затрагивающих нашу тематику, начиная с 1999 года;

- Елене Толстяковой, Валентину Синицыну и Кириллу Степанову – сотрудникам редакции журнала LinuxFormat в прошлом и настоящем;

- участникам Джуйки, форумов POSIX.ru и UNIXforum.org – поимённый их список превысил бы эту книгу по объёму.

Отдельная благодарность – моим коллегам и старым товарищам: Сергею Голубеву, Владимиру Попову, Владимиру Родионову. Все затронутые в книге вопросы многократно обсуждались с ними в реале и виртуале.

И самая большая благодарность, последняя по счёту, но первая по значению: Ирине Киттель, известной как Алиса Деева, вдохновившей меня на завершение этой работы, начатой более десяти лет назад. Со сборником её жизнеутверждающих стихов и разножанровой «малой» прозы заинтересованный читатель может ознакомиться в **Библиотеке Блогосайта**.

Часть I. История систем

В части первой рассказ пойдёт об истории операционных систем, начиная с предпосылок возникновения BSD и Linux'a. При этом история первых будет доведена почти до наших дней , а история второго – до появления первых его дистрибутивов.

Глава первая. Вопросы праистории

Было бы заманчиво, следуя заветам дедушки Ленина, свести праисторию свободных UNIX-подобных систем к трём истокам и трём составным частям. Однако реальная история чего бы то ни было гораздо сложнее (и интересней), нежели нам пытались внушить классики марксизма и их преложители. Так что с какой стороны ни смотри, а истоков и составных частей Linux'a оказывается куда больше трёх. И к тому же не всегда можно однозначно сказать, где кончаются истоки и начинаются составные части. Если, конечно, не внять мнению резонных людей из Одессы и не признать, что составные части начинаются именно там, где кончаются истоки.

Вступление

В истоках современного мира FOSS лежит целый ряд явлений, исторически независимых, но тесно переплетавшихся во времени. Настолько тесно, что порой трудно определить, где кончается одна история и начинается другая. Если, конечно, не внять мнению резонных людей из Одессы и не признать, что другая история начинается именно там, где кончилась одна.

Оценивать относительную роль каждого из этих явлений в становлении FOSS – занятие бессмысленное: без любого из них FOSS в современном его виде не мог бы ни возникнуть, ни развиваться. Тем более, что все они оказались настолько тесно переплетенными между собой, что разделить их подчас не легче, чем расплести судовой канат. И потому, следуя совету резонных людей, и не будем пытаться этого делать.

Так что дальнейшая история окажется весьма нелинейной. Она распадается на несколько отдельных сюжетов, протягивающихся параллельно, как бы независимо друг от друга – но в тесном взаимовлиянии. Повествователь лишен средств передать это литературными средствами. Остаётся только вспомнить слова, которые часто встречаются в исландских сагах:

События описаны здесь так, как будто они случились друг за другом, хотя некоторые из них случились одновременно.

Вообще, понятие времени в исландских сагах трактуется очень своеобразно: продолжительность его измеряется не какими-то внешними единицами, основанными на вращении небесных тел, а исключительно внутренней насыщенностью событиями. Например, в сагах обычен такой оборот:

Некоторое время после этого ничего не происходило.

Саги – не только литературные произведения, но и исторические источники, хотя и весьма своеобразные. И большая часть того, что в них описывается, происходило на самом деле. Так что иногда по косвенным данным (например, по сопоставлению с собственно историческими источниками из сопредельных стран, содержащих датировки в общепринятом для нас ныне формате) удаётся определить абсолютную продолжительность того «некоторого времени», когда не происходило ничего, заслуживающего быть помещённым в контекст сюжета саги. И оказывается, что такие периоды могли длиться годами, а то и десятилетиями. Для авторов и читателей саг они выпадают из течения времени, находятся вне его. А может быть, как предполагал Стеблин-Каменский, в это время для них времени просто не существовало...

Но потом действие начинает разворачиваться, как отпущенная пружина, – и время появляется вновь: в короткий по абсолютному (то есть внешнему) исчислению его отрезок событий впрессовывается столько событий, что их хватает на целую сагу. Время, измеренное по внутренней, событийной шкале, оказывается чрезвычайно длинным.

И здесь я должен подчеркнуть, что всё сказанное в предыдущих абзацах, относится к истории не только операционных систем, описываемой в этой части, но, в не меньшей степени, дистрибутивов и интерфейсов, составляющих предмет частей второй и третьей, соответственно. Вообще, вся история FOSS и состоит из чередования таких насыщенных событиями моментов и периодов межвременья,

осложняясь собственными сюжетными разветвлениями и наложением внешних факторов. Чем и напоминает исландскую сагу.

Для чего нужно столь подробное введение? Обосную словами Мэтта Диллона, в прошлом одного из ключевых разработчиков FreeBSD, а ныне создателя её форка – операционной системы DragonFlyBSD:

Самой большой ошибкой, которую может допустить программист, является игнорирование истории, и это именно та ошибка, которую сделали разработчики многих других современных операционных систем... Я плохо переношу тех, кого история не учит.

Кроме того, история ОСей мира FOSS просто чрезвычайно увлекательна, а местами и полна драматизма, как хороший приключенческий роман. Надеюсь, на страницах этой книги мне удастся хоть в какой-то мере передать те чувства, которые испытывал сам, когда с нею знакомился.

Истоки и составные части

Последуем совету упомянутых выше резонных людей и рассмотрим истоки и составные части Linux'a одним списком, предоставив читателю самому решать, где кончается одно и начинается другое. Итак, начнём перечень.

Во-первых, это академическая и университетская Computer Science эпохи «больших машин», в частности, работы по искусственному интеллекту.

Во-вторых, сеть ARPANET и все сопряжённые с ней явления, из которых вырос в итоге современный массовый Интернет. Ибо только нынешние «одноглазые» полагают, что Интернет был придуман для того, чтобы они могли сидеть «ффконтакте».

В-третьих, это корпоративный UNIX и организации по упорядочиванию его стилей, иначе говоря, стандартизации.

В-четвёртых, это Берклиада – история, полная драматизма, в которой, как и в поэме Гомера, впервые переплелись перечисленные выше пряди. Но о ней будет подробно говориться в следующей главе.

В-пятых, это общественные движения – Open Source Software и Free Software. Здесь я немного отступаю от хронологии – второе формально предшествовало первому. До de facto первое явление существует как минимум со времён Ньютона и Лейбница.

В-шестых, это эволюция аппаратных платформ – от первых интерактивных рабочих станций до «народных» x86-совместимых компьютеров.

Как видите, получается куда больше, чем сакральная троица, которую так любят бог и большевики. И это при том, что я не уверен, вспомнил ли я все истоки и все составные части – что не страшно, в этом цикле удастся осветить далеко не все даже из перечисленного.

Причём о некоторых истоках и составных частях я умолчал сознательно – например, об эпохе Eniac'a и БЭСМ'ов, ибо нет у меня о них достаточно информации. Надеюсь, что...

... Кто-нибудь услышит, Снимет и напишет, Кто-нибудь помянет...

... участников тех давних проектов. Они заслуживают этого не меньше, чем создатели первого реактора... Да и чем многие другие, которых тоже ещё не помянули.

А теперь вернёмся к нашим истокам и их составным частям. Поскольку, как уже было сказано, намеченные сюжетные линии развивались параллельно во времени, рассказывать эти истории постараюсь тоже параллельно – и не только в этой главе.

Computer Science и ARPANET

История академической Computer Science уходит в начало 60-х, время появления первых компьютеров, способных к интерактивной работе. Хотя они далеко ещё не были персоналками – но ведь ранее машины, как помнят читатели «Понедельника», почему-то начинавшегося тогда «в субботу», работали исключительно в режиме пакетных заданий. И влиять на это не могли не только маги вроде Кристобая Хозевича и Фёдора Симеоновича, но даже всемогущие научные администраторы, товарищи Камноедов и Лавр Федотыч.

Место же зарождения этой науки условно определим как крупнейшие американские университеты – Массачусетский Технологический Институт (MIT), Йель, Университет Карнеги-Меллона, Стэнфорд, Калифорнийский университет Беркли. Историческим центром этого движения долгое время была лаборатория искусственного интеллекта MIT (MIT AI – Artificial Intelligence). В недрах MIT AI родился, судя по многим свидетельствам, и термин «хакер» – так называли друг друга те, кто способен был «врубиться» в компьютерные науки. Но эта тема столь жёвана и пережёвана, что на ней мы останавливаться не будем

Работы же по созданию отказоустойчивой правительственной связи США, как нетрудно догадаться, начались по инициативе Министерства обороны этой страны. Ибо имели целью создание надёжной системы передачи информации на случай советского ядерного удара. Финансирование осуществлялась через ARPA – Агентство передовых исследовательских проектов (Advanced Research Projects Agency), которое позднее, без лишнего лицемерия, было переименовано в DARPA, с добавлением слова Defense (в данном контексте – Оборонных проектов). Запомним последнюю аббревиатуру – позднее эта организация сыграет немалую роль в нашей предыстории.

Непосредственная реализация системы связи была возложена на ряд американских университетов – Калифорнийский, Университет штата Юта, Стэнфорд. Потому что, как оказалось, кроме университетских хакеров из сферы Computer Science, разрабатывать и поддерживать её было попросту некому. А эти «ребята, за ту же зарплату», не только выковали электронный щит своей Родины в виде сети ARPANET (по имени организации-кормильца), но, будучи истинными учёными, воспользовались случаем в интересах науки. А именно – наладили бесперебойные каналы обмена информацией между своими Alma mater, создав таким образом сообщество ARPANET – прообраз грядущего Интернет-сообщества.

Первый сеанс связи в рамках проекта ARPANET состоялся 29 октября 1969 года в 21 час по местному времени. И оказался не вполне удачным: удалось передать только три символа, после чего сеть рухнула. Однако уже через два часа работоспособность её была восстановлена (учитесь, нынешние провайдеры!), и передача

завершилась успешно. Интересно, что контроль передачи осуществлялся почти тем же методом, который изображён в фильме «Волга-Волга»: не с помощью рупора, конечно (от Лос-Анжелеса до Пало Альто докричаться проблематично), но по телефону.

Сеть ARPANET очень быстро охватила не только университеты, участвовавшие в её разработке, но и многие другие учебно-научные заведения Америки, а потом и сопредельных стран, став таким образом международной коммуникационной магистралью для обмена научной информацией. Правда, скоро её в этой роли сменила сеть Национального научного фонда США (NSF – National Science Foundation), создавшего свою сеть, NSFNet, обеспечивавшую большую пропускную способность. Именно на её базе и был создан современный Интернет.

Зарождение UNIX

Зарождение UNIX, как и сообщества Computer Science, также связано с появлением компьютеров, пригодных к использованию в интерактивном режиме, что создало предпосылки к разработке тех самых систем разделения времени, допускающих как бы одновременное исполнение нескольких задач (time sharing), которые пришли на смену машинам, работавшим исключительно в пакетном режиме. Одной из первых таких систем была CTSS (Compatible Time Sharing System).

Без академической составляющей, представленной в данном случае MIT, не обошлось и здесь. В развитие CTSS в 1965 году фирмами AT&T и General Electric вместе с MIT был начат проект по созданию истинно многозадачной и многопользовательской системы, которая получила имя Multics. По замыслу она была столь прогрессивной, что в те времена оказалась нереализуемой, и в 1969 году проект был закрыт, оставив среди его участников тоску по интерактивной работе и идею системы разделения времени, вскоре воплотившуюся в UNIX.

Правда, сама ОС UNIX вышла из корпоративных недр компании AT&T, сотрудниками которой являлись его создатели – бывшие участники проекта Multics. Однако это ни в коей мере не была корпоративная разработка: Кен Томпсон и Деннис Ричи разрабатывали

её для собственных потребностей – это был первый в истории IT пример создания «системы для себя». В противоположность, например, системе VAX/VMS от фирмы DEC, которая претендовала на звание «системы для всех».

Правда, понятие «все» в случае с VAX/VMS охватывало весьма узкий круг, даже не столько лиц, сколько организаций. Но остаётся фактом, что система VAX/VMS разрабатывалась не для личного использования. Это наложило отпечаток не только на неё, но и предопределило судьбу её прямого потомка – Windows NT/etc.

UNIX же разрабатывался на голом энтузиазме, без особой поддержки дорогих руководителей, причём на аппаратном обеспечении, подлежащем списанию в утиль.

Впрочем, усилия разработчиков были оценены должным образом, и достаточно быстро: в 1983 году Томпсону и Ричи была присуждена премия Тьюринга – самая престижная награда в информационной сфере. Которую по значимости можно сравнить с премией, учреждённой некогда Альфредом Нобелем «за выдающиеся научные исследования, революционные изобретения или крупный вклад в культуру или развитие общества». Что поделать – Нобель не мог и предполагать, что информационные технологии окажут на развитие общества не меньшее влияние, чем изобретённый им динамит.

А в 1999 году Томпсон и Ричи удостоились одной из высших наград государства, гражданами которого они являются^[1]: – Национальной медали в области технологий (National Medal of Technology, ныне – National Medal of Technology and Innovation). Которую им лично вручил человек, прославившийся обсуждением вопроса, является ли оральный секс основанием для обвинения в лжесвидетельстве. Ну и прочими мелочами, типа приказа о бомбардировке Югославии. От чего, впрочем, награда эта не становится менее почётной...

На дальнейшую судьбу UNIX огромное влияние оказали юридические коллизии тогдашнего текущего момента. Незадолго до создания этой системы корпорация AT&T подверглась антимонопольному преследованию, в результате чего претерпела поражение в правах – на деятельность её был наложен ряд ограничений. В частности, она не имела права торговать

программными продуктами, в число коих попадала и новорождённая UNIX.

Разумеется, материнская корпорация постаралась пристроить к делу создание своих сотрудников – в частности, UNIX с его инструментарием использовался в AT&T для подготовки технической и патентной документации. Что, кстати, представляет собой типичную пользовательскую задачу. И скажите мне теперь, что UNIX не пригоден для применения конечными пользователями.

Однако, как уже было сказано, в силу юридических ограничений AT&T не могла сделать из UNIX коммерческий продукт. И потому исходники этой системы, начиная с 1974 года, стали распространяться в университетах – в образовательных, как это тогда задумчиво называлось, целях. На условиях по тем временам достаточно либеральных, в том числе, и просто явочным порядком, лично Брайаном – люди с психологией сталинских наркомов, которые могли сказать: «под мою ответственность», встречались не только в Советском Союзе...

Передача UNIX в университетские структуры не была свободным распространением в том смысле, который вкладывается ныне в понятие FOSS. Хотя система, точнее, тогда ещё не более, чем её прототип, и передавалась в исходных текстах с правом их изучения, модификации, доработки и прочего потрошения.

Однако, во-первых, все эти действия требовали обладания лицензией на исходный код UNIX, которая передавалась AT&T вместе с ней самой и её исходниками, но – за деньги, хотя и не очень большие по масштабам американских организаций середины 70-х годов прошлого века. В личное же пользование лицензии на UNIX тогда ещё не приобретали.

А во-вторых, и это главное, условия лицензии не допускали дальнейшего свободного распространения ни системы целиком, ни отдельных её компонентов, содержащих исходный код UNIX. Что, собственно, и создало сюжет дальнейшей детективно-технологической интриги.

Однако до развёртывания сюжета грядущего технологического детектива было ещё далеко. А пока университеты с радостью приобщались к новой операционной системе, в которой были реализованы все передовые идеи того времени. И к тому же в

принципе способной функционировать практически на всем спектре тогдашнего оборудования. Напомню, что речь идёт о середине 70-х годов прошлого века: Стив Джобс еще не помышлял о продаже калькулятора и использовал родительский гараж по прямому назначению, а Билл Гейтс не освободил мир своим MS DOS'ом от засилья CP/M.

Выйдя за стены Bell Labs, UNIX зажил самостоятельной жизнью, крепко окопавшись в той же университетско-академической среде Computer Science. Одним из её центров в данном случае оказался Калифорнийский университет Беркли – учреждение, известное всем, интересовавшимся историей как точных наук, так и их влиянием на нашу жизнь.

Получив, благодаря профессору Бобу Фабри (Bob Fabry), в 1974 году ОС UNIX вместе с её исходниками и лицензией на их использование, университет Беркли поддержал и развил традицию «систем для себя», свойственную первоначальному UNIX. Но об этом – в следующей главе. А пока – о «железных» предпосылках.

«Железные» предпосылки

Как было сказано в первом разделе этой главы, именно эволюция аппаратных платформ обеспечила почву для развития всех остальных сюжетов. Так что пришло время поговорить о «железе».

Материалы, касающиеся истории вычислительной техники, принято вести с тех времён, «когда компьютеры были большими». Нашу же историю мы, напротив, начнём с того момента, когда компьютеры начали становиться маленькими, за что и получили имя миникомпьютеров.

Впрочем, самым главным был не габарит новых машин (размером с 1-2 бытовых холодильника), а то, что они допускали интерактивное взаимодействие с пользователем – ввод задач посредством клавиатуры и вывод результатов на тот же телетайп (а потом и на экран монитора).

На одном из таких миникомпьютеров, PDP-7 производства фирмы DEC, и был разработан первоначальный UNIX. Который, впрочем, быстро утратил связь с родительской платформой: после того, как основная системы часть была переписана на языке Си (специально созданном для разработки этой ОС), возникли условия для

относительно легкого ее портирования на любое «железо». И долгое время миникомпьютеры различных типов (в основном PDP-11 и пришедшие им на смену во второй половине 70-х машины серии VAX), как наиболее демократические платформы того времени, оставались основной средой для разработки и использования UNIX.

Попавший в Беркли UNIX также первоначально был инсталлирован и работал на 16-битных миникомпьютерах PDP-11, и программные наборы 1BSD и 2BSD (собственно системами, как говорилось ранее, их назвать было еще нельзя) разрабатывались на них и для них.

Однако в 1977 году на свет вышли первые миникомпьютеры VAX, уже 32-битные. Разумеется, первоначальный UNIX, ставший к тому времени вполне кросс-платформенным, обзавелся для них соответствующей версией, носившей имя UNIX/32V. Однако и берклианская ветвь не осталась в стороне от прогресса: 3BSD – первая целостная система из Беркли уже в 1979 году была портирована на VAX, причём с эффективным использованием всех его аппаратных возможностей, в частности, виртуальной памяти. Именно тогда и сложилась та самая парадоксальная ситуация, о которой я говорил выше: пользователи VAX-машин вынуждены были получать (то есть покупать) лицензию на использование 32V, однако устанавливали и применяли на практике 3BSD, а затем и 4BSD.

Однако миникомпьютеры доживали свой век – на смену им серверы и рабочие станции на фантастически мощных по тем временам RISC-процессорах, объединённые в сети с клиент-серверной архитектурой. Они уже имели вид, подобный обычным персоналкам – с системным блоком, монитором и клавиатурой, а затем и мышью, а рабочие станции к тому же находились в индивидуальном пользовании.

Первые рабочие станции выпустила в 1981 году фирма Apollo. Они были основаны на процессоре Motorola 68000, имели сетевые интерфейсы и объединялись в клиент-серверные сети. Однако операционной системой для них был не UNIX, а собственная разработка – Aegis, позднее переименованная в DOMAIN/OS. Впрочем, ни под тем, ни под другим именем с UNIX она была несовместима, хотя, как говорят, и была с ней сходна внешне, в частности, своим командным интерфейсом.

Машины Apollo, носившие название DN100, были сами по себе дороги, а разработка софта для них, вследствие своеобразия закрытости операционки, обходилась ещё дороже. Что открывало путь к созданию альтернативных решений – более дешёвых и совместимых.

И такое решение появилось мгновенно. В том же 1981 году аспирант Стэнфордского университета Энди Бэчтольшайм (Andy Bechtolsheim), участвуя в работах по созданию сети для своей альма-матер, из отходов производства собрал машину на том же процессоре Motorola 68000. А в 1982 году он вместе Винодом Хосла (Vinod Khosla) основал компанию SUN, что первоначально расшифровывалось как *Stanford University Networks*, но быстро обрело «солнечную» символику – интересно, не читали ли основатели **Луч света в тёмном царстве?** Чуть позже третьим к ним присоединился Скотт МакНили (Scott McNealy).

«Трое появились не случайно» – они поставили себе целью создать мощный, но относительно дешёвый компьютер, основанный на стандартных промышленных компонентах. Затея эта удалась – и на протяжении 80-х годов одна за другой выпускаются машины Sun-1, Sun-2, Sun-3, основанные на процессорах Motorola 68XXX.

Стандартизация аппаратной части требовала столь же стандартной операционной системы, каковая могла основываться только на UNIX. И разработкой её занялся Билл Джой, тот самый, который, как мы увидим в главе второй, до того активно участвовал в создании BSD. И нет ничего странного в том, что именно она и была положена в основу новой ОС.

То был отнюдь не единственный пример портирования UNIX на платформу 68XXX. Мало кто ныне помнит, но, например, Apple разрабатывала собственную версию ее, под названием AUX, задолго до MacOS X – чуть не сразу после появления первого Mac'а.

Однако UNIX портировался не только – и не столько – на машины с процессором 68XXX. «Дурной пример» Sun оказался заразительным – и этим путём пошли такие разработчики собственных платформ, как IBM, Hewlett-Packard, DEC. Именно они стали и разработчиками собственных же вариантов UNIX.

А что же собственно персоналки, именовавшиеся в те годы IBM PC-совместимыми компьютерами? И ныне для большинства пользователей ассоциирующиеся с самим понятием компьютера.

А поначалу – ничего. Первый широко распространившийся персональный компьютер, собственно IBM PC и развивавший его линию IBM PC/XT, базируясь на внутренне 16-разрядных процессорах Intel 8088 и 8086, работать под исходно 32-битной UNIX не мог, как не способна была на это и персоналка следующего поколения, IBM PC/AT на процессоре Intel 80286.

Только появление в 1985 году первого 32-разрядного процессора от Intel – 80386, дало возможность использовать UNIX на дешевых и общедоступных персоналках. Так что именно здесь пролегла магистральная линия массовой компьютеризации – по всему миру шло триумфальное шествие Советской власти (то есть, пардон, Intel-совместимых PC).

Но под чем же работало все это аппаратное богатство? Да в подавляющем большинстве – под MS DOS, 16-разрядной операционной системой, созданной ещё для первых IBM PC и несущей в себе массу неустранимых ограничений: принципиальную однозадачность, отсутствие многопользовательского доступа, возможность использовать «по прямому назначению» лишь 640 Кбайт оперативной памяти, примитивную организацию файловой системы, не менее примитивные средства работы в текстовом режиме – единственно возможным силами «черного» DOS.

Конечно, предпринимались многочисленные попытки заретушировать «родимые пятна» DOS. Разрабатывались надстройки над ней, способные использовать вес физический объём оперативной памяти и многозадачность, такие, как QuaterDesq и Geoworks. Которые включали также и системы работы в графическом режиме. Некоторые пользовательские DOS-приложения (табличные процессоры Lotus 1-2-3 и QuattroPro, текстовый редактор WordPerfect) обзаводились собственными средствами управления памятью и графическими интерфейсами.

Вся эта многочисленная DOS-косметика была либо неудачной, либо не получила распространения. Конечно, существовала и альтернатива ей – разрабатывавшаяся в IBM операционка OS/2, первая 32-разрядная ОС, специально написанная для PC. Однако и она, не

смотря на весьма прогрессивный базис, не приобрела широкой популярности. О причинах этого можно было бы вспоминать долго – не последней оказалось исключительно пофигистское отношение производителя к своему детищу.

Сложилась парадоксальная ситуация: «народная», то есть общедоступная, платформа не имела адекватной «народной» же операционки, способной использовать её возможности. И в следующей главе мы проследим зарождение и развитие первой претендентки на этот титул.

Увы, претензии эти оказались не реализованными. А «свято место» массовой операционки для настольных персоналок оставалось пусто вплоть до 1995 года – появления Windows 95. Началась эра гегемонии платформы Wintel (то есть машин на Intel-совместимых процессорах под управлением ОС Windows). И гегемония эта практически не поколеблена и по сей день. Не смотря на несколько попыток борьбы с ней в 90-е годы. Не смотря на тщившиеся стать народными Mac'и во всех их проявления. Не смотря на все успехи последних лет в продвижении Linux. Но об этом – позднее.

Глава вторая. Берклиада

История возникновения и раннего развития BSD-систем, а в дальнейшем наиболее популярной из них – FreeBSD, поистине эпична сама по себе. И к тому же она оказалась поворотной точкой в дальнейшей судьбе всех открытых и свободных операционок. И потому заслуживает подробного рассмотрения, которое, начавшись в этой главе, будет продолжено в главах пятой и седьмой.

«Похищение Елены Прекрасной»

В разделе о зарождении UNIX предыдущей главы упоминался профессор Фабри, который принёс свет UNIX'овой мысли в стены Университета Беркли. Где система попала в условия открытого общения специалистов в области Computer Science самого разного ранга, от профессоров до аспирантов – именно такой статус имели во второй половине 70-начале 80-х годов прошлого века Билл Джой (Bill Joy, в последующем, как мы уже видели, один из основателей компании Sun), Маршалл Керк МакКузик (Marshall Kirk McKusick), Озалп Бабаоглу (Özalp Babaoğlu). Их усилиями, вкупе с другими сотрудниками университета, система UNIX медленно, но верно превращалась именно в то, чем она стала ныне. Достаточно сказать, что на счету «ранних берклианцев» разработка системы управления виртуальной памятью, концепции сокетов для взаимодействия между процессами, текстовый редактор `vi`, ставший в лице своего клона Vim неотъемлемой частью всех UNIX-подобных систем, и командная оболочка C-shell (`csh`), положившая начало интерактивным методам работы в командной строке.

Нам, избалованным мощными и красивыми текстовыми редакторами для графического режима (или, по вкусу, изошрёнными возможностями нынешнего Vim'а), современными командными оболочками типа `bash` и `zsh`, трудно сейчас оценить, какую роль в дальнейшем развитии UNIX-подобных систем сыграли `vi` и `csh`, выглядящие сегодня столь невзрачными.

Сотрудники Беркли оказались первыми и в организации распространения результатов своих работ. Этой цели служила Berkely Software Distribution или, сокращённо, BSD – система распространения разработанного в университете софта на магнитных лентах, от которой в конечном итоге происходит всё многообразие форм BSD- и Linux-дистрибуции.

Первые выпуски BSD (1BSD и 2BSD), вышедшие в 1978 году, ещё не представляли собой цельных систем, а содержали лишь набор утилит и приложений собственной разработки. О какой-либо системной целостности можно говорить, начиная с 3BSD (1979 год) – правда, целостность эта в значительной мере была обусловлена включением компонентов собственно UNIX.

Однако именно выпуск 3BSD послужил причиной тому, что команда UNIX-разработчиков Беркли получает в 1980 году грант упоминавшегося в прошлой главе DARPA с целью разработки протокола передачи данных для сети ARPANET, который ныне известен как протокол TCP/IP.

Практически одновременно с получением гранта DARPA Бобом Фабри формируется команда CSRG (Computer System Research Group), которая объединила всех трудящихся университета Беркли (и не только его), связанных с развитием берклианской ветви UNIX. Начиная с октября 1980 года, на протяжении двух с небольшим лет эта группа последовательно выпускает 4BSD, а затем 4.1BSD в нескольких версиях: собственно 4.1BSD – июнь 1981 года, 4.1a, 4.1b и 4.1c (1982—начало 1983 года).

Модель распространения BSD выглядит весьма запутанной для нас, незнакомых с американским юридическим крючкотворством. Все собственно Берклианские разработки распространялись хотя и не бесплатно, но за минимальные деньги (лента 1BSD, например, стоила 50 долларов), причём дальнейшее их использование было практически свободным, в духе позднейшей BSD-лицензии.

Однако те же разработки в составе цельной работоспособной системы, содержащей UNIX-код, требовали лицензирования последнего, что приводило к удорожанию на порядки. Дело доходило до ситуаций, которые кажутся нам смешными: организации покупали лицензию на использование UNIX у Bell Labs, но заказывали и использовали более функциональную систему из Беркли.

Такое положение вещей, противоречащее здравому смыслу, не могло продолжаться вечно – и скоро мы узнаем, каким образом оно разрешилось. Но сначала прервём линейность истории, обратившись к вопросу, откуда пошёл свободный софт.

Начало свободного софта

Говоря в главе первой об истоках и предпосылках истории FOSS, я упомянул общественные движения Open Source Software и Free Software, однако больше не прибавил о них ни слова. Настало время восполнить это упущение.

Под открытым и свободным программным обеспечением, для которого закрепилась аббревиатура FOSS, понимается два близкородственных, но не вполне идентичных понятия – программное обеспечение с открытыми исходными текстами (Open Source Software) и собственно свободное программное обеспечение (Free Software).

Читатель, знакомый с современным положением дел вокруг так называемого СПО, вправе задать мне вопрос: почему названия из предыдущего абзаца стоят именно в таком порядке? Ведь каждому ребёнку известно, когда Ричард Столлман, известный в миру как RMS, создал Фонд Свободного Программного обеспечения и основал проект GNU, и когда из уст Эрика Рэймонда, не менее известного как ESR, впервые прозвучало словосочетание Open Source Software. Попробуем разобраться.

Движение Open Source организационно оформилось в 1998 году. Однако зародилось оно очень давно – и в тех же академических кругах Computer Science. Собственно, первоначально никакого движения не было – а была лишь обычная, принятая в любой науке, практика свободного обмена результатами своей работы. Благо, ARPANET, а затем и Интернет предоставили к тому практически неограниченные возможности. Да и необходимости в движении не возникало – никакого другого софта, кроме открытого, просто не было в природе.

Об оформлении прототипа движения Open Source можно говорить с начала Берклиады, когда Университет Беркли стал распространять результаты своих работ открыто (то есть с доступом к исходным текстам) и свободно (то есть без ограничений на дальнейшую

модификацию и распространение), под лицензией, которая получила имя лицензии BSD.

Суть её была очень проста: с исходниками BSD UNIX можно было делать всё, что угодно, кроме как приписывать себе их авторство. Правда, была там и так называемая «оговорка о рекламе» – требование упомянуть регентский совет Университета Беркли в любом производном продукте, но со временем она была изъята.

Поскольку усовершенствования первоначальной UNIX, пришедшие из Университета Беркли, были очень существенными, результатом этого было расщепление UNIX на две ветви – проприетарную UNIX от AT&T, за которой со временем закрепилось название System V, и BSD UNIX, распространявшуюся свободно. Впрочем, в силу открытости берклианских разработок, они быстро были инкорпорированы и в System V (начиная с её Release 4, говорить от первоначальной UNIX уже не приходится).

Обе ветви UNIX, System V и BSD UNIX, сосуществовали мирно, подобно капитализму и социализму. Однако лишь до поры, до времени – пока не появилась юридическая возможность коммерческого распространения UNIX, само это слово (в форме UNIX) не стало торговой маркой, соответствие которой должно сертифицироваться, – короче говоря, пока не запахло «баблом».

И вот тут-то формальные правообладатели UNIX вспомнили, что в составе BSD-системы имеется некоторое количество кода, являющегося их «интеллектуальной собственностью», и затеяли судебный процесс против Университета Беркли, о чём речь пойдёт в скором времени.

А пока заметим, что важной вехой в становлении движения Open Source и как технологического, и как идеологического явления стала разработка оконной системы X, начатая в 1984 году, чему будет посвящена отдельная страница.

Тем временем Ричард М. Столлман, сотрудник той самой MIT AI, боролся с прикручиванием принтера от HP к своей системе. И боролся безуспешно – поскольку товарищи от Хьюлетта и Паккарда отказались предоставить ему исходники на своё *firmware*. Что привело Столлмана к убеждению – закрытые исходники суть тормоз прогресса, и все программное обеспечение должно быть открытым и свободным. Начался крестовый поход за освобождение софта.

К середине 80-х годов прошлого тысячелетия RMS создаёт Фонд свободного программного обеспечения (FSF – Free Software Foundation), начинает проект GNU – воспроизведение функциональности UNIX «с чистого листа», но в свободном исполнении, а главное – формулирует принципы Free Software: свобода использования, свобода изучения и модификации, свобода распространения.

Знакомый велосипед, не правда ли? Да, именно на таких условиях распространялись результаты работ сообщества Computer Science (как, впрочем, и любого иного научного сообщества). Новым в принципах RMS, нашедшим своё выражение в разработанной под его руководством (и с участием профессиональных юристов) лицензии GPL (General Public License), было только одно: любая программа, использующая код, защищаемый GPL, должна распространяться на тех же условиях – ныне, присно и во веки веков...

В рамках проекта GNU (что расшифровывается просто – GNU is not UNIX) были разработаны функциональные аналоги всех классических UNIX-утилит и пользовательских приложений, из которых важнейшим (и до сих пор единственным незаменимым) оказался компилятор языка Си (gcc – GNU C Compiler).

Участники проекта GNU, работавшие первоначально на чистом энтузиазме, за считанные годы воссоздали все системное окружение полноценной ОС. За одним единственным исключением – ядра. Но тут уже начинается история из главы четвёртой. А мы продолжим нашу Берклиаду.

Возвращение в Беркли

Оно ознаменовалось тем, что в августе 1983 года была выпущена система 4.2BSD – та самая, на разработку которой собственно и был получен грант DARPA. К этому времени Билл Джой, сыгравший большую роль в разработке предыдущих версий, покинул Университет Беркли и стал соучредителем новой компании Sun Microsystems (о чём – в следующей главе). На первые же роли в проекте BSD вышли Майк Карелс (Mike Karels) и Керк МакКузик.

Система 4.2BSD аккумулировала в себе как все ранние достижения берклианской мысли, так и разработки, выполненные уже

в рамках CSRG и как бы «порционно» появлявшиеся в последовательности версий 4.1BSD. Из которых главнейшими были протокол TCP/IP и новая файловая система FFS (Fast File System).

О значении TCP/IP много говорить не приходится: если вы читаете эти строки, значит, тем или иным образом имеете доступ в Интернет. Так вот, без TCP/IP ничего этого не было бы: ни Интернета, ни «одноглазикофф ффконтакте», ни самой этой книги.

А чтобы понять значение FFS, достаточно вспомнить что все файловые системы современных UNIX'ов, как свободных, так и проприетарных, берут своё начало именно от неё, если не прямо, то опосредованно, через развитие заложенных в ней принципов.

Так что система 4.2BSD не только предопределила направление развития всех последующих представителей BSD-семейства, но и оказала большое влияние на UNIX «чистой линии». А также – будущей героини нашего повествования, ОС Linux.

Линия BSD рано начала давать боковые отростки. Из них одними из главных, в рамках нашего повествования, стали микроядерные операционки, в первую очередь ядро Mach, разрабатывавшееся в Университете Карнеги-Меллона, а затем – в университете штата Юта. Некогда оно рассматривалось как прообраз операционных систем будущего, однако на практике возлагавшихся на него надежд не оправдала. Сам по себе проект Mach давно прекратил своё развитие, так что главная слава Mach оказалась посмертной. Ибо под его воздействием Энди Танненбаум написал свою игрушечную систему MINIX, которая вдохновила Линуса Торвальдса на написание Linux.

Пора опять возвращаться к магистральной линии развития BSD. Каковая после выхода 4.2BSD, оказавшейся переломной в развитии этой системы, приобрела плавно поступательный характер. Новые релизы появляются относительно редко: выход 4.3BSD датируется июнем 1986 года, а её последовательных инкарнаций – 4.3BSD-Tahoe и 4.3BSD-Reno – июнем 1988 и началом 1990 года соответственно.

Выход следующего релиза, 4.4BSD, который готовился как квинтэссенция всей предшествующей Берклиады, был запланирован на 1993 год. И действительно произошёл почти в установленные сроки. Однако ему суждено было стать и последним в ряду всех систем линии 4.xBSD: потому что в интервале 1990-1993 года

случилось несколько событий, которые в своей совокупности изменили весь ход истории свободных операционных систем.

Интермедия

Итак, мы остановились на моменте выхода 4.3BSD и двух её последовательных инкарнаций – 4.3BSD-Tahoe и 4.3BSD-Reno. Базовой платформой для всех них был VAX. Однако в 4.3BSD-Tahoe обособлены машинно-зависимые и машинно-независимые части кода, что создавало предпосылки для грядущего портирования на иные архитектуры, планировавшиеся в версии 4.4. А 4.3BSD-Reno и была прототипом этой грядущей ветки, предназначенным для обкатки намечавшихся новшеств.

Параллельно с основными выпусками 4.3BSD было подготовлено ещё два как бы дополнительных – 4.3BSD Net1 (март 1989 года) и 4.3BSD Net2 (июнь 1991 года). Основываясь на 4.3BSD-Tahoe и 4.3BSD-Reno соответственно, они содержали исключительно компоненты, разработанные в Беркли и полностью освобождённые от какого-либо кода первоизданного UNIX. И потому могли распространяться свободно как в бинарном виде, так и в виде исходных кодов.

Название выпусков 4.3BSD Net1 и Net2 связано с тем, что они замышлялись как подборки инструментария для работы с сетями – главным образом, по протоколу TCP/IP. Таково было пожелание пользователей, нуждавшихся в этих средствах, но по тем или иным причинам не испытывавших потребности в лицензировании собственно UNIX-кода. Однако, как мы увидим далее, значение этих выпусков скоро переросло поставленные первоначально скромные цели.

И 4.3BSD Net1 стал первой системой из Беркли, которая распространялась под лицензией BSD (ещё в первом её варианте, включавшем «оговорку о рекламе»).

Номинальная цена за ленту 4.3BSD Net1 была установлена в 1000 долларов. Однако, поскольку лицензия это не запрещала, далее копии ленты могли распространяться совершенно свободно, копироваться, устанавливаться на любое количество машин, передаваться и даже выкладываться на анонимные ftp-сервера. Что, разумеется, и

происходило – однако, по свидетельству очевидцев этой истории, немало организаций не сочли западло для себя заплатить указанную сумму. Причём не столько ради получения самого кода – его, как уже сказано, можно было получить и бесплатно, сколько для финансовой поддержки проекта.

Подобная практика распространения продолжалась и после выхода 4.3BSD Net2. И опять с тем же результатом – несмотря на возможность откровенной и вполне законной халявы, нашлось немало контор и даже частных лиц, которые выложили 1000 баксов за обладание дистрибутивной лентой. Среди таковых оказался и Грег Лией – в последующем один из ключевых разработчиков FreeBSD.

Факт столь массового спроса на 4.3BSD NetX тем более примечателен, что ни первый, ни второй её выпуск не содержал самодостаточной, загружаемой ОС, а включал только системное обрамление и комплекс утилит, в первую очередь, для работы с TCP/IP. И пользователи, кем бы они ни были, организациями или частными лицами, покупали её на свой страх и риск, так как превращением её в законченную ОС они должны были озаботиться сами.

В ходе подготовки выпусков 4.3BSD Net1 и Net2 обнаружилось, что проприетарного (то есть патентованного) кода первоизданного UNIX, права на который к тому времени перешли к USL (UNIX Systems Laboratory – дочерняя компания AT&T, созданная специально для продвижения этой системы) в составе берклианских UNIX'ов осталось не так уж и много. И родилась идея создания полностью открытой, свободно распространяемой операционной системы BSD. Правда, даже в наиболее полном выпуске 4.3BSD Net2 недоставало нескольких ключевых фрагментов, которые превратили бы его в полноценную операционную систему, полностью свободную от наследия UNIX. Их и следовало воспроизвести в первую очередь.

Примерно в это же время прекращается финансирование проекта BSD со стороны DARPA. Есть подозрение, что причиной тому послужил распад мировой системы социализма – всё в жизни имеет свою обратную сторону, даже крах коммунистической идеологии. И хотя CSRG просуществовала ещё несколько лет (как структурное подразделение, она была расформирована в 1995 году), ряд её сотрудников начал подыскивать себе другие занятия.

В числе их оказались Билл Джолитц (Bill Jolitz) и Линна Джолитц (Lynne Jolitz). Они поставили своей целью, во-первых, воспроизвести недостающие звенья между 4.3BSD Net2 и полноценной ОС, а во-вторых, портировать новообразованную систему на ту самую демократическую платформу того времени – на i386.

Обе задачи были успешно решены в течении полугода после выпуска 4.3BSD Net2. И в результате в январе 1992 года свет увидела работоспособная система под названием 386BSD, первая из всех берклианских систем, полностью свободная от проприетарного кода, и первая же, адаптированная для машин с процессором i386, что и было вынесено в её титулатуру.

Распространялась система 386BSD исключительно по сети, как в откомпилированном виде, так и в исходниках, и сразу, несмотря на содержащиеся в ней ошибки, приобрела популярность среди народных масс. Следствием этого стало появление большого количества исправлений, дополнений и улучшений исходной системы, которые составили корректирующий комплект, получивший неофициальное название patchkit (набор заплаток), делающий 386BSD пригодной к практическому использованию.

Некоторое время «заплаточный» проект развивался усилиями энтузиастов при координации Билла Джолитца. Однако довольно скоро выяснилось, что он не может уделять ему столько времени, сколько требовала обработка многочисленных поступающих патчей. И на некоторое время «заплаточный проект» оказался заброшенным.

FreeBSD: рождённая свободной

И тогда из среды энтузиастов системы выдвинулась группа координаторов «заплаточного» проекта, получившего условное название 386BSD 0.5 или 386BSD Interim. В их числе были Джордан Хаббард (Jordan Hubbard, ныне работающий в фирме Apple директором по технологиям UNIX), Нейт Вильямс (Nate Williams) и Род Граймс (Rod Grimes), оказавшиеся самыми последовательными приверженцами patchlit'ов. Они разработали промежуточный снапшот системы, очищенный от «заплаточных» излишеств.

Однако планы «тройки по борьбе с басмачами» (пardon, заплатками) были нарушены, когда Джолитц окончательно прекратил

поддержку своего проекта, не оставив ясных указаний насчёт того, что уже сделано, и что надлежит сделать далее. Но это уже не смогло остановить развитие проекта. А примкнувший к нему Дэвид Гринмен (David Greenman) придумал для него и имя – FreeBSD.

Вахта же Хаббарда выразилась в том, что он наладил контакт с компанией Walnut Creek CDRom, образовавшейся незадолго до этого (в 1991 году) для распространения всякого рода Free- и Shareware (а также и собственно Free Software) на компакт-дисках. Если вспомнить, что до превращения CD-приводов в стандартный атрибут настольных персоналок оставалось ещё несколько лет, можно оценить степень новаторства этой фирмы.

Общепринятым способом распространения сколько-нибудь объёмного программного обеспечения (за исключением коммерческого «коробочного») в те годы были публичные ftp-сервера. Однако доступ к Интернету, по крайней мере, для частных лиц, был тогда ещё большей экзотикой, нежели CD-привод в индивидуальном десктопе. И Хаббард предложил совершенно новую по тем временам идею – распространение дистрибутива операционной системы на компакт-дисках. Это если и не предопределило успех проекта, то немало ему способствовало.

В результате этого сотрудничества в декабре 1993 г. совместные усилия проекта FreeBSD и фирмы Walnut Creek обрели зримое воплощение в виде FreeBSD 1.0, распространявшейся не только с ftp-серверов, но и на компакт-дисках. Таким образом, FreeBSD стала одним из пионеров в этом, ныне столь привычном для нас, способе распространения дистрибутивов.

FreeBSD 1.0 включала в себя компоненты из BSD4.3 Net 2, 386BSD и её «заплаточного» проекта, а также ряд утилит, разработанных в рамках проекта GNU. Она имела большой успех, который был закреплён и развит выпуском в мае 1994 года версии FreeBSD 1.1.

Технологический детектив

Система 386BSD и её наследница FreeBSD были не единственными попытками создания BSD, свободной от проприетарного кода. Ещё один вариант был реализован созданной в

1991 году фирмой BSDI (Berkeley Software Design Incorporated) – но уже как коммерческий.

Фирма BSDI занялась разработкой собственной BSD-системы, взяв за основу всё ту же ленту 4.3BSD Net2. Возникшая в результате система получила имя BSD/386 (в дальнейшем она была известна как BSDi и BSD/OS) и стала распространяться в бинарном виде вместе с исходниками по цене 995 долларов под первым вариантом лицензии BSD.

Упоминание Калифорнийского университета и Регентского совета как создателей и владельцев распространяемой системы, присутствовавшее в первом варианте BSD-лицензии, делало фирму как бы сопричастной последнему – тем более, что она была образована бывшими сотрудниками CSRG. Среди них был и Ричард Стивенс (Richard Stevens), главный разработчик BSD/OS, известный также как автор книг по UNIX и протоколу TCP/IP (он скончался в 1999 году в возрасте 48 лет).

Не менее важным, чем причастность BSDI к Калифорнийскому университету, обстоятельством для дальнейших событий оказалось то, что её система позиционировалась, как UNIX, и заказ её можно было осуществить, обратившись по номеру телефона, содержащему слово UNIX. А слово это стало к тому времени торговой маркой, под владением USL, дочерней фирмы AT&T. Которая как раз в это время получила, наконец, право коммерческого использования UNIX. И результаты не замедлили воспоследовать, поскольку, как уже говорилось ранее, в воздухе отчётливо запахло «баблом».

Первые претензии со стороны USL, однако, касались только компании BSDI и затрагивали лишь рекламную сторону дела: использование последней торговой марки UNIX без соответствующего лицензирования и «вводящего в заблуждение» телефонного номера. Обе они были не лишены резона и немедленно удовлетворены: номер был снят, а соответствующие службы компании BSDI переформулировали свои рекламные материалы, популярно объясняя потенциальным покупателям, что BSD/386 является не совсем UNIX'ом.

Однако вслед за этим в USL вспомнили, что в составе BSD-систем имелось некоторое количество кода, являющегося их «интеллектуальной собственностью», и вчинили уже настоящий

судебный иск. Сущность его сводилась к тому, что BSDI, кроме проприетарного кода UNIX, распространяет фирменные секреты USL, чем наносит оной непоправимый финансовый урон, и к требованию прекратить продажи BSD/386.

В ответ BSDI отвергла претензии по поводу чистоты кода пресловутых шести файлов, а по поводу всего остального (то есть того, что составляло содержимое выпуска 4.3BSD Net2) перевела стрелки на Калифорнийский университет, указав, что распространяла их код в полном соответствии с BSD-лицензией.

Поскольку добиться успешного решения суда в «деле о шести файлах» показалось USL проблематичным, она переформулировала иск, включив в число ответчиков, кроме BSDI, также и Калифорнийский университет, а содержание его распространив на всю BSD-систему в виде 4.3BSD Net2, требуя теперь запрета на распространение и этой последней,

В таких случаях сначала проводится предварительное слушание, определяющее, может ли иск составить предмет рассмотрения в суде, которое и происходит при положительном ответе на этот вопрос.

Подготовка к предварительному слушанию заняла несколько недель, в течении которых обе стороны развили бурную деятельность. Как пишет Керк МакКузик,

Штат CSRG перешёл от написания кода к написанию нескольких сотен страниц материалов, которые были использованы в юридических сводках.

Наконец, в декабре 1992 года состоялось предварительное слушание, которое проводил судья Федерального суда в Нью-Джерси (штат, в котором располагалась штаб-квартира USL), Диккинсон Р. Девебуа – по причинам, которые станут ясны через несколько строк, имя его должно быть упомянуто в ряду создателей и разработчиков BSD и FreeBSD. Он не принял немедленного решения по иску, а решил подробнее рассмотреть материалы. Это заняло у него шесть недель, по прошествии которых было вынесено решение: большинство обвинений USL отклонялось, за исключением двух пунктов, касавшихся авторских прав и возможности утраты фирменных

секретов. И, кроме того, было предложено рассматривать дело в суде штата, а не в федеральном суде.

Это судьбоносное решение было вынесено в пятницу вечером. А уже в понедельник утром Калифорнийский университет вчинил компании USL встречный иск, касавшийся нарушения USL лицензии BSD, под которую подпадал заимствованный ими из BSD-систем код. То есть при распространении UNIX в сопроводительной документации не упоминался Калифорнийский университет как разработчик и собственник заимствованного кода (а бесспорных заимствований из BSD в SVR4 было немало). Вот тут и сыграла свою роль та самая «оговорка о рекламе» в первоначальной версии лицензии BSD, за которую она подвергалась нападкам со стороны пуристов Free Software, начиная с Ричарда Столлмана.

Встречный иск в суде Калифорнии предопределил бы место для всех судебных разборок, если таковые последовали бы на уровне штата: по американским законам все дела по соответствующему уровню должны проходить в одном штате, дабы сторона, располагающая большими финансовыми ресурсами, не могла пооткрывать дела против менее состоятельной стороны во всех штатах сразу, ведь проезд даже и в Америке кое-чего стоит...

Однако скоро накал страстей спал. В 1993 года USL вместе со всеми её торговыми марками и правами, реальными и мнимыми, была куплена у AT&T фирмой Novell. Рэй Нурда (Ray Noorda), бывший тогда её CEO, выразился в том смысле, что предпочитает конкурировать на рынке, а не сквалыжничать в суде. И постарался оказать максимально возможное воздействие на руководство USL, дабы решить вопрос полюбовно.

К слову замечу, что Рэй Нурда, обеспечив славу Novell, как ведущей компании в области сетевых технологий («ну кто же не помнит старика Нетваря?»), через пару лет покинул её и основал фирму Caldera, на протяжении ряда лет выпускавшую весьма прогрессивный дистрибутив Linux – Caldera OpenLinux. Он отошёл от дел на рубеже тысячелетий и скончался в 2006 году, в возрасте 82 лет. Ему не суждено было увидеть того юридического шоу, которое устроила по поводу собственности на код UNIX SCO – компания, в которую преобразовалась основанная им Caldera. Иска, почти зеркально повторившего дело USL vs Berkeley, но ещё менее

обоснованного и завершившегося с существенно более печальными последствиями для истца. Воистину, история мстит забывшим её тем, что имеет обыкновение повторяться.

Но это было ещё далеко в будущем. А пока, несмотря на всю запутанность дела, в конце концов, соглашение было достигнуто. По его условиям из 4.3BSD Net2 были удалены фрагменты кода, признанные частной собственностью USL (по некоторым данным – три файла из примерно восемнадцати тысяч), в некоторых файлах были сделаны изменения, в иных же – добавлено уведомление об авторских правах USL. И в таком виде система BSD получила право на свободное распространение.

Цена свободы

Казалось бы, детективная история разрешилась вполне благополучно, не так ли? Однако на этот счёт существуют неоднозначные мнения. Согласно Керку МакКузику (а он был тогда связан именно с разработкой 4.4BSD в рамках CSRG), воссоединение всех берклианских побегов в лоне едином вызвало лишь кратковременную задержку в их разработке, которая в итоге оказалась

... благом, поскольку она заставила различные группы повторно синхронизировать наработки, сделанные за три года с момента первого выпуска CSRG Networking Release2.

Джордан Хаббард, который тогда занимался разработкой непосредственно FreeBSD, смотрит на ситуацию тех дней не столь оптимистично, полагая, что система 4.4BSD-Lite

... была в прямом смысле light, в частности, потому, что группа CSRG удалила большие куски кода, необходимого для создания реально загружающейся системы (по причине различных лицензионных требований), и фактически, порт 4.4BSD для платформы Intel был очень неполным.

Как можно заключить из слов Хаббарда, в тот момент катастрофа проекта казалась неизбежной – легким движением руки цельная и

работоспособная система превратилась в симпатичнейшего уродца. Но, как сказал один из героев Профессора, «приключения никогда не кончаются». И участники проекта FreeBSD приступили

... к сложнейшей задаче буквально пересоздания системы с нуля на основе абсолютно новой и довольно неполной системы 4.4BSD-Lite.

Реинкарнация недостающих фрагментов заняла около года. И в итоге 22 ноября 1994 года было объявлено о выходе первой версии возрождённой FreeBSD – 2.0, которая, несмотря «на множество недотёсанных углов», снискала значительный успех. А главное – к лицензионной её чистоте не смог бы придраться ни один сутяга. Именно она положила начало традиции, не прерывающейся и поныне.

Тем не менее, момент, благоприятный для «народной системы для народной платформы», был упущен: ниша эта оказалась плотно занята, во-первых, главной системой для простого народа в лице Windows 3.1/3.11, а чуть позже и Windows 95. А на роль системы альтернативной, для народа не совсем простого, выдвинулся Linux в лице первых своих дистрибутивов: Slackware, Debian, Red Hat, чуть позже Suse.

Существует мнение, что если бы BSD (еще не разделившаяся на Net- и FreeBSD) не погрязла бы в тяжбе с AT&T и получила бы свободу в конце 80-х – начале 90-х годов, то в разработке Linux не было бы никакой необходимости. Несмотря на свою пылкую любовь к BSD-системам во всех их проявлениях, не могу с этим согласиться: если бы Linux'а не было – его следовало бы изобрести. Потому что без него (и внедрённого Линусом в IT-индустрию метода разработки софта, о котором я говорил ранее) жить было бы скучно...

Тем не менее, хотя FreeBSD и задумывалась как система для пользователей, пусть и весьма специфических – но с сегодняшней колокольни превращения компьютера в бытовой прибор все пользователи того времени выглядят специфическими, – стать таковой в те дни ей было не суждено. Однако она снискала популярность среди Интернет-провайдеров, что и предопределило её репутацию как серверной системы на многие годы вперед. Стереотип этот начинает

ломаться лишь в последние годы, что мы и увидим при рассмотрении дальнейшей истории этой системы.

И хотя главная интрига нашего исторического сюжета позади, в будущем FreeBSD будет еще один драматический поворот, о котором мы расскажем в своё время. А сейчас зададимся вопросом – почему же эта ОС не стала системой для пользователя? Ведь, казалось бы, наконец претендент на звание народной ОС для народной платформы определился. Однако эта роль – стать народным десктопных UNIX'ом – выпала на долю системы, которая в это самое время зародилась в других краях и на другой почве. Со временем ответ на этот вопрос придёт. Но прежде мы вернёмся немного назад, к моменту вскоре после расщепления первозданного UNIX'a на ветки System V и BSD.

Глава третья. Возвращаясь к UNIX'ам

Хотя изложение истории проприетарных UNIX'ов не является предметом данного сочинения, они развивались параллельно с «UNIX'ами» свободными, и обе ветви оказывали друг на друга влияние. Так что время от времени нам придётся обращаться к этой сюжетной линии. Один из таких моментов наступил.

Серенада солнечной долины

Вернёмся к тому моменту, когда Билл Джой взялся за разработку ОС для только что появившихся компьютеров Sun. В основу своей системы он положил 4.1BSD, разработчиком которой перед тем являлся. А во главу угла её поставил ориентацию на поддержку компьютерных сетей и интеграцию с ARPANET, а в последующем – и Интернет.

Именно Биллу принадлежит знаменитый лозунг: **Сеть – это Компьютер** (The Network Is The Computer). Что вполне может быть инвертировано в: **Компьютер – это сеть**. Правда, выдвинут он был позднее, в середине 90-х годов, уже в эпоху Интернета.

Поддержка сетей в SunOS базировалась на ставших тогда почти стандартными протоколах Ethernet и TCP/IP. Именно из недр Sun вышли понятие виртуальной файловой системы – VFS, реализация сетевой файловой системы – NFS, множество средств удалённого доступа (RPC, Remote Procedure Calls) и так далее.

Далее, в SunOS большое внимание уделялось поддержке масштабируемости систем. Подразумевалось, что одна и та же ОС должна работать как на многопроцессорных серверах, так и на любых подключённых к ним рабочих станциях. С этим тесно связана и поддержка многопроцессорности, столь востребованная в наши дни. Конечно, и масштабируемость, и поддержка многопроцессорности не были уникальными для SunOS – этим же путём шли разработчики всех коммерческих UNIX. Однако Sun здесь была в числе первых – это раз. И во-вторых, именно её разработки оказали наибольшее влияние на развитие поддержки многопроцессорности в FOSS-операционках.

И наконец, SunOS изначально была тесно интегрирована со средствами поддержки графического интерфейса пользователя. Причём – средствами собственной разработки. Первым из них была SunView (Sun Visual Integrated Environment for Workstations, изначально SunTools) – оконная система, возникшая едва ли не одновременно с самой ОС в первой половине 80-х годов. От всех последующих оконных систем, ориентированных на UNIX, она отличалась тем, что большая её часть поддерживалась ядром. Кроме того, она включала в себя набор стандартных пользовательских приложений, таких, как текстовый редактор, почтовый клиент, инструменты для настройки. И, наконец, она являлась частью стандартной поставки операционной системы. По-видимому, именно в SunOS впервые была реализована интеграция операционной системы, графического интерфейса и пользовательских приложений, получившая дальнейшее развитие не только в мире UNIX.

На смену ей в середине 80-х годов пришла NeWS (*Network extensible Window System*) – более сложная система, основанная на PostScript. Однако получить широкого распространения она не успела.

Наконец, в конце 80-х годов для SunOS была разработана оконная среда OpenWindows. Сначала она поставлялась как отдельное дополнение, а потом была включена в состав операционной системы.

На протяжении 80-х – начала 90-х годов были выпущены версии SunOS с 1-й по 4-ю, работавшие на машинах Sun-1, Sun-2, Sun-3. Однако в конце 80-х годов начинается смена аппаратной платформы: процессоры Motorola 68XXX в серверах и рабочих станциях заменяются на «камни» собственной разработки и производства, RISC-процессоры SPARC. Для которых разрабатывается и новая операционная система, названная Solaris 2. В отличие от предшественницы, она основывалась не на BSD-линии, а на «классической» System V (SVR4). Параллельно происходит отказ от графических средств собственной разработки, сменяясь ставшими стандартными для UNIX мира оконной системой X и рабочей средой CDE.

Некоторое время обе архитектуры со своими операционками сосуществовали, причём SunOS научилась поддерживать Sparc'i. Последний её релиз (SunOS 4.1.4) вышел в ноябре 1994 года. Тем не

менее, название SunOS по сей день сохраняется за ядром OS Solaris (и OpenSolaris).

Однако тут мы уже чрезмерно углубились в 90-е годы. А ведь SunOS была не единственным представителем проприетарных UNIX'ов. О которых пойдёт речь в следующей главе.

В оковах проприетаризма

Из предыдущих страниц могло создаться впечатление, что всё развитие UNIX вращалось вокруг университета Беркли и компании Sun. Разумеется, это не так. Просто там проходила генеральная линия развития тех систем, которые позднее составят мир FOSS. Основные же события, связанные с развитием проприетарных ветвей UNIX, происходили в более иных местах.

Я не буду говорить о них подробно – во-первых, потому, что они лежат за гранью нашей главной темы. А во-вторых, просто плохо знаю их историю – надеюсь, она найдёт своих летописцев. Тем не менее, несколько слов о проприетарных UNIX'ах сказать необходимо.

Начать с того, что не забрасывала свое детище сама AT&T во всех ее проявлениях. Правда, разработка системы плавно перемещалась от Bell Labs к иным подразделениям, типа группы поддержки UNIX и т.д., однако в мои цели не входит описание истории этой корпорации, и потому названия их я опускаю.

Из недр AT&T (в обобщенном понимании этого слова) последовательно выходят System III, System V, System V Release 2 (SVR2) и, наконец, System V Release 3 (SVR3); этой системе, разработанной в 1987 г., суждено было стать последним представителем чистой линии первоизданного UNIX. Ибо уже System V Release 4 (SVR4), появившаяся в 1989 г., включила в себя множество новшеств из разработки 4BSD – интеграцию с TCP/IP, поддержку FFS, берклианскую реализацию механизма виртуальной памяти, и многое другое.

В частности, легший в последствии в основу стандарта POSIX шелл Корна, несмотря на совместимость с первоизданным шеллом (Bourne Shell), заимствовал из C-Shell такие особенности, как управление заданиями, историю команд, поддержку псевдонимов и т.д.

(впрочем, развитие шеллов – совершенно отдельная история, о которой мы поговорим, когда придёт время – и не в этой рубрике).

Благо, как уже было сказано, берклианские разработки распространялись в соответствии с принципами открытой науки – то есть практически свободно. И если AT&T не всегда заимствовала их на уровне реализаций (то есть непосредственно кода), то идейное их влияние на первозданный UNIX было несомненным.

UNIX от AT&T, был лицензирован многими компаниями – производителями оборудования, оценившими, вслед за Sun, портируемость этой ОС: IBM, Hewlett-Packard, DEC, SGI и еще несколько забытых. Базируясь на SVR3, а позднее – на SVR4, они адаптировали систему под собственные архитектуры, создав такие варианты UNIX, как AIX от IBM, HP-UX, Digital UNIX (позднее Tru64), IRIX, соответственно. Все это были коммерческие операционки, однако, будучи жестко привязанными к аппаратуре фирм-производителей, в свободной продаже (подобно нынешним Windows) они не присутствовали.

В третьих, чисто софтверные фирмы подняли мутный вал UNIX'ов уже сугубо коммерческих, то есть предназначенных для продажи помимо оборудования.

Первыми на этом поприще отметились компании Interactive Systems и Santa Cruz Operations (очень косвенный предок скандально прославившейся ныне SCO). Не имея собственных аппаратных платформ, они адаптировали UNIX под платформы общераспространенные, коими были сначала PDP, а затем и IBM PC. Кажется, именно SCO UNIX стала первой представительницей семейства UNIX, портированной на машины с процессором Intel (то есть на архитектуру i386).

Как ни странно, в числе первых здесь оказалась и Microsoft. Ею была создана система XENIX для тех же i386-х машин. По словам видевших ее, это было нечто вроде однопользовательской реализации UNIX – как всегда, Microsoft и тут, подобно товарищу Ленину, пошла своим путем...

В результате ни одной из этих разработок не досталось народной любви – вследствие высокой стоимости и малого количества приложений, а для XENIX еще и урезанной функциональности. Хотя SCO UNIX получила довольно широкое распространение в банковской

сфере. В частности, её потомки трудятся в Сбербанке России чуть ли не по сей день.

Каждая из перечисленных (и забытых) мною компаний создавала сугубо свой UNIX, не совместимый с другими. В итоге, кроме двух базовых системы – System V и BSD, расцвело не менее дюжины вариантов, в той или иной мере различающихся между собой. Мною были упомянуты далеко не все из них, а лишь те, которые уцелели и поныне. Или – те, о которых мне довелось в свое время хоть что-то прочитать. На самом деле UNIX-клонов в те времена было гораздо больше.

Так что можно было констатировать, что в конце 70-х и в 80-х годах прошлого века UNIX развивался в соответствие со знаменитым лозунгом Великого кормчего китайского народа – «Пусть расцветают все цветы!» И, по заветам Председателя Мао, назрела настоятельная необходимость в упорядочивании стилей работы. Но об этом мы поговорим в следующем разделе, посвящённом битве за стандарты.

Битва за стандарты

Как следует из предшествующего изложения, со временем различия между клонами UNIX становились все более значительными. Во-первых, в основе существовавших её вариантов лежали разные базовые системы – SVR3, SVR4, 4BSD.

Во-вторых, каждый разработчик считал своим долгом либо адаптировать систему под возможности собственной аппаратной платформы, либо просто внести те или иные усовершенствования. А поскольку все существовавшие тогда системы, кроме BSD, были не только проприетарными, но и закрытыми, усовершенствования эти слабо согласовывались между собой. И, в любом случае, реализовывались различным образом.

Это ставило под угрозу один из краеугольных камней UNIX-идеологии – портируемость приложений. А традиции писать программы под абстрактный UNIX никто не отменял. И начался процесс, который, вслед за товарищем Мао, можно назвать «борьбой за упорядочивание трех стилей работы». Однако председателю КПК было проще – в данном случае речь шла не о трех, а едва ли не о тридцати трех стилях...

Тем не менее, процесс упорядочивания пошел. Первый шаг в этом направлении резонно было бы сделать основоположнику UNIX. И AT&T создала документ, описывающий спецификации, которым должна соответствовать система, претендующая на звание UNIX – SVID (System V Interface Definition, то есть определение интерфейса System V). И даже разработала программу, которая проверяла претендента на соответствие этим спецификациям – System V Verification Suite.

Нетрудно было предположить, что в этих спецификациях нашли отражение только те особенности BSD, которые были инкорпорированы в каноническую System V. А ведь BSD была столь же полноводным источником, из которого черпали все производители UNIX. Их такое положение дел не очень устроило. И потому было создано несколько организаций, разрабатывающих свои версии стандартов для операционных систем, расширенные по сравнению со SVID.

Наибольшее признание из них получил стандарт POSIX – Portable Operation System Interface based on UNIX, разработанный международной организацией под названием IEEE (Institute of Electrical and Electronics Engineers, Inc.). То есть было принято, что любая операционная система, претендующая на звание UNIX (или UNIX-совместимой), должна соответствовать соглашениям, описанным в основополагающих документах стандарта. Как мы увидим со временем, именно этими документами воспользовался Линус Торвалдс при создании Linux.

Крёстным отцом термина POSIX стал Ричард Столлман. Произносится это слово как *позикс* – это специально подчеркивалось во **Введении в POSIX.1**:

Ожидается произношение «поз-икс» как «позитив», а не «по-сикс». Произношение опубликовано в целях обнародования стандартного способа ссылки на стандартный интерфейс операционной системы.

Слово Portable в его названии первоначально означало, что соответствующая POSIX-спецификациям система может быть перенесена (возможно, с минимальными модификациями) на любое

компьютерное «железо». Однако со временем не менее важным оказался несколько другой аспект этого термина: любая прикладная программа, написанная в соответствии со стандартами POSIX, теоретически может быть перенесена, то есть портирована, на любую ОС POSIX-совместимого семейства. И нужно отметить – это один из тех редких случаев, когда теоретические ожидания блестяще подтвердились практикой.

Стандарты POSIX были приняты в 1988 году и зафиксированы в виде серии регулярно обновляемых документов (общим числом под два десятка), в которых описываются спецификации отдельных компонентов систем:

- Base Definitions volume (XBD) – определение терминов, концепций и интерфейсов, общих для всех томов данного стандарта;
- System Interfaces volume (XSH) – интерфейсы системного уровня и их привязка к языку Си, где описываются обязательные интерфейсы между прикладными программами и операционной системой, в частности – спецификации системных вызовов;
- Shell and Utilities volume (XCU) – определение стандартных интерфейсов командного интерпретатора (т.н. POSIX-shell), а также базовой функциональности UNIX-утилит;
- Rationale (Informative) volume (XRAT) – дополнительная, в том числе историческая, информация о стандарте.

Стандарты POSIX жестко определяли базовую функциональность систем и приложений. Каковая, ради сохранения совместимости, и не должна изменяться. В то же время расширению функциональности они отнюдь не препятствовали. Так, любая из предусмотренных для POSIX-систем команда должна была иметь определенный набор опций, предназначенных для выполнения базового набора действий. Однако никто не препятствует введению для данной команды дополнительных возможностей, реализуемых через опции, стандартом не предусмотренные. Важно только, чтобы первоначальная функциональность команды оставалась неприкосновенной.

Именно на стандарты POSIX в первую очередь и опирался Линус Торвалдс, создавая свою ОС по мотивам MINIX, о чём скоро и пойдёт речь.

Глава четвёртая. Рождение Linux

Пока, как было сказано в предыдущих главах, в Беркли и его окрестностях поневоле занимались юридическим крючкотворством, а в мире проприетарных UNIX'ов пытались выполнить заветы Великого Мао по упорядочиванию стилей, на другом конце света, в Финляндии, некий студент по имени Линус Торвальдс размышлял, что же ему делать с только что приобретённым IBM PC/386. И, как ни странно, результаты его размышлений оказали не меньшее влияние на нашу историю, нежели многолетний труд исследователей, финансируемых правительством мировой державы, и инженеров на окладе крупнейших компьютерных фирм.

Впрочем, редкий линуксописатель не описывал рождественскую сказку о том, как бедный студент копил деньги на 32-битный компьютер, а потом сочинял программу терминального доступа к удалённой университетской машине, которая затем превратилась в полноценную ОС. Известна она и в версии от создателя этой ОС – самого Линуса Торвальдса. Так что пересказывать её в очередной раз я не буду. А попробую, в меру своего понимания, выявить предпосылки появления Linux. Правда, и к вопросу о бедном студенте мне придётся ещё вернуться.

Предпосылка первая, «железная»

О первопричине появления Linux я уже говорил – она «железная», и вызвана триумфальным шествием Советской власти (то есть, пардон, Intel-совместимых PC). И главную роль здесь сыграло появление в 1985 году первого 32-разрядного процессора от Intel – 80386. Что дало возможность использовать UNIX на дешёвых и общедоступных персоналках. А начавшееся вслед за тем производство процессоров Intel 80486 вплотную приблизило по производительности эти самые «писюки» к рабочим станциям на RISC-процессорах.

Надо сказать, что параллельно с платформой на процессорах Intel и их клонах, таких, как AMD, Cyrix, UMC, существовали и другие варианты персоналок, на иной аппаратной базе – процессоре Motorola

680x0. Ими были Macintosh и Amiga, с их System # и AmigaOS, соответственно. И обе эти платформы были не прочь претендовать на высокое звание «народного» компьютера. Особенно первая – в силу ряда причин, как технических, так и маркетинговых, у неё, казалось, были на то все основания.

Однако обе эти платформы в конечном счёте оказались в... нет, не там, где вы подумали в меру своей испорченности, а в нише. Точнее, каждая в своей: Macintosh снискал любовь той прослойки, которую в советское время обзывали творческой интеллигенцией, а Amiga получила широкое распространение в узких кругах артистической, в первую очередь музыкальной, богемы.

Не в последнюю роль в расползании по нишам и Mac'a, и Amiga сыграла цена конечных решений. Что плавно подвело нас ко второй предпосылке.

Предпосылка вторая, «денежная»

Как это ни печально признать русским интеллигентам, но второй (по счёту, но не по значению) предпосылкой для возникновения Linux'a была цена. Потому что как раз в начале 90-х годов машины с 32-битными процессорами от Intel (а затем от AMD и Cyrix) стали доступны народу.

Причины этого явления уходят ещё в 80-е годы. Когда IBM выпустила свой первый PC, ещё 16-битный, руководство фирмы не относилось к этой затее совсем серьёзно, и не рассчитывало «нарубить капусты» на этом рынке. В результате все спецификации архитектуры были открыты, и не имелось ни технических, ни юридических препятствий к клонированию этих машин.

Чем немедленно воспользовались многие производители, сначала американские, а затем европейские и восточноазиатские. И очень скоро количество клонов (так называемых IBM-совместимых машин) превысило число оригинальных IBM PC-XT и PC-AT. А потом была выпущена (в 1986 году) и первая машина на 32-битном процессоре от Intel (80386). И выпустила её вовсе не IBM, а фирма Compaq – она получила имя Compaq DeskPro 386.

Тут в IBM спохватились, что рынок, созданный их стараниями, от них уходит – и разработали новую архитектуру, также на процессоре

x86, но с закрытыми спецификациями, исключаящими клонирование – PS/2. Однако было поздно: рынок был заполнен стандартными PC. К рубежу 80-х и 90-х годов их не производил только ленивый – начиная от компьютерных гигантов типа Hewlett-Packard и заканчивая дядюшкой Мяо с Малой Арнаутской улицы острова Тайваня. К которому вскоре присоединился его родич – дядюшка Ляо с её продолжения на континентальном Китае. И с тех пор эти два дядюшки (особенно второй) обеспечивают мир большей частью комплектующих для компьютеров и даже готовыми системами – под какими бы торговыми марками последние не продавались.

Результатом был лавинообразный обвал цен. Первый IBM PC-XT стоил более трёх тысяч долларов, а для полноценной работы требовал ещё и доукомплектования на треть этой суммы. Ко второй половине 80-х годов относятся слова Линуса Торвальдса:

Раньше было такое правило, что все компьютеры начального уровня стоили две тысячи.

В первые же 90-е годы порог вхождения в новый дивный мир компьютеров снизился до тысячи баксов. То есть компьютеры не скатились ещё до цены ширпотреба, как ныне. Но все, кому такие машины нужны были «по делу», были в силах приобрести их, причём не только «у них», но и «у нас».

Предпосылка третья: «студенческая операционка»

В числе тех, кому такие машины были действительно нужны, и кто мог, тем или иным путём, ими обзавестись, оказались и студенты, обучавшиеся по специальностям, связанным с Computer Science. Но вот с операционной системой для 32-битных компьютеров у студентов оказалось хуже: FreeBSD ещё не начала своё свободное плавание, а коммерческие UNIX'ы стоили несколько дороже, чем необходимое для их работы «железо». Да и не густо было с ними: на архитектуре i386 были способны работать только SCO UNIX и Xenix – кастрированный UNIX от Microsoft.

И тут в канву нашего сюжета вписывается «игрушечная» ОС MINIX. Она была явлена миру в январе 1987 года Эндрю Таненбаумом

(Andrew Stuart Tanenbaum), профессором Университета Вриье, Амстердам, Нидерланды. Где преподавал он ни что иное, как Computer Science, хотя и был по специальности физиком.

Университетское образование в области компьютерных наук в 80-е годы прошлого века базировалось преимущественно, если не исключительно, на UNIX. Что, как явствует из сказанного ранее, создавало для студентов известные трудности.

Так вот, Таненбаум вёл в именованном университете курс UNIX, к которому написал собственный учебник – Operating Systems: Design and Implementation. Но изучать UNIX без системы – все равно, что обучаться музыке без инструмента. А с инструментом-то как раз и была напряжёнка. И ему не осталось ничего другого, как такой инструмент изготовить. Им-то и стала ОС MINIX (в дальнейшем получившая имя MINIX 1), вышедшая в свет в 1987 году.

Это была маленькая и компактная операционка, работавшая на машинах с архитектурой i386. Доступность MINIX усугублялась ещё и тем, что ее можно было скомпилировать даже в 16-битном варианте, и в этом качестве она становилась пригодной к использованию не только на PC-AT (80286), но даже, как говорят, на XT'шках, то есть на машинах с процессором 8086/8088.

Распространялась она исключительно как сопроводительный материал к упомянутому выше учебнику. Весь комплект, по свидетельству Линуса Торвальдса, стоил 169 долларов при заказе по почте. Что на самом деле не так дорого: в те годы на Западе, только-только переставшем загнивать, ни одно специализированное книжное издание не стоило дешевле 100 баксов. Так что фактически основная, если не вся, затратная часть для пользователя приходилась на книжку, да и дискеты были не так дешёвы. Сама же ОС как таковая могла рассматриваться в качестве бесплатного приложения к книге и носителям. И, во всяком случае, это было несоизмеримо дешевле тех тысяч долларов, в которые обходилась лицензия на любой из существовавших тогда проприетарных UNIX'ов. Требовавших, к тому же, сущей безделицы в виде соответствующей рабочей станции в несколько десятков тысяч.

Разумеется, ОС MINIX распространялась в сопровождении исходных текстов, предназначенных для изучения и потрошения. Необходимость в котором возникла очень скоро.

Дело в том, что, предназначенная исключительно для учебных целей, ОС MINIX в принципе не была приспособлена для выполнения каких-либо реальных задач. Однако шаловливые студенческие руки (и не только студенческие) так и чесались прикрутить ее к чему-либо пригодному для практического использования. В результате система очень быстро обросла всякого рода патчами, из которых главным был патч от австралийца Брюса Эванса. После наложения этих патчей система становилась способной выступать как платформа разработчика. Именно на такой патченной системе Линус Торвальдс спустя несколько лет начнёт создавать свою операционную систему.

Однако сама по себе MINIX по прежнему распространялась исключительно в первозданном виде – как чисто учебная система, и лишь в сопровождении книги (или, напротив, сопровождая книгу). То есть, будучи открытой, она не была свободной. Ибо права на MINIX принадлежали издательству Prentice-Hall, выпустившему учебник Таненбаума. В сущности, правовой статус MINIX был точно таким же, как и обычной книги. Что, однако не мешало тому, что на протяжении десяти, а то и более, лет по ней учились поколения студентов как до Торвальдса, так и после него.

Надо заметить, что Таненбаума нельзя рассматривать только как предтечу Линуса, а его систему – как трамплин для его разработки. Кроме упомянутой выше *Operating Systems: Design and Implementation* (в переводе: *Операционные системы: разработка и реализация*), его перу принадлежат:

- *Computer Networks* (Компьютерные сети);
- *Modern Operating Systems* (Современные операционные системы);
- *Structured Computer Organization* (Архитектура компьютера);
- *Distributed Systems: Principles and Paradigms* (Распределённые системы. Принципы и парадигмы).

Все они по праву относятся к классике жанра IT-литературы, выдержали по несколько изданий (*Computer Networks* и *Structured Computer Organization* – аж по пять), и переведены на многие языки. В том числе и на русском языке они изданы издательством «Питер» в серии «Классика Computer Science».

Труды Эндрью Таненбаума не ограничиваются ОС MINIX и перечисленными выше книгами. Тут уместно вспомнить проекты Амоеба – разработанную «с нуля» открытую микроядерную распределённую операционную систему, и Globe – распределённую объектную систему, предназначенную для поддержки миллиардов пользователей «в мировом масштабе».

главной разработке Таненбаума, MINIX, судьба также уготовила вторую жизнь. Долгое время она продолжала эволюционное развитие в качестве учебной системы – были выпущены версии MINIX 1.5 (1992 год) и MINIX 2 (1997 год), представлявшие собой «песочницы» для начинающих юниксоидов. Однако кардинал лелеял коварные замыслы: превратить MINIX в полноценную операционную систему, реализующие его представления о том, какой должна быть современная ОС. А заодно – сделать ее свободной в полном понимании этого слова: ведь «несвобода» предыдущих версий объяснялась не жадностью профессора, а спецификой издания и распространения.

Результатом явился анонс новой операционки, MINIX 3, который состоялся 24 октября 2005 года. И о которой я расскажу подробнее в главе восьмой.

Герой выходит на сцену

Итак, толчком для написания Линусом собственного ядра послужила MINIX – «студенческая» операционка Энди Танненбаума, с помощью патчей приспособленная для выполнения практической работы.

Однако сам Линус не занимался «доведением MINIX до ума». Не использовал он также и код какой-либо из реализаций UNIX или BSD. Он воссоздал функциональность ядра UNIX с нуля – руководствуясь описаниями системных вызовов, данными в соответствующем стандарте POSIX. И потому Linux не является ни клоном System V, ни клоном BSD – хотя в ней и использована схема инициализации в стиле первой, да и идейное влияние второй, безусловно, имело место быть. Как было сказано в предыдущей главе, Линус Торвальдс, создавая свою ОС по мотивам MINIX, опирался почти исключительно на стандарты POSIX.

Linux создавался на машине с процессором i386 для архитектуры Intel и первоначально – только для неё. Более того, долгое время Линус вообще сомневался, что его система когда-либо сможет быть портирована на любую иную аппаратную платформу. И потому соответствие стандартам в данном случае преследовало целью не переносимость Linux самого по себе, а в первую очередь возможность компиляции в этой ОС всего ранее созданного программного ассортимента для UNIX и POSIX-совместимых систем вообще.

Лично Линусу принадлежит честь разработки ядра Linux и файловой системы ext (то есть Extended – расширение для файловой системы Minix). В качестве среды для работы он выбрал bash – командную оболочку, разрабатываемую в рамках проекта GNU. Для сборки своего кода был использован компилятор gcc (GNU C Compiler), а главной общесистемной библиотекой функций языка Си выступала GNU-реализация ее, glibc. Все прочее системное окружение ядра – комплекс пакетов, который можно назвать Base Linux – также в основном происходит из проекта GNU. Да и при выборе политики распространения Линус в конце концов остановился на лицензии GPL – детище Ричарда Столлмана и его Фонда свободного программного обеспечения (FSF).

На основании сказанного выше часто полагают, что ОС Linux должна на самом деле именоваться GNU/Linux. Правильно ли это?

По моему скромному мнению, нет. Конечно, роль программного обеспечения, разработанного в рамках проекта GNU, для развития Linux как пользовательской платформы переоценить трудно. Однако не проект GNU ухватился за столь недостающее ему ядро. Напротив, это Линус для обеспечения работы своего ядра использовал отдельные компоненты из GNU-арсенала. В полном, к слову сказать, соответствии с духом и буквой GPL и движения FSF. Впрочем, те, кто считает нужным подчеркнуть роль компонентов GNU в составе Linux, вполне могут это делать – и делают.

Добавлю ещё, что неотъемлемой чертой Base Linux является альтернативность его комплектации. И потому ОС Linux – не только (а может быть, и не столько) ядро и набор базовых программ, но в первую очередь алгоритм для построения такого набора. И создание такого алгоритма – второе, после написания кода ядра, великое достижение Линуса.

Наконец, Линус оказался создателем уникального метода разработки масштабных проектов Open Source, того самого, который Эрик Реймонд позднее назовёт методом большого базара. Впрочем, справедливости ради следует отметить, что в данном случае и он изобрёл велосипед – аналогичный способ привлечения дармовой рабочей силы использовал Том Сойер в своих «Приключениях». Однако, если инструментами Тома были сердцевина от яблока и крыса с привязанной к хвосту верёвкой, чтобы удобнее размахивать ей над головой, то орудием Линуса оказался Интернет.

Рождение Linux дало толчок к окончательному оформлению движения Open Source, несколько обособившемуся от сообщества Free Software – хотя и по сей день это существенно пересекающиеся множества. Но, если апологеты FSF, во главе с Ричардом Столлманом, декларируют, что всё программное обеспечение должно быть свободным, исходя из моральных и идеологических соображений, то для сторонников Open Source характерен более прагматический подход. Их принцип – открытое программное обеспечение следует использовать не потому, что оно открытое, свободное или бесплатное. А потому, что оно просто лучше проприетарного. В том числе – и в следствие публичной экспертизы, реализуемой именно благодаря внедрённому Линусом методу Тома Сойера.

Глава пятая. Берклиада, тур второй

NetBSD и OpenBSD: первый форк в благородном семействе

Уделив на предыдущих страницах столько внимания FreeBSD, я невольно оставил в тени её родную сестру – NetBSD. А между тем она была первой в ряду свободных ОС BSD-клана.

Кроме того, существует мнение, не лишённое оснований, что именно NetBSD воплощает в себе дух первозданного UNIX *par excellence*. По крайней мере, это верно в отношении максимально полной независимости от аппаратной части: в отличие от FreeBSD, первоначально ориентированной только на «демократическую» платформу i386, NetBSD исходно разрабатывалась как кроссплатформенная: с первого для своего существования она поддерживала рабочие станции HP 9000 и Sun, компьютеры Amiga и Macintosh, полузабытые машины PC532 (на процессорах серии NS32000), а также, конечно, обычные персоналки с i386 и выше.

Наконец, история NetBSD также не лишена драматизма. И если драматизм в развитии FreeBSD носил, как мы видели, детективно-юридический характер, то здесь можно говорить скорее о драме идей, завершившейся... скоро мы увидим, чем она завершилась.

Ранее, в главе второй, я уже упоминал, что будущая NetBSD отделилась от проекта 386BSD в начале 1993 года, в тот момент, когда он оказался заброшенным своим создателем, Биллом Джолитцем, а «заплаточная группа» – будущие разработчики FreeBSD, ещё не развернули свою деятельность.

Ядро новой группы составили Крис Деметрио (Chris Demetriou), Тео де Раадт (Theo de Raadt), Адам Гласс (Adam Glass) и Чарльз Ханнам (Charles Hannum). Разрабатываемая ими система получила имя NetBSD, предложенное Тео – как раз в это время Интернет начал широко распространяться в узких пока кругах, особо приближённых к разработке.

Первым релизом новой системы считается версия 0.8, вышедшая в апреле 1993 года. Однако истинную мультиплатформенность она

обрела в версии 1.0 – правда, та последовала не очень скоро, осенью 1994 года.

А вскоре, в декабре того же года, внутри группы перворазработчиков разгорелся конфликт: Тео разошёлся во мнениях относительно дальнейшего развития системы с остальными членами группы. И дело закончилось тем, что весной 1995 года ему был закрыт доступ к дереву исходных текстов NetBSD.

Это расхождение, как идейное, так и чисто личное (переписка по данному вопросу была опубликована де Раадтом в сети), и послужило причиной первого форка в BSD-клане. Поскольку исходники NetBSD были полностью свободны, Тео взял их за основу, модифицировал согласно своим представлениям о том, «как надо», и основал проект, получивший имя OpenBSD.

Во главу угла новой системы были положены два аспекта: свобода от любых компонентов, могущих ограничить её распространение, и безопасность. Последняя неразрывно была связана с криптографией. И, дабы избавиться от оков, налагаемых законами США на распространение «сильных» криптографических технологий, Тео покидает Калифорнию и перебирается в Канаду, куда некогда эмигрировала из ЮАР его семья, дабы откосить сыновей от службы в армии антинародного режима апартеида. Там, в почти родном городе Калгари, университет которого Тео закончил до работы над NetBSD, и обосновывается штаб-квартира нового проекта.

В результате, после нескольких внутренних тестировочных версий, в октябре 1996 года, рождается новая ОС – выходит первый официальный релиз OpenBSD 2.0. С тех пор полугодовой релиз-цикл выдерживается неукоснительно – очередные версии появляются каждую весну и осень.

Однако завершим историю NetBSD. В последующие годы она была портирована на всё «железо», которое может запускаться, и немножко – на то, которое запускаться не способно. Чтобы убедиться в этом, достаточно посмотреть «лист совместимости» на сайте проекта – в них обнаружатся и VAX, и Sun Sparc, и RISC-системы от Hewlett-Packard, и DEC Alpha, и PowerPC, и Amiga, вкуче с мало кому ведомыми Acorn, Atari, Sharp, и так далее, и так далее, и так далее... Список столь обширен, что PC-платформа как-то просто теряется в середине его.

В качестве системы пакетного менеджмента в NetBSD в 1997 году была принята `pkgsrc`, разработанная по образу и подобию портов FreeBSD, но почти сразу также приобретшая кросс-платформенный характер. Кроме родной ОС, она официально поддерживается для многих UNIX-подобных операционных систем: Solaris, Linux, Darwin (Mac OS X), FreeBSD, OpenBSD, IRIX, AIX, DragonFlyBSD, HP-UX, QNX. Правда, это не значит, что в них она широко используется: во всех этих ОС есть собственные развитые средства управления пакетами. В частности, в OpenBSD, ответившейся до возникновения `pkgsrc`, была просто заимствована система портов из FreeBSD. Относительно Linux мне известно несколько попыток прикрутить `pkgsrc` к Slackware. Лишь в DragonFlyBSD `pkgsrc` долгое время была принята в качестве штатной, но об этом мы поговорим в следующей главе. А в этой – вернёмся к FreeBSD и очередному перелому в её истории.

FreeBSD: десятилетие спокойствия

Мы оборвали историю FreeBSD 22-го ноября 1994 года – дне, когда было объявлено о выходе FreeBSD версии 2.0, после чего оценили, во что же этой ОС обошлась её свобода. На нынешней же странице посмотрим, как события развивались дальше.

Начиная с выхода первой «настоящей» версии FreeBSD (то есть 2.0), сложилась модель разработки этой операционной системы, реализуемая и по сей день. Впрочем, она была в значительной мере унаследована от стиля работы CSRG и свойственна всем системам берклианской линии.

С первого дня существования проекта все его компоненты (исходные тексты ядра, системного и пользовательского окружения, документация, багрепорты, архивы рассылок) существуют в открытом виде, и доступ к ним может получить любой желающий через веб-сайт или ftp-сервер проекта и их зеркала по всему миру.

Все участники разработки FreeBSD объединяются в три кольца. Первое, как бы внешнее, кольцо включает в себя многочисленных добровольцев со всего мира, работающих над отдельными компонентами системы – начиная от ядра и до сопровождения портов, а также занимающихся составлением и переводом документации. Разработчики (как, впрочем, и всё остальное прогрессивное

человечество) имеют свободный доступ к дереву исходных текстов системы, но вносить в него изменения непосредственно не могут: свои наработки они должны передавать «по команде» для утверждения.

Утверждением занимаются члены второго кольца — так называемые коммитеры (committers). Кроме контроля над деятельностью разработчиков, они и сами занимаются разработкой какой-либо из подсистем FreeBSD и могут вносить изменения (как свои, так и курируемых ими разработчиков) в соответствующие ветви дерева исходных текстов.

Однако полномочий на изменение дерева исходников в целом не имеют и коммитеры — это привилегия ядра команды (core team), в функции которых, кроме разработки собственных узлов системы, входит также отслеживание изменений, вносимых коммитерами, и разрешение противоречий между ними, буде таковые возникают. Иными словами, на них возложен учёт модификаций системы и контроль над её целостностью.

Раз сложившись, модель разработки функционировала и функционирует по сей день. И сохраняющееся единство системы на протяжении почти полутора десятков лет доказало её действенность. Конечно, кое-какие коллизии возникали — но их мы рассмотрим, когда подойдёт время.

А пока вернёмся немного назад, к началу истории собственно FreeBSD, и посмотрим, что же послужило причиной её почти мгновенной популярности.

Первой составляющей успеха новой операционки стала утилита `sysinstall` — программа установки и сквозного конфигурирования системы, почти не изменившаяся по сей день за исключением мелких, но приятных усовершенствований. Ныне она кажется архаичной и часто подвергается критике со стороны пользователей Linux, привыкших в последние годы к красивым графическим инсталляторам и конфигураторам. Однако не будем забывать, что она появилась до первых аналогов в Linux — в нём в те годы ничего подобного не было ни в одном дистрибутиве.

Более того, под влиянием `sysinstall` возникла не только программа установки практически одновозрастной Slackware — влияние её сказывалось ещё и в начале нынешнего тысячелетия, непосредственно — в инсталляторах таких дистрибутивов, как CRUX и

Archlinux, косвенно – в установочной программе Zenwalk’a. Что же до сквозных графических конфигураторов, то первого из них, Drackconf из Mandrake, Linux’у пришлось ждать около пяти лет (первые варианты YAST, упомянутого на странице про Suse, функционировали в текстовом режиме).

Ничуть не менее важной составляющей FreeBSD была система портов и пакетов. Ибо это была первая в истории FOSS цельная система сборки и установки программ с автоматическим разрешением их зависимостей. Вспомним, что одновозрастная Slackware таковых не имела вообще (и, штатно, не имеет и по сей день), а rpm и dpkg на большее, чем сообщение о нарушении зависимостей, способны не были. Как, впрочем, не способны и по сей день – до появления изощрённых механизмов apt и его последователей было ещё очень и очень далеко.

Таким образом, основные особенности, определившие потенциал FreeBSD, в том числе, и как пользовательской платформы, были заложены уже в её первой «настоящей» версии. Почему же она не реализовалась в этом качестве, уступив пальму первенства Linux’у? Тайна сия велика есть, хотя некоторые предположения на этот счёт сделать можно.

Вспомним, кем были первые пользователи первых FOSS-систем. Это были, с одной стороны, разработчики их же самих, с другой – сетевые администраторы и Интернет-провайдеры. А для первых более свободная и динамичная модель разработки Linux’a, видимо, казалась более привлекательной, нежели более иерархическая и «камерная» модель FreeBSD. С другой стороны, обеспечиваемые последней стабильность и предсказуемость оказались более востребованными именно администраторами, для которых надёжность была важнее фронтирности.

А потом уже работал просто стереотип мышления: за FreeBSD закрепилась репутация серверной платформы, тогда как от Linux’a ждали «поворота лицом к конечному пользователю». И, надо сказать, стереотип этот работает и по сей день.

Однако я опять отклонился от генеральной линии. Успех первой версии FreeBSD был закреплён выходом версии следующей, получившей номер 2.05 и ликвидировавшей те самые «недотёсанные углы», о которых упоминал Хаббард.

Дальше время опять замедляет свой ход. Впереди были долгие годы плавной эволюции. Примерно два-три раза в год выпускается новая версия системы (2.1.x, затем – 2.2.x), она обрастает приложениями и утилитами (значительная часть которых происходит из проекта GNU и Фонда свободного программного обеспечения), совершенствуется ядро, улучшается (как это ни странно для, казалось бы, чисто американской по происхождению системы) интернациональная поддержка.

В ноябре 1996 г. происходит событие, определившее структуру развития FreeBSD на долгие годы (с некоторыми оговорками – до сего дня): ветка 2.x.x была выведена из активной разработки, получив имя STABLE. Отныне, вплоть до последнего релиза (2.2.8 в ноябре 1998 года), в ней лишь исправляются ошибки и вносятся мелкие безопасные изменения. А все долговременные и принципиально новые разработки концентрируются в версии 3.0-CURRENT. Каковая претворяется в STABLE в октябре 1998 г.

Начиная с ветки 3 (версия 3.4, судя по архивам), начинаются первые попытки портирования FreeBSD на архитектуры, отличные от i386. Первым претендентом на портирование стали машины с процессорами DEC Alpha, доживавшие тогда свои последние дни. Тем не менее, поддержка этой платформы осуществлялась долгое время, пока не была прекращена с выходом 7-й ветки.

С этого времени и вплоть до ответвления единовременно развивается две ветки FreeBSD – STABLE, предназначенная для широкого применения, и CURRENT, ориентированная главным образом на разработчиков и энтузиастов. Так, в январе 1999 г. обособляется ветка 4.0-CURRENT, обретшая статус стабильной в марте 2000 г.

Ветке 4.X суждено было стать самой «долгоиграющей» во всём дереве развития FreeBSD. Правда, на протяжении только 2000 года вышло ещё три её релиза – 4.1, 4.1.1 и 4.2, выступавшие в роли своего рода обкаточных для всех новшеств этой ветки. Которая в дальнейшем стабилизировалась, и последующие её версии, вплоть до последней, 4.11, вышедшей в январе 2005 года, содержали в основном исправления и косметические изменения. И кстати, версия 4.11 поддерживалась более двух лет после её выхода, а практически, насколько я знаю, используется чуть ли не по сей день.

FreeBSD: переломный год

Однако закулисно на протяжении трех лет шла незаметная, но большая работа по коренному изменению FreeBSD, завершившаяся появлением в январе 2003 года первой версии новой, 5-й ветки. В нарушение описанной выше закономерности, она очень долго (вплоть до версии 5.3), не получала статуса STABLE, а позиционировалась как «ново-технологический релиз». В сущности, стабильной она так и не стала, выступая скорее как прототип грядущей ветки 6.

Тем не менее, выход в свет 5-й ветки FreeBSD был, пожалуй, самым революционным событием со времен ее появления как самостоятельной операционной системы. Почему год её выхода я и назвал годом переломным — конечно, не столь драматичного, как события 1991-1993 годов, но тоже сопровождавшегося не только приобретениями, но и потерями.

Именно в 5-й ветке FreeBSD, начиная с самой первой пред-релизной версии, предназначенной для разработчиков (июнь 2002 года), был заложен тот потенциал, который обусловил её применимость в качестве современной настольной операционки для конечного пользователя. Это и последовательно модульный подход, позволяющий обходиться без пересборки ядра для поддержки важных для пользователя особенностей, и файловая система устройств, облегчающая работу с устройствами «горячего подключения» (такими, как USB-накопители, сканеры, цифровые камеры), и эффективная работа с ATA RAID (а в дальнейшем и с винчестерами Serial ATA), и многое другое.

В 5-й ветке был продолжен и курс на кросс-платформенность: с первого же релиза (5.0) в ней появляется поддержка AMD64, Sparc64 и IA64 (Merced, он же Itanium), несколько позднее, в версии 5.5 (май 2006 года) — PowerPC, то есть практически всех 64-битных платформ. Если вспомнить, что объект первого портирования FreeBSD, процессор Alpha, также был 64-битным, то это можно рассматривать как подготовку к эпохе 64-разрядных вычислений. Иначе довольно трудно было бы объяснить усилия, затрачиваемые на поддержку архитектур или мёртвых, как Alpha, или отмирающих, как PowerPC (обратим внимание, что выход порта для него произошёл через год после перехода Apple на «камни» от Intel), или, наконец, поставляемых

обычно с собственными операционными системами, как Sparc64 и Itanium, и вряд ли предоставляющих широкое поле для инсталляций FreeBSD.

Шестая ветка FreeBSD, появившаяся в ноябре 2005 года, не была столь богата инновациями, а скорее являла собой логическое продолжение тенденций, заложенных в ветке 5.X. Хотя многие из её новых особенностей оказались очень важны для конечного пользователя. Среди них – поддержка высоких разрешений в так называемой графической консоли, обратнo портированная из DragonFlyBSD (о которой речь пойдет в следующей главе), возможность работы с популярными файловыми системами Linux – ReiserFS и XFS (правда, в режиме «только для чтения»). Наконец, версии 6-й ветки демонстрировали значительное повышение быстродействия, особенно – операций с файлами. До этого за новшества, привнесённые в 5-ю ветку, приходилось расплачиваться снижением скорости работы по сравнению с веткой 4-й, в ряде случаев – весьма существенным.

Интересно, что FreeBSD 6-й ветки в варианте для архитектуры AMD64 демонстрировала некоторый прирост быстродействия относительно своей 32-битной сестры – не очень значительный, но всё-таки видимый невооружённым тестами взглядом. В отличие от Linux, дистрибутивы которого в своих тогдашних 64-битных инкарнациях были как минимум не быстрее 32-битных, а то и медленней на некоторых операциях, в первую очередь файловых. Вполне возможно, что в этом и сказалась «тренировка» на неактуальных, казалось бы, чисто 64-битных платформах...

Тем не менее, и 6-я ветка не производила впечатления долгожителя, хотя последняя её версия (6.4, появившаяся в ноябре 2008 года) пользовалась статусом промышленного, хотя и «староватого» (legacy) релиза, и поддержка её осуществлялась довольно долго.

Новым рубежом в развитии FreeBSD стала ветка 7 (февраль 2008 года), в которой, начиная с первых пре-релизных версий (с осени 2007 года), поддерживается файловая система ZFS – одно из самых перспективных явлений в этой сфере, разработанное фирмой Sun для своей операционки Solaris в 2004 году. И хотя поддержка ZFS во FreeBSD долгое время имела статус экспериментальной, и эта

файловая система не рекомендовалась к промышленному применению, было ясно, что её стабилизация в рамках этой ОС – не более, чем вопрос времени.

И это время наступило: в середине сентября 2009 года, в канун выхода релиза 8-й версии, Павел Давидек (Pawel Dawidek) объявил о готовности ZFS во FreeBSD к промышленной эксплуатации. Однако до повсеместного её внедрения было ещё далеко: ZFS не поддерживалась на стадии установки программой `sysinstall` – и соответственно, на ней нельзя было расположить корень файловой иерархии. Фактически о её окончательном утверждении во FreeBSD можно говорить, начиная с 10-й ветки – финальный её релиз вышел за несколько дней до окончания редактирования этой книги.

Оглядываясь вокруг

А теперь я опять вынужден отступить от хронологической последовательности и хотя бы беглым взглядом окинуть события, происходившие одновременно с переломом в развитии FreeBSD и дальнейшими событиями.

В отличие от Linux, FreeBSD изначально не сегментировалась на множество дистрибутивов (тех самых, которые будут предметом рассмотрения второй части), хотя время от времени она давала боковые побеги, например, PicoBSD – вариант 3-й ветки на одной дискете.

Далее, существовало (и частично существует по сей день) несколько проектов создания LiveCD на основе FreeBSD – в основном специализированных сборок для системных администраторов. Они отражали всё ту же общую тенденцию – доминирование в развитии FreeBSD серверного направления. Хотя и здесь она была теснима мало-помалу, с одной стороны, соплеменным Linux'ом, с другой – классово чуждым Windows, но сохраняла твёрдые позиции.

А вот о настольных применениях этого сказать нельзя. Если Linux понемногу пробивал дорогу на пользовательские десктопы, то FreeBSD, похоже, к этому и не стремилась – по крайней мере, до недавнего времени. Статистика заходов на сайты, тематически связанные с UNIX и Open Sources, показывает, что доля FreeBSD среди клиентских машин ничтожно мала.

Однако несколько более удачных попыток изменить сложившееся положение было предпринято. Первой можно считать выход весной 2005 года PC-BSD – как легко догадаться из названия, варианта BSD для персонального использования. Она базировалась на актуальных в данный момент версиях FreeBSD и была снабжена красивым (и удобным) графическим инсталлятором с внутренними средствами автоматического конфигурирования применительно к наличествующему оборудованию, что позволяет в считанные минуты развернуть полноценную рабочую станцию с KDE и его приложениями, в том числе графическими и мультимедийными.

Особенностью PC-BSD являлся собственный формат пакетов, резко рвущий с традициями UNIX в отношении зависимостей – необходимые библиотечные функции встраиваются непосредственно в бинарный пакет, а не вызывались из внешних слинкованных библиотек. Однако она унаследовала и традиционные для FreeBSD методы обновления системы в целом, а также систему портов для установки приложений.

Проект PC-BSD не остался одиноким на ниве пользовательских десктопов, производных от FreeBSD: считанные месяцы спустя (лето 2005 г.) аналогичный по сути, но несколько иначе реализованный проект был объявлен под именем DesktopBSD.

Следует подчеркнуть, что ни PC-BSD, ни DesktopBSD не являлись отдельными дистрибутивами в том понимании, в каком этот термин применяется к вариациям на тему Linux'a. И тем более это – не самостоятельные системы, поскольку и та, и другая после установки превращаются в самую обычную (и полноценную) FreeBSD. Точнее, это – именно дистрибутивы в буквальном смысле слова, то есть способы распространения операционной системы FreeBSD, адаптированные для конечного «десктопного» пользователя.

главное, чем и PC-BSD, и DesktopBSD отличались от своей материнской системы – это их программами инсталляции. Если для FreeBSD в этом качестве на протяжении многих лет применялась текстовая (псевдографическая) программа sysinstall, то в основу установщиков ее «юзерофильных» разновидностей лег BSD Installer.

Это – совершенно самостоятельный проект, цель которого, как нетрудно понять из названия, – разработка универсального установщика для любых BSD-систем. Отличительная его особенность

— в том, что собственно низкоуровневая его часть может быть надстроена различными текстовыми или графическими интерфейсами. Последние и использованы в PC-BSD и DesktopBSD. Текстовый же вариант инсталлятора был использован в DragonFlyBSD (см. следующую главу).

Существовала также попытка Эндрю Тернера (Andrew Turner) прикрутить (в рамках программы Google's Summer of Code 2005) BSD Installer и к собственно FreeBSD – взамен sysinstall. Впрочем, развития она не получила – последнюю версию сборки FreeBSD с этим инсталлятором датируется маем 2006 года.

Наконец, в рамках все той же программы Google's Summer of Code (теперь уже – 2007) Иваном Ворасом (Ivan Voras) была предпринята еще одна попытка одеть инсталлятор FreeBSD во фрак – посредством программы `finstall`. В отличие от всех предыдущих вариантов, она основывалась не на движке BSD Installer, а на собственном back-end'e, и имела построенный на библиотеке Gtk front-end, запускавшийся с LiveCD из среды XFce. Сама же устанавливаемая им система была самой обычной FreeBSD текущей (current) версии для архитектуры i386. Правда, проект этот был быстро и полностью заброшен.

Интересные побеги на древе FreeBSD – гибридные системы FreeBSD/Linux, то есть попытки использования её ядра в обрамлении иной инфраструктуры, заимствованной из различных дистрибутивов Linux. Таких проектов некогда было два: Debian GNU/FreeBSD и Gentoo/FreeBSD.

Проект Debian GNU/FreeBSD первоначально существовал в двух вариантах: `libc5-based Debian GNU/FreeBSD` и `gnu-libc-based Debian GNU/FreeBSD`. Оба они использовали ядро FreeBSD и пользовательское окружение проекта Debian, в частности, его репозитории и систему управления пакетами по механизму `apt`.

Первый проект, как нетрудно догадаться, в качестве главной системной библиотеки для портируемых приложений использовал BSD Libc – родную для всех BSD-систем. Пользовательское окружение (так называемый `userland`) в нём тоже было унаследовано от FreeBSD. Последнее обновление проекта датировалось апрелем 2002 года, после чего он прекратил своё развитие, и притом навсегда – в том числе и потому, что физически погиб его сервер. Хотя из всех «гибридов» он выглядел наиболее логичным, представляя собой, в сущности,

FreeBSD Distributions, в котором место традиционных портов заняла инфраструктура apt. Эта модель, как будет показано в главе девятой, была успешно реализована в системах Nexenta и Dyson.

Второй же из упомянутых проектов, известный под названием Debian GNU/kFreeBSD, участь сия миновала – более того, он ныне имеет статус официально поддерживаемого в рамках мегапроекта Debian. В отличие от предыдущего, здесь в качестве основы для пользовательских приложений выступает стандартная для Linux библиотека glibc (GNU C library), а большая часть родного юзерланда заменена GNU-аналогами: буква **k** в имени системы, видимо, призвана подчеркнуть, что от FreeBSD в ней было взято только ядро. Хотя с портированием инфраструктуры Debian существовали (да и существуют) проблемы, сама система, по утверждению её разработчиков, являлась вполне работоспособной. Впрочем, на меня она произвела впечатление химерической смеси бульдога с носорогом...

Проект портирования на ядро FreeBSD инфраструктуры Gentoo примечателен тем, что сам по себе дистрибутив Gentoo Linux был создан под сильным влиянием FreeBSD: в частности, портежи Gentoo представляли собой первоначально адаптацию портов FreeBSD к ядру и окружению Linux'a. И первая попытка обратного портирования системы портежей на FreeBSD была предпринята Грантом Гудьером (Grant Goodyear) через год после обретения Gentoo стабильного статуса, в сентябре 2003 года.

В последующем на базе этого развился самостоятельный проект Gentoo/FreeBSD, который то умирал, то вновь и вновь подвергался гальванизации. В январе 2007 года он был заморожен, а все его исходники удалены с зеркал проекта Gentoo. Причиной послужила несовместимость лицензий на отдельные компоненты BSD-системы с лицензией GPL, под которой распространяется Gentoo. И хотя принципиальная сторона этой проблемы была благополучно разрешена, ясных указаний о дальнейшей судьбе проекта я не обнаружил.

Тем не менее, FreeBSD легла в основу и совершенно самостоятельной операционной системы, корни которой уходят в тот самый год переломный 2003 год.

Где-то в середине июня 2003 г. Мэтт Диллон (Matt Dillon), известный, помимо всего прочего, и существенным вкладом в разработку системы виртуальной памяти FreeBSD, вместе с группой товарищей объявил о начале работы над новой ОС BSD-семейства DragonFlyBSD, о которой я расскажу в следующей главе. А пока настало время подвести

Предварительный итог...

Всем печально известна судьба якобинцев: Им теперь уж ни выпить, ни сытно поесть. Коль хотите вы в жизни чего-то добиться, То уроки истории надо учесть

... рассмотрения длинной и насыщенной истории операционной системы FreeBSD. Посмотрим, какие уроки мы можем извлечь из нее, дабы не уподобиться якобинцам из старой песенки студентов-историков Удмуртского государственного университета, фрагмент из которой приведён эпиграфом этого раздела.

Первый вывод – технологического плана. Хотя FreeBSD по анкетным, так сказать, данным и моложе Linux'a, за спиной у нее – долгая история совместного с UNIX развития. То есть она – система с прошлым. Что имеет и свои минусы, и свои плюсы. Не могу отказать себе в удовольствии процитировать Мэтта Диллона (Matthew Dillon):

Хотя некоторые считают BSD «старой» операционной системой, те из нас, кто работает над ней, видят ее скорее системой со «зрелым» кодом. *Лев Кертман*

Итак, первый вывод из истории FreeBSD – что это система сложившаяся, устоявшаяся, где-то даже консервативная. Но при этом – постоянно развивающаяся и совершенствующаяся.

Кроме того, FreeBSD (и ее предтечи) возникла и развивалась в университетской среде, не просто высококлассными программистами, но людьми с неслабой теоретической подготовкой. Следствием чего явилась исходная продуманность ее архитектуры. И опять упомяну Диллона:

В мире программирования алгоритмы становятся более важными, чем код, и именно из-за академических корней в BSD изначально большое внимание уделялось проработке алгоритмов.

И это – второй вывод из рассмотренной истории, который также можно отнести к технологии.

Третий же вывод, также следующий из академического происхождения FreeBSD, носит, условно говоря, гносеологический характер. Ученые (по крайней мере, те, кто заслуживает неругательного значения этого слова) – люди, основным стимулом деятельности которых является удовлетворение собственного любопытства. И FreeBSD, как творение академических исследователей, – это система, идеально для такого удовлетворения подходящая. Как сама по себе, так и как инструмент исследования в иных научных областях, в том числе – и далеких от Computer Science...

Последний же вывод из истории имеет отношение к тому, что называют затертым словосочетанием «общечеловеческие ценности»: мужество разработчиков, создавших FreeBSD, несмотря на все препоны юридического и организационного характера, более чем заслуживает уважения.

FreeBSD на Руси

Осталось сказать несколько слов об истории FreeBSD (и её предшественниц) на Руси. Несколько слов – не потому что история эта коротка. Отнюдь – существуют косвенные данные, что первые BSD-системы появились ещё в Советском Союзе, и еще до официального выхода самой FreeBSD. Достоверных сведений о тех временах далеких, теперь почти былинных, очень немного. Однако то, что примерно к 1995-1996 году сервера большинства крупных Интернет-провайдеров России работали под управлением этой операционной системы, можно считать более-менее установленным. Тогда же, вероятно, появились и первые зеркала проекта. Да и дистрибутивы FreeBSD на дисках в исполнении Walnut Creek стали доступны, по крайней мере, в Москве, также не позднее 1998-1997 года.

Характерно, что один из самых старых и по сей день самых обширных сайтов по UNIX-тематике – Opennet.ru, и BSD-минипортал его – один из самых полных источников информации по рассматриваемой тематике.

Из остальных Интернет-ресурсов того времени хотелось бы отметить фундаментальные работы на сайте Ивана Паскаля, заметки по FreeBSD Андрея Лаврентьева, материалы Игоря Сысоева, не потерявшие своего значения и по сей день. Ну и немалую роль сыграло первое систематическое руководство по FreeBSD на русском языке, написанное Андреем Федоровым.

Правда, о настольном использовании FreeBSD в те годы никто и не помышлял. Не была она избалованна и вниманием прессы. Хотя первые публикации о BSD-системах в компьютерных журналах не намного отстали во времени от статей про Linux – вспоминаем статью Вадима Колонцова «ОС BSD жила, живет и будет жить» (Открытые системы, № 3, 1997). Однако долгое время они выглядели редкой вкрапленностью даже на фоне не частых тогда материалов о Linux. Хотя были среди этих вкраплений и очень яркие сочинения. Например, статья того же Вадима Колонцова под характерным заглавием – «Я живу во FreeBSD». Сайт, на котором она была размещена, давно прекратил своё существование, однако при желании можно обнаружить её в переразмещенном виде.

Ряд статей о BSD-системах вообще и FreeBSD, в частности, появлялся в короткий период существования UNIX-раздела в журнале Byte/Russia (1999-2000 годы); к сожалению, ни одна из них ныне в Сети недоступна...

Переломным с точки зрения популяризации FreeBSD в широких массах стал 2002 год. Во-первых, тогда был начат (и к настоящему времени практически завершен) проект тотального перевода официальной документации по этой ОС, размещавшийся первоначально на freebsd.org.ua, а ныне доступный на главном сайте проекта.

Во-вторых, начиная с 2002 года, статьи о FreeBSD заняли прочное место на страницах общеконьюктерных журналов.

И, наконец, в-третьих, в конце 2002 – начале 2003 года подряд вышли первые русскоязычные книжки по FreeBSD, две переводные (очень похожие, так как принадлежат перу одних и тех же авторов –

Майкла Эбена и Брайана Таймэна) и одна отечественная (вашего покорного слуги). На протяжении 2003-2004 годов к ним прибавилось еще несколько изданий – Родерика Смита, Майкла Лукаса, а затем, в 2006 году, фундаментальная книга Керка МакКузика и Джорджа Невилл-Нила «FreeBSD: архитектура и реализация».

Конечно, среди изобилия книг о Linux, вышедших в последние годы, эти достижения книгоиздателей кажутся скромными. Однако следует учитывать, что FreeBSD – не Linux, и не распадается на такое множество дистрибутивов, каждый из которых требует отдельного описания.

Росло тем временем и количество русскоязычных Интернет-ресурсов, посвященных BSD-системам вообще и FreeBSD, в частности. Конечно, их число не поражает воображение, как количество Linux-сайтов. Однако к упомянутым ранее ресурсам присоединился специализированный bsdportal.ru.

На момент сочинения предыдущей онлайн-версии этого материала (2009 год) можно было констатировать, что FreeBSD заняла в настольном секторе прочное, хотя и весьма скромное, положение. Которое, впрочем, было вскоре утрачено. Но это уже не история, а скорее хроника текущего момента.

Глава шестая. Последний взлёт индивидуального ОСетворчества

Все операционные системы, о которых шла речь в предыдущих статьях, как и практически все их дистрибутивы, своими корнями уходят в прошлое тысячелетие. Однако и тысячелетие нынешнее, едва начавшись, ознаменовалось рождением новых операционок.

Об одной из них, системе, возникшей на наших глазах, на наших же глазах угасшей, но возродившейся вновь, хотя и под другим именем, я и хочу рассказать в этой статье. А в конце её попробую в очередной раз извлечь уроки из истории – тем более, что тема эта будет продолжена в ближайших главах. Итак, наша сегодняшняя героиня – AtheOS, позднее известная как Syllable.

Пролог

История горячего финского парня Линуса Торвальдса, в одиночку из ничего сочинившего операционную систему, знают все, интересовавшиеся историей ОСетростроения. А вот история норвежского парня Курта Скавена (Kurt Skauen, за точность транскрипции не ручаюсь) широкой известности не получила даже в узких кругах. Хотя Курт занимался своей разработкой ещё более в одиночку и ещё менее из ничего. Возможно, потому, что он парень ещё более горячий, вследствие чего его деятельность не имела таких последствий.

Однако начнём по порядку. Все системы, описанные ранее в этом цикле, так или иначе, генетически или парагенетически, связаны с первозданным UNIX'ом.

Так, все BSD-клоны, в сущности, ни что иное, как UNIX, очищенный от проприетарного UNIX-кода. MINIX, о которой упоминалось ранее и к которой мы вернёмся в ближайшее время, представляла собой модельную (или «игрушечную» систему UNIX). Linux же исторически – попытка воспроизведения функциональности UNIX-систем, вообще не используя код UNIX, а опираясь только на стандарты. И даже Hurd, в котором декларируется отход от принципов UNIX-архаики, подчинен единой идее: сделать все, как в UNIX, но

иначе. То есть в полном соответствии с известным рекурсивным высказыванием Ричарда Столлмана: GNU – GNU is Not UNIX. Правда, к счастью, всё, что до сих пор сделано в рамках проекта GNU, от этого меньшим UNIX'ом не стало. По крайней мере, пока.

Возникает вопрос – все ли в мире свободных ОС прямо и непосредственно происходит от UNIX? Как выясняется, не всё. И примером этому – некая свободная альтернативная операционная система, названная создателем AtheOS. Об этимологии её имени могу только гадать – но у меня оно ассоциируется и со славным городом Афинами, и с Афиной Палладой. Дальнейшую ассоциативную цепочку читатель легко построит сам.

Чем была AtheOS

Создателем AtheOS от начала и до конца всей истории выступал один-единственный человек – ранее упомянутый Курт Скавен. Согласно его декларации, AtheOS – своего рода *tabula rasa* (цитирую – «new clean desktop OS»), разработанная с нуля. То есть – не потомок UNIX, в отличие от BSD, и не реинкарнация ее, подобно Linux. И даже POSIX-совместимость Курт не возводил в абсолют, хотя и более-менее его стандартам следовал. В этом отношении напрашивается аналогия AtheOS с QNX, отнюдь не UNIX'ом, но местами сходной с ним операционкой. История которой, кстати, тоже весьма интересна, но мной не рассматривается по причине недостаточного знакомства.

Разработка AtheOS была начата Куртом во второй половине 90-х годов. Однако о своём создании он заявил миру весной 2000 года, разместив в открытом доступе её исходники под лицензией GPL (тогда ещё только за номером 2). А в начале 2001 (то есть уже однозначно в XXI веке) года под AtheOS был портирован Apache и сайт проекта <http://www.atheos.cx/> заработал под управлением её же самой. И работал ещё несколько лет после прекращения разработки, без всякого участия автора. Так что всю короткую, но яркую историю AtheOS можно целиком считать принадлежащей к третьему тысячелетию.

AtheOS функционировала на любых Intel-совместимых процессорах, причем с очень эффективной поддержкой мультипроцессорных архитектур. Система написана почти целиком на

Си – ассемблерная часть ядра составляет чуть больше 20 Кбайт. И потому теоретически она повязана с Intel-архитектурой не больше, чем любая иная POSIX-совместимая система.

Одна из отличительных особенностей AtheOS – поддержка в ядре графического интерфейса пользователя, основанного на архитектуре клиент-сервер, но отличного, тем не менее, от оконной системы X, привычной всем пользователям UNIX. Вместе с тем поддерживается и стандартный интерфейс командной строки в лице типичных UNIX’овых Shell’ов (штатно – bash, но и zsh был на эту ОС портирован). Ну и вообще декларируется поддержка, хотя и не полная, всяческих стандартов (типа POSIX).

Как она получалась...

Всё это было прочитано мной в далёком 2001 году. И вызвало желание ознакомиться с системой вживе. Разумеется, первым действием к тому было получение системы с сайта разработчика. Основной её комплект включал:

- образы двух загрузочных дискет;
- образ дискеты с данными, под коими имеется ввиду базовый набор компонентов;
- собственно систему в виде единого тарбалла объемом около 20 Мбайт;
- небольшую, но вполне внятную документацию, посвященную описанию инсталляции системы и параметров загрузки ядра.

Кроме этого, на сайте (в отдельном каталоге) имелся набор дополнительных пакетов (также в виде tgz-архивов), несколько ограниченный, но оригинальный по подбору: средства разработки (gcc, automake и подобные), web-сервер Apache, редактор emacs, основные UNIX-утилиты типа grep, gawk и т.д., включая даже Midnight Commander.

Как устанавливалась...

Для установки системы требовался винчестер со свободным разделом или не размеченным пространством, какой-либо носитель с

файловой системой FAT (раздел диска или, например, Zip) и три трёхдюймовые дискеты. На FAT-носитель помещался базовый файл, на дискеты, посредством rawrite (в DOS/Windows) или dd (в UNIX/Linux), – образы загрузочных дисков и дискеты с данными.

Далее следовало выполнить загрузку с первой дискеты (вторая запрашивалась по ходу дела), после перехода в графический (VGA) режим требовалась дискета с данными. И тогда на экране появлялось цианидно-зелёное рабочее поле с единственным окном терминала, в котором была запущена командная оболочка bash (точно такая же, как в Linux того времени).

Все последующее было не просто, а очень просто. Для начала в bash запускалась программа DiskManager и на пустом пространстве целевого диска выделялся раздел под родную файловую систему afs (AtheOS File System. Разумеется, если не жалко, можно было уничтожить какой-либо из разделов существующих).

Программа создания разделов, как и все в этой системе, работала в графическом режиме (текстовый режим в AtheOS отсутствовал как класс) и была весьма удобной в обращении. Правда, номенклатура накопителей в ней, как это в обычае среди «крутых пацанов», отличалась от любой другой. Иерархия каталогов в AtheOS также значительно отличалась от типичной для большинства UNIX-систем. Но ко всему этому нужно было просто привыкнуть.

После этого на разделе или диске создавалась (командой format) файловая система afs и две точки монтирования – для FAT-устройства с базовым файлом и для afs-раздела для системы собственно. Установка же последней осуществлялась банальной распаковкой (командой tar с соответствующими опциями) базового тарбалла.

И как работала

Теперь оставалось только обеспечить загрузку новообретённой системы. Загрузчиком её являлся самый обычный GRUB. И потому посредством стандартного текстового редактора (в качестве коего выступал jed – к счастью или несчастью, но никакого vi не было и в помине) правился его конфигурационный файл.

Затем система перезагружалась (обязательно с помощью комбинации из трех пальцев, но никак не Reset'ом) с первой дискеты и

при появлении меню GRUB'a переводилась в режим его редактирования. Тут следовало указать новый корневой раздел, после чего сделанные изменения записать в MBR. После этого, вынув дискету, можно было загрузить AtheOS уже нормальным образом. При этом система практически мгновенно переходила в графический режим, и после авторизации перед глазами возникал рабочий стол с пузырьчатыми обоями, в углу которого сиротливо ютились пиктограммы для запуска программ: файлового менеджера, браузера, терминала, утилиты настройки (Prefs) и пары-тройки системных мониторов.

Штатный набор приложений выглядел бедновато, но мог быть расширен за счёт дополнительных пакетов, правда, тоже не очень многочисленных. Это выполнялось в два приема: сначала пакет распаковывался из архива, а потом регистрировался в базе данных специальной утилитой. В отличие от всех тогдашних UNIX-подобных систем, дополнительные пакеты устанавливаются каждый в свой подкаталог каталога /usr, а не раскидывал свои файлы по древу многочисленных bin'ов, lib'ов и прочих man'ов. Не будем обсуждать, хорошо это или плохо – но ныне такой подход практикуется в PC-BSD и некоторых дистрибутивах Linux.

Что еще остается добавить? Утилита конфигурирования Prefs позволяла настроить разрешение экрана и глубину цвета, выбрать экранные шрифты (в качестве системных используются шрифты True Type) и раскладку клавиатуры – они представлены для большинства европейских языков, хотя русского среди них не было, как и кириллических экранных шрифтов, хотя русская locale имелась.

Не знаю, удалось ли мне в своём рассказе передать то чувство лёгкости, быстроты, компактности, аккуратности интерфейса, простоты установки и использования, которое испытывал при общении с AtheOS действующий пользователь Linux или BSD. Но она вызывала именно такие эмоции. Конечно, на тот момент времени её нельзя было рассматривать как полноценную универсальную ОС для практической деятельности. Хотя уже тогда резонные люди утверждали, что применяться как платформа для разработки она могла. А её потенциал как системы для настольного использования просматривался достаточно явно.

Что же касается утверждения автора об отсутствии связи AtheOS с UNIX – можно констатировать: он постарался, чтобы его систему нельзя было бы спутать с Linux или FreeBSD. Однако несомненно, что идеологически он следовал именно пути UNIX, а не, скажем, традициям DOS или Windows. Хотя в те годы AtheOS часто сравнивали с AmigaOS или BeOS.

Увы, потенциал AtheOS так и не был реализован. Разработки Курта прекратились в начале 2002 года, последнее обновление сайта датировалось осенью 2003-го. Хотя, повторяю, сайт был доступен ещё долгое время, пока на нём не появилось сообщение об окончании срока регистрации домена. Ныне сайт символически восстановлен в качестве своего рода мемориала по тому же адресу – правда, дальше первой страницы пройти по нему нельзя.

О причинах прекращения разработки в Сети ходили противоречивые слухи. В частности, писали о том, что Курт увлёкся любительским пилотированием и потерял интерес к AtheOS. А поскольку в ходе активной его разработки он, в отличие от Линуса, не особенно привлекал к нему посторонних разработчиков, проект оказался «бесхозным», и Афина Паллада не пришла ему на помощь.

Так что столь интересный и потенциально многообещающий проект можно было бы считать мертвым. Однако чувство печали – ведь с уходом чего-то хорошего становится как-то грустно – заставляло меня время от времени наведываться на сайт проекта в надежде увидеть там какие-то подвижки. Пока вдруг по наитию не набрал слово «AtheOS» в поисковой строке Google.

Эпилог

И тут в очередной раз обнаружилось, что приключения никогда не кончаются, по крайней мере – в мире Open Source (надеюсь, вы не забыли, что AtheOS распространялась по лицензии GPL?). Так вот, угасание проекта вызвало, видимо обиду не только у меня: на одном только SourceForge я обнаружил тогда пять разработок, выводящих свою генеалогию из системы Курта.

Правда, по настоящему действующей была только одна, носившая имя Syllable. Но зато это было настоящее развитие, продолжающееся и

поныне, хотя далеко не теми темпами, как при Курте. Так что хочется верить – история AtheOS еще не окончена.

Я же пока сделаю выводы из истории прошедшей. главный из них таков: даже в наш век, когда одни разработчики Open Source пользуются поддержкой крупных фирм, другие просто работают в них за зарплату, а третьи сами образуют коммерческие фирмы, четвёртые создают большие и устоявшиеся сообщества, а пятые вообще уповают на поддержку родного государства, разработка новой, оригинальной и эффективной ОС силами индивидуала-энтузиаста оказывается возможной. Причём разработка в короткий срок – в отличие от развиваемых сообществом долгостроя, на которые не будем указывать пальцем.

Конечно, и это второй урок нашей истории – до определённого предела. После которого требуется привлечение соратников и единомышленников. Курту Скавену это не удалось – или это не входило в его планы. Но если проект интересен не только его автору, соратники и единомышленники появляются сами, даже если автор потерял всякий интерес к своему созданию. И это – третий урок, который мы в очередной раз извлекаем из истории Open Source.

Так что AtheOS, как это и сказано в заглавии статьи, можно считать предпоследним взлётом индивидуального ОСетворчества. Почему предпоследним? Не только потому, что было бы обидно, если бы он оказался последним. Но и потому, что в двух ближайших статьях цикла мы увидим более иные примеры его. Причём примеры более оптимистичные.

Героинями следующих двух статей цикла также будут ОС третьего тысячелетия – DragonFlyBSD и MINIX3. Хотя первая из них рассматривается обычно как форк FreeBSD, а вторая – как развитие той самой «игрушечной» MINIX, которая вдохновила Линуса Торвальдса на создание своей терминальной программы, обе они нынче являют собой вполне самостоятельные системы. И история их в этом качестве целиком принадлежит уже нашему времени. А судьба их оказалась более счастливой, нежели у героини сегодняшнего рассказа.

Глава седьмая. «Драгонада» Мэтта Диллона

Как было отмечено в прошлой главе, новые операционные системы возникают на наших глазах возникает не каждый день. И, главное, не каждая из возникших систем активно развивается, а не забрасывается по прошествии недолгого времени или входит в состояние стагнации. В этой главе я расскажу о счастливой судьбе такой системы, пусть не завоевавшей мир, но вполне благополучной.

Как догадались многие читатели, речь пойдёт о DragonFlyBSD, ответвлении FreeBSD, для которой отсчёт времени пошёл в июне 2003, и я за которой я следил с самого начала. И которую начал использовать с того момента, как она к тому стала пригодна.

«Железная» ретроспектива

Начало нулевых годов стало одним из переломных моментов в развитии информационной сферы. Вычислительные мощности обычных настольных персоналок достигли такой страшной силы, что сполна, а то и с лихвой, перекрывали потребности подавляющего большинства пользователей. По крайней мере тех, чьим основным занятием было создание контента – до начала эры тотального его потребления оставалось ещё несколько лет.

И тут обнаружилось, что, впервые с момента появления персональных компьютеров, софтверный сектор индустрии не смог выполнить свое сакральное предназначение – встряхнуть пользовательские кошельки на перманентный апгрейд своих машин, что обеспечивало бы финансовые источники для дальнейшего развития сектора хาร์ดверного. И развитие базового компьютерного «железа» – процессоров и чипсетов, определяющих лицо платформы, – если и не прекратилось, то резко затормозилось.

Конечно, время от времени производители продолжали бодро рапортовать об очередном повышении тактовых частот процессоров, увеличении их кэша до совершенно невообразимых, некогда вполне достаточных для основной памяти, объемов, включении дополнительных наборов инструкций под замысловатыми названиями,

встраивании в чипсеты поддержки всего, чего можно, и даже того, чего, как недавно казалось, нельзя – например, 3D-графики. Однако накал страстей вокруг всего этого был не тот, что в во второй половине 90-х. А отчёты о тестировании процессоров и материнских плат стали напоминать просмотр фотофиниша на стометровке.

главное же, что, подобно рекордам в стометровке, достижения «камнестроителей» все меньше волновали широкие пользовательские массы. Ведь от медведя не убежит и олимпийский чемпион, а успеть за пивом в магазин перед его закрытием способен любой мало-мальски тренированный человек. Так и с компьютерами: настольно-пользовательские задачи в большинстве случаев стали решаться средствами любого подручного «железа», купленного за пределами лавки древностей. А задачи «тяжелые» по прежнему требовали более чем всех ресурсов персоналок, и потому решались обычно не на них (по крайней мере, разумными людьми).

Правда, стимуляции пользовательского интереса для производители в первые нулевые предложили два архитектурных решения, продаваемые как революционные. Первым (по времени) была архитектура Pentium 4. Она обеспечивала рост тактовой частоты процессоров, поражающий воображение пользователя, тянущегося к бумажнику. И к тому же перспективы роста казались тогда безграничными.

Правда, на счет безграничности жизнь довольно скоро внесла свои коррективы. И разговоры о том, что эта технология позволит играючи достичь тактовой частоты в 10 гигагерц, как-то стихли сами собой. Однако чисто случайно оказалось, что гигагерцев, в отличие пряников, как раз и хватило на всех (кроме тех, которым, как уже говорилось, их не будет хватать всегда).

Вторая, столь же революционная, новинка – 64-разрядные вычисления. Вспомним терью статью цикла: каким прорывом в светлое будущее были 32-битные процессоры для PC, те самые первые «трешки», которые сделали возможным портирование на эту архитектуру UNIX и, в конечном счёте, появление Linux. Повторилась ли история на новом витке диалектической спирали?

Увы, отрицательный ответ был получен практически мгновенно. Потому что в те далекие уже годы аппаратура PC едва поспевала за софтом – 32-битные ОС разменивали уже второй десяток лет своего

существования, и приложений, использующих 32 разряда на полную катушку, было вдоволь. В описываемый же момент их в пользовательском сегменте просто не было по одной простой причине – не востребованности. К слову сказать, почти нет их и по сей день. Ибо единственная ниша пользовательских приложений, где 64 бита хоть как-то задействованы – параноидальная криптография.

Так что усилия «камнестроителей» пропадали бы втуне. Если бы ещё не одно новшество, о котором я сознательно не упоминал ранее – Hyper Threading, то есть виртуальная мультипроцессорность. Каковая в некоторых (правда, весьма редких) задачах давала вполне даже реальный прирост производительности. Правда, он мало значил для пользователей, работающих преимущественно интерактивно. Но весьма способствовал производительности труда применителей – тех, кто, в силу врожденной лени отдавал предпочтение всякого рода скриптам, пакетным заданиям и прочим средствам автоматизации.

Однако Hyper Threading был не более чем суррогатом истинной мультипроцессорности. Своего рода мультипроцессорность для бедных, но гордых. И потому, сказавши **А**, производитель процессоров неизбежно должны открыть рот для произнесения **Б**. То есть переходить к собственно мультипроцессорным конфигурациям в пользовательском сегменте.

Разговоры о двухпроцессорных пользовательских десктопах возникали неоднократно. Кое-кому из читателей памятно, как дешевенькие Celeron'ы первого разлива можно было вставлять в относительно не очень дорогие двухпроцессорные материнские платы, получая таким образом нечто вроде «народного суперкомпьютера».

Правда, первая волна «народной мультипроцессорности» была очень быстро пресечена производителем. Однако идея мультипроцессорности для народа продолжала витать в воздухе – никаким иным способом создать впечатление прогресса уже не удавалось (к слову сказать – не удаётся и по сей день). И первый шаг в этом направлении сделала, насколько мне известно, IBM со своим процессором Power4 – в то время абсолютным рекордсменом по «чистому» (то есть тестовому) быстродействию. В том числе и благодаря тому, что имели варианты с двумя и более процессорными ядрами в едином корпусе.

Сами по себе процессоры Power4 (как и пришедшие им на смену Power5) ориентировались на индустриальный сектор. Однако на базе их были созданы процессоры G5 – сердце тогдашних Mac’ов, имевших, в том числе, и двухъядерный вариант.

Правда, пользователям PC’шек (а мы говорим в основном о них) от этого было бы ни холодно, ни жарко. Однако здесь «камнестроители» не заставили себя ждать: и AMD, и Intel очень быстро анонсировали, а затем и воплотили в реальность, свои двухъядерные решения, стоимость которых вполне вписывалась в рамки «суперкомпьютера для народа». По крайней мере, в лице лучших его представителей.

Так что пользователи оказались перед выбором между традиционными одноядерными процессорами с большей тактовой частотой или процессорами двухъядерным – с меньшей (если оставаться в рамках одного бюджета). Как я уже говорил, рост тактовых частот упёрся в потолок целесообразности: сколь бы велик он ни был (а тут имелся ещё и потолок технологический), адекватного прироста производительности он уже за собой не влёт. Но могли ли пользователи рассчитывать на хоть какой-то выигрыш в производительности от многоядерности?

Софтверная перспектива

Исходя из общих соображений было очевидно, что ожидать двукратного увеличения быстродействия от самого факта удвоения числа процессоров (или их ядер) не приходится. Во-первых, на «железном» уровне два и более процессоров будут совместно использовать какие-то общие ресурсы компьютера – память, кэши, шины и так далее.

Во-вторых, неизбежны были потери быстродействия за счет «накладных расходов» – согласования операций, выполняемых на разных процессорах.

В-третьих, системные и прикладные задачи, выполняемые на многопроцессорной машине, должны допускать их распараллеливание, иначе любое увеличение количества «числодробителей» доставит мало радости пользователю.

И, наконец, в-четвертых, эффективность многопроцессорных конфигураций в значительной мере определялась стилем выполнения пользовательских задач. Очевидно, что преимущественно интерактивные методы работы от удвоения «камней» выиграют весьма мало – в любом случае тут узким местом окажется пресловутый человеческий фактор.

Решение проблем многозадачности на «железном» уровне было задачей производителей аппаратуры. А вот минимизация же «накладных расходов» и распараллеливание задач относились уже к сфере разработчиков софта, в первую очередь – системного. Хотя в последнем случае роль создателей программ прикладных ничуть не меньше. Ну а эффективное использование достижений тех и других – это уже вахта пользователей.

И нужно сказать, что пользователи UNIX-подобных операционных систем, в силу самой специфики их работы и укоренившихся привычек, казались подготовленными к многозадачности лучше других. И были способны получить от нее больший выигрыш.

Ведь что происходит на типичной пользовательской UNIX? На ней постоянно что-то компилируется, архивируется и разархивируется, кодируется и декодируется, бэкапится и восстанавливается. И все это – параллельно, и, в большей или меньшей степени, без интерактивного участия применителя. Озаботившегося, разумеется, заблаговременно, скриптами для запуска своих задач, выводом полученных данных в логи и прочие файлы, и так далее – за интерактивным режимом остается только просмотр результатов. И, конечно же, их обдумывание.

Так что вырисовывалась заманчивая картина – все это изобилие параллельно работающих задач выполнять действительно параллельно, раскидав по разным процессорам. Дело оставалось за малым – воплотить её в кодах.

Изначально создатели UNIX (и ранних его клонов) ни о какой многопроцессорности не помышляли. И один из краеугольных камней его идеологии – концепция монолитных процессов, выполняемых на одном процессоре квазипараллельно, за счет квантования времени, – казалось бы, препятствует реализации распараллеливания задач по разным «камням».

Тем не менее, когда многопроцессорные серверы и рабочие станции стали реальностью в индустриальном секторе, в дополнение концепции процесса была создана и концепция т.н. нитей, или потоков (threads). Это – части процесса, выполняемые параллельно и почти независимо друг от друга (в том числе и на отдельных процессорах), разделяющие, тем не менее, ресурсы составленного из них процесса. То есть собственного контекста, в том числе и отдельного пространства памяти, они не имеют, почему носят ещё и имя легковесных процессов (light weight process) – обычные UNIX-процессы в этом случае можно называть «тяжелыми».

Само по себе понятие нитей возникло задолго до UNIX – чуть ли не со времен Очакова и ламповой электроники. И уже тогда были выявлены существенные недостатки этой концепции. Однако за истекшие годы ничего лучшего для поддержки мультипроцессорности придумано не было.

Как уже говорилось, проблема мультипроцессорности встала в первую очередь в индустриальном секторе. Где по ряду причин (в том числе и исторических) традиционно преобладали проприетарные представители UNIX-семейства. И разработчики последних доблестно эту проблему разрешили. Можно спорить, где она была решена лучше, где – не так хорошо, однако общепризнанно: масштабируемость многие годы был главной отличительной чертой (и главным козырем) и AIX от IBM, и Solaris от Sun, и прочих их братьев-конкурентов.

Свободные UNIX-совместимые ОС, как мы помним по первой статье цикла, разрабатывались преимущественно или в университетско-академической среде, или просто энтузиастами-любителями, как правило, на подручном оборудовании. Среди которого многопроцессорные суперкомпьютеры встречались не так уж и часто (солнце народной мультипроцессорности ещё не показало из-за горизонта своих первых лучей). И потому долгое время поддержка многопроцессорности была слабой стороной и Linux, и BSD-систем (по крайней мере, для платформы Intel и совместимых).

Движение свободных операционок в корпоративный сектор, в первую очередь в качестве серверов разного рода, поставило перед ними задачи многопроцессорной поддержки и масштабируемости. И задачи эти постепенно решались: в том или ином виде многопроцессорные конфигурации давно поддерживаются ядром

Linux и FreeBSD, затем такая поддержка появилась в NetBSD и OpenBSD. Тем не менее, ни одна из этих ОС не дотягивала ещё до масштабируемости проприетарных UNIX'ов.

Правда, в мире свободного софта испокон веков развивалось другое течение, косвенно связанное с многопроцессорностью – так называемые микроядерные ОС. Идея их – в том, чтобы максимально возможную часть обязанностей ядра (взаимодействие с устройствами, файловыми системами и т.д.) вынести в пользовательское пространство памяти, оставив за ядром только коммуникационные функции. Теоретически рассуждая, это должно было бы обеспечить упрощение устройства системы, легкость её портирования на новые архитектуры (в том числе и многопроцессорные), а также возможности масштабирования.

Из микроядерных решений наибольшее признание получило ядро Mach, разработанное в Университете Карнеги-Меллона, а затем развивавшееся в Университете Юты. Оно легло в основу ряда проектов разработки свободных ОС – самым известным из них долгое время был Hurd, разработка которого затем была переведена на другое микроядро – L4. Что, однако, не приблизило проект к состоянию, пригодному для применения. Однако существовали и другие попытки создания свободных микроядерных ОС – проекты xMach и Yamit. И оба они своё развитие прекратили.

Таким образом, можно констатировать, что судьба свободных микроядерных проектов была (и остаётся) печальной. Что, помимо их не востребоваемости, объясняется ещё и технологическими причинами: судя по всему, разработчики не смогли обеспечить приемлемую производительность своих систем: ведь упрощение устройства ядра влечёт за собой усложнение межпроцессных коммуникаций.

Как ни странно, удачные реализации микроядерной архитектуры имели место быть в проприетарном секторе: на ранних версиях Mach основывалась знаменитая система NEXTStep, видевшаяся лет 15 назад платформой фантастического будущего. А предпоследняя, 3-я, версия Mach легла, вместе с системными службами FreeBSD, в фундамент современной MacOS X.

Таков был исторический фон, на котором развернулись описанные далее события.

У истоков новой системы

Итак, где-то в середине июня 2003 г. Мэтт Диллон (Matt Dillon), вместе с группой товарищей сообщил о начале работы над новой ОС BSD-семейства – DragonFlyBSD. Возник сайт проекта – <http://www.dragonflybsd.org> и репозиторий её исходников, созданный 16 июня 2003 года – этот день можно считать именинами системы. Репозиторий этот основывался на кодовой базе FreeBSD 4-й ветки, имевшей статус стабильной (хотя в то время уже всю развивалась ветка 5-я, вбирающая в себя все инновации BSD-мира).

Новая система получила и собственный тотем – стрекозу, что должно символизировать легкость и быстроту её. Кстати, разработчики не гнушаются и сокращенных названий своей системы – DragonFly и даже DFBSD, к которым, вслед за ними, время от времени, будем прибегать и мы.

Может возникнуть (и многократно возникает) вопрос: для чего нужна ещё одна BSD-система? Разве не вдоволь насмотрелись мы на изобилие Linux-дистрибутивов, чтобы и BSD-системам желать той же участи?

Вопрос этот, конечно, носит сугубо риторический характер. Ведь если новые операционные системы создаются – значит, это кому-то нужно. И каждая такая система (если она, конечно, действительно нова и оригинальна) привносит в наш мир что-то свое, увеличивая, тем самым, сложность его и разнообразие.

А в оригинальности DragonFlyBSD отказать невозможно. Ибо, при практически полном внешнем сходстве с прототипом (FreeBSD 4.X), «внутри» у нее всё было другое: управление памятью и процессами, представление о драйверах устройств и виртуальной файловой системе, вплоть до нового типа файлов – вариантных символических ссылок (varsims).

В основу DragonFly была положена модель легковесных нитей ядра (LWKT – Light Weight Kernel Threads), что само по себе и не ново. Новым стал механизм планирования нитей – вместо единого планировщика (sheduler) их было введено несколько, по числу процессоров. Нити привязаны к своим процессорам изначально, однако допускается передача выполнения нити с одного процессора на

другой при некоторых особых условиях. Данные отдельных нитей могут быть кэшированы независимо для каждого процессора.

Подобно системам с микроядерной архитектурой, в DragonFly максимум функций ядра вынесен из его пространства памяти в пользовательское пространство (userland). В первую голову это относится к драйверам устройств – таким образом достигается рост как производительности, так и надежности системы, резко уменьшая вероятность её «падения» под воздействием неправильно работающего драйвера.

Это повлекло за собой отказ от традиционного для UNIX механизма системных вызовов (каковой лишь эмулируется в целях совместимости). Его место занял механизм сообщений (messages) и их очередей, т.н. портов (ports), подобный применяющемуся в микроядре March, упоминавшемся выше.

При этом DragonFly не является микроядерной ОС – базовые функции по прежнему возлагаются на ядро (и размещаются в его пространстве памяти). Однако почти всё прочее может быть безболезненно собрано в качестве модулей юзерланда.

Таким образом, можно видеть, что основные инновации DragonFly ориентированы на работу в многопроцессорных системах. А вопрос о том, есть ли что делать простому пользователю на многопроцессорном, с позволения сказать, компьютере, мы рассмотрели уже в ходе обзора хардверной ретроспективы.

Далее, важно, что если матушка DragonFly, FreeBSD, изначально предназначенная только для архитектуры i386, все более эволюционировала в сторону кроссплатформенности (в 5-й ветке к поддержке древней Alpha был добавлен Sparc, а затем и PowerPC), то наша «стрекоза» возвращается на исходные рубежи. И единственной поддерживаемой архитектурой в ней является Intel-совместимая – на тот момент только 32-битная (64-разрядный вариант долго находился в состоянии разработки).

Такое ограничение в плане поддерживаемого «железа» может показаться отступлением от истинного UNIX Way. Однако на момент выхода DFBSD сбылось мрачное пророчество, высказанное почти двадцать лет назад в одном компьютерном журнале:

Через десять лет все платформы, кроме IBM PC, уйдут в небытие

И все остальные архитектуры в качестве настольных платформ полностью утратили актуальность. Разработчики DragonFly считались с этой реальностью: в их тогдашних планах переноса на другие архитектуры не было (нет его и сейчас). Что компенсировалось возможностью оптимизации под платформу, единственно значимую практически. Это дало свои плоды – по визуальному быстродействию в настольных условиях DragonFly со дня своего зарождения существенно опережала FreeBSD как 5-й, так и 4-й ветки.

Наконец, в DragonFly на уровне ядра поддерживался механизм, напоминающий prelinking (предварительное связывание с разделяемыми библиотеками) – насколько мне известно, особенность почти уникальная. И обещавшая значительный прирост скорости загрузки (а возможно, и быстроты исполнения) сложных программ, связываемых со множеством библиотек.

Всё сказанное выше было технологическим обоснованием для того, чтобы отнестись к DragonFly не просто как к ещё одному BSD-клону. Но это подкреплялось и субъективным фактором – личностью организатора проекта.

К моменту начала работы над DragonFly Мэтт Диллон был широко известен (в узких кругах) благодаря трем разработкам: Си-компилятору для платформы Amiga (именно из этой ОС пришла в DragonFly идея «ядерного прелинкинга»), утилите dcron и, главное, системе управления виртуальной памятью во FreeBSD. Не то чтобы он был единственным автором последней, однако вклад его в эту тему был одним из определяющих современный облик FreeBSD. Да и к аналогичной подсистеме ядра Linux он приложил руку.

Что немаловажно, в специальной статье (присутствующей в официальной документации FreeBSD) Мэтт сумел описать архитектуру виртуальной памяти языком, понятным для широких масс трудящихся. Очень рекомендую к прочтению – во введении к ней высказано немало интересных мыслей общего характера. Тем более, что она доступна и в русском переводе. А пока позволю себе вторично процитировать её фрагмент:

Самой большой ошибкой, которую может допустить программист, является игнорирование истории.

И дальнейшая история показала, что в DragonFly ошибки истории прошедшей были учтены.

Некоторое время проект развивался как бы закулисно. Конечно, все желающие ознакомиться с прототипом системы могли свободно получить её исходники с сайта проекта через CVS и развлекаться с ними в свое удовольствие (нужно ли говорить, что DragonFly распространялась и распространяется на условиях лицензии BSD?). Однако в виде, пригодном для установки простыми смертными, она не существовала.

Так продолжалось до мая 2004 года, когда один за другим начали появляться iso-образы CD бета-версий DragonFly. Они не имели ещё инсталлятора: следовало, руководствуясь документацией (вполне, впрочем, ясной, хоть и англоязычной), вручную разметить диск, создать файловые системы, перенести на них с дистрибутивного CD необходимые каталоги и произвести ещё кое-какие манипуляции (типа создания файлов устройств и настройки стартовых сервисов). Задача была не то чтобы сверхъестественно сложной – но и не вполне тривиальной.

А затем... Затем, в июне 2004 гjlф, появился пре-релиз DragonFly, точнее, DragonFlyBSD 1.0RC1. От своих бета-предшественников он отличался тем, что уже имел инсталлятор – BSD Installer, разработанный в рамках самостоятельного проекта как универсальный установщик для любых BSD-систем. И впервые опробованный именно на DragonFly.

Надо заметить, что уже в те далекие времена (в масштабах истории DragonFly) установка этой ОС проходила без малейших осложнений (в том числе и на «железо», на которое FreeBSD устанавливалась с трудом). Однако к использованию система была пригодной лишь условно. Ибо, кроме базиса (соответствующего FreeBSD Distributions), не содержала почти ничего – ни Иксов, ни прекомпилированных пакетов (за исключением нескольких консольных утилит типа cvsup и cdrtools), ни собственной системы пакетного менеджмента.

Нужно сказать, что пре-релизная стадия для DragonFly оказалась очень короткой: уже в 11 июля 2004 года было объявлено о выходе релиза – DragonFlyBSD 1.0-RELEASE. Правда, и он просуществовал недолго: как это нередко бывает, в него вкралось несколько мелких, но весьма неприятных ошибок, которые были выявлены мгновенно и столь же быстро исправлены в корректирующем релизе 1.0A.

Начиная с этого момента, DragonFly можно было считать более-менее пригодной к использованию. Сам по себе дистрибутив по прежнему не включал ни пакетов, выходящих за рамки базовой системы, ни системы портов. Однако с самого момента выхода релиза прекомпилированные пакеты для DragonFly можно было найти на двух самостоятельных сайтах.

При этом жизнь не стояла на месте, и дальнейшая работа над системой не прекращалась: с интервалом в 3-5 дней на ftp-сервере проекта и его зеркалах появлялись текущие снапшоты, одни из которых позиционировались как рабочие, другие же – как экспериментальные.

Двигалось дело и с портированием сторонних приложений. Первоначально оно осуществлялось с помощью адаптированной системы портов FreeBSD. Однако позднее разработчики пошли другим путем: прикрутили в DragonFly pkgsrc – кроссплатформенную портообразную систему, заимствованную из проекта NetBSD. И таким образом в распоряжении пользователя новой ОС сразу оказалось все изобилие открытого и бесплатного софта, портированного на эту платформу – а надо отметить, что на NetBSD он портирован в очень значительной своей части.

Первоначально предполагалось, оба варианта – не более, чем временные паллиативы, и в рамках проекта DragonFly будет разработана собственная пакетирования – насколько можно судить по отрывочным указаниям того времени, похожая на систему PBI, реализованную позднее в PC-BSD. Однако затем эта идея была оставлена, и до недавнего времени pkgsrc была единственной системой управления пакетами.

Такова была ранняя, короткая, но насыщенная событиями история операционной системы DragonFlyBSD на момент годовщины её первого релиза. В последующие годы в ней появилось немало новшеств, как косметических (например, настоящая графическая

консоль – аналог режима frame buffer в Linux), так и весьма кардинальных. Среди последних необходимо отметить:

- собственную реализацию виртуальной файловой системы, обеспечивающей доступ к практически всем файловым системам UNIX-подобных ОС

- оригинальную файловую систему Hammer – точнее, интегрированная система размещения данных, по своим принципам сходная с возникший чуть раньше ZFS и несколько более «юной» btrfs.

Статическое создание файлов устройств сменилось динамической файловой системой устройств devfs. Со временем DragonFly была портирована на архитектуру x86_64. И, наконец, для управления пакетами она снова вернулась к собственной модификации портов FreeBSD – системе dports. А сразу же по появлении системы пакетного менеджмента pkg-ng внедрила её у себя – чуть ли не раньше, чем возник первый официальный репозиторий для родительской операционки (то есть всё той же FreeBSD).

Однако предпосылки всего этого были заложены в первый год существования системы. Ныне её история не окончена – но выходит за рамки рассматриваемого периода.

Глава восьмая. MINIX 3: вторая жизнь

Вступление

Кто же из линуксоидов не знает старика MINIX'a? Эта миниатюрная учебно-показательная UNIX-подобная ОС была создана четверть века назад профессором Эндрю Таненбаумом. Предназначалась она для вразумления студентов и приобщения их к идеалам UNIX на самой демократической платформе всех времён и народов – на IBM PC-совместимых компьютерах. Она уже фигурировала в нашей истории – в разделе о зарождении Linux. Ибо именно MINIX вразумил Линуса Торвальдса настолько, что он занялся сочинением своей терминальной программы, которой суждено было превратиться в Linux.

В том же разделе была описана и история ОС MINIX, сначала 1, а потом и 2, которую можно рассматривать как предысторию героини сегодняшнего рассказа. И которая завершилась анонсом MINIX 3, состоявшимся 24 октября 2005 года. Насколько повторяясь, подчеркну: это была не просто следующий релиз прежних MINIX'ов, а совершенно новая операционная система, и цифра «3» здесь – не номер версии, а часть её имени собственного. Таненбаум неоднократно подчеркивал, что сходство ее с предшественниками – лишь в первом компоненте названия, а различие между MINIX 1/2 и MINIX 3 не меньше, чем между Windows 3.1 и Windows XP.

Отныне MINIX 3 более не рассматривалась как учебно-показательная разработка, а позиционировалась как всамделишная операционная система общего назначения, предназначенная, в перспективе, для широкого класса устройств, в том числе и встраиваемых. Это символизировалось и сменой правового статуса системы: отныне она распространялась под лицензией BSD.

Официальный сайт проекта – minix3.org. Интересно, что буквально через несколько месяцев после анонса MINIX 3, 1 февраля 2006 года, Романом Игнатовым был создан русскоязычный ресурс по этой системе – minix3.ru, который успешно развивается и по сей день.

Отличие новой системы заключалось ещё и в модели разработки. MINIX 1 и MINIX 2 были фактически личными творениями Эндрю Таненбаума, а все дополнения к ней, вроде ставшего знаменитым патча Брюса Эванса (именно с его дополнением использовал MINIX Линус Торвальдс для работы над своей будущей операционкой), носили, в силу условий распространения, сугубо неофициальный характер.

К разработке же MINIX 3 Таненбаум с самого начала привлёк широкий круг участников – от своего соавтора по второму изданию учебника Альберта Вудхалла до студентов, аспирантов и сотрудников Университета Врије, а также волонтеров. Состав участников проекта, по понятным причинам, был весьма текучим, и перечислить их всех поимённо не представляется возможным. Однако нельзя не отметить среди них наших соотечественников – упомянутого выше Романа Игнатова и Евгения Иванова.

О микроядрах

В MINIX 3 воплотилось представление Таненбаума и его соратников о том, какова должна быть современная операционная система. Однако, чтобы говорить нѐм, надо опять обратиться к истории – на этот раз к истории микроядерных операционок (краем этот вопрос был затронут в предыдущей главе).

Каждый школьник-линуксоид знает, что MINIX с самого своего рождения представляла собой микроядерную ОС. А вот какая он, эта микроядерность?

Для начала зададимся вопросом, а что же такое ядро вообще? Традиционно ядро определяется как программа, обеспечивающая взаимодействие всего остального системного и прикладного софта с аппаратной частью компьютера, и распределение его ресурсов между приложениями. В соответствии с этим ядро функционирует в отдельной области памяти, которая так и называется – пространством ядра. Память же, отводимая под все остальные программы, именуется пользовательским пространством; протекающим там процессам доступ в пространство ядра закрыт, и как-либо влиять на ядро, в том числе и негативно (вследствие ошибок в программе или злого умысла), они не могут. Но все процессы внутри пространства ядра

взаимодействуют друг с другом, и ошибка в одном из компонентов может повлечь за собой тяжкие последствия, вплоть до краха системы.

Понятие аппаратуры компьютера, однако, оказывается двояким. С одной стороны, это те узлы, без которых машина в принципе не может функционировать – процессор и память. Эта сторона взаимодействия охватывается такими службами ядра, как обработчик прерываний, средства запуска и останова процессов, планировщик задач, механизм межпроцессного взаимодействия.

С другой стороны, в понятие аппаратной части включаются и внешние, по отношению к системе процессор/память, устройства, от видеокарт и жестких дисков до принтеров, сканеров, сетевых адаптеров и многого, многого другого.

Более того, у этой аппаратной медали есть еще и третья сторона – сервисы доступа к файловым системам, сетевым протоколам и так далее. Они представляют собой связующее звено между собственно внешними устройствами и пользовательскими программами. Например, сервисы доступа к файловым системам обеспечивают возможность взаимодействия между дисковыми устройствами, несущими файловые системы, и обращающимися к ним приложениями.

Ядра, обеспечивающие все три рассмотренные функции, именуются монолитными. И они вполне успешно функционировали до тех пор, пока внешних устройств и сервисов было мало. Однако со временем количество и тех, и других стало расти, как снежный ком. Вспомним, сколько на наших глазах появилось только критически важных устройств, таких, как новые дисковые интерфейсы или файловые системы.

В результате ядра стали катастрофически увеличиваться в размерах. Что влекло за собой а) непроизводительные расходы ресурсов, в первую очередь памяти, и б) рост числа ошибок, обусловленный огромными объемами «ядрёного» кода, недоступного восприятию человека.

С первой болезнью научились бороться посредством модульного подхода: многие части ядра, обеспечивающие работу отдельных внешних устройств (т.н. драйверы устройств) или ядерных сервисов (в Linux их подчас также называют драйверами, например, драйвер файловой системы «имя рек»), могут не встраиваться жестко в

исполняемый файл – образ ядра, а подключаться к нему в виде внешних модулей. И ныне ядра всех активно развивающихся UNIX-подобных систем, таких, как Linux или BSD, являются модульно-монолитными (подчас их для краткости называют и просто модульными).

Внешние модули могут как загружаться в память при старте системе, так и подгружаться в работающей системе, по необходимости. Нередко они обладают и способностью выгружаться, когда эта необходимость пропадёт. Однако в любом случае модули функционируют в пространстве ядра, так что их введение проблемы общей устойчивости не решает: криво написанный драйвер устройства все равно сохраняет способность обрушить систему. А поскольку, как уже говорилось, сложность кода ядра растёт, возрастает и вероятность появления ошибок, в том числе и критичных для работы системы.

Один из «косметических» вариантов решения проблемы был предложен Мэттом Диллоном и реализован им в системе DragonFlyBSD, о которой рассказывалось в предыдущей главе. Прошедшие с её появления годы доказали жизнеспособность этой реализации. Однако проблемы «большого» монолитного ядра это не снимало.

Для кардинального решения проблемы устойчивости ядра и были придуманы микроядра. Идея их – в том, чтобы, по словам Таненбаума, «вынести ядро за пределы ядра». То есть оставить в ядре только средства управления базовой аппаратурой, а драйверы устройств и сервисы выделить в отдельные программы, функционирующие в пользовательском пространстве памяти. При необходимости они обращаются к функциям ядра через специальные процедуры, но влиять на него каким-либо образом не могут. Такой подход приводит к повышению надёжности системы, но снижает производительность, поскольку требует дополнительных накладных расходов.

Собственно, идея микроядра появилась очень давно, чуть ли не одновременно с самыми первыми UNIX'ами. Однако долгое время производительность машин не позволяла эффективно использовать микроядра в составе практически применяемых систем. Тем не менее, различных их реализаций было создано очень много. Время от времени то или иное микроядро пропагандировалось как основа

операционной системы будущего, но удачных реализаций законченных микроядерных систем оказалось довольно мало.

Среди удачных решений на базе микроядра наибольшей известностью пользуется QNX. Правда, эта система лишь недавно начала то приоткрывать свой код, то закрывать его обратно, и потому об её устройстве известно немного. Да и как ОС общего назначения она никогда не позиционировалась.

Из свободных микроядерных реализаций наибольшую известность приобрело микроядро Mach, разрабатывавшееся вплоть до второй половины 90-х годов университетами – сначала Карнеги-Меллона, а затем штата Юта. Различные его версии в разное время составляли основу законченных систем, как свободных – BSD Mach (она же xMach) или Yamitt, так и проприетарных – NeXT и продолжившей её дело MacOS X. Все они имели в своем составе микроядро Mach, поверх которого запускалась драйверно-сервисная часть от ядра BSD, собранная в виде отдельного модуля.

Впрочем, из всех перечисленных систем только BSD Mach можно назвать по настоящему микроядерной, так как у него BSD-окружение ядра функционировало в пользовательском пространстве. И у NeXT, и у MacOS X BSD-окружение запускалось в том же пространстве ядра, так что микроядерными их можно назвать только по имени. А Yamitt в том виде, в каком я её наблюдал, просто не запускалась вообще.

Наконец, как мы видели, начиная с 1987 года и по настоящее время существует MINIX 1/2, основанная на собственной реализации микроядра. Каковую, в рамках очерченных для неё задач, можно считать удачной. А вот будет ли сопутствовать удача её потомку – микроядру нашей героини, MINIX 3, рассудит история.

Собственно о MINIX 3

Для начала заметим, что Таненбаум, по его же словам, не ставил самоцелью создание именно микроядерной ОС: задачей его команды было просто построение системы надежной и безопасной. Другое дело, что решение этой задачи им всегда виделось именно в микроядерной архитектуре. Почему он и предложил таковое, причем в виде, кардинально отличающемся от всех предлагавшихся ранее реализаций.

Отличие первое – микроядро MINIX 3 самое микроядреное микроядро в мире. Из него вычищено все, кроме перечисленных ранее компонентов, таких, как обработчик прерываний, средства запуска и останова процессов, планировщик задач, механизм межпроцессного взаимодействия; правда, почему-то в ядро включён и один из сервисов – сервис часов. В результате все это хозяйство укладывается менее чем в 4000 строк кода – и только оно исполняется в пространстве ядра.

Второе отличие – драйверная и сервисная части, вычлененные из ядра, разделены между собой. В результате образуется знаменитая четырехслойная модель – метафора, в основании которой лежит ядро, надстраиваемое «драйверным слоем», которое, в свою очередь, перекрывается «сервисным слоем» и венчается «слоем пользовательских программ».

Кроме того – и это третье отличие, – каждый драйвер и каждый сервис представляет собой отдельный процесс в пользовательском пространстве, аналогично обычным пользовательским приложениям. В результате ни один драйвер и ни один сервис, как бы криво они не были написаны, не в состоянии обрушить всю систему – точно также, как в любом UNIX'e это (почти) никогда не могут сделать обычные пользовательские приложения. Не могут повлиять они и на соседние процессы, так как напрямую взаимодействовать они не могут, а вынуждены при необходимости обращаться к ядру.

Наконец, четвертое, и, пожалуй, главное: сервер реинкарнаций. Это процесс, выступающий родительским по отношению к процессам всех драйверов и сервисов. Которые он запускает при старте, а в дальнейшем отслеживает состояние. Если процесс какого-либо драйвера или сервиса в силу неких причин самопроизвольно «умирает», он запускает его вновь. Если один из драйверов или сервисов начинает вести себя «нехорошо», сервер реинкарнаций в состоянии убить соответствующий ему процесс и тут же запустить его заново, обеспечивая, таким образом прозрачное для пользователя самовосстановление системы при отказе почти любого драйвера устройства или системной службы.

Очевидно, что такая, достаточно сложная, схема взаимодействия драйверов и системных служб не может не привести к некоторой потере производительности по сравнению с обычными системами, где драйверы и сервисы сосуществуют в едином пространстве памяти, вне

зависимости от того, пользовательском или «ядерном» (а подчас еще и вместе с ядром). То есть – неизбежно должны вызвать снижение быстродействия системы. В каких масштабах? Исследования команды Таненбаума дают ответ на этот вопрос, и к нему я еще вернусь. Но сначала сделаю маленькое отступление на тему, что такое быстродействие операционной системы вообще и с чем его едят, то есть – чем меряют.

Когда обсуждается проблема быстродействия любых ОС, первым делом обычно вспоминают о скорости загрузки. Вероятно, потому, что ее проще всего измерить: достаточно посидеть с секундомером перед несколькими машинами с разными операционками или дистрибутивами, чтобы потом уверенно утверждать о их сравнительном быстродействии.

Однако имеет ли скорость загрузки системы к быстродействию ее при реальной работе? Отнюдь. Достаточно вспомнить, что MS DOS 3.3 на IBM PC/XT грузилась быстрее, чем любой Linux на любом супер-мега-Ivy Bridge. Перефразируя слова Сергея Образцова, измерение скорости загрузки ОС, строго говоря, даже к скорости загрузки ОС никакого отношения не имеет. Потому как зависит скорость загрузки в первую очередь от количества подгружаемых модулей и стартовых сервисов. Так что измерение её на самом деле меряет радиус кривизны рук пользователя, меру его лени или, напротив, количество свободного времени, которое он способен выделить на доведение системы до ума. А в последнее время, с распространением SSD-накопителей, скорость загрузки ОС вообще измеряет толщину кошелька пользователя или степень его жадности.

Не лучше и с тестами на реальных приложениях под разными ОС. Например, со столь любимым сравнением скорости обработки запросов web-сервером под Linux и FreeBSD, на основании чего делается вывод о превосходстве одной операционки над другой. Кстати, и не помню даже, кого над кем, да это и не важно. Потому что сразу же возникает закономерный вопрос: а что меряется в этом случае? Сравнительное быстродействие ОС? Или все-таки качество реализации конкретной версии Apache или, например, MySQL под ту и другую систему?

В общем, отдав в свое (не такое уж давнее) время дань увлечению всякого рода тестированием (это занятие было бы точнее

классифицировать как пузометрию или ... ну, то, что в этнографической литературе называют «сравнением мужей», подробно описанному в книжке: **Мир FOSS. Заметки гуманитария**), я пришел к стойкому убеждению, что в большинстве случаев это либо измерение аршином с точностью до ангстрема, либо, по изящному выражению Таненбаума, сравнение яблок с апельсинами.

И тем не менее, методика тестирования, предложенная командой Таненбаума, производит впечатление. Во-первых, она (команда) поставила себе целью оценить влияние на быстродействие системы одного-единственного фактора: выноса драйверов за пределы ядра и перемещения их в пользовательское пространство. И потому тестирование быстродействия MINIX 3 проводилось... на MINIX 2. Каким образом? Очень просто: в качестве сравнительных объектов использовались каноническая MINIX 2, с одной стороны, и она же, пересобранная с удалением из ядра драйверов устройств и еще некоторыми модификациями, что фактически превратило её в MINIX 3.

Во-вторых, в качестве тестов выполнялись процедуры, скорость которых действительно зависит от ОС исключительно или очень существенно: время исполнения системных вызовов, скорость чтения из файла и записи в файл, а также чтения непосредственно из блочного устройства (винчестера). Тесты с реальными приложениями тоже проводились – но предельно простыми (в смысле – мало подверженными посторонним по отношению к ОС влияниям): пересборка образа системы и набора контрольных тестов POSIX, а также обработка текстового файла утилитами типа sed, grep.

Результаты оказались парадоксальными. Разумеется, квази-Minix 3 проиграла MINIX 2 по всем статьям. Но давайте посмотрим, где и насколько.

По чисто «ядерным» тестам вроде исполнения системных вызовов отставание первой составило 12%, по тестам на файловых операциях и запросах к базам данных – 8-9%, по тестам на реальных приложениях – 6%. Последнее на современных машинах практически равнозначно отсутствию визуальной разницы вообще.

Чего нет у рыб?

Результаты тестов, о которых я только что говорил, были опубликованы в работе «A Lightweight Method for Building Reliable Operating Systems Despite Unreliable Device Drivers» ещё в январе 2006 года. Они показали, что катастрофического провала производительности при переходе к «чистому» микроядру не наблюдается. И, следовательно, идея, положенная в основу архитектуры MINIX 3:

- имеет право на существование, и
- заслуживает дальнейшего развития.

Так что дело оставалось за малым: воплотить идею во что-то, реально работающее. А до этого на момент публикации результатов тестов, MINIX 3 было также далеко, как омарам – нетрадиционным способом пробраться до города Пекина.

Обычно, когда описывают особенности некоей системы, разговор начинается с того, что в ней есть. Я же нарушу традицию, и расскажу о том, чего с MINIX 3 не было. И даже не в момент её анонса, а более чем год спустя, на рубеже 2006-2007 годов, когда мне довелось познакомиться с ней воочию – в актуальной на тот момент версии 3.1.2.

По поводу возможностей, отсутствующих в MINIX 3 на момент моего с ней знакомства, хочется сделать небольшое литературное отступление.

Выдающийся русский филолог и писатель Лев Успенский (автор, в числе многого прочего, замечательной книжки «Занимательная топонимика») в своих воспоминаниях рассказывает, как в студенческие годы сдавал экзамен по какой-то биологической дисциплине. И на вопрос, вынесенный в заглавие этого раздела, ответил: «У рыб нет монокля и полного собрания сочинений Шпильгагена». А на уточняющий вопрос, знает ли он, чего ещё нет у рыб, сказал: «Знаю. Но монокля и собрания сочинений Шрильгагена у них точно нет». За что и был удостоен отличной отметки.

С MINIX 3 – случай аналогичный. Монокль к установочному ее диску не прилагается, и полное собрание сочинений Шпильгагена на нем не присутствует (да и вряд ли вообще существует в оцифрованном виде). Однако, как и с рыбами, список отсутствующих в MINIX 3 функций моноклем и Шпильгагеном далеко не исчерпывается.

Итак, в MINIX 3 отсутствовали:

- поддержка огромного количества современного оборудования – от шины USB до интерфейса SATA, а видеоподсистема обеспечивала работу только в VESA-режиме;
- возможность динамической линковки приложений с функциями системных библиотек;
- поддержка каких-либо файловых систем, кроме своей собственной – даже доступ к ISO 9660 осуществлялся через устройство, которое у людей располагается обычно чуть ниже спины;
- поддержка виртуальной памяти.

Ясное дело, что с прикладным софтом дело обстояло не лучше. В свежееустановленной системе имелся набор классических UNIX-утилит в реализации, примерно соответствующей стандарту POSIX, то есть далеко не самых богатых возможностями. Конечно, эту проблему можно было частично решить путём доустановки дополнительных пакетов (а их уже тогда было). Но вот с поддержкой устройств, файловых систем и прочего системного инвентаря рядовой пользователь ничего поделать не мог – это была вахта разработчиков.

И они стояли её доблестно: постепенно в MINIX 3 появилась поддержка виртуальной и разделяемой памяти, иных файловых систем, вплоть до подсистемы FUSE с экспериментальной поддержкой NTFS. Обрастала она и драйверами устройств, расширялся круг портированных приложений, в том числе за счёт задействования системы pkgsrc – той же самой, что была принята на вооружение в DragonFlyBSD.

Описывать хронологию всех этих изменений в деталях я не буду: заинтересовавшийся читатель легко может отследить их по новостному разделу официального сайта или его русскоязычного его гомолога. На последнем, кроме того, можно видеть, как крепла поддержка в MINIX 3 русского языка – сначала усилиями отечественных добровольцев, а потом и на официальном уровне.

И ныне, пожалуй, в MINIX 3 не найти разве что, действительно, монокля и полного собрания сочинений Шпильгагена – большинство остальных атрибутов полноценной операционки в ней имеется. Что внушает оптимизм относительно его дальнейшего развития.

Заключение

Вот о будущем всех трёх родившихся на наших глазах операционных систем я и хотел бы сказать несколько слов под занавес. С одной стороны, и DragonFlyBSD, и MINIX 3 развиваются – может быть, не такими темпами, как хотелось бы их поклонникам, но поступательно и необратимо. Правда, о Syllable этого не скажешь – её жизнь протекает ни шатко, ни валко. Но относительный успех хотя бы двух систем из трёх, на фоне бурного развития Linux»а, показателен. Это при том, что разработчики Free- и других BSD тоже не сидят сложа руки.

С другой стороны, Год Великого перелома, о котором речь пойдёт в одной из следующих статей цикла, смешал карты: сначала интенсивная десктопизация Linux, а затем внедрение её инкарнации – Android»а , казалось бы, не оставил для всех других операционных систем места под солнцем.

Однако у этой медали, как ни странно, есть и третья сторона: Linux буквально в последнее время пошёл по пути, от которого в восторге далеко не все применители, майнтейнеры и разработчики. И кто знает, не сделает ли это востребованными более иные операционки, как давно существующие, так и недавно рождённые. Однако это область, где кончается история и начинается политика, да ещё и приправленная предсказаниями и пророчествами.

Глава девятая. UNIX: хождения в народ

Как бы ни хотелось апологетам абстрактной свободы вообще и свободы софта откреститься от внешнего мира, с его презренными проприетарными разработками, результаты их деятельности сосуществовали с последними в едином пространстве. И сосуществование между этими общественно-технологическими системами было если и не всегда мирным, то обычно взаимонаправленным. И, подобно тому как Linux проникал в enterprise-среду, проприетарные UNIX'ы предпринимали попытки внедриться на пользовательские десктопы. Зачем и почему? Не трудно ответить.

Зачем нужен десктопный UNIX

На протяжении многих лет из всех закоулков FOSS-мира слышались радостные вопли о UNIX'е с человеческим лицом. А с распространением Linux'a они стали перемежаться стонами о неготовности его к десктопу.

Зачем надо готовить Linux к десктопу? Исторический опыт, в том числе и отражённый в этой книжке, однозначно свидетельствует, что система, пришедшая в дома на пользовательские десктопы, неизбежно рано или поздно окажется в промышленном секторе.

Это хорошо иллюстрируется случаем с Windows: её 95-я инкарнация, появившись на пользовательских десктопах сначала как платформа для запуска игрушек, быстро проникла на десктопы и офисных работников. А с выходом сестры во интерфейс – Windows NT 4.0 – оказалась и достаточно массовой средой для серверов рабочих станций. И с тех пор только укрепляла свои позиции по всем фронтам.

Напротив, enterprise-систему, на пользовательские десктопы не попавшую, столь же неизбежно ждёт увядание и в её основной, промышленной, нише. Забегая вперёд, скажу, что такова была судьба проприетарных UNIX'ов, таких как Tru64, HP-UX, Sun Solaris и ряд ныне забытых: иные мертвы, остальные существуют по инерции, на

старых корпоративных контактах. И одна из причин том – недостаточное внимание десктопному сектору.

Как показывает история, системам, не преуспевшим на десктопах, мало чего светит и на противоположном полюсе пользовательского пространства – на всякого рода гаджетах. И тут достаточно вспомнить блистательный взлёт PalmOS и Symbian, не имевших никаких десктопных традиций – и их медленное, но неуклонное отступление под натиском Windows Mobile/CE, перешедшее затем в паническое бегство.

В меньшем масштабе история повторилось и при появлении нетбуков: модели с предустановленным Linux'ом быстро исчезли из прайс-листов. И не вследствие козней «Империи зла», а из-за отсутствия спроса. Отдельные же сохранившиеся реликты – не более чем платформы для установки пиратской Windows.

Подозреваю, что в ближайшее время следует ожидать и ещё одного витка истории. С появлением Windows 8, в том числе и в мобильной модификации, сдадут свои позиции гаджеты на Android'е, казалось бы, властвующем здесь безраздельно. Ибо длинные руки Microsoft'а дотянулись до святой святых без-win'ного мира: до ARM-процессоров. И тут магия имени сработает в очередной раз: потребители гаджетной продукции, снова предпочтут устройства с системой, хотя бы именем похожей на ту, что стоит на их настольных персоналках. А поскольку Android за всё время своего доминирования на гаджетах так и не порадовал, в отличие от былых PalmOS и Symbian, своими технологическими достоинствами, к ним присоединятся и многие из тех, кого действительно можно назвать пользователями.

Я далёк от мысли считать себя самым умным. И уверен, что сказанное выше задолго до меня понимали серьёзные дяди из фирм, разрабатывающих проприетарные UNIX'ы. Почему время от времени можно было наблюдать попытки экспансии последних на рабочие столы применителей, о чём нынче не очень любят говорить вслух – или просто напрочь забыли.

Блеск и нищета персональных Workstation'ов

В этом разделе я расскажу о почти забытой истории – первых попытках «десктопизации» рабочих станций на RISC-процессорах под управлением проприетарных UNIX, происходивших в 90-х годах прошлого столетия. И надо сказать, некоторые предпосылки к тому имелись.

С одной стороны, в те времена былинные под UNIX развивалось некоторое количество обычных пользовательских приложений. Лучший текстовый процессор всех времен и народов – WordPerfect, замечательная электронная таблица Wings, в которой было реализовано большинство достижений, позднее приписанных Excel'у, Frameworks – единственная настольная издательская система, пригодная для вёрстки очень больших и очень сложно структурированных документов, – все они имели и версии под большинство проприетарных UNIX-систем. А уж что касается специализированных приложений, типа ГИС или имидж-процессоров, то они и разрабатывались под UNIX, а лишь затем портировались на Windows NT или получали аналогов под свободные UNIX-подобные системы.

С другой же стороны, в это время начала стираться грань между рабочими станциями и персональными компьютерами. С древних времён существует мнение, что рабочая станция – это что-то могучее и серьёзное, а персональный компьютер – слабое и легкомысленное. На рубеже 80-х и 90-х годов прошлого века оно имело некоторые основания: в рабочих станциях мощные высокоскоростные (в плане мегагерц) процессоры с RISC-архитектурой – в персональных компьютерах CISC-процессоры с тактовой частотой, дай бог переваливающей за 10 МГц. В «работягах» – мегабайты памяти, в «персоналках» – роковые 640 КБ, в лучшем случае до мегабайта, из которого треть невозможно использовать без дополнительных ухищрений. «Работяги» комплектуются мощными видеосистемами и крупноформатными мониторами высокого разрешения – в «персоналках» гнездятся CGA/EGA-адаптеры, к которым подсоединены соответствующие мониторы о четырнадцать дюймах. В рабочих станциях...

Да, собственно, на этом сравнение можно заканчивать: сказанного, казалось бы, достаточно, чтобы возместить между «работягами» и «персоналками» Железную Стену, разделяющую Мир

Гуманного Воображения и Мир Страха перед будущим, как писали некогда классики. Не зря же приобщившиеся рабочих станций ласково называли персональные компьютеры «писюками» (причём вне зависимости от того, шла ли речь о собственно «плебейских» IBM PC или претендующих на аристократичность Macintosh'ах). Однако возгордились они рано, потому что ситуация менялась прямо на глазах.

Уже появление первых процессоров Intel 80486 с их внутренними RISC-инструкциями ознаменовало начало «стирания грани» между «персоналками» и «работягами» в номинальной вычислительной мощности, а Pentium и Pentium II процесс усугубили. И к рубежу тысячелетий можно было говорить о полном «замазывании пропасти», как бы ни пытался этому воспрепятствовать профессор Выбегалло. А поскольку в это же время происходило стремительное вымирание почти всех аппаратных платформ некогда легендарных рабочих станций, то ныне рабочая станция – это обычно, за исключением отдельных реликтов, тот же компьютер на базе архитектуры x86, что и «персоналка», и различия их – в сфере применения, определяющей детали конфигурации.

На моей памяти в 90-х годах прошлого века было три попытки проникновения рабочих станций на пользовательские десктопы. Первым на этом поле решил сыграть Hewlett-Packard со своей RISC-платформой HP-PA (Precision Architecture) и UNIX-операционкой HP-UX. Где-то в 1993-1994 годах появились сообщения о создании пользовательского варианта таких машин, в несколько урезанной (с точки зрения памяти и дискового пространства) комплектации, но зато по вполне «пользовательской» цене, сопоставимой со стоимостью тогдашних PC-брендов типа IBM или Compaq. В начале 1995 года машины эти демонстрировались на первой отечественной выставке UNIXExpo – и, должен заметить, впечатление производили изрядное, особенно в отношении 3D flying'a («облёта» виртуального трёхмерного ландшафта) и воспроизведения видео в реальном времени – на PC и то, и другое были ещё практически недоступно. Однако дальнейшего развития это начинание не получило, о причинах чего скажу чуть позже.

Вторым вступила в игру компания DEC с компьютерами на базе процессоров Alpha – первыми в те времена с точки зрения номинального быстродействия и к тому же первыми 64-разрядными.

Эти машины были способны работать под управлением нескольких ОС – собственных VAX/VMS и DEC UNIX (Tru64 UNIX), а также Windows NT; существовал тогда уже и порт Linux под процессоры Alpha (кстати, вообще первый порт этой ОС на платформу, отличную от i386).

Компания DEC пошла другим путем, нежели Hewlett-Packard: она предоставила возможность сторонним производителям собирать машины на базе своих процессоров и материнских плат, прочие же компоненты Alpha-компьютеров к тому времени (на дворе был 1997 год) уже использовались стандартные, те же самые, что и для PC. Я знал две или три фирмы в Москве, собиравшие на заказ такие машины, также по цене, не превосходящей высококлассные персоналки. А комплектовались они, по желанию заказчика, любой из поддерживаемых ОС – правда, за отдельную мзду. Но не возбранялось приобретать их без ОС вообще, и устанавливать на них Linux своими силами.

В третьем раунде борьбы за пользовательские десктопы (гонг прозвучал в самом конце 90-тых) выступила компания Sun. Она также обратилась к услугам сторонних партнёров, в числе которых были и крупные российские компьютерные фирмы (например, R-Style). Они продавали самые обычные компьютеры Sparc, комплектовавшиеся из экономии стандартной для PC видеосистемой (а не собственным видеоадаптером 3D Creator, который, будучи и сам не дешевым, требовал дорогостоящего фирменного монитора) и IDE-дисками. В качестве предустановленной ОС на них имел место быть, разумеется, Solaris.

Все три проекта потерпели фиаско. Причин к тому было несколько:

- и дороговизна стульев (пardon, UNIX-десктопов) для трудящихся всех стран, привыкших уже к дешёвому и относительно качественному самосбору, утратив привычку «меряться брендами»;
- и неудачность комплектации: так, HP-PA в «пользовательском» исполнении скорее напоминал сетевую рабочую станцию, мало пригодную к автономному существованию; сфера же применения «самосборных» Sparc вообще оказывалась неясной – для пользовательских десктопов они были дороговаты, до серверов не

дотягивали из-за дисковой подсистемы, а как серьезные графические станции не проходили из-за слабой видеосистемы;

- и постепенное прекращение развития большинства пользовательских приложений для проприетарных UNIX;
- и, наконец, просто утвердившаяся среди конечных пользователей привычка к Windows всякого рода.

Собственно, единственным из трех перечисленных проектов, имевшим шансы на успех, был вариант Alpha. Этому способствовала

- и политика, ориентированная на «конструктора-самоделкина» (вплоть до сборщика-индивидуала – такая возможность в Москве тоже имелась);

• и самая демократичная из всех трёх рассмотренных вариантов цена: когда в конце 1997 года я всерьёз размышлял о покупке такой машины «навалом», то высчитал, что она, при прочих равных, обошлась бы на 500 баксов дороже, чем самосборный компьютер на Pentium-II;

- и, наконец, наличие порта самой демократичной ОС – Linux – именно её устанавливали на «самосборные альфы» все известные мне их обладатели.

Конечно, можно было бы порассуждать на темы альтернативной истории – о том, что было бы, если бы платформа Alpha с ОС Linux на борту во второй половине 90-тых годов завоевала бы хоть какую-то популярность. Но история не имеет сослагательного наклонения. И единственное, что мы можем извлечь из нее, – это память об ошибках.

Sun Solaris: глотки свободы

И такие уроки постаралась извлечь фирма Sun. К рубежу тысячелетий она осознала, что попытка «десктопизации» UNIX на собственной аппаратной платформе обречена на провал. Каковы бы ни были достоинства последней (в данном случае – платформы Sparc), не будучи массовой, по соотношению цена/производительность она и близко не сможет конкурировать в «настольном» сегменте с машинами на архитектуре x86. Что, собственно, и показала история с «народными» Sparc'ами: даже в таком обстуганном с разных сторон виде они оказывались через чур дорогими для среднего десктопа и недостаточно мощными – для десктопа класса high end.

И тогда, подозреваю, что в головах Sun'овских «кардиналов» зародились те же коварные замыслы, что несколько раньше взлелеял Билл Гейтс, а несколько позже воспроизвёл Марк Шаттлворт. И которые с успехом раскусил... нет, не Шарль де Баас де Кастельморо, известный нам под ником д'Артаньян. А всего лишь – (не очень) скромный автор этих строк.

Замыслы были просты: если не получается продвинуть в массы ОС с собственной платформой (а без продвижения в массы говорить о массовости и, следовательно, перспективе удешевления последней смешно), то почему бы не продвинуть туда же ОС на платформе массовой? В расчёте на то, что массы, привыкнув к новой ОСи на «своей» платформе, по мере своего информационного и финансового роста ринутся скупать то «железо», для которой эта ОС была создана.

Здесь можно воспользоваться поэтическим обобщением Алисы Деевой (см. сборник её **Стихов и прозы** в **Библиотеке Блогосайта**):

Чтоб тайны приподнять завесу Скажу: хотя все бабы
стервы, Но путь к постели поэтессы Лежит через её
шедевры.

Применители же по своей природе всё больше стервецы (не считите за дискриминацию по гендерному признаку, но стерв среди них мало). И путь к их кошельку лежит через их десктопы. И ещё в 90-х годах фирма Sun начала пролагать этот путь посредством своей ОС.

Проложение это происходило параллельно с попыткой внедрения своей аппаратной платформы, хотя предпосылки к нему были заложены задолго до этого – с выходом в 1993 году версии Solaris для платформы x86. Некоторое время она существовала на правах бедного родственника – вычислительная мощь родной архитектуры Sparc настолько превосходила тогда любые Intel-совместимые процессоры, что в использовании на них весьма прожорливой к ресурсам ОС в промышленных масштабах казалось мало осмысленным. И потому было принято решение распространять версию для x86 бесплатно для некоммерческого использования.

Правда, и тут поначалу Solaris'у на десктопах ничего не светило. Один вид её листа совместимости с оборудованием вызывал скупую мужскую слезу даже у применителя Linux'a, не избалованного тогда

поддержкой аппаратуры. В частности, все видеокарты, которые поддерживались использовавшимся в те годы в Solaris коммерческим X-сервером, уже являли собой музейные экспонаты. Да и последний не отличался дружелюбием к пользователю, особенно не нативному носителю английского.

Так что большой популярности Solaris для x86 не снискал даже в условно-бесплатном варианте. Тем более, что фирма Sun, как истинный хозяин своего слова, то давала его в отношении бесплатности Solaris, то забирала его обратно. Пока, наконец, не было принято кардинальное решение: открыть исходники всего, чего можно открыть без патентных осложнений со сторонними производителями. Что и было сделано в июне 2005 года.

Для открытых исходников Solaris была изобретена собственная лицензия, именуемая CDDL (Common Development and Distribution License). Она основана на известной MPL (Mozilla Public License) и, подобно последней, признаётся FSF в качестве свободной. Однако ряд особенностей делает её несовместимой с GPL – то есть невозможным использование в одном проекте кода под обеими лицензиями. Тут есть немало крючкотворских заморочек, вникать в которые – выше разума обычного человека. Однако остаётся фактов – эта лицензия породила ряд коллизий, одна из наиболее показательных будет рассмотрена в следующей главе.

Однако просто открытием исходников Sun не ограничилась. Следуя примеру некоторых коммерческих дистрибутивов Linux (о чём пойдёт речь во второй части нашей книги), она расщепила свою ОС на две ветви – проприетарную, собственно Solaris, и свободную – OpenSolaris. И тут надо подчеркнуть, что в основной своей части первая была столь же открыта, как и вторая. И она также могла использоваться бесплатно. Отличие же заключалось в том, что собственно Solaris включала в себя некоторый дополнительный проприетарный (и закрытый) софт. Кроме того, «законный» покупатель ОС Solaris получал техническую поддержку от Sun Microsystems, которой скачиватель бесплатного варианта был, разумеется, лишён.

С тех пор ситуация изменилась. И после ответвления свободного потомка от проприетарного Solaris в этом потомке использовались те же самые Xorg, интегрированная среда GNOME, Openoffice.org и

другие приложения, что и в любом дистрибутиве Linux или BSD-системе. Это давало надежду на возможность использования OpenSolaris в мирных целях. Насколько они оправдались — мы увидим в одном из следующих разделов этой главы. А пока обратимся к тому, чем завершилась история собственно OpenSolaris.

Мир без Солнца, или бессмертна ли мафия?

Смутные слухи о том, что фирма Sun решила продаться общественным работникам, циркулировали в Сети задолго до того, как этот факт свершился. Причём в качестве ответственного работника называли разные компании. Когда же оказалось, что в этой роли выступила Oracle, возникли опасения за судьбу свободных проектов, находившихся на Sun'итарном иждивении. А в их числе были Openoffice.org, MySQL, VitruaBox, ряд средств разработки, не говоря уже о собственной ОС – OpenSolaris, которая нас сейчас собственно и интересует. Не загнутся ли они под чутким руководством Ларри Эллисона? – думалось в народе.

Впрочем, оснований волноваться за большинство из них не было ни малейших. Так, MySQL вполне могла выступить «легковым» дополнением к собственно СУБД Oracle. OpenOffice.org не могли бросить, как единственный «полноценный» офисный пакет из числа свободных, ввиду его явной востребованности конечным пользователем. VirtualBox и Sun Studio etc. представляли интерес для множества разработчиков, и потому их тоже легко не прикрыть.

В первую очередь опасения касались OpenSolaris, ибо возникал резонный вопрос, касающийся её проприетарного собрата, то есть собственно Solaris: а нужна ли будет Oracle ещё одна ОС, в дополнение к собственному клону RHEL? И если нет – то необходимости в тестовой площадке для неё (а именно в этом качестве OpenSolaris и выступала в свои самые «солнечные» дни) не будет тем более. Сама же она, за время своего «свободного плаванья» не достигла ни полностью законченного состояния, ни критической массы пользователей и разработчиков – той, что сделало бы её способной это плавание продолжать.

Ныне мы знаем, что опасения эти были не беспочвенны. Судьба всех курировавшихся Sun свободных проектов оказалась различной, но

в общем и целом не трагичной, и в том или ином качестве они продолжают свое развитие. Все, кроме OpenSolaris: этот проект можно считать полностью свёрнутым: 14 августа 2010 года Oracle официально объявила о прекращении её разработки как свободной системы. Отныне она стала доступной только в бинарном виде и на условиях Oracle Technology Network Developer License, то есть только для разработки и тестирования программ под неё же саму.

Этим новый владелец OpenSolaris подписал ей смертный приговор: если она и была чем интересна потенциальным пользователям – то своими инновациями. А превратившись в Solaris... не для бедных даже, а для нищевродов, питающихся объедками с барского стола, она стала не интересной никому. Как, подозреваю, вскоре станет не интересен и «большой» Solaris – срок его жизни отмерен временем контрактов на техподдержку крупных заказчиков.

Правда всё может сложиться и не столь трагично, но об этом – в следующем разделе. А пока извлечём из всего сказанного мораль, свежую и оригинальную: строить серьёзные корпоративные и тем более государственные решения можно только на базе систем, давно и хорошо поддерживаемых своими сообществами. Ибо самый раскомерческий фирмач может в одночасье продаться своему другу, врагу, конкуренту или просто прохожему. И только мафия сообщества – бессмертна. До тех пор, пока сообщество существует, разумеется...

Дети Солнца

Как было сказано в одном из предшествующих разделов, после открытия исходников SunOS и большей части её сопутствующего обрамления, от генеральной линии партии Solaris, по образу и подобию Red Hat с его Fedora, и SUSE с её openSUSE, ответвился комсомольский побег — OpenSolaris. Который, согласно известной песенке, призван был с молодым задором пробивать дорогу для почёта старикам. Впрочем, фрайер этот танцевал под музыку не долго – до продажи его «партийного куратора» фирме Oracle.

Однако открытие исходников SunOS и её обрамления спровоцировало также и несанкционированную старшими партийными товарищами активность в виде клонов. Причём первый из них, ShilliX, был представлен Георгом Шиллингом (известным

разработкой утилиты cdrecord) буквально через несколько дней после данного события. Вслед за чем появились такие клоны (или дистрибутивы?) OpenSolaris, как BeleniX, Nexenta и MilaX.

Затем последовало приобретение Sun'a фирмой Oracle и закрытие разработки OpenSolaris -- ей на смену пришёл бесплатный, но не открытый Solaris «для нищелюбивых». Что вызвало появление свободного форка – illumos, продолжавшего развития SunOS, и разрабатываемого на его основе дистрибутива OpenIndiana, явившейся преемницей OpenSolaris.

Это была, так сказать, генеральная линия комсомола, не примирившегося с засильем партийных геронтократов из официального Solaris'a и классово чуждых буржуинов Oracle. Однако наряду с ней образовался и левый уклон.

Выше я упоминал клоны (или дистрибутивы?) OpenSolaris. Из них SchilliX как-то развивается и по сей день -- но это система, ни в коем случае не предназначенная для конечного пользователя (честно говоря, я так и не понял, для кого она предназначена, кроме её собственного разработчика). Прочие же системы не пережили продажи Sun'a и фактического сворачивания головного проекта, OpenSolaris. Но если BeleniX и MilaX тихо канули в Лету, то Nexenta породила сразу две линии систем.

Первой была Nexenta сама по себе -- своеобразная надстройка инфраструктуры Debian над SunOS и её userland'ом. Однако и она как самостоятельный продукт прекратила своё развитие довольно быстро. Ныне это своего рода основа для NexentaStor -- специализированной коммерческой системы для хранилищ данных, использующей преимущества ZFS. А вот вторая линия её развития имела неожиданное продолжение -- в виде ориентированной на десктопы системы StormOS, которая своей быстротой и компактностью вызвала ассоциации с теми дистрибутивами Linux'a, которые я некогда назвал системами быстрого развёртывания.

Система StormOS меня некогда очень заинтересовала, однако век ей был отпущен недолгий: уже через год на её официальном сайте сообщения о дальнейшем развитии стали носить очень уклончивый характер. И я почти забыл о ней -- мало ли в бозе усопших дистрибутивов UNIX-подобных систем мне довелось видеть на своём веку. Но однажды, зайдя на полумёртвый сайт проекта StormOS, я

неожиданно обнаружил на нём следы очередного восхода – DysonOS, возникший в сентябре 2012 года.

По агентурным данным, разработчиком Dyson является наш соотечественник, Игорь Пашев. И она представляет собой продукт дальнейшей интеграции Solaris и Linux в его Debian-ипостаси. Если в Nexenta и StormOS (как и в современной OpenIndiana) Debian'овские механизмы накладываются на ядро и userland от Solaris'a, то в Dyson последний планомерно замещается своими GNU-аналогами. В настоящее время от исходной системы, кроме ядра и специфичных служб обеспечения (SMF, DTrace, ZFS) сохраняется также собственная LibC, которую автор в ближайшее время планирует заменить на glibc.

Разработка Dyson находится на достаточно ранней стадии, и кое-что из критически важного для конечного пользователя в ней не реализовано. Однако самое интересное в этой системе -- то, что в существующем виде она работает. И есть шанс, что она будет развиваться и дальше, храня наследие Solaris в мире FOSS.

Глава десятая. Из истории файловых систем

Файловые системы и, шире говоря, системы размещения данных вообще – неотъемлемая часть операционных систем. И потому их историю уместно рассказать в этой части книги. К тому же развитие ОС и их систем размещения данных – вещи, очень тесно связанные друг с другом.

Вводные слова

Одна из главнейших задач при работе на компьютере – манипулирование данными: создание, модификация, копирование, перемещение и так далее. И тут первое дело – это организация их размещения. Это понятие включает в себя широкий круг частных вопросов – схемы дисковой разметки, управления дисковыми массивами и логическими томами, файловые системы и их монтирование в файловую иерархию. Они тесно связаны между собой, но традиционно решаются каждая с помощью собственного инструментария.

Однако с некоторых пор в UNIX-подобных операционках получили распространение интегрированные системы размещения данных, объединяющие в себе и файловые системы, и задачи управления массивами и томами, и даже, частично, задачи разметки дисков. Такие системы, как мы скоро увидим, существовали очень давно – со времен доисторического UNIX’а, хотя и в проприетарном исполнении. Ныне среди них представлены и системы свободные, хотя и распространяемые под разными, не всегда совместимыми, лицензиями.

Вне зависимости от того, реализуется ли размещение данных путем отдельных инструментов или интегрированных систем, оно должно обеспечивать выполнение ряда требований. Которые были сформулированы ещё в старом советском анекдоте. Как известно еще с те, атеистических, времен, Господь Бог, создавая человека, хотел сделать его умным, честным и партийным. Но оказалось, что даже он,

при всём своём всемогуществе, не смог ему дать больше двух качеств вместе.

Аналогично и с системами размещения данных: разработчики хотели бы видеть их быстрыми, надёжными и простыми в обращении. Давайте посмотрим, удалось ли им превзойти Господа.

Дисковая разметка

Говорят, что во времена далекие, теперь почти былинные, файловых систем не было: информация на носители записывалась побитно, без всякой организации в именованные её наборы. Впрочем, такой способ записи данных применялся и много позднее – например, при резервном копировании на стримерные ленты. Можно обходиться без файловых систем и при записи на стандартные блочные устройства – винчестеры, SSD, компакт-диски. Однако в большинстве случаев данные на носителях блочного типа организуются в виде файлов, а файлы объединяются в файловые системы – плоские, как в древнем DOS'е, древовидные, как во всех UNIX-подобных операционках, или, так сказать, «многодревные», как в Windows. Каковые могут быть созданы непосредственно на носителе как raw-устройстве, но обычно накладываются на дисковые разделы.

До недавнего времени в Linux'е применялась разметка в MS-DOS-стиле, предполагающая возможность разбиения диска на четыре раздела, называемых первичными [primary partitions]; один из них может быть определён как расширенный раздел [extended partition], внутри которого по «матрёшечному» принципу можно создать логические разделы, максимальным числом до 63.

Разметка в MS-DOS-стиле преобладает в дистрибутивах Linux'а и по сей день. Однако всё большее распространение получает разметка в GPT-стиле. Среди её преимуществ – возможность создания на диске до 128 абсолютно равноправных (то есть не разделяющихся на физические и логические) разделов. А в случае использования винчестеров «продвинутого» формата [Advanced Format] и SSD, размер блоков которых равен 4 КБ, она обеспечивает оптимальное выравнивание границ разделов.

Исторически сложилось так, что одному разделу соответствовала одна файловая система. Соответственно, и выходить за границы

несущего их устройства файловые системы не могли. И если требовалось работать более чем с одной файловой системой на одном физическом накопителе (а в UNIX-подобных ОС это почти всегда так), то был необходим тщательный расчет дискового пространства для каждой из них: ошибки в расчетах влекли весьма неприятные последствия, вплоть до необходимости переразбиения диска и переустановки ОС вообще.

Правда, дисковые разделы могут не только разделяться, но и объединяться в программные массивы или в группы томов, о которых мы сейчас и поговорим.

Массивы и логические тома

Задача объединения носителей информации особенно актуальна при использовании нескольких физических накопителей, и особенно при их добавлении в работающую систему. В элементарном исполнении это делалось просто (по крайней мере, в UNIX-подобных ОС): второй (новый) накопитель просто размечался по соответствующей для данной ОС схеме, на нем создавалась новая файловая система определенного типа, которая монтировалась в общую файловую иерархию. Однако выход за границы существующего раздела и диска для файловой системы был по-прежнему невозможен.

Для решения задачи объединения физических носителей в единое логическое устройство и «размазывания» по ним файловых систем традиционно используется два основных способа: RAID (Redundant Array of Independent Disks – избыточный массив независимых дисков) и LVM (Logical Volume Manager – менеджер логических томов).

RAID'ы существуют трёх видов – аппаратные, квази-аппаратные (так называемые Fake RAID) и чисто программные (Soft RAID). Первые дороги и на десктопах почти не встречаются; работа вторых под Linux'ом часто проблематична, так что речь пойдёт в основном о третьих. Впрочем, с точки зрения логики это роли почти не играет.

Логически в любом из RAID'ов несколько дисков (а в Soft RAID – и дисковых разделов) могут просто слиться воедино (Linear RAID), при записи на них может осуществляться расщепление данных [stripping], что приводит к ускорению дисковых операций (RAID Level

0); на объединенных разделах можно создать различные формы избыточности, обеспечивающей восстановление данных при отказах дисков. Из таких избыточных массивов чаще всего используется полное дублирование (RAID Level 1, он же mirror) или избыточность за счет контрольной суммы (RAID Level 5). Наконец, возможно и совмещение стриппинга с дублированием.

RAID любого типа и уровня может разбиваться (и обычно разбивается) на разделы, которые уже несут на себе файловые системы. И, таким образом, позволяют размещать их на нескольких физических устройствах. Однако они не решают второй проблемы размещения данных – необходимости расчета потребного для них дискового пространства и его перераспределения при необходимости.

Этим целям служит технология LVM, объединяющая физические носители в группы логических томов, разделяемых на собственно логические тома, которые, в свою очередь, разбиваются на экстенды – объединения физических блоков дисковых устройств. Логические тома предстают перед операционной системой как обычные разделы, каждый из которых может нести свою файловую систему. При этом технология LVM даёт возможность при необходимости перераспределять физическое пространство носителей между ними посредством добавления или отнятия экстендов на лету, не только без переразметки дисков, но и без перезапуска системы.

Технология LVM может обеспечить, как и RAID Level 0, стриппинг данных между физическими томами с целью повышения быстродействия файловых операций. А в сочетании с Soft RAID позволяет и создавать массивы с полной (зеркалирование) или частичной (за счёт контрольных сумм) избыточностью, повышающей надёжность. Таким образом, LVM выполняет оба поставленных условия: слияние дискового пространства, в том числе и вновь подключаемых накопителей, и возможность его перераспределения между существующими файловыми системами, да ещё и с бонусом в качестве повышения быстродействия. Комбинация же LVM и Soft RAID позволяет и повысить надёжность. Казалось бы, чего ещё не хватает для счастья?

Чтобы ответить на этот вопрос, следует вспомнить тезис Господа Бога об уме, честности и партийности. То есть в нашем случае – о

быстроте, надежности и простоте использования. В соответствие с чем видим:

- либо быстрое и простое решение на основе RAID Level 0, не блещущее надёжностью;

- либо надёжное решение без ощутимой потери быстродействия на основе одного из RAID с избыточностью, не являющееся, однако, эталоном простоты; влекущее, кроме того, ещё и потерю дискового пространства вплоть до пятидесятипроцентной (в случае RAID Level 1);

- либо, наконец, относительно надёжное и потенциально быстрое решение при использовании технологии LVM – однако о простоте здесь можно забыть сразу: если установить LVM позволяет инсталлятор почти любого современного дистрибутива, то управление логическими томами и по сей день задача не из самых тривиальных.

К тому же мы забыли о файловых системах. А они вносят свою, и немалу, лепту в соотношение «ума, честности и партийности».

Файловые системы

Изначальная файловая система Unix носила имя s5fs (то есть файловая система System V). По свидетельству современников, была она отменно медленной, да еще и не допускала имен файлов из более чем 14 символов. А поскольку в те времена в Университете Беркли также разрабатывалась версия Unix (именовавшаяся BSD Unix – предтеча всех современных BSD-систем), то берклианцы решили поправить дело. И создали в 1983 году файловую систему, названную FFS (Fast File System, где первое слово символизировало ее быстроту сравнительно с s5fs).

Поскольку FFS, как и прочие разработки берклианцев, была свободной, создатели проприетарных Unix'ов не погнушались включить поддержку FFS в очередную версию канонической System V. А уже от нее и произошло, прямо или косвенно, большинство современных файловых систем UNIX, как проприетарных, так и свободных. Хотя для нашего повествования имеет значение только UFS из FreeBSD.

Вообще говоря, UFS расшифровывается просто как файловая система Unix (Unix File System). И под этим именем известны и

файловые системы других ОС этого семейства, отнюдь не идентичные UFS из FreeBSD (например, файловая система SunOS и Solaris). Так что некоторая неопределенность терминологии имеет место быть.

Однако нас интересует только собственно UFS из FreeBSD. Ибо именно в ней впервые появилось понятие группы цилиндров с собственной структурой. Они призваны были обеспечить минимизацию перемещения головок винчестера и, соответственно, быстродействие файловых операций.

Другим новшеством UFS, появившемся несколько позже, стал парадоксальный механизм Soft Updates, приводящий к росту одновременно и надёжности, и быстродействию файловой системы. Он, с одной стороны, выполнял контроль за последовательностью зависимых обновлений (тем самым способствуя целостности состояния файловой системы и, следовательно, её надёжности). С другой же стороны, при включении Soft Updates происходила группировка нескольких обновлений в единую атомарную операцию синхронного обращения к диску. Что и вызывало рост быстродействия файловых операций.

И действительно, по производительности UFS+Soft Updates была вполне сопоставима с файловой системой Linux'a – ext2, хотя последняя по умолчанию работала в асинхронном режиме, а UFS – в частично синхронном (с немедленной записью метаданных и отложенной – блоков данных). А в отношении надёжности они были просто не сопоставимы: в те далёкие годы, да ещё и не иобильные с точки зрения бесперебойников, полный развал ext2 при «мёртвом» зависании или сбое питания был делом достаточно обычным, тогда как с UFS у меня такого не случалось ни разу.

В 5-й ветке FreeBSD на смену UFS пришла её усовершенствованная, 64-битная, модификация. Которая привнесла немало удобств (в частности, выполнение операции fsck в фоновом режиме на смонтированных файловых системах), однако ценой этого было быстродействие – точнее, его потеря. Не случайно именно тогда от FreeBSD отпочковалась DragonFly с её файловой системой Hammer, правда, находящейся тогда только в голове создателя, Мэтта Диллона.

В Linux'e же поначалу использовалась файловая система MINIX, созданная Эндрю Таненбаумом для своей одноимённой ОС на базе классической s5fs, без всякого влияния берклианских наработок. Она

имела массу ограничений, в частности, на максимальный размер в 64 МБ (вследствие своей 16-битности), на длину имени файла (14 символов). Что и послужило причиной для её усовершенствования, выполненных Линусом Торвальдсом одновременной с работой над ядром. Что в итоге вылилось в файловую систему extfs (Extended File System), уже 32-битную, с обычными для этого класса поддержкой разделов до 2 ГБ и длины имён файлов до 256 символов. Хотя MINIX долгое время использовался в Linux'e для его загрузочных дискет, а поддерживается ядром чуть ли не по сей день.

Однако и на extfs творческая мысль разработчиков не остановилась. Для ликвидации некоторых присущих ей ограничений, например, в атрибутах файлов, в 1993 году Реми Кард (Remi Card) разработал улучшенную её модификацию, названную ext2. Именно она стала на долгие годы стандартной для ОС Linux. И используется по сей день, оставаясь эталоном быстродействия.

Думаю, каждого, кто начинал знакомство с Linux'ом во времена безраздельного господства файловой системы FAT в DOS/Windows, ext2fs поражала скоростью выполнения всех файловых операций. Что было обусловлено эффективностью их кэширования и полностью асинхронным режимом работы. За что, как уже только что было сказано, приходилось платить надёжностью – сбой системы по любой причине влёл за собой тяжкие последствия, вплоть до полного разрушения файловой структуры. Но и даже когда до полной катастрофы дело не доходило, отказы (например, по питанию) вызывали за собой долгую и нудную процедуру проверки целостности файловой системы.

Для решения проблемы надёжности файловой системы (они, хотя и в различной степени, касались всех UNIX'ов) предлагались различные механизмы – Soft Updates, о котором только что говорилось, был одним из них. Однако магистральная линия развития файловых систем пошла по пути так называемого журналирования. Суть его в том, что сведения о файловых операциях записываются в специальный файл журнала до того, как эти операции будут фактически выполнены. Это дает возможность после любого сбоя «откатить» файловую систему до последнего непротиворечивого состояния. Обратной стороной чего, как обычно, является снижение быстродействия –

различное для отдельных файловых систем и видов файловых операций.

Эпонимом журналируемых файловых систем стала JFS, разработанная IBM для собственных RISC-станций и своего же варианта UNIX — AIX (1990-й год). Затем, в варианте JFS2, она была портирована сначала на серверную версию OS/2 (конец 90-х годов), бившуюся тогда в последней стадии агонии, а вслед за тем — и на Linux, разумеется. Где она и прижилась под именем просто JFS. И стала, кажется, первой «чужой, но породнившейся» файловой системой, поддержка которой была официально включена в каноническую версию ядра — точной даты не помню, но где-то аккуратно рядом с рубежом тысячелетий.

Однако широкого признания Linux-реализация JFS не снискала — ни тогда, ни в последствии. В частности, и вследствие исключительной медлительности — в этом отношении она могла поспорить только с UFS2 из FreeBSD. Причина заключалась в том, что в Linux-версии разработчики отказались от собственного кеширования файловых операций (наличествовавшего в реализациях JFS для AIX и OS/2), положившись на то, что эта функция будет выполняться ядром.

Следующей журналируемой файловой системой в Linux стала ReiserFS, разрабатывавшаяся специально для этой ОС Хансом Рейзером (Hans Reiser) и сотрудниками его фирмы Namesys, начиная с конца 90-х годов. Официальную поддержку в каноническом ядре она обрела с выходом его версии 2.4.

Коньком ReiserFS (кроме собственно журналирования) была работа с **очень** большими массивами **очень** маленьких файлов — то есть файлами меньше логического блока файловой системы (обычно в диапазоне от 512 байт до 4 Кбайт). А таких в любой UNIX-системе действительно несчётное множество. Типичный пример — дерево портов FreeBSD или портежей Gentoo.

В большинстве файловых систем для таких мини-файлов существует как свой inode (информационный узел, содержащий метainформацию о файле), так и блок данных, что приводит как к расходу дискового пространства, так и снижению быстродействия файловых операций. В частности, именно в этом причина катастрофической задумчивости файловой системы FreeBSD (как

старой, UFS, так и новой, UFS2) при работе с собственной же системой портов.

В файловой системе ReiserFS в таких случаях отдельные блоки под данные не выделяются -- она умудряется записать данные файла непосредственно в область его же inode. За счет этого и дисковое пространство экономится, и быстродействие возрастает -- буквально в несколько раз по сравнению со всеми прочими файловыми системами. Повторяю, речь идёт сейчас только об операциях с **очень** маленькими файлами -- на всех прочих такого превосходства, разумеется, нет и близко.

Такое обращение с мелкими файлами ReiserFS послужило причиной возникновения легенды о ее ненадежности. Ибо при крахе файловой системы (то есть разрушении служебных областей) данные, размещенные совместно со своими inodes, вместе с ними же и пропадают -- причем безвозвратно. Тогда как в тех файловых системах, где inodes и блоки данных всегда разобщены в разных областях дискового пространства, данные теоретически можно восстановить. Так, для ext2/ext3 даже существуют средства, позволяющие это сделать.

Однако, как и всякая легенда, эта лишь производит впечатление достоверности. Во-первых, безвозвратная потеря данных относится лишь к очень маленьким файлам. Среди пользовательских таковых практически не бывает, а все прочие же легко восстанавливаются из дистрибутива.

Во-вторых, говоря о возможности восстановления данных из блоков, утративших привязку к своим inodes, я не случайно употребил определение «теоретическая». Потому что на практике это занятие чрезвычайно трудоемкое, не дающее гарантированного результата. Каждый, кому приходилось этим заниматься, согласится, что предаться ему можно только от полной безысходности. И это относится ко всем файловым системам Linux. Так что этим аспектом при выборе файловой системы можно пренебречь.

По суммарному быстродействию ReiserFS была однозначно быстрее всех остальных одновозрастных ей журналируемых файловых систем, а по некоторым показателям могла с успехом потягаться и с ext2, бывшей тогда эталоном быстродействия.

Что же до надёжности – повторяю, панические слухи на этот счёт не имели под собой оснований. В тестах на отключение питания (а во времена её становления я такие тесты проводил регулярно, и не для развлечения – таковы были мои тогдашние жизненные условия) она вела себя ничуть не хуже любых других журналируемых файловых систем.

Главная же проблема ReiserFS была в другом – в совместимости. Сначала она «из коробки» поддерживалась очень малым количеством дистрибутивов, и совсем не поддерживалась никакими другими ОС, даже соплеменными BSD. Только в DragonFly довольно быстро реализовали модуль её поддержки в режиме «только для чтения» – но отношение этой операционки к другим файловым системам всегда было особым.

Проблема с совместимостью для ReiserFS возникла и в последние годы. Если раньше она «ещё не поддерживалась» многими дистрибутивами, то теперь один за другим дистрибутивы её «уже не поддерживают». Видимо, скоро она исчезнет и из ядра Linux'a.

Чтобы более не возвращаться к этому вопросу, замечу, что на протяжении ряда лет Ханс Рейзер и фирма Namesis разрабатывали файловую систему Reiser4. Это не была очередная версия ReiserFS (развитие той остановилось на версии 3.6.X), а существенно новая разработка, в детали которой вдаваться сейчас не буду: до полной пригодности к практическому применению она доведена не была, и теперь уже, наверное, никогда не будет. О некоторых причинах можно догадаться, прочитав соответствующий раздел в книге **Мир FOSS. Заметки гуманитария**, имеющей место быть в **Библиотеке Блогосайта**.

Зато идеалом с точки зрения совместимости оказалась ext3fs – журналируемая модификация классической ext2: представленная на рубеже тысячелетий Стивеном Твиди (Stephen Tweedie), она, однако, получила поддержку в ядре Linux'a не сразу, а даже позже, чем ReiserFS. Однако после этого была включена практически во все дистрибутивы и, следовательно, могла быть прочитана почти на любой Linux-машине. Более того, она осталась полностью совместимой с ext2 по формату. То есть прародительница превращалась в ext3 лёгким движением руки – точнее, добавлением к ней журнала с помощью утилиты tune2fs не только без остановки машины, но даже без

отмонтирования целевого носителя. Не менее проста была и обратная процедура – ext3 можно было просто перемонтировать без журналирования, и тогда она становилась самой обычной ext2. Утилиты для работы файловой системой (создания, проверки и так далее) для ext2 и ext3 также были одними и теми же.

В ext3 (и это была её особенность) предусматривалось три режима работы:

1. журналирование с обратной записью (writeback), когда в файл журнала записываются только изменения метаданных файлов;
2. последовательное, или обычное (ordered), задеиствуемое по умолчанию – также с журналированием только метаданных, но с группировкой связанных с ними данных в единую транзакцию;
3. полное журналирование (journal), когда все изменения и метаданных, и блоков данных фиксируются в журнале, а только потом уже переносятся на диск.

Теоретически считается, что от первого режима к третьему возрастает надёжность файловой системы, но уменьшается быстродействие. В отношении быстродействия – вопрос спорный: Дениэл Роббинс (Daniel Robbins) приводит случаи, когда режим полного журналирования оказывался быстрее не только последовательного, но даже журналирования с обратной записью. По моим наблюдения, показатели быстродействия ext3 были вообще невоспроизводимы и от режима журналирования не зависели вовсе. И в любом случае уступали интегральной скорости работы в ReiserFS (не говоря уже о ext2), будучи сопоставимыми с UFS (не UFS2!) при включении Soft Updates.

Отступление для тех, кто не знает: Дениэл Роббинс был не только создателем дистрибутива Gentoo, но и автором большого количества технических статей, публиковавшихся на сайте IBM для разработчиков свободного софта. Среди этих статей (кстати, обладающих незаурядными литературными достоинствами) было *Руководство по продвинутым файловым системам*, известное в русском переводе Владимира Холманова. А размещались эти переводы первоначально на сайте линуксоидов города Ярославля, созданном в незапамятные времена Александром Благиным и существующем, как ни странно, до сих пор.

Что же до надёжности, то ext3 в режиме полного журналирования снискала репутацию абсолютно «непробиваемой» системы для использования на серверах, а в режиме последовательном считалась оптимальной для настольного применения.

Спорить на счёт устойчивости ext3 не буду. Но моё личное мнение по этому вопросу таково: устойчивость файловой системы вообще зависит от личного везения применителя. Особенно если речь идёт о сравнении ext3 и ReiserFS: жалоб на развал при аварийном завершении работы по поводу первой я слышал не меньше, чем относительно второй.

По собственному же опыту – банальное выключение питания в ходе обычной пользовательской работы, как правило, безболезненно переносят все журналируемые файловые системы, кроме, в некоторых случаях, XFS, о которой скоро пойдёт речь.

Тем не менее, как я уже сказал, репутация часто оказывается самым весомым фактором – и на протяжении долгого времени ext3 предлагалась по умолчанию инсталляторами большинства дистрибутивов.

Ныне, однако, ext3 почти вышла из употребления. Если ext2 всё ещё нередко применяется в ряде специальных случаев (например, для носителей небольшого объёма или там, где журналирование в принципе не имеет смысла), то в амплуа файловой системы общего назначения нынче выступает ext4. Она была разработана Эндрю Мортонем (Andrew Morton) в качестве экспериментальной в 2006 году, и в 2008 принята в ядро Linux'a как стабильная. С тех пор она развивается и поддерживается командой разработчиков, из которых наиболее известен Теодор Цо (Theodore Ts'o).

Подобно ext3, ext4 представляет собой дальнейшее развитие линии ext2, но уже без сохранения совместимости на уровне формата и инструментария. В ext4 предусматривается те же три режима журналирования, что и в ext3, с теми же названиями и особенностями. Однако в ней добавлен и четвёртый – режим работы без журнала вообще. Считается, что в этом режиме она наконец смогла отобрать у ext2 лавры рекордсмена по интегральному быстродействию.

Однако я забежал вперёд – ext4 не стала ещё достоянием истории и, похоже, не собирается это делать в обозримом будущем. Потому что следующей журналируемой системой по очередности появления в

Linux'е стала XFS. Хотя создана она была много раньше – в 1994 году, усилиями фирмы Silicon Graphics, для её собственного варианта UNIX'a – IRIX. И была одной из первых 64-разрядных файловых систем.

К весне 2001 года относится открытие исходников XFS под лицензией GPL и разработка Linux-реализации. Долгое время её поддержка ядром этой ОС обеспечивалась только сторонними патчами – официально она была включена в ядро только в 2004 году. Да и после этого какой-то период по умолчанию поддерживалась не всеми дистрибутивами.

Но постепенно XFS утвердилась в Linux'е в качестве стандартной, чему способствовали её особенности, хорошо вписавшиеся в изменившиеся реалии окружающего мира. Если ReiserFS лучше всего показывала себя в обращении с маленькими файлами, то «коронкой» XFS была работа с **очень** большими файлами на **очень** больших файловых системах. Что и не удивительно, если вспомнить о её происхождении: фирма Silicon Graphics (ныне SGI) издревле была ориентирована на профессиональные средства в области графики и мультимедии, данные которых требовали большого объёма дисковой памяти и организации её в виде больших файлов. К середине нулевых годов это оказалось востребованным и в Linux'е.

Однако это было не единственной причиной распространения XFS: она (с некоторыми оговорками, о которых я скажу чуть позже) отличалась также впечатляющей производительностью и высокой надёжностью. Причём в полном блеске показывала своё быстроедействие на многодисковых устройствах (multiple devices), то есть на аппаратных и программных RAID'ах, которые тоже получили широкое распространение в это время.

Однако, говоря ранее о быстроедействии и надёжности XFS, я не случайно сделал оговорку. Дело в том, что это единственная файловая система, в которой теоретические ожидания (обратная корреляция между этими факторами) воплощается в действительность. В ранних версиях её Linux-реализации, которые действительно были завидно быстры, имело место так называемое «обнуление файлов» при сбое питания.

Затем это безобразие пытались побороть, включив в XFS по умолчанию опцию write barriers, от сей напасти избавляющую. Что,

однако, привело к падению быстродействия на некоторых операциях. В частности, удаление большого количества маленьких файлов (а в работе с маленькими файлами XFS не блистала и без этой опции) стало происходить просто удручающе медленно.

При этом проблема потери данных при сбоях до конца решена не была, хотя и не стояла так остро. Однако, с учётом и старых воспоминаний, отношение к XFS местами было настороженным. Хотя эта система уже давно предлагается по умолчанию в инсталляторах некоторых дистрибутивов, причём отнюдь не самых «мультимедийных» или «промышленных», таких как Zenwalk или Salix.

В итоге, однако, работа с мелкими файлами была оптимизирована за счёт заимствования из ext3 отложенного журналирования – хотя реальное воплощение этого ожидается только в светлом будущем ядра Linux версии от 3.3. Что же до потери данных – это решается ещё проще: разработчики предлагают больше внимания уделять состоянию аппаратуры, в частности, и приобретению источников бесперебойного питания, и поменьше слушать страшных баек об исчезнувших в результате сбоя непреходящих ценностей.

Отступление. Интересно, что это перекликается с позицией Google. Как-то в сеть просочилась информация, что на серверах этой компании используется исключительно ext4 – и в режиме без журналирования (without journal). Как я недавно говорил, теоретически это должно обеспечивать максимальную производительность. Что же до неизбежного в таком случае риска нарушения целостности файловой системы при сбоях – в Google предпочитают бороться с этим путём обеспечения качественного электропитания. Впрочем, надо заметить, что своё решение Google как панацею от всех бед отнюдь не пропагандируют. Видимо, опять таки помня, что водка, легко доступная министру, не всегда по карману не только бичу, но даже простому инженеру.

Изменение отношения к XFS совпали с изменением модели её развития. Фирма-создатель, ныне носящая имя SGI, постепенно отстранялась от дальнейшей её разработки – в последние годы она осуществлялась в основном силами программистов, по совместительству являющихся сотрудниками Red Hat. В конце концов SGI отказалась от поддержки XFS вообще, и ныне эта файловая

система продвигается Red Hat'ом. В частности, она будет файловой системой по умолчанию в RHEL 7.

Рассказ о традиционных файловых системах уместно закончить упоминанием файловой системы Tux3. Это экспериментальная файловая система, развиваемая Дениэлем Филиппсом (Daniel Phillips) на протяжении уже пятнадцати лет, но так никогда и не анонсировавшаяся в качестве окончательного релиза. Отличительной её особенностью является версионная модель. То есть каждый файл в ней существует в виде нескольких разновременных вариантов, что в случае сбоев выполнять откат до последнего работоспособного.

Впрочем, когда мы эту файловую систему увидим – не ясно. Её разработчик лет пять назад в одном интервью высказался по сему поводу очень оптимистично: это случится раньше, нежели портирование на Linux файловой системы Hammer из DragonFly (о ней будет сказано позднее). Учитывая, что с тех пор никто вроде бы и не начинал работы по продвижению Hammer'a в Linux – времени у Дениэла ещё вдоволь.

Резюмирую затянувшийся базар о файловых системах. Можно видеть, что с точки зрения простоты использования ни в одну из файловых систем Linux'а бросить камень рука не подыметься: создание и монтирование их никаких трудностей не сулит. Так что требование «партийности» для них выполняется, пожалуй, при любых соотношениях «ума» и «честности». Но эта ситуация сохраняется, пока мы не начинаем комбинировать «ум, честность и партийность» файловых систем с аналогичными качествами систем управления RAID'ами или с LVM. В частности, вследствие многоуровневости всех этих решений. И очевидно, что удлинение «цепочки» уровней в любом случае приводит к снижению надежности: чем больше в ней звеньев, тем вероятней отказ всей цепи.

И тут-то и возникает вопрос: а нельзя ли уменьшить количество уровней, сделать систему более «плоской»? Попыткой ответа на этот вопрос и стали интегрированные системы размещения данных.

Из истории систем размещения

Не в интересах правды, а истины ради нужно заметить, что комплексные системы размещения данных – отнюдь не порождение

мира FOSS, их корни лежат в недрах проприетаризма. И первой из них была, видимо, файловая система Veritas (или VxFS), разработанная фирмой Veritas Software и представленная миру в 1991 году. Она же претендует на звание первой в истории мироздания журналируемой файловой системы. Хотя, как говорилось в предыдущем разделе мне известно, JFS – эпоним всех журналируемых ФС – в своей реализации для AIX появилась в 1990 году, так что вопрос приоритета в отношении журналирования остаётся не вполне ясным.

VxFS является основной файловой системой в HP UX, работает также во всех ныне живущих проприетарных UNIX'ах и теоретически может применяться и в Linux'е. Однако о практических примерах такого применения, по крайней мере в не очень промышленной обстановке, я не слышал: VxFS является системой проприетарной и весьма дорогой.

VxFS тесно интегрирована с менеджером логических томов – VxVM. Благодаря чему в ней возможно изменение (в любую сторону) размера файловой системы «на лету», включение различных режимов использования томов – стриппинг данных, их зеркалирование, а также комбинации того и другого, создание избыточных массивов по типу RAID Level 5, изменение внутренней организации данных без остановки работы. Всё это позволяет VxFS (в сочетании с VxVM) претендовать на звание комплексной системы размещения данных.

Впрочем, не меньше к тому оснований было и у AdvFS – файловой системы, разработанной к 1993 году фирмой DEC для своего проприетарного варианта UNIX, именовавшегося сначала OSF/1, затем Digital UNIX, и завершившего свою жизнь под именем Tru64 UNIX. Судьба её была печальной. Снискав заслуженное признание на своей родной платформе DEC Alpha под управлением указанной ОС, она после покупки DEC фирмой Compaq оказалась в загоне. А после того, как Compaq, в свою очередь, был поглощён фирмой Hewlett Packard, использовавшей для своего UNIX'а на платформах HP PA и Itanium только что упомянутую VxFS, AdvFS оказалась совсем не при делах.

В результате HP сделала щедрый дар сообществу свободного софта вообще и Linux-сообществу в особенности: в середине 2008 года исходники файловой системы AdvFS были открыты под лицензией GPLv2 – ради максимальной совместимости с ядром Linux. С предложением использовать их в этой ОС если не в качестве

системной целостности, то как богатую технологическую базу. Правда, оговорка, что сама HP не заинтересована в дальнейшем развитии AdvFS, заставляла опять вспомнить народную присказку: «Возьми, боже, что мне не гоже».

Да и предложение несколько запоздало: как мы скоро увидим, к тому времени интенсивно развивались и ZFS, и Hammer, и btrfs.

Однако, помимо исходников, HP предоставила также доступ ко всей документации – благодаря чему об AdvFS при желании можно узнать больше, чем о любой другой проприетарной файловой системе для UNIX-подобных операционок. Это избавляет меня от необходимости описания её особенностей. Замечу только, что среди них мы увидим все черты развитой комплексной системы размещения данных. Те самые, которые мы наблюдаем при рассмотрении устройства, например, ZFS. К обзору истории которой и пора перейти.

Начало истории ZFS

Разработчики ZFS поставили себе честолюбивую цель: создать систему хранения данных, которая отвечала бы всем трем критериям сформулированного ранее идеала. Разработка её проводилась в компании Sun Microsystems, командой под руководством Джеффа Бонвика (Jeff Bonwick) и Мэттью Аренса (Matthew Ahrens). Первоначально название ZFS рассматривалось как аббревиатура от Zettabyte File System, но быстро стало просто условным именованием. Его можно интерпретировать как последнюю точку в развитии файловых систем вообще. И в последующем мы увидим: это недалеко от истины.

Результаты работы над ZFS были представлены миру в августе 2004 года. А в 2006 году она была включена в штатный состав OS Solaris 10 (релиз 6/06). То есть, подобно своим предшественницам, она также была проприетарным продуктом. И пользователям свободных UNIX-подобных систем поначалу от её существования было ни холодно, ни жарко. Однако период камерного существования ZFS продолжался недолго – уже в ноябре 2005 года, то есть до включения в Solaris, её поддержка была интегрирована в открытый её вариант, OpenSolaris. Ибо она основывалась на том же ядре SunOS 5, что и коммерческий прототип.

Исходники ZFS распространяются, как и собственно OpenSolaris, под лицензией CDDL (Common Development and Distribution License). Эта лицензия, базирующаяся на Mozilla Public License (MPL), не влияет на общую лицензию проекта, в состав который включены CDDL-компоненты. И потому оказывается совместимой с большинством свободных лицензий. За исключением... какой? Правильно, GPL во всех её проявлениях.

Разумеется, ZFS была задействована в клонах openSolaris, таких, как BeleniX, SchilliX и, в первую голову, в Nexenta OS. Правда, последняя развивалась в направлении коммерческой системы хранения данных, а о числе пользователей остальных можно было только гадать.

Некоторое время ZFS была доступна пользователям Macintosh'a – в Mac OS X Leopard от осени 2007 года. Правда, ходившие перед её выходом слухи, что она будет там файловой системой по умолчанию, оказались несколько преувеличенными: поддержка ZFS оказалась опциональной и лишь в режиме «только для чтения». А в последующих версиях ОСей семейства кошачьих вообще исчезла и, видимо, уже не возродится.

Так что для широких народных масс ZFS по-прежнему оставалась недоступной. Пока... пока её не портировали под FreeBSD в 2007 году, и официально не включили её поддержку в 7-ю версию этой ОС, вышедшую в начале 2008 года. В чём, как и в дальнейшем её развитии, основная заслуга принадлежит Павлу-Якубу Давидеку (Pawel Jakub Dawidek) и Ивану Ворасу (Ivan Voras). Правда, до недавнего времени ZFS нельзя было задействовать при установке FreeBSD средствами её штатного инсталлятора и конфигуратора sysinstall. Однако это без труда можно было осуществить в дальнейшем руками. В том числе и разместить на ZFS корень файловой иерархии. А с выходом FreeBSD версии 10.0 она стала доступна применителям этой ОС, что называется, «из коробки».

С самого начала поддержки ZFS во FreeBSD появилась и возможность задействовать её, что называется, «искаропки», в десктоп-ориентированном клоне последней – PC-BSD. А с переходом FreeBSD, начиная с версии 9.0, на новую программу установки – BSDInstall, эта функция распространилась и на материнскую систему.

Успех ZFS во FreeBSD, где она стала если не главной файловой системой, то добилась равноправия с UFS2, послужил примером для

других BSD-систем. Так, ныне ZFS поддерживается в NetBSD – эта работа была начата Оливером Голдом [Oliver Gould] летом 2007 года в рамках акции Google Summer of Code. А в 2009 году Адам Хамсик [Adam Hamsik] интегрировал её код в ядро NetBSD. Правда, насколько я понимаю, использование ZFS в этой операционке рекомендуется только в экспериментальных целях.

Наконец, одно время в списках рассылки DragonFlyBSD активно обсуждался вопрос о портировании ZFS и на эту операционку. Потом, правда, разговоры эти стихли – вероятно, в связи с активной разработкой файловой системы Hammer, обладающей во многом аналогичными возможностями. Однако, учитывая лёгкость адаптации к DragonFlyBSD любых сторонних файловых систем, можно не сомневаться, что поддержка ZFS на уровне обмена данными будет включена в неё тогда и если (или если тогда), когда (и если) это кому-то понадобится.

Таким образом, пользователям большинства BSD-систем ZFS или уже доступна как нативная, или может стать доступной в ближайшее время.

Из истории юриспруденции

А что же Linux, спросите вы меня? Как обстоит дело с поддержкой ZFS в самой массовой из свободных UNIX-подобных операционных систем нашего времени? А вот с Linux'ом все оказывается гораздо сложнее. Ибо не зря поминали мы выше лицензию CDDL. Которая сама по себе очень даже свободная, и не накладывает почти никаких ограничений на распространение защищаемых ею программ.

В частности, не запрещает CDDL и коммерческого распространения производных продуктов в виде бинарников, без открытия исходных текстов. Как известно, не накладывает такого ограничения и лицензия BSD, почему включение кода поддержки ZFS в любые BSD-системы и проходит юридически безболезненно, как мы только что видели на примере FreeBSD.

А вот с лицензией GPL обеих актуальных версий (v2 и v3) CDDL входит в диалектическое противоречие. Ибо любые продукты, производные от программ под GPL, вне зависимости от формы

распространения, должны сопровождаться исходными текстами. Что делает юридически невозможным включение кода поддержки ZFS непосредственно в ядро Linux, распространяемое, как известно, на условиях GPLv2.

Кроме того, невозможность включения в ядро Linux кода поддержки ZFS объясняется тем, что GPL требует распространения всех основанных на ней продуктов под GPL же, тогда как CDDL – сохранения её для «своих» компонентов.

Правда, часть кода ZFS была открыта под GPL с тем, чтобы соответствующий патч можно было включить в загрузчик Grub. Это обеспечило возможность загрузки Open Solaris непосредственно с ZFS-раздела. Однако оказалось недостаточным для полноценной реализации этой системы, которую можно было бы распространять под данной лицензией.

Впрочем, не будучи юристом, ломать голову над лицензионными вопросами не буду, и моим читателям не советую, ибо понять это всё равно невозможно. А достаточно лишь запомнить, что всеми резонными и юридически подкованными людьми признано, что поддержки ZFS в ядре Linux быть не может.

Таким образом, сложилась абсурдная, с точки зрения здравого смысла, ситуация: два программных продукта под свободными лицензиями (обсуждать вопрос, какая из них «свободней другой», мы сейчас не будем), созданные друг для друга, как Huggies и... э-ээ... место пониже спины (дальнейшие события показали, что технических сложностей при портировании ZFS на Linux практически нет), невозможно было использовать в составе одного проекта. По крайней мере, для законопослушных граждан, чтущих... нет, не уголовный кодекс, а принципы свободного программного обеспечения.

И, разумеется, здравомыслящие люди попытались эту ситуацию разрешить. И первая такая попытка была предпринята ещё в 2006 году в рамках Google Summer of Code. Основывалась она на поддержке ZFS через FUSE (Filesystem in Userspace). Поскольку модуль FUSE работает как пользовательское приложение, необходимости во включение кода ZFS в ядро Linux нет, что снимает все юридические вопросы. Однако встают вопросы другие – производительности и устойчивости.

Проект ZFS-FUSE развивается по сей день, хотя и не очень быстрыми темпами. Правда, находясь в стадии хронической бета-версии, он до сих пор рассматривается как сугубо экспериментальный. Да и в любом случае в таком виде ZFS выполнять свои функции – быть надёжным хранилищем данных большого объёма – скорее всего, не сможет.

Так что ZFS-FUSE нельзя считать кардинальным решением вопроса с этой системой размещения данных в Linux. А на то, что в его ядро будет встроена собственная реализация ZFS, рассчитывать не приходится.

ZFS on Linux: технология против крюкотворства

И тем не менее, решение этой проблемы нашлось – и решение столь же изящное, сколь и очевидное. Его предложил весной 2010 года Брайан Белендорф (Brian Behlendorf), некогда один из основных разработчиков web-сервера Apache. Он создал модуль поддержки ZFS, который собирается и может распространяться отдельно от ядра, сохраняя прародительскую лицензию CDDL. А поскольку последняя, как уже говорилось, является лицензией «пофайловой», этим самым обходится антагонистическое противоречие – запрет на распространение продуктов, в которых смешан код, лицензируемый под CDDL и GPL.

На базе разработки Брайана возникло сразу два проекта. Первый осуществлялся индийской компанией KQ Infotech, которой уже в сентябре 2010 года удалось выпустить работоспособный, пригодный для тестирования патч Linux-ядра с реализацией файловой системы ZFS. А в январе следующего, 2011, года появилась финальная его версия, доступная тогда в исходниках и в виде двоичных пакетов для Fedora 14, RHEL6, Ubuntu 10.04 и 10.10.

Однако весной того же года KQ Infotech была куплена фирмой STEC, занимающейся производством SSD-накопителей, каковых, впрочем, в наших палестинах никто не видел. И работы по дальнейшему развитию нативной поддержки ZFS были свёрнуты. Хотя исходники модуля и сопутствующих компонентов до сих пор доступны, последнее их обновление происходило около назад. А информации о дальнейшей судьбе проекта с тех пор не появлялось.

Однако сам Брайн продолжал свою работу – вместе с сотрудниками Ливерморской национальной лаборатории, каковая, будучи в подчинении Министерства энергетики США, занимается не только вопросами ядерного оружия (эвфемизмы вроде Минсредмаша в ходу не только в бывшем Советском Союзе), но и разработкой суперкомпьютеров. В результате скоро возник проект ZFS on Linux, в рамках которого модуль поддержки ZFS и сопутствующие утилиты поддержки, портированные из Solaris – так называемый SPL (Solaris Porting Layer), были доведены до ума, и к началу 2011 года стали пригодны для использования в экспериментальном режиме. А к настоящему времени, несмотря на формальное сохранение статуса *release candidate*, порт ZFS on Linux можно считать готовым к практическому применению.

Правда, майнтейнеры основных дистрибутивов не торопились включать поддержку ZFS в свои системы даже в качестве дополнительных неофициальных пакетов. Подозреваю, что не столько из косности и лени, сколько из-за очередной сложности: видимо, по всё тем же лицензионным ограничениям модули `zfs` и `spl` приходится привязывать к фиксированной версии (и даже конкретной сборке) ядра Linux. Что, при регулярных, даже корректирующих, обновлениях последнего требует и их пересборки.

Тем не менее, разработчики проекта воплотили результаты своей работы в виде дополнительного (так называемого PPA) репозитория для Ubuntu. А также сочинили подробные инструкции по собственноручной сборке пакетов в форматах RPM и Deb (ссылки можно найти на странице проекта).

Достаточно подробно включение ZFS описано в Gentoo Wiki. А майнтейнеры её клона, дистрибутива Sabayon, прославившиеся своей склонностью к экспериментам, включили поддержку ZFS почти «искаропки»: соответствующие модули подгружаются при старте с LiveDVD и могут быть опробованы в «живом» режиме. Хотя штатного способа установки системы на ZFS в инсталляторе этого дистрибутива, всё из-за тех же юридических заковык, и не предусмотрено. Но нет и причин, препятствующих любому благородному дону установить этот дистрибутив на ZFS в любом виде, хоть и корня файловой иерархии. Если ему этого хочется, конечно.

Часть II. История дистрибутивов

Глава одиннадцатая. Начало дистрибуции

Прошлую статью я завершил на обсуждении вопроса, что же такое придумал Линус, и не GNU ли его Linux. В религиозные вопросы по сему поводу вдаваться не будем. А лучше посмотрим, что же именно Линус придумал (не считая метода разработки, который придумал Том Сойер).

Был ли дистрибутивом первозданный Linux?

Общеизвестно, что Линус придумал ядро операционной системы имени... нет, не и минеральных источников, а имени себя. И это правда, чистая правда – но не вся правда. Потому что Линус придумал ещё и файловую систему ext (расширение файловой системы MINIX, которая позднее воплотилась в ext2). Кроме того, им или с его подачи был разработан набор низкоуровневых утилит для работы с ядром, его модулями, файловой системой ext – он получил имя linux-utils. Наконец, в рамках реанимированного Линусом метода Тома Сойера Вернер Альмесбергер разработал загрузчик ядра Linux – Lilo (Linux LOader), который затем, до появления GRUB'a, успешно выступал в качестве мультисистемного.

Именно этот комплекс, (почти) способный к самостоятельному существованию, и можно назвать операционной системой Linux в самом узком смысле слова. Однако он существовал не в безвоздушном пространстве. Ибо, с одной стороны, требовал средств управления – им, в силу некоторых причин, стала командная оболочка bash. А с другой – его требовалось чем-то собирать, и в этом качестве выступил компилятор gcc вместе с набором сопутствующих ему инструментов (binutils, make и так далее). И то, и другое было разработано в рамках проекта GNU – что и служит по сей день основанием для именования нашей ОС как GNU/Linux.

Подобно первозданному UNIX'у, Linux изначально являлся типичной «системой для себя». Более того, исходно единственным его назначением была разработка самого же себя – никаких других целей Линус перед собой поначалу не ставил. Да и первые пользователи

Linux'а устанавливали (точнее, собирали) систему для того, чтобы её изучать и, по возможности, совершенствовать. Так что ни в каких дополнительных компонентах, кроме ядра, утилит оформления и инструментария для их сборки, необходимости не возникало.

Установка Linux в те «времена старинные, теперь почти былинные» была задачей не вполне тривиальной даже для опытного компьютерщика (а иные его и не пользовали). И в формирующемся тогда же Linux-сообществе возникла идея облегчить им эту процедуру. В результате чего родилось понятие дистрибутив Linux. Это – система комплектации ядра ОС и его оформления дополнительными программами и способ её распространения. Она предполагает наличие программы-установщика и средств управления пакетами, то есть теми самыми дополнительными программами.

И уже через несколько месяцев после обнародования Линусом исходников первой (0.01) версии своего ядра, в начале 1992 года, появляются первые наборы программ, которые можно считать прототипами позднейших дистрибутивов Linux – MCC Interim Linux и TAMU. Они представляли собой комплекты разработчика, включающие в прекомпилированном виде ядро, командную оболочку, компилятор со средствами сборки, а также основные пользовательские утилиты, что позволяло развернуть работоспособную систему на «чистой» машине, не несущей никакой иной ОС.

Начало начал

В итоге в октябре 1992 года на свет появляется комплект, который можно назвать первым в истории настоящим дистрибутивом Linux. Он носил имя SLS (Softlanding Linux System) и был разработан Питером Мак-Дональдом. Помимо ядра Linux и утилит оформления, дистрибутив SLS включал в себя оконную систему X и средства работы с сетью, то есть был уже вполне пригоден для конечного пользователя. Правда, не следует забывать, что конечными пользователями Linux в те годы были исключительно его же разработчики.

Дистрибутив SLS просуществовал недолго – последняя его версия вышла в 1994 году. Однако он лёг в основу целой линии дистрибутов,

протянувшейся в наши дни яркой нитью, и потому о нём стоит сказать подробней.

Дистрибутив SLS распространялся преимущественно на трехдюймовых дискетах, в количестве 20-30 штук. Образы дистрибутивных дискет можно было получить по Сети (у нас – практически только по служебным каналам), а также заказать на CD (хотя CD-приводы в то время на пользовательских машинах были не меньшей экзотикой, чем Интернет на дому).

Одной из знаковых особенностей SLS была схема инициализации в BSD-стиле – хотя в дальнейшем в большинстве дистрибутивов майнстрима возобладал стиль System V, которую Линус заимствовал из первозданного UNIX'a.

Формат бинарных пакетов в SLS был предельно прост – tar-архив, компрессированный с помощью Gzip или compress, возможно – с постинсталляционным сценарием. Для установки и удаления пакетов использовалась утилита sysinstall – предтеча всех последующих систем пакетного менеджмента. Которая не только разворачивала архив и инкорпорировала его компоненты в файловую систему, но и фиксировала его в специальной базе данных – на предмет последующего удаления, если таковое потребуется. Хотя о контроле зависимостей тогда речи ещё и не возникало.

Прекращение разработки SLS связывается с его переходом на формат бинарных файлов ELF вместо общепринятого тогда в Linux и вообще в UNIX формата a.out. Хотя ELF был более «прогрессивен», нежели a.out, тогда это оказалось шагом преждевременным. Но, возможно, дело было просто в потере интереса разработчика к своему производству – ситуация, с которой мы ещё не раз столкнёмся при знакомстве с историей Open Source.

О SLS ныне мало кто помнит, однако роль его в дальнейшем дистроении трудно переоценить: именно он лёг в основу старейшего дистрибутива из числа доживших до наших дней – Slackware.

Slackware: первый шаг к Linux'у для всех

Итак, дистрибутив SLS умер. Но душа его жила. Ещё в период его активного развития Патрик Фолькердинг принял SLS за основу своей Linux-системы, названной Slackware, первая версия которой была

обнародована 17 июля 1993 года и с тех пор успешно развивается по сей день.

Именно со Slackware началась и история Linux-дистрибуции в организационном, так сказать, аспекте. Сразу же после своего появления Slackware, помимо обычных сетевых каналов, начала распространяться на CD известной медиа-фирмой Walnut Creek.

Slackware в своём внутреннем устройстве унаследовала первозданную простоту SLS. И не только унаследовала – именно простоту Патрик возвёл в основополагающий принцип построения системы. Реализация его выразилась в сохранении BSD-стиля инициализации, простого формата пакетов, и «идеологически обусловленного» отказа от контроля их зависимостей.

Создававшиеся чуть позже (но в масштабах эпохи – практически одновременно) дистрибутивы Debian и Red Hat пошли по прямо противоположному пути: всё более усложняющаяся со временем схема инициализации в стиле System V, включение максимально большого количества метаданных в структуру пакетов и все более изощрённые формы контроля их зависимостей.

Новшествами Slackware были:

- собственная программа инсталляции – меню-ориентированная, работающая в псевдографическом режиме, похожая по виду и родственная по духу создававшейся в то же самое время утилите sysinstall из FreeBSD;

- выделение категорий пакетов – базовой системы (A), консольных приложений (AP), средств разработки (D), оконной системы X и ее приложений (X и XAP, соответственно), и так далее;

- набор утилит для управления индивидуальными пакетами, не предусматривающего, однако, никакого контроля зависимостей.

Время показало провиденциализм подхода Патрика – Slackware живёт и развивается вот уже 15 лет, не поступаясь своими принципами, сохраняя редкую по нынешним временам компактность, лишь обновляя версии ядра, компоненты базовой системы и приложений. Сохраняется и устойчивый круг пользователей этого дистрибутива.

Отступление. Многие линуксоиды моего и более старшего поколения начинали свою дорогу в Linux со Slackware – и ничуть об этом не жалеют, вне зависимости от того, какие дистрибутивы бы они

не использовали в дальнейшем. Знакомство с этой системой дает совершенно неоценимый опыт, позволяющий найти пути для решения любых проблем в любых других дистрибутивах. И потому крылатая фраза «изучая Slackware, ты изучаешь Linux» имеет под собой все основания.

Исходный «примитивизм» формата пакетов Slackware не мешает использовать в этом дистрибутиве весьма развитые средства пакетного менеджмента, обеспечивающие учёт и автоматическое разрешение зависимостей. Поскольку внутри самих пакетов никакой метайнформации не содержится, для хранения её можно использовать любую внешнюю базу.

Этой особенностью формата пакетов Slackware активно пользуются все, у кого возникает к тому желание или необходимость, приспособивая к ней любые системы управления пакетами. Так, мне доводилось слышать об удачных попытках применения в Slackware системы портов, заимствованной из FreeBSD. Для Slackware поддерживается система `pkgsrc` – портообразная система, разработанная первоначально для NetBSD. На базе синтеза Slackware и `pkgsrc` активно развивается несколько дистрибутивов, например, Voltalinux и Draco GNU/Linux.

Механизм `apt-get`, обеспечивший славу Debian, а в дальнейшем немало способствовавший и популярности семейства Ubuntu, также был адаптирован для использования в Slackware: здесь он получил название `slapt-get`. На основе синтеза Slackware и пакетного менеджера `raspm`, происходящего из Archlinux (кстати, во многом – идейного наследника Slackware), возник дистрибутив Frugalware.

Разработан был для Slackware и собственный менеджер пакетов, `swaret`. Однако в штатный комплект дистрибутива он, вследствие заветов Патрика, так и не вошёл. Потому что для собственно Slackware в качестве основных средств управления пакетами Патрик по прежнему признает только два: установку официальных бинарников с помощью штатных утилит, при ручном разрешении зависимостей, и компиляцию из исходников – в случае, если нужных пакетов нет в репозитории или их официальная сборка по тем или иным причинам не удовлетворяет пользователя. Всё остальное – от глюкавого. То есть дистрибутив этот должен рассматриваться не столько как законченная

система, сколько как каркас для конструирования системы собственной.

Debian: второй шаг к пользователю

Дистрибутиву Slackware не долго пришлось оставаться в гордом (почти) одиночестве на своём тернистом пути к пользователю. Скоро этот путь пришлось делить на троих – сначала с Debian, а затем и с Red Hat.

История дистрибутива Debian особенно интересна по четырём причинам. Первая – ее длительность и насыщенность событиями, давшими большое количество инноваций, аккумулированных мировым дистрибутированием. Вторая причина – Debian стал основоположником универсалистской тенденции в дистрибутировании, стремившейся охватить весь океан свободного софта. Третья – Debian стал прародителем максимального количества активно развивающихся клонов. И, наконец, четвертая – некоторые из его потомков сравнились по популярности с родителем, а один (Ubuntu) даже превзошёл его (не это ли мечта всех родителей?).

Debian – или, точнее, Debian GNU/Linux, разработчики настаивают именно на таком его именовании, – был создан в 1993 году Яном Мёрдоком (Ian Murdock), и его название образовано сочетанием имен его жены Деборы (Debora) и самого автора – в то время он был студентом Университета Пэрдью (Purdue). Однако очень быстро вокруг Debian выросло сообщество пользователей и разработчиков, и проект приобрёл общественное значение.

Основной идеей раннего (1993– 1995 гг.) Debian были – модуляризация авторских пакетов, сборка этих модулей в качестве дистрибутивных пакетов с детальным описанием их зависимостей, утилита dpkg для управления оными в масштабе одного отдельно взятого пакета. И, под занавес первого акта, dselect – первая система пакетного менеджмента, достойная претендовать на звание именно системы и представляющая собой front-end к dpkg, обеспечивающий автоматическое разрешение зависимостей и установку целевых наборов пакетов. Эти тенденции получили развитие в дальнейшем.

Универсализм Debian проявился на следующем этапе его развития, начиная с 1996 года, когда Яна, ушедшего после окончания

университета на службу мировому капиталу (в компанию Progeny), на посту лидера проекта сменил Брюс Перенс – известный адепт Open Source, автор многочисленных публикаций на эту тему и, по совместительству, – тогда еще и немалый чин в компании Pixar. Каковая, к слову сказать, поучаствовала и в поддержке проекта Debian – в том числе, как мы увидим чуть дальше, и идеями.

Будучи по долгу службы связанным со всякого рода мультимедийным софтом, Брюс резонно решил включить его в состав дистрибутива. А там уже был один шаг до того, чтобы инфраструктура Debian охватила практически все изобилие свободного программного обеспечения. Так что с тех пор количество пакетов в дистрибутиве нарастало от версии к версии чуть ли не в геометрической прогрессии, и к настоящему времени с трудом поддаётся исчислению.

Кроме того, на время лидерства Брюса пришлась разработка документов **Принципы Свободного Программного Обеспечения Debian** и **Общественный контракт Debian**, а также создание принципов контроля качества включаемых в дистрибутив пакетов. Наконец, именно он предложил систему кодовых имён версий дистрибутива (Potato, Woody и другие) – это были персонажи из мультфильма Toy Story, выпущенного компанией Pixar (вот оно, идейное воздействие масс-медиа!). При этом имя Sid, которое носил соседский мальчик, портящий игрушки, было навсегда закреплено за разрабатываемой версией – как символ того, что разработчики новой версии программы в процессе своей работы подчас вынуждены временно «подпортить» версию старую.

В период 1996– 1999 года Debian, в частности, благодаря политике контроля качества, завоевал признание как серверная платформа и система для технически грамотных пользователей (читай – в первую голову, для разработчиков). При этом он счастливо совмещал в себе качества «системы для себя» и «системы для всех». Первая сторона вопроса обеспечивалась программой `dpkg`, вторая же стала возможной благодаря ее надстройке – `dselect`.

Одновременно продолжали развиваться универсалистские тенденции дистрибутива – не только вглубь океана Open Source Software, но и вширь – переносясь на архитектуры, отличные от i386. В интервале 1996– 1999 года Debian был портирован на платформы

68XXX, Alpha, затем – Sparc и PowerPC, Intel64 (так называемый Merced) и AMD64.

Важнейшей, наверное, вехой в развитии Debian (и не только его) стал выпуск весной 1999 года версии 2.1 Slink (Slinky – это такая собачка из того же мультика). И судьбоносность ее определяется тем, что в нее впервые был включён apt – универсальный инструмент для управления пакетами, который и создал позднее условия для широкого распространения Debian-клонов.

Значение apt переоценить трудно – он не только был портирован в дистрибутивы, использующие формат пакетов rpm, не только послужил прообразом для многих других систем управления пакетами, претендующими на универсальность (yum, urpmi), но и оказался своего рода связующим звеном между пакетными дистрибутивами и системами Source Based, поскольку обеспечивал не только установку бинарных пакетов, но и их построение (вплоть до тотальной пересборки системы, подобно сакраментальному make world из FreeBSD). Впрочем, все это стало ясно много позднее, по крайней мере, широким пользовательским массам.

Не случайно именно к 1999 году относятся первые попытки создания на базе Debian Систем Быстрого Развёртывания, таким, как Storm Linux и Corel Linux. Но это история, до которой мы доберёмся ещё не скоро.

Red Hat: совсем для всех?

Если Slackware продолжил исконно UNIX'овую традицию систем для себя, а Debian являет собой первый пример дистрибутива, развиваемого сообществом и для сообщества, то следующим шагом дистрибуции стало создание дистрибутива, претендующего быть системой для всех.

Ибо тем временем обозначилась первая сфера практического применения Linux за пределом круга разработчиков программного обеспечения – сервера сетевых служб, в том числе – web-сервера. Это вызвало к жизни вторую волну дистрибутивов (правда, по времени она практически пересеклась с первой – но в те героические годы счет вёлся на месяцы, если не на дни). И первой ласточкой её стал Red Hat, который и создавался как дистрибутив «для всех» – хотя, конечно, под

понятие «все» тут попадали в первую очередь администраторы компьютерных сетей (время Linux'а для конечного пользователя ещё не пришло). Но важно, что Red Hat представлял собой не набор для конструирования собственной системы, как Slackware (да и Debian в те годы, до разработки apt, также скорее предполагал собственное конструирование, нежели готовое решение), а попытку создания системы, работающей «из коробки».

Было время на Руси, когда

Говоришь Linux – подразумеваешь Red Hat. Говоришь
Red Hat – подразумеваешь Linux.

Дистрибутив Red Hat – третий в ряду ныне живущих патриархов дистрибутов, после Slackware и Debian (хотя, повторяю, приоритет тут исчисляется первыми месяцами). Он разрабатывается с 1993 года, в октябре 1994 года появилась первая общедоступная бета-версия, а в мае 1995 года – первый официальный релиз.

В отличие от Slackware, созданного и развивавшегося кустарём-одиночкой с персональным компьютером, и Debian, вокруг которого быстро сложилось сообщество разработчиков, за Red Hat с самого начала его разработки стояла одноимённая коммерческая компания. Основали её Боб Янг (Bob Young) и Марк Юинг (Marc Ewing) в 1993 году, имея целью поставить свободное слово на службу мировому капиталу.

Происхождение названия дистрибутива (красная шляпа) и соответствующего логотипа объясняют тем, что Юинг в студенческие годы рассекал по колледжу в дедушкином шапо соответствующего колёру. Хотя Янг объясняет его тем, что красный цвет в дзэн-буддизме символизирует всякие хорошие качества.

Это была первая попытка монетизации свободного софта: сам дистрибутив распространялся свободно, в соответствии с лицензией GPL, и бесплатно (по цене носителей и доставки), деньги же предполагалось извлекать из технической его поддержки. А поскольку оплачивать таковую обычно готовы не частные лица, а организации, то Red Hat изначально был ориентирован на корпоративную сферу – во-первых, и на дружелюбие к пользователю – во-вторых.

Правда, следует учесть, что хотя шире стал круг пользователей, но связь их с народом не стала особенно ближе: расширился он за счёт сисадминов, до приобщения к Linux'у пользователей «простых» оставалось ещё несколько лет.

Настоящие сисадмины, как известно, люди занятые – с одной стороны, и ленивые – с другой. Возиться с конструированием системы им обычно не с руки. И потому Red Hat с самого начала отличался простотой установки – максимально возможной в те времена.

Значительную роль в упрощении процедуры установки и поддержки сыграл формат пакетов RPM (что тогда расшифровывалось как Red Hat Package Manager) и одноимённая утилита для манипулирования такими пакетами, способная отслеживать зависимости и сообщать об их нарушении (но ещё не разрешать их автоматически). По сравнению с молчаливым пакетным инструментарием из Slackware, способным установить неработоспособную, из-за нарушения зависимостей, программу, это казалось большим прогрессом.

Происхождение системы rpm (будем понимать под этим и набор утилит, и формат пакетов, с которыми они работают) теряется во мраке веков. В первых версиях Red Hat использовалась система RPP, обеспечивающая установку пакетов одной командой, проверку зависимостей и запрос информации о них. Однако сборка пакетов для неё требовала существенной модификации исходников, что было напряжно для разработчиков.

Параллельно раннему Red Hat некоторое время развивался дистрибутив Bogus, ныне мало кому известный. В нём имелась собственная пакетная система – PMS (Package Management System), написанная Рикардом Файтом (Rikard E. Faith). Она обладала слабым механизмом запросов информации о пакетах, а проверка их зависимостей просто отсутствовала. Но зато пакеты для PMS можно было собирать непосредственно из исходников, без всякой их модификации.

В ходе подготовки 2-го релиза Red Hat Рикард Файт вместе с Дугом Хоффманом (Doug Hoffman) по контракту с поименованной компанией написали систему PM, вобравшую в себя лучшие особенности RPP и PMS. Хотя практически она так и не была задействована, но послужила одной из основ для RPM.

Собственно система RPM была создана Марком Юингом и Эриком Троэном (Erik Troan), основываясь на всех достижениях предшественников – RPP, PMS и PM. Вариант её, подготовленный для тестовых версий второго релиза, быстроты ради был написан на Perl'е, что создавало ряд проблем, например, при загрузке с дискеты (а в те времена это было достаточно обычным способом старта Linux'а). И непосредственно к выходу релиза Red Hat 2.0 система была полностью переписана на C, база данных пакетов перепроектирована для пущей надёжности и быстродействия, и создана библиотека rpmlib для использования функциональности RPM сторонними разработчиками. Иными словами, система RPM приобрела практически тот вид, в каком мы знаем её ныне, подвергаясь с тех пор только корректировке ошибок и косметическим доделкам.

Система RPM и одноимённый формат, став штатными и общедоступными в релизе Red Hat 2.0, вышедшей в сентябре 1995 года, сразу завоевали популярность и вне родительской системы. Вскоре они были использованы в Caldera Linux, Suse, Mandrake и многих других – и об этом будет говориться в следующих сериях.

Глава двенадцатая. Лицом к пользователю

Red Hat, проторивший путь дистрибутивам для всех с коммерческой поддержкой, не долго оставался на нём одиноким: вскоре у него появились последователи. И первым среди них стала Suse.

Suse: по стопам Red Hat

Дистрибутив Suse изначально назывался S.u.S.E. – эту аббревиатуру производят от имени компании, разработчика и распространителя его (Gesellschaft für Software- und System-Entwicklung), хотя в источниках можно встретить и другие трактовки его имени. Что, впрочем, не важно – форма его несколько раз менялась (S.U.S.E, SUSE, Suse), а в конце концов в чистом виде оно просто вышло из употребления.

Компания S.u.S.E. начинала свою деятельность с распространения уже имевшихся дистрибутивов – SLS и Slackware, а также издания руководств по UNIX и Linux. Однако вскоре ей захотелось иметь дистрибутив собственный, каковой и был создан на имеющейся кодовой базе названных систем. Позднее этот путь станет типичным для большинства коммерческих Linux-компаний.

Suse, как коммерческий продукт вышеозначенной фирмы, начал развиваться практически одновременно с Red Hat, в 1994 году. Сначала это был самый обычный дериват Slackware с германской локализацией и прикрученной универсальной системой установки и конфигурирования – YAST, призванной облегчить труд администратора, сведя его к простановке галочек в соответствующих полях.

Таким образом, Suse стал первым дистрибутивом со сквозной системой установки и автоматизированного конфигурирования. При этом ручные настройки путем прямой правки конфигов – единственный способ доведения до ума материнской Slackware, – отменялись напрочь: попытки ручного вмешательства в Suse влекли в те годы последствия непредсказуемые.

Постепенно Suse всё больше отдалялась от родительской системы, заимствуя многие особенности Red Hat. В частности, в неё были внедрены формат пакетов RPM и система инициализации в стиле System V (в Slackware, как известно, исконным был BSD-стиль инициализации). И в результате уже через пару-тройку лет Suse стала настолько своеобразной системой, что и сам папа Патрик не догадался бы о её происхождении.

Ещё одной особенностью Suse стало повышенное внимание к графической системе, место которой в Linux'е 'к тому времени практически безраздельно заняла свободная инкарнация Иксов — XFree86. Тесные контакты с разработчиками последней обеспечил дистрибутив поддержкой самых современных тогда видеочипов — в те времена для каждого графического чипа (или серии близких чипов) предназначался собственный X-сервер.

Свидетельствую как очевидец: запустить Иксы на считавшихся тогда крутыми видеокартах типа ATI Match32/64 в Slackware или Red Hat часто удавалось только после прикручивания к ним соответствующего X-сервера, вытасченного из Suse.

Бизнес-модель Suse строилась несколько по иной схеме, нежели у Red Hat. В частности, этот дистрибутив включал в себя ряд закрытых проприетарных компонентов (в первую очередь — ту же систему YAST). И в «полноразмерном» виде бесплатно не распространялся — для свободного скачивания была доступна функционально ограниченная версия. Она же распространялась первыми системами онлайн-торговли по цене носителя и доставки. А полноценный дистрибутив в коробочном исполнении продавался за немалые по масштабам тех лет деньги — от ста долларов и выше. Однако он сопровождался объёмной (и очень качественной) печатной документацией, которая сама по себе составляла львиную долю стоимости коробки, и атрибутикой — в частности, значком хамелеона, исполненном как правительственная награда.

Видимо, непосредственная продажа дистрибутивов и составляла основную статью дохода фирмы. Хотя и от технической поддержки в корпоративных масштабах она тоже, насколько я знаю, не отказывалась.

В скором времени Suse стал дистрибутивом номер один не только в Германии, но и практически по всей Европе, заняв на этом

континенте ту же нишу, что и Red Hat в Америке. И жил долго и счастливо, пока не был куплен фирмой Novell — но тогда это было делом далёкого будущего.

Caldera: предыстория грядущего

Лавры Red Hat не давали покоя майнтейнерам и на Американском континенте. И из всех, возделавших их в то время, наибольшая удача выпала на долю дистрибутива Caldera — правда, удача, так сказать, посмертная, в виде памяти, и не очень лестной. Хотя как раз собственно дистрибутивом такая память и не заслужена. Для понимания этого нам опять придётся вернуться назад и даже свернуть в сторону — в мир проприетаризма. А именно — к истории компании Novell, которая уже фигурировала в нашем рассказе в связи с освобождением FreeBSD.

Возникнув в 1979 году, компания эта на протяжении долгого времени с миром FOSS никак не пересекалась, разрабатывая и продвигая свою проприетарную сетевую систему Netware. И продвинула её настолько, что одно время Netware и корпоративные сети воспринимались как близнец-братья.

Но в начале 90-х годов прошлого века компанию, как некогда сиятельного Камильбека из «Повести о Ходже Насреддине», поразило хватательное рвение. Она в массовом масштабе скупала операционные системы (DR DOS — продвинутый аналог MS DOS), компоненты офисных пакетов — текстовый процессор WordPerfect и табличный процессор QuattroPro, получает права на распространение настольной СУБД Borland Paradox. В числе прочего ею были приобретены также права на исходный код и торговую марку UNIX (оставшиеся как бы бесхозными после разделения их прародителя — AT&T). На базе чего была создана система UNIXWare, но это части других историй — былой и грядущей.

«Офисное» направление деятельности Novell особого успеха не имело, как и её UNIX-бизнес. В результате один из основателей фирмы, Рэй Нурда (Ray Noorda), ушел в отставку, а все новоприобретения — распроданы в розницу: офисные пакеты — фирме Corel, где они составили пакет Corel Office, агония которого продолжается чуть не по сей день, UNIXWare — SCO (то есть Santa

Cruz Operations), дополнив её собственную систему, получившую отныне имя SCO OpenServer.

Рэй же Нурда основал новую компанию – Caldera Systems, которая, в частности, приобрела у Novell DR DOS. Однако основным направлением её деятельности стала Linux-дистрибуция.

Первоначально Caldera Linux представляла собой цельнотянутый Red Hat, дополненный некоторыми не вполне свободными компонентами, типа рабочего стола Looking Glas и средств интеграции с сетями Novell Netware. Однако вскоре этот дистрибутив обзавёлся собственным графическим инсталлятором (одним из лучших для своего времени) и вообще приобрёл своеобразие, отмеченное новым собственным именем – Caldera OpenLinux.

Свободная версия Caldera OpenLinux представляла собой очень компактный, аккуратно укомплектованный дистрибутив, широкому использованию которого препятствовали только слабые средства интернационализации. А коммерческая версия включала в себя немало проприетарных продуктов, вплоть до Wabi (средства эмуляции Windows от фирмы Sun), текстового процессора WordPerfect и векторной «рисовалки» CorelDRAW, работавших в режиме эмуляции, офисного пакета StarOffice (в то время ещё закрытого), и так далее.

Наконец, в первых годах нового тысячелетия Caldera становится одним из соучредителей альянса United Linux – наряду с Suse (тогда это была Европа), Turbolinux (Япония) и Connectiva (Бразилия); таким образом, альянс этот охватил чуть не все континенты (африканской Ubuntu тогда ещё и в проекте не было). Но это уже тоже начало иной истории.

А пока резюмируем: Red Hat, Suse и Caldera образовали к концу 90-х годов «могучую кучку», создавшую предпосылки для продвижения FOSS-систем в корпоративный сектор.

Магия Linux

Итак, на протяжении 1994-1997 годов дистрибутивы Linux обрастали «дружественными к пользователю» инсталляторами, средствами сквозного конфигурирования и пакетного менеджмента, включали в себя пользовательские, в том числе офисные, приложения. Предпринимались и первые попытки интернационализации. Однако от

конечного пользователя эти дистрибутивы оставались не менее далеки, чем декабристы с Герценом — от народа.

Впервые о Linux'е для конечного пользователя можно говорить, начиная с 1998 года, когда Гаэль Дюваль создал дистрибутив Mandrake (ныне – Mandriva). Основной его идеей было объединение Linux'а и графической среды KDE. Разработчики Mandrake были первыми, кто решился на такой шаг, не смотря на тогдашнюю неясность лицензионной политики в отношении библиотеки Qt, на которой KDE основывался: пуристы от FSF и, вместе с ними, основные майнтейнеры дистрибутивов, включая Red Hat, полагали лицензии этих программ не соответствующими идеалам свободного программного обеспечения. И попросту игнорировали единственную в те времена по настоящему интегрированную пользовательскую среду.

Собственно говоря, первая версия Mandrake представляла собой самый обычный Red Hat (от которого унаследовала номер версии – 5.1) и KDE, уже тогда имевшим большой набор штатных пользовательских приложений.

Второй особенностью Mandrake была попытка поддержки «из коробки» экзотического в те годы «железа», начиная от звуковых карт и кончая телетюнерами и цифровыми камерами.

Третья изюминка Mandrake – интернационализация: пожалуй, это был первый дистрибутив, декларировавший поддержку всех языков мира (что в итоге послужило предпосылкой для появления «русского Линукса»). Все это надолго приковало к Mandrake сердца пользователей, не имеющих прямого отношения к разработке программ или администрированию сетевых служб.

Роль Mandrake в приобщении первой волны конечных пользователей к миру Linux (и UNIX-подобных операционок вообще) переоценить трудно. В том числе и потому, что этот дистрибутив обеспечивал, более или менее удачно, возможность не только работать, но и развлекаться – не особо утруждая себя настройкой аппаратных средств. Не случайно в многочисленных опросах на форумах о первом в жизни дистрибутиве Mandrake (а теперь и Mandriva) уверенно лидировал вплоть до появления Ubuntu.

Таким образом, Mandrake можно считать практически первым дистрибутивом, поставившим во главу угла интересы конечного пользователя. Под его влиянием шаги в этом направлении

предприняли и ветераны дистрибуции, такие, как Red Hat и Suse – они обзавелись красивыми графическими инсталляторами, предлагавшими predetermined варианты установки типа пользовательской мультимедийной станции, офисной машины и так далее.

Немалую роль в обращении Linux'а к конечному пользователю (сиречь офисному и домашнему, профессионально не связанному с IT) сыграло директриса развития офисного пакета StarOffice. Созданный немецкой фирмой StarDivision первоначально для OS/2, он быстро был портирован на все существовавшие тогда платформы и операционки, претендовавшие, хотя бы чисто теоретически, на звание настольных – в том числе и на Linux. И хотя не являлся тогда ни открытым, ни свободным продуктом, был доступен, при определённых условиях, для бесплатного использования. А по своей функциональности, опять же со множеством оговорок, приближался (или стремился приблизиться) к MS Office, ставшему безраздельным властителем столов конторских служащих (Lotus Office и WordPerfect Office к концу тысячелетия уже отошли в область преданий).

Исторической правды ради нужно заметить, что StarOffice был не первым офисным пактом для Linux: этот титул по праву принадлежит пакету Applix одноимённой, также немецкой, фирмы. Каковой, правда, обладал рядом недостатков, в частности, не способен был, без хирургического вмешательства, работать с кириллицей. Да к тому же не был ни открытым, ни свободным, ни даже бесплатным.

Были и другие попытки создания программ офисного назначения. Тут можно вспомнить и Siag – прототип офисного комплекта, состоящий из текстового процессора и электронной таблицы, и простой монофункциональный текстовый процессор Ted, и Lux – попытку облечь TeX в клерковский костюм с галстуком. Все они канули в Лету забвения – только Abiword сохранился в составе эвентуального GNOME Office (хотя изначально к GNOME никакого отношения не имел). Впрочем, и Lux продолжает свое развитие – но уже скорее в качестве программы вёрстки «на скорую руку», нежели универсального word-процессора.

В результате всех этих процессов – и популяризации самой системы, и появления пользовательских приложений для нее, – на рубеже 1998-1999 годов в широких кругах околокомпьютерной общественности заговорили о появлении «Linux'а с человеческим

лицом». Казалось бы, этот самый «очеловеченный» Linux имеет все шансы прочно окопаться на пользовательских десктопах, заменив в этом качестве Windows (о прочих пользовательских платформах, за исключением MacOS, к тому времени забыли).

Началось явление, вошедшее в историю ушедшего тысячелетия как Linux-бум. Оно, в свою очередь, вызвало к жизни новые дистрибутивы, уже прямо заявленные в качестве пользовательских десктопов – такие, как Corel Linux, распространявшийся по схеме коммерческого софта. Впрочем, ни народной любви, ни более-менее приемлемой популярности они не снискали: по настоящему о коммерческих (или квазикommerческих) дистрибутивах заговорят только через несколько лет.

Назад, в будущее: Gentoo и другие

Однако скоро выявилась и обратная сторона любого user-ориентированного дистрибутива «для всех»: оказалось, что у каждого из их разработчиков были свои представления о том, что же нужно конечному пользователю для полного счастья.

Одни полагали, что счастье достижимо только в среде KDE, другие – что истинно счастливым юзера может сделать только идеологически правильный GNOME. Ну а третьи решали вопрос кардинально, и помещали в дистрибутив все, что только можно. И юзерофильные дистрибутивы стали пухнуть, как на дрожжах. К тому же графические инсталляторы этих систем, облегчая, с одной стороны, установку, с другой – навязывали пользователю предопределённые свыше наборы приложений. О назначении коих этот самый пользователь рисковал никогда не узнать – просто из-за их изобилия. А количество user-ориентированных средств конфигурации стало, по меткому выражению Владимира Попова, превышать число конфигурируемых параметров.

И вот тут многие пользователи начали вспоминать, что они еще и администраторы собственных компьютерных систем – пусть даже в масштабе одного отдельно взятого десктопа. И, с легкой руки Клиффорда Вольфа (создателя дистрибутива Rocklinux) в обиход вошел термин – «дистрибутив, дружественный к администратору». Началась эра популярности дистрибутивов Source Based.

На этой волне появились и упомянутый выше Rocklinux (исторически первый дистрибутив из этой серии), и LFS (Linux from Scratch) – набор рецептов по сборке собственной Linux-системы с нуля, созданный Герардом Бикмансом, и вариации на тему Sorcerer, и CRUX с Archlinux.

Однако наибольшую известность на этом поприще снискал Gentoo Дэниеля Роббинса. Не в последнюю очередь – благодаря прекрасной документированности процесса установки, позволяющей, строго следуя директивным указаниям, собрать индивидуализированную, в том числе оптимизированную под наличное «железо», систему даже относительно малоопытному пользователю. А система портежей, родившаяся под идейным влиянием портов FreeBSD, позволяла очень гибко наращивать функциональность системы установленной.

Правда, мало для кого Gentoo оказался первым дистрибутивом – миссия его заключалась скорее в повышении общей квалификации пользователей, прошедших через Red Hat, Suse или Mandrake, и разочарованных их дружелюбием, местами навязчивым до неприличия. И с точки зрения понимания устройства системы опыт пользователя Gentoo уступает, пожалуй, только сборщику LFS.

Ну а роль самого Дэниеля Роббинса в пропаганде Linux'а также переоценить трудно. Кроме своего дистрибутива, он стал и автором многочисленных ярких статей о самых разных аспектах устройства этой ОС – и о файловых ее системах, и о программных RAID-массивах, и о приемах работы в командной оболочке. Увы – весной 2005 года он поступил на службу классовому врагу: по окончании аспирантуры ни в одной фирме, связанной с Linux и Open Source, не нашлось для него должности с достойной зарплатой. Тем не менее, в среде майнтейнеров родного дистрибутива он был подвергнут анафеме. Правда, скоро выяснилось, что «Карапетяны в неволе не размножаются», и Дэниел с Microsoft распрощался. А проект Gentoo по прежнему живёт и развивается.

На пути к гармонии

Однако и тут все оказалось не так гладко, как виделось поначалу. Будучи исходно типичным представителем «дистрибутивов для себя»,

Gentoo весьма мало подходил на роль системы общего пользования. Установка его (точнее, сборка из исходников) вполне могла длиться сутками, после чего требовалась еще и ручная доводка – правкой конфигов в текстовом редакторе. А выигрыш от оптимизации «под железо» постепенно оказывался все более иллюзорным, нивелируясь ростом общей вычислительной мощности компьютерного парка. И среди пользователей началась неясная сначала тоска по прошлому – простым решениям «из коробки», позволяющим развернуть систему и начать работать в ней за считанные часы, а уже потом, при необходимости и по возможности, шлифовать ее до немыслимого совершенства (а какой линусоид к совершенству не стремится?).

И тут наступил час систем, которым на всем протяжении своего развития удавалось счастливо балансировать на зыбкой грани между дистрибутивами «для себя» и «для всех», между универсализмом и индивидуализмом, не склоняясь ни к «популизму» Mandrake сотоварищи, ни к «кастовости» Slackware (а позднее и Gentoo).

И здесь в первую голову нужно назвать одного из отцов дистрибуции – Debian. Разработка в 1999 году apt (Advanced Packaging Tools) – универсального набора инструментов для управления пакетами, – сделала его лидером в пакетном менеджменте. Не случайно apt был использован в rpm based дистрибутивах (впервые – в бразильской Connectiva, затем – в российском Altlinux). И, более того, послужил прототипом для универсальных систем управления rpm-пакетами, таких, как yum (Red Hat/Fedora, ASPLinux) и urpmi (Mandrake).

И именно поэтому Debian дал в нашем тысячелетии наибольшее число активно развивающихся потомков. Впрочем, это уже совсем другая история.

Глава тринадцатая. Linux на Руси

Итог предыдущих глав можно подвести таким образом: на рубеже тысячелетий система, придуманная гиком для себя и, в лучшем случае, для «того парня», то есть группы товарищей-разработчиков, постепенно начала превращаться сначала в систему для корпоративных администраторов, затем – для корпоративных пользователей, и в конце концов, с появлением Mandrake, для пользователей обычных. И всё это происходило в мировом масштабе. А как обстояло дело на отдельно взятой бывшей одной шестой? Давайте посмотрим.

Предыстория русского Linux'а

Если все предыдущие очерки являли собой реконструкцию былого, составленную по литературным и сетевым источникам, то в настоящей последует описание событий, которым автор этих строк был наблюдателем, свидетелем, а отчасти и участником. Тем не менее, следуя заветам Корнелия Тацита, изложить их он постарается ещё более без гнева и пристрастия, чем всё остальное. Хотя личный характер этого и некоторых последующих очерков отрицать не буду.

Началом документированной истории Linux на Руси можно считать 29 июня 1992 года, когда случилось вот что (цитирую):

Комитетом по внешним связям мэрии Санкт-Петербурга зарегистрировано и внесено в государственный реестр Акционерное общество закрытого типа «Урабан-Софт Лтд.»

О чём было выдано СВИДЕТЕЛЬСТВО установленного образца, подписанное председателем означенного комитета, неким В.В.Путиным.

Однако примечательно оно не фамилией подписавшего. И не фактом регистрации очередного АОЗТ – в те годы общества различной степени безответственности плодились как поросята. И даже не то, что это АОЗТ с почти русским названием регистрируется Комитетом по внешним связям – учредитель его, Джон Линн Росмэн, являлся

американским гражданином. А тем, что через год, в 1993 году, она выпустила набор дискет (ещё, кажется, пятидюймовых) под названием «Открытое ядро». Этому набору суждено было стать предтечей российского дистрибутива.

Дистрибутивом в собственном смысле слова «Открытое ядро» ещё не было – оно представляло собой подборку текущих версий Slackware, FreeBSD, HURD, дополненную базовыми средствами русификации и русскоязычными материалами – переводами документации и первыми оригинальными отечественными руководствами (тогда это все еще могло уместиться на один диск). Конечно, о локализации в полном смысле слова речи не еще шло. Но уже первый компакт-диск «Открытого Ядра» (получивший номер версии 1.2) давал возможность работать с русскоязычными документами, затратив усилия, которые тогда представлялись минимальными.

Однако деятельность Урбан-Софта в то время не получила ни признания, ни известности. Да и сама она как-то забылась. Каюсь, что и я в своих предыдущих статьях на исторические темы уделил ей недостаточно внимания. Так что спасибо Олегу Садову и Сергею Голубеву, которые донесли до нас подробности событий тех давних лет.

Как ни странно, большой общественный резонанс получила статья Владимира Водолазкого – дедушки русского линуксописания. В 1994 году в журнале «Монитор» была опубликована его статья под примерным названием: *как легко и без головной боли установить Linux на свой персональный компьютер*. Журнал этот не пользовался известностью (кажется, за пределами Москвы был недоступен), и чуть ли не в следующем году прекратил своё существование, и само имя его забылось. Однако ксерокопии статьи Владимира, в сопровождении горы дискеток с Linux Slackware, ещё многие годы спустя ходили по рукам, выполняя роль вполне реального руководства по установке системы с этих дискет.

У истоков отечественной истории Linux стоят также статьи Петра Врублевского и Виктора Хименко, опубликованные в журнале Мир ПК в середине 1995 года, и посвященные установке SLS и Slackware, соответственно. В обеих статьях немало внимания уделяется вопросу,

как получить дистрибутив – тогда на Руси (да, видимо, и в Польше, откуда происходит Петр) это было задачей нетривиальной.

Впрочем, решить эту задачу попытались традиционным российским методом: в те же срединно-девяностые среди пиратских наборов типа All for Windows и All for OS/2, нет-нет, да и попадались диски All for Linux. В принципе, у тех же пиратов можно было заказать некоторые дистрибутивы «на золоте» – правда, ввиду эксклюзивности заказа, по ценам совершенно астрономическим.

Однако скоро необходимость в пиратских услугах (по крайней мере, в Москве) отпала: на прилавках специализированных салонов (Медиахаус и Электротех Мультимедиа) и больших книжных магазинов начали появляться оригинальные, симпатично оформленные дисковые боксы (на 4 или 6 дисков) от Walnut Creek и InfoMagic, включавшие один-два дистрибутива Linux (обычно – Slackware и Red Hat текущих версий), намерянное количество дополнительного софта самого разного назначения – начиная со средств разработки и заканчивая научными приложениями, краткий буклетик-руководство (на английском, разумеется, языке), и все это – по цене немаленькой, но вполне сопоставимой с заказной «болванкой от пиратов» (и, добавлю, более-менее доступной даже при зарплате научного сотрудника тех лет). Особенно впечатлял Linux Developers Kit – комплект для разработчиков из 12 дисков, изданный совместными усилиями компаний Pacific HiTech и Walnut Creek в 1997 году.

Однако перед этим, в 1996 году, произошло два знаковых события. Первое – выпуск той же фирмой Урбан-Софт второй версии «Открытого ядра» – по уже сложившей традиции, в виде компакт-диска в боксовом исполнении. Он включал в себя текущую версию Slackware во вполне рабочем виде, а также ознакомительные версии Debian, FreeBSD и даже HURD. главное же – на нём находился тогдашний Red Hat, адаптация которого к нашим условиям стала в дальнейшем основной сферой деятельности компании (со временем получившей имя Линукс Инк).

Если первые версии «Открытого ядра», как я уже говорил, не особо были замечены широкой общественностью, то версия вторая появилась в нужное время: мне довелось воочию наблюдать, как за ним охотились в больших московских книжных магазинах, типа «Библио-глобуса» и «Дома книги» – чуть ли не единственных, где его

можно было обрести. Кроме того, диск этот сыграл неожиданную (и, возможно, решающую) роль в дальнейшей истории Linux'a на Руси.

Вторым же событием 1996 года стал выход первой русскоязычной книги про Linux, написанной М.Шойхером, которая называлась: *Как установить Linux на ваш компьютер*. Правда, это была, строго говоря, скорее брошюра о восьмидесяти страницах. Однако она подробно описывала процесс инсталляции (на примере дистрибутива Slackware, компакт с которым шел в комплекте) и начальное конфигурирование системы. Немало внимания уделено было и пересборке ядра – любимому виду спорта тогдашних начинающих линуксоидов.

Руководствуясь этой книжкой (а также упомянутыми выше статьями Врублевского и Хименко), я и поставил «по уму» свой первый в жизни Linux (в варианте Slackware) – предшествовавшие эксперименты мои основывались на дискетах к статье Водолазкого и «золоте» от пиратов, и никаких практических результатов не имели. Более того, я добился локализации системы с помощью компонентов, вытащенных из «Открытого ядра», обрета способность писать русские тексты и даже распечатывать их на принтере. Впрочем, как-то использовать Linux в мирных целях мне тогда даже и в голову не приходило.

IPLabs Linux Team: начало русского Linux'a

Следующая веха на пути русского Linux'a – 1998 год, когда фирма IPLabs (точнее, ее подразделение – IPLabs Linux Team) совместно с Институтом логики (на самом деле это были одни и те же люди – Алексей Новодворский, Алексей Смирнов и Юрий Девяткин с коллегами) вдруг, ни с того, ни с сего, начала распространять дистрибутивы Linux.

Однако началась вся эта история с очередной неслучайной случайности. Как свидетельствует Алексей Новодворский (известный в сети как AEN), в 1996 году им с сыном в руки попал компакт-диск... чего бы вы думали? Того самого «Открытого ядра», о котором шла речь в предыдущем разделе. Их внимание привлекли два дистрибутива – Red Hat и Debian, и они разыграли их в орлянку. В итоге Алексею достался Red Hat, Петру – Debian. Так оно и повелось – дистрибутивы

rpm based оказались основной сферой деятельности сначала IPLabs Linux Team, а затем и выросшей из него Altlinux.

Но это будет потом. А сначала IPLabs Linux Team распространяла (в заказном исполнении, на «золоте») чуть ли не весь спектр тогдашних дистрибутивов Linux – как популярных в те годы, так и мало известных даже в узких кругах, но представлявшихся перспективными. Среди них были и дистрибутивы-ветераны – Red Hat, Slackware, Debian, и BlackCat – разработка товарищей из соплеменной Украины (о котором я скажу чуть позже), и обаятый ореолом таинственности Stampade – один из предтеч будущего Gentoo. Как будет сказано в главе семнадцатой, планировалось и распространение SUSE.

Однако в силу стечения ряда обстоятельств главным объектом распространения IPLabs Linux Team стал юный на тот момент Linux Mandrake, о котором говорилось в прошлой главе – произведение француза Гаэля Дюваля сотоварищи.

Первая версия Linux Mandrake в исполнении IPLabs Linux Team (за номером 5.1) не носила еще гордого имени RE (то бишь «русская редакция») и весьма точно воспроизводила прототип (подобно тому как тот, в свою очередь, был тогда точным клоном Red Hat, дополненным KDE). Однако она уже содержала серию пакетов русификации, последовательная установка которых позволяла выполнить почти полную локализацию системы.

Так что это событие знаменовало, как выяснилось в ходе последующих событий, зарождение стройной концепции локализации (и, шире говоря, интернационализации), воплотившейся в следующем произведении IPLabs Linux Team – Linux Mandrake 6.0 RE, увидевшим свет летом 1999 г. Оно было реализовано в виде двухдискового набора с кратким, но чрезвычайно информативным печатным руководством (автором его был Алексей Новодворский). Сама система имела русифицированную программу установки: выбор русского в качестве языка инсталляции автоматически приводил к корректной и полной локализации, включая и возможность работы с кириллицей в т.н. «неправильных» приложениях (существовало в те годы такое понятие).

А в самом начале 2000 года увидел свет Linux Mandrake 7.0 RE – уже практически полностью самостоятельный дистрибутив,

унаследовавший от прародителей только программы установки и конфигурирования Mandrake, совместимость с форматом пакетов rpm и, частично, с файловой системой Red Hat. Он выходил в тиражном виде — как тогда говорили, на «серебре», в боксовом исполнении. И имел два варианта — полный, четырехдисковый, и краткий, однодисковый. Первый сопровождался традиционно информативным руководством по установке и использованию системы.

С Linux Mandrake в исполнении IPLabs (начиная с версии 5.1, начало 1998 года) началась моя практическая работа в этой ОС. До этого я, при наличии толики свободного времени, увлеченно устанавливал и переустанавливал различные дистрибутивы (Slackware, Red Hat, Caldera, Suse), пересобирав ядро, настраивал Иксы, прикручивал русские буквы, — но никаких реальных задач придумать не мог. Может быть, потому, что все время уходило на настройку системы. И тут выяснилось, что в Mandrake (тем более RE) можно работать сразу после установки — благо и соответствующие задачи появились: создание контент-сайтов разной, преимущественно научной, направленности. Оказалось, что Linux идеально подходит для этих целей. В результате через год совместного использования Linux и Windows (в 1999 году) последняя была снесена с моего диска — и с тех пор никогда более не появлялась на моих машинах.

Параллельно с Mandrake RE IPLabs Linux Team осуществляла и другой проект — подготовку к изданию дистрибутива Debian GNU/Linux. Каковой и был осуществлен осенью 2000 года, практически сразу вслед за выходом оригинальной версии 2.2, получившей имя безвременно скончавшегося разработчика Джоэля Клекера. Именно Debian в исполнении IPLabs Linux Team стал первым российским коробочным дистрибутивом этой системы, сопровождавшимся уже не буклетом, а полноценным руководством по установке и настройке (включая русификацию), написанным Петром Новодворским — как вы помните, некогда при жеребьёвке он вытянул именно его.

Впрочем, магистральная линия развития Русского Linux'а пролегла не здесь: в те годы никто не рассматривал Debian как систему для широких народных масс. Тогда как Mandrake и в оригинальной, и в русской редакциях позиционировался именно так.

К слову сказать, сама фирма IPLabs, основным профилем которой была сборка компьютеров и торговля комплектующим, была первой в России, решившейся на продажу машин с предустановленным Linux'ом – так называемых Linux Box. Правда, начинание это тогда развития не получило. Ибо далеко опередило своё время.

Параллельно с деятельностью IPLabs Linux Team развивались еще два локализованных дистрибутива украинского происхождения (но, тем не менее, русский язык поддерживавших) – KSI Linux Сергея Кубушина и Black Cat Леонида Кантера и Александра Каневского. Первый представлял собой, наверное, один из первых примеров т.н. деривата Red Hat. То есть, не будучи прямым его клоном, он использовал пакеты его формата и в какой-то мере был совместим по файловой иерархии. Black Cat же позиционировался разработчиками как Red Hat с улучшенной поддержкой кириллицы и исправлением замеченных ошибок.

Судьба украинских дистрибутивов была различной. Разработчики Black Cat влились в российскую компанию ASPLinux, и результаты их работы были инкорпорированы в одноименном дистрибутиве. Сергей Кубушин же, вследствие воистину детективной истории, связанной с переделом собственности на Украине, вынужден был эмигрировать в США, и развитие его дистрибутива прекратилось навсегда.

Новые игроки на Linux-поле

В середине 2000 года впервые заговорили и о другом претенденте на должность русского дистрибутива для конечного пользователя. Им стал вышеупомянутый ASPLinux – точный клон Red Hat с улучшенной поддержкой кириллицы. Все новые и новые редакции бета-версии его с завидной регулярностью (чуть ли не еженедельно) стали появляться на сайте одноименной фирмы в виде iso-образов. Процесс этот длился далеко не один месяц и логически завершился осенью 2001 года выходом ASPLinux 7.1 – первым по настоящему коробочным российским дистрибутивом, сопровождавшимся подробной печатной документацией: руководством по установке, руководствами пользователя и администратора, автором коих по случаю оказался ваш покорный слуга.

Однако перед этим, практически одновременно, весной 2001 года, случилось два события, во многом предопределившие развитие Linux на Руси. Во-первых, было объявлено о включении разработчиков Black Cat в команду ASPLinux – шаг логичный, поскольку оба дистрибутива позиционировались одинаково. Во-вторых, IPLabs Linux Team была преобразована в фирму ALTLinux, что сопровождалось выпуском дистрибутива Linux Mandrake RE Spring 2001. Ему суждено было стать последним представителем русской линии Mandrake – в дальнейшем Altlinux выпускала дистрибутивы уже под собственным именем.

Таким образом, к концу 2001 года сложились две линии дистрибутивов с национальной спецификой – Altlinux и ASPLinux, вокруг каждой быстро сформировались сообщества пользователей.

Полноты картины нужно заметить, что никуда не исчез и третий игрок на этом поле – компания Linux Ink (преобразованная из того самого УрбанСофта, который выпускал серию «Открытых ядер»). Она также занималась распространением кириллизованного Red Hat (ее комплекты так и назывались – Red Hat Cyrillic Edition), однако влияние ее и по опубликованным материалам, и по личным контактам с пользователями, прослеживается в те годы очень слабо. Не сформировалось вокруг нее и устойчивого сообщества пользователей.

Были в те годы и другие, малоизвестные, попытки создания дистрибутивов с национальной спецификой. Так, московская компания CPS (дистрибьютор ряда известных софтверных фирм) распространяла Corel Linux – один из первых коммерческих клонов Debian, который в свое время позиционировался как лучшая Windows, чем сама Windows. Дистрибутив этот шёл в сопровождении пакетов русификации системы, заимствованных из разработки Петра Новодворского для Debian. Однако популярности он не снискал (как, впрочем, и сам Corel Linux в мировом масштабе).

Наконец, существовало несколько проектов создания «русской Slackware», один из которых (под названием Fregate) был даже доведен до стадии iso-образа. Однако все они создавались на чистом энтузиазме и прекращались по исчерпанию оногo.

Горизонты раздвигаются

Так что, казалось бы, дальнейшее развитие Linux в России будет определяться сосуществованием национально-специфических дистрибутивов Altlinux и ASPLinux. Однако этого не произошло. И причин тому несколько.

Во-первых, сам Linux к тому времени приобрел подлинно интернациональный характер, и процедура прикручивания к нему русскоязычных атрибутов (шрифтов, раскладок клавиатуры, спеллинга) не была уже столь сложной задачей. Благо и описаны эти процедуры были многократно – как в сетевых источниках, так и в журнальных статьях, и даже в толстых книгах.

Во-вторых, эпоха rpm based дистрибутивов заканчивалась – на Руси, как и в мире, все большее внимание пользователей привлекали дистрибутивы Source Based.

И в-третьих, неожиданно в России стал доступным практически весь Linux-мир в многообразии представляющих его дистрибутивов. В чем немалая заслуга двух фирм, созданных в 2000 году – Linuxcenter и Linux-online, сферой деятельности которых была онлайн-торговля дистрибутивами на CD в очень широком ассортименте (в том числе и весьма экзотичными). Нельзя преуменьшить и роль улучшения коммуникаций, по крайней мере – в Москве и ряде других городов и весей, а главное – некоторое снижение тарифов на подключение к Интернету.

В результате всего сказанного перед пользователями Linux, как действующими, так и начинающими, оказалось обширное пространство для выбора системы. Чем они и не преминули воспользоваться.

Единение с миром

Где-то с середины 2002 года началась эра популярности Gentoo, сначала в мире, а потом и в России. Разумеется, это дистрибутив, плохо приспособленный для российских просторов, так как эффективное его использование требовало хорошего коннекта. И потому сразу же были предприняты попытки сборки такого его варианта, который позволял бы при установке обходиться без подключения к сети или ограничиваться минимальным объемом скачиваемых программ.

Важно отметить, что ни один из проектов распространения Gentoo на пост-советских пространствах не претендовал на звание самостоятельного дистрибутива даже в отдаленной перспективе: время безудержного клонирования чего бы то ни было становилось достоянием истории. Нет, целью их было просто облегчение доступа пользователей к интересной системе во всем ее богатстве.

Характерно, что возникший в то же время rpm-based дистрибутив Linux XP, позиционировавшийся как клон ответвившейся перед тем Fedora, не снискал даже доли признания своих прародителей. И популярности ему не добавила даже национальная специфика – Linux XP представляет собой разработку упомянутой выше фирмы Linux-online. Не приобрел всенародной любви и проект НПО Сеть «русской Slackware» под названием MOPSLinux. Следствием чего было тихое угасание обоих.

Вообще очень интересно проследить динамику пользовательских предпочтений. И некоторые возможности к тому имеются. Время от времени на различных сайтах проводятся опросы на тему – какой дистрибутив вы используете? Так вот, в первые годы тысячелетия в таких опросах безусловно лидировал Red Hat и его прямые и косвенные потомки, при стабильной, по подчиненной доле Slackware и Debian. В опросах же середины десятилетия за майку лидера спорят следующие системы: Gentoo, Slackware, Archlinux и FreeBSD, последнее время к ним приближаются и Debian с Ubuntu.

Мой рассказ об истории Linux'а на Руси подошел к концу. Вместе с историей русского Linux'а вообще. Это не значит, что развитие этой ОС в России прекратилось – напротив, число ее пользователей всё множится. Просто, с одной стороны, российские дистрибутивы первой волны – Altlinux и ASPLinux – вошли в круг мирового дистрибутирования полноправными его участниками (хотя и не первого эшелона). С другой же – российские пользователи не замыкаются на отечественных дистрибутивах, используя все те же системы (и примерно в тех же пропорциях), что и мировое сообщество Open Source вообще.

Так что казалось бы, что говорить о каком-то специфическом русском пути развития Linux уже не приходится – и в нашей стране система эта окончательно продемонстрировала свой интернациональный характер. И хотя все эти истории имели своё

продолжение — я к ним уже не вернусь, по крайней мере, в этой книге.
Почему — скажу в последнем и окончательно заключении к ней.

Глава четырнадцатая. Slackware и первые СБР

Рост популярности Linux привел к появлению ее вариантов, ориентированных на так называемого конечного пользователя. Для снижения порога «вхождения в тему» были придуманы системы, снабженные простым в использовании инсталлятором, позволяющим сразу же получить готовую к употреблению систему с некоторым, безальтернативно устанавливаемым, набором пользовательских приложений и настройками, более или менее подходящими абстрактному настольному юзеру. То есть системы быстрого развёртывания. Однако перед их рассмотрением необходимо вернуться немного назад, на зарю дистрибуции.

Slackware и потомки

Как было сказано в главе одиннадцатой, дистрибутив Slackware представлял собой не столько как законченную систему, сколько как каркас для конструирования системы собственной. И эта его особенность активно использовалась. В результате Slackware стала плодотворной прародительницей клонов: на сегодняшний день на Distrowatch зарегистрировано 65 её производных. Правда, активно развиваемых среди них около дюжины, но в прошлом их число было куда больше.

Среди клонов Slackware можно выделить:

- дистрибутивы, базирующиеся на Slackware и дополненные той или иной системой пакетного менеджмента, например, упомянутые выше Voltalinux и Draco GNU/Linux, использующие pkgsrc от NetBSD, или Frugalware, в котором применяется rpm, заимствованный из Archlinux;
- LiveCD общего (Slax, Klax, Wolvix) или специализированного (Blin) назначения;
- порты Slackware на аппаратные платформы, отличные от i486 (SLAMD и Bluewhite – на AMD64, Slackintosh – на PowerPC;
- варианты Slackware с «национальной окраской» (Karamad – с иранской, JoLinux – с бразильской, Nonux – с голландской, и так

далее).

Конструкторский характер Slackware способствовал тому, что на ней базировалось изобилие разного рода специализированных систем, которые условно можно объединить под названием «Linux на дискете». Правда, ныне, с широким распространением LiveCD, появлением «Linux на флэшках» и отмиранием 3-дюймовых дисководов, «дискеточные» Linux'ы представляют интерес исторический – как напоминание о временах, когда деревья были большими, а дистрибутивы – маленькими.

Первые СБР

Рост популярности Linux привел к появлению ее вариантов, ориентированных на так называемого конечного пользователя. Для снижения порога «вхождения в тему» были придуманы системы, снабженные простым в использовании инсталлятором, позволяющим сразу же получить готовую к употреблению систему с некоторым, безальтернативно устанавливаемым, набором пользовательских приложений и настройками, более или менее подходящими абстрактному настольному юзеру.

Эта концепция наиболее последовательно проводится в системах семейства Ubuntu и их бессчётных производных. Однако, как ни странно, пионером тут была Slackware. Точнее, её клоны: в плане поворота «поворота лицом к пользователю» они оказались если и не «впереди планеты всей», то в первых рядах дистрибьютеров.

Потому что одним из первых опытов в направлении «безальтернативной» пользовательской установки был, видимо, Vector Linux, разработанный на базе Slackware Робертом Ланге (Robert S. Lange) и Дарреллом Ставемом (Darrell Stavem) на самом рубеже тысячелетий – в том самом приснопамятном 2000-м году,. Уже в первой версии этого дистрибутива, вышедшей в июне 2000 года, была реализована концепция установки интегрированной рабочей среды (в данном случае KDE) с фиксированным набором пользовательских приложений, необходимых и, более или менее, достаточных для решения стандартных задач офисного или домашнего десктопа.

Правда, этот дистрибутив производил довольно странное впечатление. С одной стороны, вроде бы всё красиво и шоколадно. Но

с другой – подборка софта выглядит весьма своеобразно. Во-первых, бросалось в глаза изобилие дублирующих приложений, что для однодискового дистрибутива представляется непозволительной роскошью. Во-вторых, несмотря на то, что в качестве десктопа по умолчанию в Vector Linux используется KDE, многие разработанные для этой среды приложения были заменены Gtk-аналогами, подчас существенно более слабыми функционально. А в-третьих... а в-третьих, не понравился он мне, вот и всё. Хотя на форумах я видел немало высказываний пользователей, не разделяющих мою точку зрения.

Тем не менее, вне зависимости от моих личных симпатий и антипатий, Vector Linux был практически первым «безальтернативно устанавливаемым» дистрибутивом не только в клане Slackware, но и в мире Linux вообще. MEPIS и Lindows, не говоря уже об Ubuntu, появились несколькими годами позже.

Были и другие попытки создания «Slackware с человеческим лицом», уж не знаю, насколько удачные, например, Kwort аргентинского происхождения.

Путь Дзэн

Однако наиболее удачным и ярким представителем «пользовательской» линии развития Slackware суждено было стать дистрибутиву Zenwalk. Возникнув в середине 2004 года под именем Minislack, свое нынешнее имя он получил в начале второго года жизни – в августе 2005-го. И имя это следует интерпретировать как серьезное стремление к постижению высших истин (Zen) – но не без доли истинно мушкетерской бесшабашности (walk). А в качестве тотема этого дистрибутива выступает самое умное и быстрое млекопитающее планеты – дельфин.

Как явствует из первоначального названия, разработчик дистрибутива – француз Жан-Филипп Гийомен (Jean-Philippe Guillemin), – ставил своей целью создать компактную систему, предназначенную для вполне конкретного конечного пользователя: себя, любимого. Свои побуждения он описывает во Вступлении к Руководству пользователя Zenwalk. Там же изложены и принципы,

которыми руководствовался Жан-Филипп при начале разработки – и которых он собирается придерживаться впредь.

Жан-Филипп оказался не одинок в своих представлениях об идеальном дистрибутиве Linux. И потому со временем вокруг проекта выросло не очень большое, но активное сообщество разработчиков – в настоящее время их около 20 человек (см. список контактов).

Интересна динамика развития дистрибутива: выход релизов не привязан какому-либо графику: новый релиз выпускается тогда, когда появляются новые версии того, что в него стоит поместить. Иногда этот срок составляет месяц, иногда полгода или год, но в среднем колеблется в пределах 2-3 месяцев. Благодаря чему в текущем релизе всегда можно найти самый актуальный на данный момент времени софт.

Модерн – вообще фирменный стиль дистрибутива Zenwalk. Так, он был первым из тех считанных дистрибутивов, которые начали было штатную, на стадии инсталляции, поддержку файловой системы Reiser4. Однако стремление к модерну гармонически сочетается в нем с сохранением стабильности: когда стало ясно, что окончательное доведение до ума Reiser4 нам в обозримом (а может быть, и в необозримом) будущем не светит, поддержка этой файловой системы была исключена.

Каждая версия дистрибутива имеет стандартную редакцию, включающую, кроме ядра и базового набора, оконную систему X, интегрированный десктоп Xfce, браузер, почтовый клиент, текстовый процессор и электронную таблицу, а также еще некоторое количество необходимых приложений – строго по одному на каждую задачу.

Стандартная редакция распространяется в виде iso-образа компакт-диска, объем которого демонстрировал завидное постоянство. Некоторое разбухание имеет место быть – не за счет разбухания дистрибутива (принцип его комплектации «одна задача – одно приложение» остался неизменным), а исключительно из-за увеличения «веса» всех его компонентов.

Большинство версий Zenwalk распространяются еще и в виде так называемой Core-редакции, образ которой тянет на 200 Мбайт, а то и меньше: объем iso-образа версии 4.8, последней на сегодняшний день, для которой имеется core-редакция, составляет всего 170 Мбайт. В ее состав входят базовые компоненты Linux и минимум консольных

приложений, без Иксов, десктопа, офисных и мультимедийных программ. Как можно видеть из таблицы, соге-редакция выпускается не для всех версий дистрибутива и, как правило, с некоторым запозданием относительно стандартной редакции.

Один из основополагающих принципов построения Zenwalk – сочетание компактности и функциональности. В нем предельно последовательно проводится идея: одна задача – одно приложение. То есть на установочном диске Zenwalk не найти изобилия десктопов и оконных менеджеров, кучи браузеров и почтовых клиентов, эшелонов аудио- и медиаплееров – всего того, что традиционно ассоциируется у нас с юзерофильными дистрибутивами. Впрочем, разработчики, кажется, и не декларируют своей чрезмерной любви к пользователю. Жан-Филипп разрабатывал его для себя, любимого, и позднее к нему присоединились те, чьи вкусы были близки вкусам основоположника.

Разумеется, содержимым диска пакетное наполнение дистрибутива не исчерпывается: дополнительные пакеты доступны, во-первых, из официальных репозиториев проекта, во-вторых, из коллекций дополнительного софта с нескольких зеркалируемых серверов, и в-третьих – из домашних каталогов участников проектов. Но в любом случае на всеохватность ни одного из этих источников рассчитывать не следует.

Недостаточное количество собственных приложений Zenwalk может быть восполнен двумя путями. Во-первых, в распоряжении его пользователей – коллекции программ, предназначенные для родительницы, Slackware.

Второй же путь наращивания функциональности дистрибутива – традиционен для пользователя любой основанной на Slackware системы. Это – самостоятельная сборка недостающих программ из исходников с созданием пакетов «родного» формата.

Системные требования для установки Zenwalk по нынешним временам более чем скромны. Пакеты его собираются под архитектуру i686, но с возможностью запуска на машинах i486, однако в качестве процессора все-таки рекомендуется что-либо класса Pentium-III. Памяти разработчики полагают достаточным 128 Мбайт, места на диске – 2 Гбайт под систему (реально установка с CD занимает 1,3 Гбайт). Требования к видеосистеме определяются текущей версией Иксов.

В век стремительного распространения 64-битных процессоров о двух, а то и четырех ядрах сборка с оптимизацией под i486 выглядит анахронизмом. Однако Патрик и его последователи, в числе коих и Жан-Филипп, знают, что делают, и результаты их деятельности говорят сами за себя: визуально Zenwalk – один из самых быстрых дистрибутивов, которые я видел в своей жизни. Хотя автор сего сочинения и осознаёт всю условность оценки визуального быстродействия, а главное, влияния его на скорость выполнения реальных задач, это греет душу.

Предназначение

Остается рассмотреть вопрос – а кому и зачем нужен еще один дистрибутив? Тем более, на первый взгляд, казалось бы, ничем особенно не выдающийся. Ведь в нем нет ни красот современных «юзерофильных» систем, таких, как современная Mandriva, ни, напротив, «крутости» Gentoo, ни простого доступа к пакетному изобилию, как в Debian, ни внешнего блеска и мощной поддержки Ubuntu и его сородичей, ни динамичности тотального обновления Archlinux... Ответ постараюсь дать в конспективной форме.

Во-первых, Zenwalk – если и не идеальный, то близкий к оптимальному дистрибутив для решения рабочих задач пользователя, не нуждающегося в «архитектурных излишествах»: каждой пользовательской задаче соответствует один, но зато простой в освоении и использовании инструмент, выполняющий свои задачи «из коробки».

Во-вторых, аскетизм его требований к аппаратуре делает Zenwalk пригодным для реанимации морально устаревших машин – причем не только в качестве специализированных монофункциональных, но и полноценных пользовательских десктопов. Ну а на современной технике он способен демонстрировать чудеса быстродействия.

В-третьих, как ни странно, – Zenwalk представляет собой отличную учебную площадку для начинающих пользователей. По крайней мере, тех из них, которые стремятся как можно скорее избавиться от своего «начинающего» статуса, и потому не гнушаются чтением man-страниц и прочей документации.

В-четвертых, Zenwalk – отличная среда для достижения собственного идеала в отношении пользовательского десктопа. Причем возвышенные цели могут достигаться в процессе решения приземлённых практических проблем – по мере возможности и необходимости.

И, наконец, в-пятых... Активная политика по продвижению Ubuntu (со всеми её разновидностями) и последовавший следствие этого фантастический успех этого семейства дистрибутивов привел к огромному наплыву новых пользователей Linux, в том числе и таких, которые еще вчера и слова-то такого не слышали. Что, казалось бы, хорошо – не за это ли боролись мы долгие годы? Но, с другой стороны, для многих из начинающих пользователей Ubuntu и Linux стали такими же близнецами-братьями, как Ленин и Партия. Я уж не говорю о возросшем уровне некомпетентности, точнее, о воинствующем нежелании уровень своей компетентности повышать. Так что Zenwalk выступает в этом море как один из островков, на которых найдут пристанище те начинающие юзеры, которые хотят стать компетентными. И которые готовы затратить на это определенные усилия – пропорциональной им будет эффективность их последующей работы.

Наконец, самое распоследнее. В свое время Дуглас Кенни и Генри Бэрд написали книжку, название которой в наших изданиях обычно переводится как «Тошнит от колец». Это весёлая пародия не столько даже на сочинение Профессора (к которому авторы относились с глубоким уважением), сколько на его многочисленных последователей.

Так вот, в мире Linux сложилась похожая ситуация, которую можно назвать «Тошнит от Убунт». При всей моей симпатии к этому дистрибутиву и его ближайшим родственникам, бессчётные эпигоны Марка Шаттлворта начинают вызывать нечто вроде аллергической реакции. И что каждый третий пользователь Ubuntu, вчера водрузив эту систему на свою машину, сегодня садится описывать в блоге очередное путешествие по новооткрытой Америке на только что изобретенном велосипеде с квадратными колесами... То, что каждая вторая такая заметка завершается словами: «Например, в Ubuntu...» ... То, что слова Linux и Ubuntu уже начинают восприниматься почти как синонимы... Всё это вместе могло бы вызвать пароксизм здорового смеха, если бы не навевало столько грусти.

Но ведь мир Linux так обширен и разнообразен, и в этом его прелесть. И если он свое разнообразие утратит, сведясь к Ubuntu и её производным, то и прелесть его будет утрачена. А сам Linux перестанет быть Linux'ом. Так что нужно же писать не только об Ubuntu – а к нему, как явлению планетарного масштаба, мы вернёмся в одной из ближайших глав.

И несколько последних слов

Все те концептуальные особенности Zenwalk»а, описанные выше, показались части его разработчиков не совсем соответствующими духу первоизданного Linux»а. И в результате от него отделился проект Salix. Но о нём сейчас разговора не будет – он выпадает и за хронологические рамки этой главы, и в тему систем быстрого развёртывания не совсем вписывается.

Глава пятнадцатая. Debian: история клонирования

О дистрибутиве Debian уже говорилось в главе одиннадцатой, посвящённой началу дистрибуции Linux. И оборвалась тогда его история на моменте зарождения его первых коммерческих клонов, каковыми выступили Corel Linux и StormLinux. Ни тот, ни другой проект тогда успехом не увенчались. Правда, по разным причинам. Corel Linux, не дав мгновенного коммерческого успеха родительской корпорации, был ею тихо похерен, как нежеланный ребенок. StormLinux же, будучи самостоятельным проектом, просто скончался голодной финансовой смертью.

Но дело их не пропало. Corel Linux, подобно подкидышу в цыганскую семью, со временем претворился в бравого чавела — дистрибутив Xandros, некоторое время развивавшийся вполне успешно. Что же до StormLinux — кое-какие из заложенных в нем идей получили развитие позднее в дистрибутивах семейства Ubuntu.

Другим направлением клонирования Debian стало портирование его инфраструктуры на ядра, отличные от ядра Linux. Первой ласточкой тут стал HURD — знаменитый долгострой проекта GNU: возникает проект Debian GNU/HURD. А в дальнейшем Debian-инфраструктура (в первую очередь пакетный репозиторий и механизм получения из него пакетов через apt) были пересажены и на совсем, казалось бы, чуждую почву — ядра BSD-систем. Что, прочем, счастья им тоже не принесло.

В результате универалистские тенденции в развитии дистрибутива переросли уже прямо в имперские амбиции. И со временем Debian стал позиционировать себя ни много, ни мало, как операционную систему, низводя роль собственно ядра (Linux, HURD, какое-либо из BSD — по утверждениям идеологов проекта, это не имеет никакого значения) до незначительного винтика в ее составе.

Жизнь не подтвердила притязаний дебианистов. Воз HURD и ныне там, где был четверть века назад. Ни одной из BSD-имплантаций не сопутствовал успех — и не удивительно, ведь каждая из них имеет не только собственное системное окружение, тесно интегрированное с

их ядром и отличное от GNU, но и свою, отработанную и «притертую», систему пакетного менеджмента с хорошо развитой инфраструктурой. Наконец, сами участники проекта начали поговаривать о том, что поддерживать такое количество аппаратных платформ, большинство из которых готовится отойти в мир иной, – непроизводительная трата средств.

Мне кажется, что возникновение клонов Debian было в том числе и реакцией на имперские устремления разработчиков материнской системы. От которой в итоге ответвилось три серии производных дистрибутивов, развивающихся независимо друг от друга, но при сильном взаимовлиянии. И, что немаловажно, в значительной мере сохраняющих совместимость между собой (и со своим прародителем) не только в отношении файловой иерархии, системы инициализационных скриптов, формата пакетов и методов управления оными, но даже и на уровне пакетных репозиториях. общего, кроме формата пакетов.

Во-первых, от Debian отделились дистрибутивы коммерческого типа – Xandros (бывший Corel Linux), Mepis и Linspire (ранее весьма прославившийся как Lindows). Они включали в себя проприетарные компоненты, такие, например, как пакет CrossOver Office (средство запуска под Linux Windows-приложений), фирменные драйвера устройств и так далее. Полные версии этих дистрибутивов распространялись за деньги. Платным являлся также доступ к их обновлениям. Однако облегченные варианты всех этих дистрибутивов, содержащие только компоненты Open Source (с небольшой примесью не вполне свободных, в понимании FSF, программ), были доступны для свободного скачивания на соответствующих сайтах.

Из дистрибутивов коммерческой серии наибольшую известность, местами скандальную, снискал проект Linspire. Ибо начат он был Майклом Робертсоном в 2001 году под прозрачно-пародийным именем Lindows. Да и, честно говоря, ранние его версии действительно производили впечатление пародии на дистрибутив. В конце концов эта система, неоднократно сменив имя, слилась в творческом экстазе с тем самым Xandros»ом, упомянутом ранее. И, вместе с ним и всеми братьями по классу, упокоилась в братской могиле.

Во-вторых, Debian лег в основу знаменитого LiveCD Knoppix – одного из первых «живых» дистрибутивов (то есть Linux-систем,

способных полноценно работать непосредственно с компакт-диска, без установки на винчестер). В Knoppix впервые появилось большинство инноваций, таких, как использование сжатого образа файловой системы cloop, автоопределение оборудования, автоматическое конфигурирование сети и подстройка параметров оконной системы X, которые потом стали характерными для большинства LiveCD, а потом и «обычных» дистрибутивов. Кроме того, Knoppix содержал средства автоматического переноса самого себя на жесткий диск, после чего превращался практически в самый обычный Debian.

Наконец, в-третьих, на базе Debian образовалось немало свободных дистрибутивов общего назначения, из которых наибольшая известность суждена была Ubuntu. Именно последняя и похоронила в братской могиле все коммерческие дериваты Debian'a, о которых я только что говорил.

Глава шестнадцатая. Ubuntu и его клан

В главе четырнадцатой была описана история первых систем быстрого развёртывания, основанных на дистрибутиве Slackware – таких, как Vector Linux и особенно Zenwalk. Однако наибольшую популярность среди всех СБР (и, замечу в скобках, не-СБР тоже) суждено было снискать производным другой линии – дистрибутиву Ubuntu, базировавшемуся на Debian.

Начало истории

Основатель этого дистрибутива – южноафриканец Марк Шаттлворт [Mark Shuttleworth], в 90-х годах – один из разработчиков Debian. А по совместительству также – бывший глава бывшей Интернет-компании Thawte Consulting, занимавшейся вопросами криптографии и сетевой безопасности. Деятельность которой была столь успешна, что на закате эпохи «дот-комов» ее приобрела известная корпорация VeriSign за некую астрономическую сумму, сделавшую Марка весьма небедным человеком. После чего он повел себя не очень стандартным для акулы капитализма образом.

Что надлежит сделать порядочному человеку в таком случае? Перво-наперво, «поделиться с пацанами». И каждый из бывших сотрудников Thawte Consulting получил премию в размере немалого количества рэндов (это валюта такая в ЮАР – название её происходит от месторождения Витватерсрэнд, на котором золота было добыто больше, чем на всех остальных месторождениях за всю историю человечества).

Во-вторых, следует осуществить голубую мечту своего детства. И Марк слетал в космос туристом, оказавшись в этом качестве вторым человеком в истории Земли.

В-третьих, стоит подумать о тех, кому, мягко говоря, не повезло стать миллионерами. И Марк создает и финансирует несколько некоммерческих организаций – по развитию образования в Африке, помощи развивающимся странам, и так далее.

И, наконец, вернуться к тому, с чего начинал — в данном случае в начале всех начал оказался Linux, на котором был построен бизнес компании Thawte.

А потому Марк собирает команду для разработки собственного дистрибутива Linux. В основу которого, естественно, кладется Debian — собственно, Ubuntu поначалу и позиционировался просто как Debian с «человеческим лицом» (и несколько осовремененный с точки зрения пакетной базы). Говорят, что само слово Ubuntu на одном из африканских языков (подозреваю, что на зулусском) означает нечто подобное нашему понятию «гуманизм».

Отступление. Правда, если обратиться к историческим источникам, видно, что представления о гуманизме у зулусов и близкородственных им народов были весьма своеобразными. Так, Мативаан, вождь одного из таких племен, найдя тело убитого вождя враждебного племени, имел обыкновение вырывать у него желчный пузырь и выпивать содержимое. Он полагал, что таким образом к нему перейдут смелость и лютость павшего врага.

Распространение

Однако вернёмся к основной теме. Дистрибутив Ubuntu, созданный во второй половине 2004 года (примерно через полгода после Zenwalk), мгновенно завоевал очень широкую известность и популярность, уже в следующем году возглавив рейтинг сайта Distrowatch, считающегося одним из самых авторитетных ресурсов по теме Open Source. Хотя надо помнить, что рейтинг этот весьма условен и отражает не столько распространённость дистрибутива, сколько просто к нему интерес. Но в данном случае оказалось, что он отражал действительность.

Отчасти это было обусловлено колоритом личности Марка Шаттлворта, отчасти — связано с экзотичностью истории дистрибутива. Однако главную роль в завоевании пользовательских симпатий сыграла политика распространения дистрибутива: на сайте проекта установочные CD и Live CD можно было заказать бесплатно — с абсолютно бесплатной же доставкой в любую точку мира (даже в российскую глубинку). Думаю, это немало способствовало известности Ubuntu в нашей стране.

Одним из основных принципов Ubuntu был отказ от имперских амбиций исходного Debian, о которых говорилось в одной из предыдущих заметок. В частности, Ubuntu первоначально ограничился поддержкой лишь трех, актуальных для основной массы пользователей, архитектур – x86, amd64 и PowerPC (позднее к ним добавилась ARM, но это совсем другая история). И не ставил своей целью «спакетировать» все, что открыто и свободно, сконцентрировавшись поначалу в основном на приложениях, «интегрированных в интегрированные среды» (читать: Gnome и KDE, позднее XFce) – хотя и представления об интеграции у Ubuntu-майнтайнеров оказались достаточно самобытными.

Не менее важно, что при создании дистрибутива была сразу четко определена его целевая аудитория. Сам Марк в интервью журналу LinuxFormat (№ 2, 2005), на вопрос, для каких пользователей предназначен Ubuntu, отвечает так:

Для двух категорий. В первую входят люди, которые действительно любят свободное программное обеспечение за его качество и техническое превосходство – то есть те, кто по-настоящему предан идее open source. Они являются участниками сообщества и вкладывают свой труд, равно как и получают что-то от него взамен... Ubuntu был сделан для себе подобных – то есть для самих разработчиков.

Другая группа, которая, как мне кажется, считает открытые проекты действительно привлекательными, прямо противоположна первой. Это люди, которые знают о компьютерах совсем немного и не хотят знать ничего сложного. По сути, они всего лишь хотят использовать то, что просто нормально работает и будет делать все правильно – так, как им нужно; где они с легкостью смогут найти то, что им потребуется.

Интересно, что в том же интервью Марк отвечает и на вопрос, для кого Ubuntu не предназначен:

Средняя группа, до которой мы пока не можем добраться на данном этапе: люди, которые очень много

пользуются компьютерами. Они установили дополнительное программное обеспечение, и у них есть парочка устройств, которые они любят подключать к своим компьютерам. Их потребности слишком разнообразны и не могут пока быть удовлетворены Linux или Ubuntu. Они не являются достаточно опытными пользователями, чтобы заставить это работать, и они недостаточно прямолинейны и открыты для нас, чтобы мы смогли сделать эту работу правильно.

Иначе говоря, изначально Ubuntu был ориентирован, с одной стороны, на тех, кто сам все знает и умеет, с другой – на тех, кто ничего о компьютерах не знает, знать не хочет, но готов положиться на знающих. Тогда как промежуточная категория «полузнающих» (а это, увы, большая часть пользователей Windows) к использованию Ubuntu (да и Linux вообще) не готова.

Нарастающая популярность Ubuntu имеет и объективные причины. В двух словах, Ubuntu – это почти самый обычный Debian, использующий deb-формат пакетов и систему управления ими – apt, а также чуть модифицированный Debian Installer. И в то время более или менее сохранявший совместимость с огромным пакетным репозиторием Debian (по крайней мере, у меня в то время проблем с установкой Debian’овских пакетов в Ubuntu не возникло ни разу).

Отличие его от прародителя заключалось в том, что он комплектовался самыми свежими версиями пакетов, примерно соответствующим тестируемой [testing], а иногда и нестабильной [unstable] и даже экспериментальной [experimental] веткам Debian. Сборка пакетов осуществлялась с оптимизацией с флагом -O2, что на процессорах того времени обеспечивало несколько большее быстроедействие, чем у исходного Debian, собираемого с флагом -O1.

Вторая особенность Ubuntu – в том, что при инсталляции системы по умолчанию автоматически устанавливалась и настраивалась графическая среда. Коей, в соответствии с традициями Debian, стал Gnome.

Впрочем, выбор Gnome был обусловлен не только этим. Шаттлворт объясняет его тем, что во времена создания первой версии Ubuntu Gnome был хороший, а в KDE были одни «рюшечки и менюшечки». Однако в 2002– 2003 годах, когда затевался проект

Ubuntu, всё было с точностью до наоборот, и KDE далеко опережал Gnome по функциональности и «юзабельности», это я как очевидец свидетельствую. Так что, на мой взгляд, Марк несколько лукавит.

Дело в том, что на момент начала разработки Ubuntu уже существовало несколько базирующихся на Debian систем быстрого развёртывания, в частности, упомянутые в главе пятнадцатой MEPIS, Xandros, Lindows/Linspire. И все они в качестве рабочего стола по умолчанию (или даже единственного) использовали KDE. Так что Gnome было единственным способом выделить Ubuntu на их фоне. И, к слову сказать, последующий взлёт популярности Gnome был спровоцирован именно нарастающим распространением Ubuntu.

Появление «разновидностей»

Но поскольку Gnome – всё-таки лишь один из возможных пользовательских рабочих столов, немедленно (весной 2005 года) был создан вариант дистрибутива, использующей в качестве рабочего окружения KDE. Который логично получил имя Kubuntu. Правда, сборкой его занимался чуть ли не единственный человек, Джонатан Ридделл [Jonathan Riddell], при поддержке дюжины энтузиастов. Что не мешало рекордными по срокам сборками новейших версий KDE – напомним: это были времена расцвета 3-й ветки...

Особенностью третьего из основных, на первых порах, представителей, Edubuntu, как и следует из названия, является комплектование программами образовательного назначения.

Собственно Ubuntu, Kubuntu и Edubuntu стали первыми представителями семейства. Вслед за ними появился серверный вариант Ubuntu, лишенный не только какой-либо интегрированной среды, но и оконной системы X вообще, и Nubuntu – LiveCD для сетевого администратора.

Наконец, последним на тот исторический момент пополнением семейства стал Xubuntu – дистрибутив, в котором рабочей средой пользователя выступает Xfce.

Подчеркнем, что все представители семейства Ubuntu – это одна и та же система. И различия их проявляются только в комплектации инсталляционного CD или DVD. В случае необходимости наращивания установленной системы пакетами, на CD (DVD) отсутствующими, все

три дистрибутива обращались к одному и тому же репозиторию или его зеркалам. Поэтому, вне зависимости от комплектации исходного носителя, из пакетного репозитория можно было легко установить почти любой менеджер окон или интегрированную среду. Более того, возможна безболезненная трансформация Kubuntu, например, в Ubuntu и обратно.

Собственно Ubuntu, Kubuntu, Edubuntu, Nubuntu и Xubuntu – это, изначально, официальные члены семейства. Однако Ubuntu оказался не менее продуктивным клонопородителем, нежели предок – Debian. И потому число его побочных потомков росло с каждым днем.

Это были, во-первых, просто локализованные версии Ubuntu/Kubuntu: финская, итальянская, тайваньская и так далее. От исходных дистрибутивов они отличаются только более или менее полным переводом интерфейса и системных сообщений на соответствующие языки.

Во-вторых, практически сразу в изобилии появились национально-специфические дериваты, отличающиеся от прародителя не только языком, но и учётом особенностей национального делопроизводства. В этой части особенно отличилась Испания, во многих провинциях которой – Андалузии, Кастилии, Галисии – было создано по собственному дистрибутиву для использования в их правительственных и муниципальных учреждениях.

Наконец, третья группа клонов Ubuntu – это дистрибутивы специального назначения, нацеленные либо на определенный круг задач, либо на специфическое оборудование.

Приведенного списка достаточно, чтобы представить себе начальные масштабы «экспансии Ubuntu». Ну, а продолжение её выходит за хронологические рамки настоящей главы.

Глава семнадцатая. SUSE в истории

В главе двенадцатой история SUSE уже затрагивалась – но лишь начальные её моменты, да и то очень кратко. Ныне же есть смысл осветить её подробно и целиком – от истоков до наших дней.

Представление семейства

Сочетание символов SUSE в разное время писалось по разному и имело разное содержание. Сначала оно в форме S.u.S.E. было просто аббревиатурой от названия фирмы, занимавшейся консалтингом и поддержкой UNIX-систем. После того, как эта фирма занялась разработкой собственного дистрибутива, на него было перенесено её имя. Имя это, утрачивая расшифровку, точки и меняя регистр символов, закрепилось за дистрибутивом на долгое время – вплоть до его расщепления на коммерческую и свободную линии.

В настоящий момент коммерческая линия представлена дистрибутивом SLE (SUSE Linux Enterprise), свободная же – openSUSE. С последним тесно связан ряд проектов, таких, как:

- OBS (Open Building System, ранее openSUSE Building System) – автоматизированная система сборки пакетов не только для родного дистрибутива и соплеменного SLE, но и ряда других (Fedora, RHEL, CentOS, Mandriva);
- SUSE Studio – система автоматической сборки на базе openSUSE и SLE в соответствие с потребностями и пожеланиями пользователя;
- openQA – система автоматического тестирования созданных образов дистрибутивов;
- openFATE – система управления возможностями и пожеланиями.

Все они неразрывно связаны с дистрибутивами openSUSE и SLE. И потому ныне SUSE можно рассматривать как общее имя для семейства проектов, охватывающих все стороны развития дистрибутивов – от разработки до распространения. И целью

настоящей статьи будет описание того, как SUSE дошла до жизни такой. То есть – её истории.

Из предыстории

История SUSE уходит своими корнями в седую древность – в далёкий 1992 год. И началась она в городе Нюрнберге или, точнее, в университете Эрлангена – Нюрнберга. Когда его недавний студент – Томас Феер (Thomas Fehr) и трое студентов действующих – Бурхард Штайнбильд (Burchard Steinbild), Хуберт Мантель (Hubert Mantel) и Роланд Дюрофф (Roland Dyroff), собрались... нет, не выпить самого лучшего пива из Баварии, а чтобы учредить фирму по разработке программного обеспечения и оказанию консалтинговых услуг в области UNIX-систем.

Фирма эта получила название Gesellschaft für Software- und System-Entwicklung (Компания по разработке программ и систем). И первые два года своего существования занималась распространением только что возникших в это время дистрибутивов Linux – сначала SLS Питера Макдональда, а затем, в преддверии безвременной его кончины – Slackware Патрика Фолькердинга (подробности их истории описаны здесь и здесь). В сферу деятельности компании входило также оказание технической поддержки пользователей, преимущественно корпоративных.

В 1994 году увидела свет локализованная, то есть немецкоязычная, версия Slackware, которая получила имя собственное – S.u.S.E. Linux, и номер версии – 1.0. Оно представляет собой аббревиатуру компании-распространителя. Последнюю нельзя ещё было назвать майнтейнером и тем более разработчиком. Но вклад её в дистрибутив не ограничивался германизацией – дистрибутивный комплект из сорока трёхдюймовых дискет сопровождался весьма подробной печатной документацией. С тех пор качественная «бумажная» документация на многие годы стала визитной карточкой SUSE и служила образцом, к которому стремились многие другие разработчики дистрибутивов. В частности, на неё ориентировались сочинители документации для Mandrake Linux/RE (в последующем Altlinux) и ASPLinux.

В 1996 году пути S.u.S.E. и прародительской Slackware расходятся навсегда. В качестве причины источники приводят то, что Патрик не принимал патчи с исправлениями ошибок в его системе, в результате чего германцам приходилось повторно править их в каждой новой версии.

Однако видится и другая причина: к этому времени популярность Linux'а вообще достигла того критического уровня, когда аскетические средства установки, конфигурирования и управления пакетами Slackware, развивавшегося в качестве типичного «дистрибутива для себя» – перестали устраивать потенциальных заказчиков компании S.u.S.E. Которые желали видеть «дистрибутив для всех», подобный набиравшему тогда популярность Red Hat'у – в статье «Linux: начало дистрибуции» я уже говорил, какой смысл тогда вкладывался в понятие «все».

Начало самостоятельного плавания

Так или иначе, но в 1996 году дистрибутив S.u.S.E. Linux пустился в самостоятельно плавание. Это ознаменовалось:

- появлением собственной инсталляционной программы по образу и подобию таковой из Red Hat, считавшейся тогда эталоном дружелюбия к пользователю;
- изменением системы инициализации – с BSD-стиля, исконного для Slackware, на SysV, принятый как в первозданном Linux'е Торвальдса, так и в большинстве распространённых и тогда, и ныне дистрибутивов этой ОС; разработкой первой в истории мироздания и дистроения сквозной системы конфигурирования дистрибутива – YaST (Yet another Setup Tool, то есть «Ещё один установочный инструмент»), потомок которой, под именем YaST2, используется дистрибутивах семейства SUSE по сей день;
- сменой формата пакетов – со свойственных Slackware простых тарбаллов на заимствованный из Red Hat'а RPM, быстро ставший наиболее популярным для распространения бинарников независимыми разработчиками.

Не ручаюсь, что все эти изменения произошли одновременно – сам я свидетелем ещё не был, а однозначных указаний в Сети (за исключением YaST'а) не нашёл. Но могу определённо утверждать, что

в 1997 году, когда я впервые увидел S.u.S.E., все они уже имели место быть в этом дистрибутиве.

Эта первая оригинальная разработка компании S.u.S.E. получила номер версии сразу 4.2, хотя логика подсказывала в лучшем случае лишь версию с цифрой 2. Почему – тайна сия велика есть. В Сети мне встречалось мнение, что номер версии был взят разработчиками прямо с потолка. Однако рискну высказать иное предположение: в 1996 году увидел свет дистрибутив Red Hat версии 4.0, а затем и 4.1. Разработчики же S.u.S.E. Linux сочли, что их продукт является более «продвинутым» – а учитывая систему YaST, некоторые основания к тому у них были. И потому присвоили ему «опережающий» номер версии: Red Hat 4.2 увидит свет лишь в следующем, 1997, году.

В скором времени SuSE, утратив в 1998 году точки в своём имени, стал дистрибутивом номер один не только в Германии, но и практически во всей Европе, оккупировав на этом континенте ту же нишу, что и Red Hat в Америке. И занял, вслед за последним, второе место по распространённости в корпоративном секторе в мировом масштабе.

Бизнес-модель SuSE строилась несколько по иной схеме, нежели у Red Hat. В частности, этот дистрибутив включал в себя ряд собственных закрытых проприетарных компонентов, в первую очередь – ту же систему YaST и собственную графическую рабочую среду. Входил в него также коммерческие X-серверы – X-Accelerated и MetroX, которые тогда обеспечивали лучшую, по сравнению со свободной модификацией Иксов – XFree86, поддержку видеокарт. Ни один из этих компонентов не был доступен в исходных текстах. Хотя использование свободных X-серверов (в те далёкие времена на каждую серию видеочипов приходился свой X-сервер) и оконных менеджеров (время интегрированных десктопов ещё не настало).

В «полноразмерном» виде SuSE бесплатно не распространялась – для свободного скачивания была доступна evaluation-версия, по истечении 30 дней приобретающая функциональную ограниченность: утрачивали работоспособность YaST и графическая среда. Что, однако, не препятствовало дальнейшему использованию дистрибутива – ввиду наличия свободных альтернатив в его составе.

»Полноразмерный» дистрибутив в коробочном исполнении продавался за немалые по масштабам тех лет деньги – от 30 до 100

долларов, в зависимости от комплектации. А установочный компакт evaluation-версии распространялась первыми системами онлайн-торговли по цене носителя и доставки. В том числе и в нашей стране – именно посредством такой, ныне забытой, онлайн-фирмы я в далёком 1997 и познакомился впервые с SUSE.

Справедливости ради надо отметить, что коробочные версии сопровождались печатной документацией различного, в зависимости от цены, объёма, но неизменно превосходного качества, которая сама по себе составляла львиную долю стоимости коробки. А «старшие» коробки включали и разнообразную атрибутику – в частности, значок с изображением хамелеона, исполненный почти как правительственная награда.

Видимо, непосредственная продажа дистрибутивов в коробочном исполнении и составляла основную статью дохода фирмы. Хотя и от технической поддержки в корпоративных масштабах она тоже не отказывалась.

К технологическим высотам

Ближе к концу тысячелетия, одновременно с успехами в бизнесе, продолжается поступательное технологическое развитие дистрибутива SuSE.

Оно выражается, во-первых, в весомом вкладе в совершенствование графической системы, место которой в Linux'е и остальных UNIX-подобных ОСях к тому времени почти безраздельно заняла свободная инкарнация оконной системы X – XFree86. Тесные контакты как с разработчиками последней, так и с рядом производителей видеокарт, в том числе и профессиональных, таких, как германская фирма Else, обеспечили дистрибутив поддержкой самых современных тогда решений.

Во-вторых, осенью 1998 года SuSE, сразу вслед за Mandrake, включила в свой состав KDE – первую (и тогда единственную работоспособную) интегрированную графическую среду. Что, с точки зрения пуристов свободного софта, в частности, Ричарда Столлмана, выглядело крамолой, так как лежащая в её основе библиотека Qt распространялась не под свободной лицензией. Тем не менее, судьбы

SuSE и KDE оказались тесно связанными, и связь эта не разорвана и по сей день.

И Иксы с хорошей поддержкой «железа», и KDE как бы ориентировали SuSE в направлении десктопов – ведь к рубежу тысячелетий в прессе всё чаще стали поговаривать о Linux-буме именно касаясь настольного его применения. Что получило своё выражение в вариантах основного дистрибутива – SuSE Linux Office Desktop и SuSE Linux Desktop, имена которых говорят сами за себя.

Однако не меньшее внимание в развитии SuSE уделялось и серверному направлению. Апострофом чего стало появление в 2001 году SuSE Linux Enterprise Server (SLES), работающего не только на традиционных PC, но и на рабочих станциях IBM и даже её майнфреймах серии s/390, как 32-, так и 64-битных. Со временем он стал по настоящему кросс-платформенной системой, включив в себя поддержку архитектур Intel Itanium и x86_64.

В общем, всё было понятно и привычно: хороший дистрибутив развивался и становился ещё лучше. И продолжалось это до осени 2003 года, когда мир открытого софта облетела весть о покупке фирмы-производителя SuSE компанией Novell, известной своей сетевой операционной системой Netware. И считавшейся (обоснованно или нет – другой вопрос) одним столпов проприетаризма,

Рождение openSUSE

Объявление о том, что Novell покупает компанию S.u.S.E. вместе с её дистрибутивом, вызвало большое волнение в сообществе Open Source и опасения за будущее SuSE. Задолго до завершения сделки (как известно, такие дела с кондачка не решаются, требуя одобрения всяких антимонопольных контор) посыпались мрачные прогнозы.

Однако опасения оказались напрасными. Ибо первым деянием Novell в рамках проекта после покупки, кроме очередной коррекции имени дистрибутива (отныне и по сей день он величался SUSE или включал в своё имя этот компонент) стало открытие исходных текстов системы YaST (к тому времени в имени её добавилась цифра 2) на условиях лицензии GPL 2.

Следующий шаг, сделанный компанией Novell в 2005 году, был ещё более радикальным: единый проект был расщеплён на две ветки –

коммерческую SUSE Linux Enterprise и свободную – openSUSE. Первая осталась в ведении Novell, управление второй было целиком отдано в руки сообщества независимых разработчиков. Что, однако, не исключало тесного взаимодействия между ветвями, в том числе и участия одних и тех же лиц (например, Грега Кроа-Хартмана, известного разработчика ядра Linux) в обоих проектах.

Дистрибутивы коммерческой ветки развивались на базе уже существовавшей SuSE Linux Enterprise Server – как в традиционно серверном (SLES), так и десктопном (NLD – Novell Linux Desktop, позднее преобразованный в SLED) направлениях. Кроме того, к ним присоединился дистрибутив OES (Open Enterprise Server), интегрирующий в себе Linux и Netware, и содержащий общие для них сетевые службы.

Одним из существенных новшеств, привнесённых в десктопную линию коммерческой ветки, стало включение в SLED десктопа GNOME. Причина была в том, что незадолго до покупки S.u.S.E. Novell начала активно участвовать в проекте Mono – свободном воплощении среды разработки .Net от фирмы Microsoft. Проект же этот основывался на библиотеке Gtk и был тесно интегрирован с GNOME. В частности, и потому, что основоположником обоих проектов был один и тот же человек – Мигель де Икаса. Так что казалось естественным, что сначала, в версии 9 обоих дистрибутивов SLE GNOME оказался на равных правах с традиционным для SuSE KDE, а, начиная с версии 10, приобрёл в них статус рабочей среды по умолчанию.

Интересно, как разработчики SLE будут выкручиваться из этой ситуации теперь, с появлением радикально «улучшенного» GNOME 3 с его радикальным изменением парадигмы и ориентацией на гаджеты, мало уместной в корпоративной среде. Это касается всех разработчиков Enterprise-систем, в которых затраты на переучивание персонала корпоративных заказчиков являются существенным ограничителем на внедрение уж очень революционных инноваций. Впрочем, это заботы корпоративов и «энтерпрайзеров». Нас же больше интересует развитие свободной ветки.

Развитие свободной ветки проходило параллельно и опережающими темпами. Первой версией openSUSE, разработанной сообществом, стала 10-я, основанная ещё на наработках прежней

SuSE. Однако уже версия 10.1, под влиянием SLE, получила в качестве одного из основных десктопов, в дополнение к KDE, также и GNOME.

Правда, GNOME в качестве основного десктопа в openSUSE не прижился. И это – не смотря на выход в начале 2008 года версии KDE 4.0, принятой пользователями очень неоднозначно. Тем не менее, уже в openSUSE релизе 11.2 (2009 год) KDE, достигшая тогда вполне работоспособной версии 4.3, опять становится основным десктопом. Хотя сборки openSUSE с GNOME, как и другими интегрированными средами (XFce, LXDE) и оконными менеджерами (например, Enlightenment) продолжают регулярно выходить. Хотя только сборки с GNOME имеют официальный статус – забота об остальных находится в руках волонтеров.

Соглашение с дьяволом

Однако я забежал несколько вперед. Возвращаясь к хронологии, надо рассказать об одном из тех событий, которые потрясли мир (Open Source, разумеется – многие обитатели этого мира вообще имеют склонность быть потрясенными и потрясаемыми).

«Случилось это в осень, в ноль шестой год» (а точнее, 2 ноября 2006), когда... нет, в Питере с нагана никого не убили. Но компания Novell, к тому времени уже более двух лет являвшаяся владельцем SUSE Linux Enterprise, а также главной поддерживающей силой openSUSE, и корпорация Microsoft объявили о начале сотрудничества в технической, маркетинговой и патентной сферах.

Заявление это вызвало обвинения Novell в «сделке с дьяволом» и грозные предупреждения о том, что такие сделки добром никогда не кончаются. Каюсь, и автор этих строк в написанном в те дни очерке, посвященном сему событию, был не очень оптимистичен.

Однако ничего страшного не произошло. Сотрудничество с Microsoft в технической сфере вылилось в облегчение интеграции Linux и Windows в гетерогенных сетях, в частности, увязку фирменных служб каталогов (eDirectory и ActiveDirectory, соответственна). Маркетинговое сотрудничество имело своим следствием рост продаж SUSE Linux Enterprise в несколько раз (в сетевых источниках фигурирует цифра до 250%).

Что же до сотрудничества в патентной сфере, вызвавшее наибольшие поправки, вплоть до обвинений в государственной измене идеалам Свободы, то оно могло коснуться только корпоративных заказчиков, и только в одной слаборазвитой по части патентного законодательства стране. Да и там никаких реальных последствий не имело. А косвенно это соглашение способствовало повышению внимания к патентной чистоте кода, используемого в открытом и свободном программном обеспечении.

Парадоксы бизнеса

Правда, по прошествии некоторого времени после соглашения с Microsoft финансовые дела у Novell пошли не очень хорошо. Однако, как учит нас классическая логика, *после того* вовсе не всегда означает *вследствие того*. И причиной тут было не соглашение по поводу Linux'а – как раз Linux-бизнес компании развивался вполне успешно.

Частыми версиями фирма не баловала – за период с 2004 по 2010 год мажорных релизов вышло всего три: 9, 10, и 11. Однако в течении каждого релиз-цикла регулярно выходили сервис-паки, которые по обилию и характеру обновлений часто вполне тянули на настоящие релизы.

Да и свободная ветка в лице openSUSE продолжала радовать пользователей своим развитием. Новые релизы её выходили с регулярностью раз в 6-8 месяцев. Начиная с версии 11.2 (октябрь 2009 года) релизам стали присваиваться кодовые имена. В отличие от Ubuntu или Fedora, в качестве таковых выступали не звери, реальные или мифические, и не малопонятные словосочетания, созвучные именам великих учёных и населённых пунктов, а оттенки зелёного – исконного геральдического цвета SUSE. И дизайн каждого релиза оформлялся в соответствие с оттенком-эпонимом.

Первый «цветной» релиз был оформлен в изумрудных (Emerald) тонах, следующий, 11.3 – в бирюзовых, релиз 11.4 переливался морской волной (Celadon), а текущий, 12.1 – приобрёл цвет листьев спаржи (Asparagus), и так далее.

Так что, раз у линии SLE хватало сил на выпуск очень серьёзных сервис-паков, а у сообщества openSUSE оставалось ещё время на подбор оттенков, гармонирующих с номером очередного релиза, с

Linux'ом в компании Novell был полный порядок. А главная причина финансового упадка фирмы крылось, видимо, в том, что время Netware, на которой в значительной мере строился её бизнес, прошло окончательно и бесповоротно. И никакой интеграцией с Linux'ом эту систему было уже не реанимировать.

Продажа бессмертной души

Завершилась «упадочническая» тенденция в развитии Novell очередной покупкой, о которой было объявлено в ноябре 2010 года. Только на этот раз, как сказал бы принц Датский, ужинал не Полоний – ужинали Полония. То есть компания Novell впервые выступила не в роли покупателя.

Хотя эта роль была ей хорошо освоена ещё в 90-х года, когда, одержимая, подобно сиятельному Камильбеку из «Повести о Ходже Насреддине», хватательным рвением, она скупала всё, до чего могла дотянуться:

- операционные системы – DR DOS у фирмы Digital Research, ставшей Novell DOS;
- настольные СУБД – Paradox у фирмы Borland;
- офисные пакеты, скомпонованные перед этим той же фирмой Borland в виде сборной солянки из текстового процессора WordPerfect, электронной таблицы QuattroPro, презентационной программы Presentation.

Но на этот раз компания Novell оказалась в амплу продавца – причём самой себя. А покупателем её товара выступила корпорация Attachmate, о которой до того времени в кругах Open Source, не интересующихся «большим бизнесом», мало кто слышал.

Неизвестность всегда рождает опасения. И в который уже раз по миру Open Source поползли слухи, что за спиной этой сделки стоял всё тот же улыбающийся дьявол из Microsoft'a (которой действительно досталось акций компании Novell примерно на пятую часть их суммарной стоимости). А сама сделка имела своей целью нанести сокрушительный удар защитникам свободы, задушив одного из самых мощных и влиятельных игроков на этом поприще.

Как и в прошлых аналогичных случаях, слухи о близкой кончине SLE, а с ней и openSUSE, оказались несколько преувеличенными. По

крайней мере, за прошедшее время (а оформление сделки было завершено в начале 2011 года) никаких тому подтверждений получено не было. Как и вмешательства дьявола Microsoft в посюсторонние дела разработки коммерческой и свободной веток SUSE.

Последствие

А вот корпорация Attachmate проявить себя успела – и есть мнение (и не только моё), что с хорошей стороны. Первым проявление её деятельности было разделение бывшей компании Novell на две независимые (друг от друга) группы:

1. собственно Novell, которой достался весь груз Netware'вского прошлого и в довесок к нему – проект ORS;
2. SUSE, унаследовавшей как SLE, так и контакты с разработчиками openSUSE.

Не знаю, как это скажется на Novell, но для SUSE, по моему (и опять же не только моему) мнению, это означает возможность сконцентрироваться на развитии собственно Linux-проектов, без отвлечения на посторонние «мелочи» вроде взаимодействия с Netware.

Время не оправдало надежд пессимистов прошло ещё слишком мало. Хотя для линии SLE оно поначалу казалось мёртвым сезоном. Однако 28 февраля 2012 прошло сообщение о выходе SUSE Linux Enterprise 11 SP 2 – как я уже говорил, сервис-паки SLE по серьёзности обновлений тянут на полноценные релизы многих более иных дистрибутивов.

Этот же сервис-пак знаменателен ещё и изменением метода обновления функциональности дистрибутива. На смену модели back-портирования (включения в старые версии ядра, Иксов и других ключевых компонентов функциональности, достигнутой в новых их версиях, вышедших за время жизни текущего релиз-цикла) пришла модель forward-портирования. Отныне в дистрибутив будут включаться новые версии ядра и других базовых компонентов, которые дополняются специфичными для SUSE возможностями и модифицируются для сохранения совместимости с предыдущими версиями системы. Процесс этот будет непрерывным, следующим генеральной линией развития критически важных составляющих дистрибутива.

Ну а про интенсификацию развития openSUSE под властью Attachmate и говорить нечего. Во-первых, новые релизы выходят регулярно и, до последнего времени, с прежней периодичностью, раз в 6-8 месяцев. Правда, с одним из релизов, 12.2 вышла задержка: вместо обещанного июля он появился в сентябре. Это связывали не столько с замедлением разработки, сколько со сменой версии компилятора и рядом других технических факторов. А может, разработчики хотели подгадать к юбилею своей системы?

Во-вторых, начиная с версии 11.4 (а она вышла уже по завершении сделки), и в свободной ветке, как и в коммерческой, изменилась модель обновления дистрибутива, причём ещё более радикально, нежели в SLE. А именно, здесь появился репозиторий Tumbleweed, подключение которого превращает openSUSE в систему частичного rolling, то есть непрерывно обновляемую по скользящему графику. Причём – исключительно по желанию пользователя, насильно в «перекати-поле» его никто не превращает.

Я далёк от мысли приписывать все эти, безусловно полезные, новшества благотворному влиянию корпорации Attachmate. Однако их появление свидетельствует как минимум о том, что нынешний владелец не чинит препятствий творческой мысли разработчиков и коммерческой, и свободной ветки. Не видно с его стороны и навязывания заведомо непопулярных в народе инноваций. Так что современное положение openSUSE внушает оптимизм.

SUSE на Руси

В 90-х годах прошлого века SUSE в нашей стране была менее известной, чем Red Hat, но пользовалась устойчивой симпатией в определённых, хотя и узких, кругах. И имела хороший шанс получить более широкое распространение. Ибо в самом конце 90-х IPLabs Linux Team (та самая, которая потом стала Altlinux'ом) предприняла попытку распространять SUSE в России примерно на тех же условиях, что и Mandrake/RE. То есть – с пакетами русификации, дополнительным софтом, актуальным для наших пользователей, и русскоязычной документацией. А Алексей Новодворский и Алексей Смирнов даже съездили в Мюнхен на переговоры по этому поводу. После чего в отчёте о поездке написали историческую фразу: «Самое лучшее из

Баварии – это, конечно, не йогурт. Самое лучшее из Баварии – это, конечно, пиво».

Но, вероятно, вследствие известной любви немцев к *deutsche ordnung*, нашему бизнесу категорически противопоказанному, высокие договаривающиеся стороны к консенсусу так и не пришли. А иначе имели бы мы нынче что-нибудь вроде openSUSE/RE. Но увы – история сослагательного наклонения не имеет...

Глава восемнадцатая. Год великого перелома

Мы с вами проследили историю Linux'a, начиная от времени до его зарождения, когда появились люди, ещё не знавшие о Linux'e, но уже бывшие линуксоидами. И до года Великого перелома в его истории, каковым я полагаю 2005-й. Разумеется, условно – из общей привычки к круглым датам. Но тем не менее берусь обосновать свою точку зрения.

Этапы Большого пути

Но сначала давайте обернёмся назад и окинем взглядом вехи, пройденные Linux'ом за первые пятнадцать лет своего существования. Приводимые ниже даты также условны – каждое из указанных событий было не ежесекундным, а растягивалось на год-два.

Итак, в 1991 году Linux неожиданно появляется из головы своего создателя, подобно Афине из головы Зевса, первоначально в виде, непригодном к использованию никем, кроме своего создателя. Решавшего, при посредстве этой ОС, свою задачу – то есть разработку её же.

В скором времени Linux'у находится более широкое применение среди разработчиков – правда, разрабатывают они пока ещё только сам Linux. Ибо, прежде чем начать его использовать – его следовало собрать. Однако уже в 1992 году создаются первые дистрибутивы Linux (SLS, затем Slackware) – и он становится доступным для широких масс разработчиков, в первую очередь UNIX-программистов. А с возникновением в 1993-1994 годах Debian'a, Red Hat'a и S.u.S.E. к ним присоединяются ещё более массовые круги системных администраторов.

О Linux'e для народа некомпьютерного речи пока нет. Впервые на эту тему робко заикнулась в 1997 году фирма Caldera, выпустив в свет одноимённый дистрибутив. Однако, будучи ориентированным на совершенно определённую часть народа – офисных работников, успеха он не имел.

Так что первым по настоящему народным Linux'ом суждено было стать Mandrake, вышедшему в свет в 1998 году. И уже в году следующем он приобрёл такую популярность в узких кругах, что компьютерные журналисты прогнозировали начало Linux-бума к рубежу тысячелетий.

Прогнозы эти не оправдались, и Mandrake, быстро получивший известность, не являлся законченным решением для конечного пользователя. Нет, это был скорее экспериментальный проект, ориентированный хотя и не на профессиональных компьютерщиков, но всё-таки тех пользователей, которые успели подружиться с этим инструментом. Вот среди них-то он и снискал тогда свою заслуженную славу.

Кроме того, Mandrake послужил примером для ряда майнтейнеров, предлагавших Системы Быстрого Развёртывания (СБС) – дистрибутивы с ограниченным набором приложений (в идеале строившихся по принципу: одна задача – одно приложение), но зато устанавливающихся быстро, просто, и по окончании установки предлагающих более-менее готовую к употреблению систему.

Первая волна таких дистрибутивов, вроде Corel Linux, представляла собой попытку коммерческой эксплуатации того самого прогнозируемого, но несостоявшегося Linux-бума. Они откровенно «косили под Windows» (вплоть до смешного – пресловутой некогда Lindows) и делались без учёта «вековых традиций» Linux-дистрибутивов. И потому подверглись вполне заслуженному забвению.

Однако сама идея СБС для конечного пользователя от этого хуже не стала. И потому в первые из нулевых годов была подхвачена и развита в таких дистрибутивах, как Vector Linux и MEPIS, а затем Zenwalk. Правда, снискав заслуженную славу, опять-таки, только в узких кругах, ни один из них широкого распространения не получил. Но, наконец, осенью 2004 года появляется Ubuntu сотоварищи, ставшая знаменем новой эпохи.

Почему 2005-й?

Вот мы и подошли к тому самому 2005 году, который я называю годом Великого Перелома. И наконец начну обоснование своего мнения.

Как только что было сказано, Ubuntu появляется осенью 2004 года, когда был представлен как самый совершенный и окончательный пользовательский Linux-десктоп для всех категорий пользователей – от самых начальных ньюбов до крайних гуру. Однако поначалу «действующие» пользователи встретили его достаточно скептически. Ибо помнили ещё мутный вал первых дистрибутивов «с человеческим лицом», вызывавших естественный вопрос: если таково лицо, то какова же у них... место, где спина теряет своё название?

Так что поначалу Ubuntu пророчили ту же судьбу, что и прочим «человеколицым» дистрибутивам (каюсь, и я был в числе скептиков). Однако это был один из тех нередких случаев, когда провидцы и ясновидцы оказались не правы. Ибо интенсивная работа над ошибками (как чужими, так и своими) и несколько дезординарных маркетинговых ходов, типа «всемирной рассылки» дисков, как раз в 2005 году и принесли свои плоды: Ubuntu стал одним из самых популярных дистрибутивов Linux.

Впрочем, история Ubuntu и её триумфального шествия по пользовательских десктопам уже быларассмотрена в главе шестнадцатой. В рамках же главы настоящей нас более интересуют следствия этого триумфа. Именно они и позволяют сейчас, задним числом, увидеть, что 2005 год и был переломным в распространении этой ОС среди широких народных масс.

Последствия первые...

Таким образом, Ubuntu понадобилось всего около года для того, чтобы добиться той популярности среди узкого круга широких народных масс, к которой на протяжении более чем десятилетия стремились и Red Hat, и Suse, и Debian, и Mandrake с Mandriva.

Одним из главных следствий этого стал лавинообразный рост источников информации о Linux'е. Ибо к середине нулевых годов пыл линуксописателей первого призыва подыссяк: проблемы, которые волновали поколения пользователей, начинавших свой путь в Linux ещё в прошлом тысячелетии, и бывшие поводом для сочинения всякого рода FAQ'ов, How-to'ёв, Tips'ов и Hint'ов, ушли в прошлое. Ибо большая их часть решалась «искаропки», требуя в худшем случае

небольшой косметической доводки. О чём «старикам» писать было просто не интересно.

Однако на смену пришёл эшелон новых пользователей, открывавших для себя вещи, казавшиеся им откровениями – и спешивших поделиться своими новообретёнными знаниями. А поскольку это совпало по времени с массовым распространением блогов, а затем и микроблогов, возможностей поделиться для этого было вдоволь.

Можно сколько угодно иронизировать над сочинениями, хорошо описываемыми в старом и не очень цензурном анекдоте про Вовочку, вернувшегося 1 сентября из первого класса и радостно сообщившего родителям, как на самом деле называется пиписька. Ваш покорный слуга, в силу природной язвительности, отдал дань этому занятию. Однако время расставило всё по своим местам, и блоги не совсем тривиального содержания развились в полноценные ресурсы. Так что ныне информация о новых программах или работе в Linux'е нового «железа», скорее всего, найдётся на сайтах убунтийской тематики.

Частное, но очень важное для конечного пользователя следствие распространения Ubuntu – появление в Иксах качественных шрифтов. Точнее, не столько даже самих шрифтов, сколько механизмов их экранного воспроизведения. Если раньше для практической работы широко применялись пиксельные шрифты, а порой приходилось прибегать и к шрифтам от классового врага (причём, дабы соблюсти букву лицензии, следовало понимать толк в извращениях), то в Ubuntu качественный рендеринг шрифтов обеспечивался из всё той же коробки. И её примеру очень быстро последовали другие дистрибутеры.

Далее, Ubuntu, развивая традиции СБР, довела до логического завершения безальтернативную «установку в пять кликов», милую сердцу начинающего пользователя. И в то же время сохранила идущую от Debian'а альтернативную текстовую инсталляцию, допускающую широкое ручное вмешательство со стороны пользователя, знающего, что он делает.

Список усовершенствований, предложенных в Ubuntu пользователю любой степени предварительной подготовки, можно продолжать ещё долго. Так что остаётся только констатировать, что нововведения Ubuntu привели к повышению планки «юзерофильности» в хорошем смысле слова; или, если угодно, к

понижению порога вхождения для начинающего пользователя. Что, по закону обратной связи повлекло а собой рост числа этих самых, начинающих.

Иными словами, с появлением Ubuntu мир Linux изменился, и это — тот медицинский факт, с которым отныне должны были считаться все дистрибуторы.

... и отдалённые

И реакция дистрибуторов не замедлила последовать. И она носила самый разный характер.

Во-первых, легкость создания производных систем на базе Ubuntu повлекла за собой появление многочисленных её клонов — от простых ремиксов (каковым поначалу была, например, Xubuntu) через многие системы с «региональной» окраской до серьёзных переработок, вроде Mint'a.

Во-вторых, популярность Ubuntu стала стимулом развития прародительского Debian'a. Начиная с 2007 года, в нём появляется, наконец, графический инсталлятор, который ранее обещали на протяжении многих лет — и регулярно пролонгировали свои обещания. Интенсифицируется скорость разработки: если ранее интервал между очередными версиями мог составлять по три, то отныне релиз-цикл не превышает двух лет.

В-третьих, юзерофильные СБР начинают появляться на базе дистрибутивов, ранее в излишней любви к пользователям не замеченных. Даже на базе Gentoo, которая издревле славилась тем, что если она пользователей и любит, то очень нетрадиционными способами, возникают такие системы, как Sabaion и Calculate Linux.

Ну а уж число клонов Slackware, исповедующих идеологию СБР (а ведь именно они, в лице Vector Linux, положили начало этой традиции), множится на глазах: в некогда одинокому Zenwalk'у присоединяются сначала SalixOS, затем Slackel и Porteus.

Эпидемия любви к пользователю охватила не только дистрибутивы Linux, но и родственные системы, такие, как FreeBSD: её дериват, PC-BSD, являл собой типичную СБР со всеми её атрибутами — пользовательским десктопом и набором приложений в

виде ансамбля KDE, и даже собственным форматом пакетов и системой их управления, не требующей отслеживания зависимостей.

Даже свободная разновидность проприетарной Sun Solaris – Open Solaris по внешнему виду перестала отличаться от стандартного универсального дистрибутива Linux. А некоторые её потомки даже превзошли мамашу в этом отношении).

В конце концов инициатива в развитии пользовательской системы, в виде light-desktop, была проявлена в рамках проекта NetBSD, казалось бы, предельно далёкого от всяких «деSKTOPных шалостей». И подобные же вождения обозначили приверженцы OpenBSD. Правда, насколько я знаю, в обоих случаях дальше намерений дело не пошло.

«Десктопизация» Fedora

Однако самым главным явлением в контексте грядущих событий было то, что на место лидера в десктопном майнстриме начинает стремительно выдвигаться Fedora. На протяжении ряда лет, с самого своего возникновения в 2003 году, этот дистрибутив являлся «секретной лабораторией» Red Hat по генетической модификации Linux'a. И уже в силу своей экспериментальной природы он был не очень приспособленным к практическому применению, особенно совсем начинающими пользователями. И вдруг в одночасье он поворачивается к этому пользователю, в первую очередь десктопному, вполне человеческим лицом.

«Десктопизация» Fedora выражалась в той совокупности мелочей, каждая из которых, может быть, и не играла решающего значения. Но в их кумулятивный эффект создавал у пользователя ощущение комфорта при установке, настройке и использовании дистрибутива. Это были:

- и его визуальное быстроедействие или, как говорят, «отзывчивость» пользовательских приложений;
- и объединение всех пакетов, не имеющих официальной поддержки, но развиваемых примкнувшими майнтейнерами, в единый репозиторий – RPM Fusion;
- и отказ от «скользящей» модели обновления дистрибутива (так называемой rolling release), что, при некотором снижении

«фронтирности», греющей душу экспериментаторов, способствовало стабильности, более важной для пользователей-«практиков»;

- и, наконец, исправление мелких, но раздражающих багофич, многие из которых были унаследованы с седых времён стародавнего Red Hat.

В масштабах нашей страны очень важную роль в десктопизации этого дистрибутива сыграл проект Russian Fedora. Возникнув осенью 2008 года, он с тех пор выпускает собственные сборки дистрибутива (RFRemix) одновременно с выходом оригинального релиза. Именно в этих сборках и были впервые реализованы многие важные для пользователя мелочи, в том числе исправлялись багофичи, связанные с локализацией.

Важным моментом для конечного пользователя было не только существование RFRemix самого по себе – системы, собранной с учётом локальной специфики, но и наличие поддержки на языке родных осин. Таковая осуществлялась, во-первых, на форуме проекта, во-вторых – с помощью постоянно пополняемой русскоязычной документации, не только переводной, но и оригинальной.

При этом RFRemix продолжала сохранять полную совместимость с первозданной Fedora: одну сборку в другую можно было превратить легким движением руки – и в любом направлении. Что немаловажно для отечественного пользователя, часто испытывающего недоверие к отечественной же продукции любого рода – часто законное, но в данном случае абсолютно неоправданное. Так что, убедившись в последнем после установки оригинального дистрибутива, пользователь мог безболезненно перейти на русскую сборку. А разочаровавшись в ней, непатриотично вернуться на исходные позиции.

В общем, пресловутая «готовность к десктопу», о которой последнее время говорят не меньше, чем о «человеческом лице» Linux'a – в недалёком прошлом, и в оригинальной Fedora, и, особенно, в RFRemix возрастала от версии к версии. И достигла своего апогея в релизе 14 (ноябрь 2010 года). Который можно было рекомендовать всем пользователям без всяких оговорок, вне зависимости от задач и начальной подготовки.

Не Fedora единой...

... прирастала десктопизация: во второе пятилетие нулевых годов сдвиги в этом направлении наблюдались повсеместно.

Не стояла на месте Ubuntu, развивая свою пользовательскую инфраструктуру – ту самую, на которую редко обращают внимание. Но которая создаёт комфорт пользователя. В частности, того, который хотел бы опробовать новую версию нужной ему программы, но не имеет желания заниматься лобовой компиляцией тремя волшебными словами, либо подготовкой для сборки пакета по правилам дистрибутива.

Linux Mint, начав свою жизнь как «причёсанный» клон Ubuntu для самых крайних пользователей, сравнился по популярности с прародительницей. Не в последнюю очередь и потому, что стал предлагать серию пользовательских решений на любой, что называется, вкус. А возможно, и потому, что это один из немногих Linux-десктопов, который не стремится стать сервером: ибо рано или поздно серверные амбиции начинают одолевать почти всех майнтейнеров.

Кстати, ещё один дистрибутив, до сих пор не замеченный в «почёсывании маршалским жезлом» – PCLinuxOS. И его роль в создании пользовательских решений по заслугам ещё не оценена. А ведь он первым предложил самый простой способ создания дистрибутива «для себя, любимого» любым пользователем – путём простого снапшота работающей системы. Казалось бы, решение, валяющееся на поверхности – однако до Texstara никем не реализованное.

Очень важные в нашем контексте изменения происходили в недрах openSUSE – в частности, в виде проектов Open Build Service и SUSE Studio. Конечно, они ориентированы в первую очередь на разработчиков. Однако с их возникновением появились новые возможности для создания самых разнообразных систем, в том числе и пользовательских. Ибо OBS позволяет легко собрать какой-либо экзотический, но позарез нужный в данной ситуации пакет. А SUSE Studio – подымай выше: это средство создания индивидуализированного дистрибутива под конкретную задачу. Типологически родственное тому, что предложено в PCLinuxOS – но на порядок более универсальное.

Конец истории

Иными словами, к рубежу десятилетий сложились все условия для того, чтобы на базе Linux'a разрабатывались те самые цельные решения для конечных пользователей самой разной ориентации, о которых столько говорили с первых дней распространения этой ОС за пределы десктопа Линуса Торвальдса. И, казалось бы, дело за малым: бери и создавай.

Однако этого не случилось, и вектор развития Linux'a поменялся кардинально.

Почему? На этот вопрос можно предложить очень много ответов. Но в двух словах суть их сводится к одному: в этот момент закончилась история и началась политика. И для оценки политической ситуации придётся опять вернуться к Ubuntu.

Глава девятнадцатая. Феномен Ubuntu

О возникновении и первых шагах дистрибутива Ubuntu и его сородичей рассказывалось в главе шестнадцатой. И рассказ тот обрывался на моменте, когда Ubuntu обрела всенародную известность и популярность в широких народных массах. В чём далеко превзошла всех предшественников и современников. Но история никогда не кончается – и впереди было самое интересное. То, что позволяет сегодня оценить феномен Ubuntu во всей его полноте. Однако для этого нам опять придётся вернуться к истокам. Да и несколько поступиться принципами – ибо эта глава написана и с гневом, и с пристрастием.

Немного ретроспективы

Итак, обратим вспять время и посмотрим, как выглядел мир Linux накануне появления Ubuntu на арене истории, то есть к осени 2004 года. Надо сказать, что картина сложилась вполне благостная:

- Red Hat целиком переключился на коммерческие продукты, а в свободное от этого время экспериментируют в своей песочнице, именуемой Fedora, при участии сложившегося вокруг сообщества волонтеров;
- SUSE, недавно купленная компанией Novell, пытается расширить своё присутствие на американском рынке, и потому вынуждена идти нагав... нога в ногу с Red Hat; что, в частности, проявляется в учреждении собственной песочницы – openSUSE;
- Mandriva балансирует между банкротством и получением немислимых правительственных контрактов (или всё-таки субсидий?), хотя не прочь запустить свои щупальца и в сопредельные страны с давними своими, ещё со времён Mandrake, приверженцами, такие, как Россия и Бразилия;
- разработчики Debian ведут, правда, уже шёпотом, разговоры о мировом господстве на всех платформах (в том числе чужих, вроде разных BSD'ей, а то и вовсе несуществующих, типа Hurd), чем обещают осчастливить всё прогрессивное человечество;

- пользователи Slackware, под мудрым руководством Великого Патрика, продолжают изучать материальную часть, благодаря чему время от времени поставляют кадры «продвинутых» пользователей существующих дистрибутивов, а то и разработчиков дистрибутивов новых (таких, как Zenwalk);

- пользователи Gentoo, набравшись опыта, задумываются, а куда бы им переползти, но пока стойко продолжают перекомпилировать систему по будням, отдыхая за настройкой опций компиляции по праздникам;

- Герард Бикманс продолжает регулярно выпускать новые издания своей LFS, вокруг него по прежнему группируется могучая кучка приверженцев, развивающих его дело вширь, в виде Beyond LFS;

- создатели юзерофильных дистрибутивов с псевдокоммерческим уклоном, таких, как Vector Linux, MEPIS, Xandos, Lindows/Linspire, потеряли надежду массово развести лохов своими Linux'ми, которые «виндее всех виндей», но пока ещё рассчитывают удержать лохов уже окученных;

- разнообразные нишевые дистрибутивы и дистрибутивы «для себя»... да кто же уследит, что происходит в их мирках; разве что некоторым них, как Archlinux, удаётся попасть в запасной состав вышеперечисленной сборной – правда, как потом выяснится, ценой отказа от своего стиля игры.

В общем, идёт нормальная цивилизованная жизнь: крупные (по меркам FOSS-бизнеса) воротилы ворочаются, мелкие барыги – барыжничают, гики – гикствуют, строители светлого будущего на одном отдельно взятом десктопе – продолжают его строить, мало внимания обращая на окружающий мир.

На пресловутых «простых» пользователей все дружно забили. Разговоры о Linux'e с человеческим лицом, да ещё на каждом десктопе, ведутся чисто по привычке. Выражение «вендокапец» превращается в мантру, смысл которой забылся. Упоминание виндового апокалипсиса становится ритуальной фразой, подобной напоминанием о неизбежности победы коммунизма в мировом масштабе. В светлом будущем, разумеется, в отдалённой перспективе.

Одни из недавних приверженцев лозунга «Каждой домохозяйке – по тёплому Linux'у!» понемного переквалифицируются в управдомы...

пардон, в сисадмины. И начинают зарабатывать на жизнь знаниями и умениями, приобретёнными за время пламенного энтузиазма. Другие, возвращаясь к Windows, становятся офисными менеджерами и клерками. И вспоминают период пламенного энтузиазма как увлечение юности. Те же, кто видел свою роль не в амплуа пропагандиста, но популяризатора, просто отходят в сторону. И если и продолжают свою деятельность – то, скорее, по привычке.

В общем, идёт нормальная цивилизованная жизнь, внешне кипящая, но на самом деле устаканившаяся. И ничто, казалось бы, не в силах нарушить сложившегося благолепия. Если бы не два фактора.

Первый – те самые «простые» пользователи, о которых Linux забыл. Но которые не забыли о Linux'е. Ибо были не так уж просты, и нашли в нём среду для эффективного решения своих профессиональных задач. Или для применения к своим очень серьёзным хобби – серьёзным до той грани, где хобби смыкается с профессией. То есть те пользователи, которых я выделил в группу применителей (см. *Мир FOSS. Заметки гуманитария* в **Библиотеке Блогосайта**).

Второй же фактор – появление нашего перманентного героя, дистрибутива Ubuntu. Благодаря которому, прямо или косвенно, наши применители могли перестать задаваться вопросом: «Что ты сделал для Linux'а?». А получили не только право, но и возможность спросить: «Что Linux сделал для тебя?»

Появление героя

«Под звон мечей и зловещее пение стрел в огне пожаров вышел на арену мировой истории русский народ» – написал В.В. Мавродин книге о возникновении Древнерусского государства. Наш же герой появляется на арене мира Linux'а скорее под клацанье клавиш и подвывание кулеров. И представляется как самый совершенный и окончательный пользовательский Linux-десктоп, с помощью которого любая кухарка сможет управлять персональным компьютером. Возможно, даже не под наблюдением комиссара... то есть сисадмина. Тем самым смешивая расклад, описанный в прошлом разделе.

Правда, выяснилось это не сразу: «действующие» пользователи Linux встретили появление Ubuntu... да никак они его не встретили.

Ибо помнили ещё и 1999 год, обещавший приход Linux'a на каждый пользовательский десктоп. И первую волну юзерофильных дистрибутивов, каждый из которых представлялся как «Linux с человеческим лицом» (можно подумать, что до этого у Linux'a было не лицо, а... ещё одна спина). И то, как эти человеколицые дистрибутивы меняли имена, исчезали или владели жалкое существование, не нужные никому, даже своим создателям.

Так что те самые действующие пользователи, интересующиеся новыми дистрибутивами, поначалу пророчили и Ubuntu ту же судьбу. Должен сознаться, среди них, наряду со многими, был и автор этих строк. Однако это был один из тех нередких случаев, когда провидцы и ясновидцы, даже будучи очевидцами, оказались не правы (повторяю, это и ко мне относится). Ибо не учли, что организатор всего этого безнадёжного предприятия, Марк Шаттлворт, окажется способным на весьма дезординарные меры для продвижения своего произведения. Не пренебрегая, однако, и мерами вполне тривиальными.

Самой дезординарной мерой была организация приёма заказов на компакт дистрибутива безвозмездно, то есть даром – с бесплатной же почтовой доставкой по всему миру, в том числе по России. И это действительно работало – даже в самых удалённых городах и весях нашей необъятной родины.

Правда, по поводу этой акции скептики поначалу иронизировали: нынче каждый линуксоид имеет возможность бесплатно заказать себе подставку под пивную кружку. А ведь зря иронизировали: в результате о Linux'e узнала масса людей, прежде о его существовании и не подозревавших. И немало представителей этой массы дистрибутив хотя бы опробовали. А кое-кто из опробовавших так к нему и прикипел.

Это была одна из причин почти мгновенного роста популярности Ubuntu. Вторая же, как я говорил – вполне тривиальна: интенсивная «работа над ошибками», и не только своими. Ubuntu изначально позиционировался как очередной Linux с человеческим лицом, с одной стороны, и концентратор самого свежего софта – с другой. В плане первого вопроса были учтены все ошибки прежних попыток «очеловечивания» Linux'a. И в итоге разработчикам удалось если не найти оптимум между «настройкой с паяльником и осциллографом» и

«молчаливыми визардами для полных идиотов», то вплотную к нему приблизиться.

Направление работ по второму вопросу очевидно: использование самых свежих версий софта всегда потенциально чревато ошибками в оном – и ошибки эти следовало исправлять. Или не допускать – путём сознательного ограничения «степени свежести» – ведь программы, в отличие от осетрины, бывают свежести весьма разной. И в итоге в Ubuntu не стало никакого особого гипермодерна – она основывалась на репозиториях Debian тестируемой ветки, пригодность к использованию которой в десктопных условиях общепризнана.

В результате уже через год, к осени 2005, обнаружилось, что Ubuntu – вполне зрелая система, пригодная к применению «искаропки» пользователем любого уровня. Разумеется, не без некоторых шероховатостей, касавшихся в первую очередь локально-зависимых вещей, но это было вполне естественно: обеспечить равную поддержку всех языков, от зулусского до русского, за столь короткий срок физически невозможно. Да и лечилось всё это достаточно просто.

Всё сказанное выше было причиной того, что за пару лет количество пользователей Ubuntu достигло не просто какой-то значимой цифры – оно превзошло число пользователей всех прочих дистрибутивов, вместе взятых.

Не менее, чем количество пользователей Ubuntu, показателен их состав в сравнении с более иными дистрибутивами. Так, в многочисленных опросах о первом дистрибутиве Linux на протяжении первой половины нулевых годов неизменно, и с большим отрывом, лидировала Mandrake/Mandriva. Но те же опросы о текущем дистрибутиве показывали, что после успешного старта с Mandriva изрядное число пользователей перетекало на другие дистрибутивы.

Для второй же половины нулевых годов картина стала совершенно иной. Место лидера в опросах о первом дистрибутиве прочно заняла Ubuntu. И в то же время процент пользователей, оставшихся верными этому выбору, был неизменно высок.

Кроме того, было (и есть) немало действующих пользователей Ubuntu, для которых этот дистрибутив был не первым, и даже не пятым. Тех, кто прошли и ручную настройку Slackware, и тотальную компиляцию Gentoo, и роман «Ядро и мир» от FreeBSD, а кое-кто – и

сборку LFS. И чьё сердце успокоилось в казённом доме – на тихой и уютной Ubuntu.

Интересно также, что среди пользователей Ubuntu высок процент тех, кто не имеет к компьютерам ни малейшего отношения – ни по долгу службы, ни по велению души. А разве что по жизни вынужден ими пользоваться. Тогда как среди пользователей иных дистрибутивов процент этот исчезающе мал. Более того, среди моих личных, реальных и виртуальных, знакомых (а круг и тех, и особенно других у меня весьма широк) вообще нет людей, не работающих в околокомпьютерных сферах или просто не интересующихся компьютерами как хобби, которые использовали бы какой-либо дистрибутив Linux'a. Разумеется, если этот Linux – не Ubuntu.

Так что буквально за пару лет Марку Шаттлворту, фирме Canonical, примкнувшим к ним независимым разработчикам и, не в последнюю очередь, активным пользователям – создателям сайтов и авторам блогов убунтийской тематики, удалось превратить, казалось бы, рядовую «человеко-мордастую» поделку в самый популярный и распространённый дистрибутив планеты.

Не будем пока оценивать это в терминах великого советского поэта, автора знаменитой поэмы о «хорошо» и плохо», а примем как медицинский факт. И посмотрим, что же из этого получилось.

Реакция

Итак, Ubuntu понадобилось всего года два для того, чтобы добиться той популярности среди узкого круга широких народных масс, к которой на протяжении полутора десятков лет стремились и Red Hat в свою ещё десктопную пору, и Debian во время своих самых широких имперских притязаний, и Mandrake с Mandriva при всей своей перманентной фронтирности. Как же прореагировали на это явление дистрибутисты?

В первом приближении ответ очень просто: по разному. Для начала появление Ubuntu, в силу её развитой инфраструктуры, спровоцировало волну клонов: сначала официальных и полуофициальных вариантов со своими рабочими средами, затем – локализованных версий и версий, ориентированных на национальную специфику, а также специализированных систем (см. LXF #155). В

сущности, если не любая кухарка, то почти любой квалифицированный пользователь в состоянии собрать свой дистрибутив на базе Ubuntu. Другое дело, что как раз квалифицированные пользователи понимают бессмысленность этого занятия...

Иной была реакция со стороны ветеранов дистрибуции. Первый вал упрёков посыпался со стороны разработчиков Debian'a. Каковой сводился к двух основным пунктам: а) заимствованию всего из прародительского дистрибутива и б) несовместимости с ним на уровне пакетов и конфигов. Не смотря на то, что это положения во многом взаимоисключающие, некоторый повод к ним Ubuntu давала.

Да, Ubuntu действительно почти целиком основана на пакетной базе из репозитория Debian – tested и частично unstable. Да, в некоторых случаях пакеты из Debian'a не желали устанавливаться в Ubuntu, и чуть в большем количестве случаев – наоборот. Да, со временем Ubuntu приобретала всё большую дистрибуцифичность, а с переходом на схему инициализации upstart вообще отделилась от предка. Однако со временем ситуация по ряду позиций менялась.

До сих пор «официальная» часть репозитория Ubuntu в основном (кроме собственных разработок, типа того же upstart'a и среды Unity повторяет репозитории Debian'a. Но существует и неофициальная часть инфраструктуры Unity – репозитории PPA (Personal Package Archive) и инструмент для работы с ними – Launchpad. Так вот, PPA-репозитории – неисчерпаемый кладёз пакетов самого разного назначения. И все новинки свободного софтверостроения в первую очередь появляются именно в них. Так что для упрёка в паразитировании Ubuntu на Debian'e не остаются никакой почвы. К тому же нынче в отношении бинарной совместимости пакетов для Debian'a и Ubuntu достигнут консенсус. Что же до дистрибуцифичности – тут уж ничего не поделаешь: любой активно развивающийся дистрибутив рано или поздно приобретает свою специфику.

С другой стороны, Ubuntu косвенно дала Debian'у очень немало. Во-первых, здоровая спортивная злость подстегнула разработчиков последнего, и они существенно сократили релиз-цикл. Во-вторых, видимо, из той же спортивной злости, были реализованы наконец некоторые задумки, обещанные и ожидаемые... даже не три года, а

существенно больше, например, графический инсталлятор. А в-третьих и главных, популярность Ubuntu вызвала рост интереса и к родительской системе. И хотя я уже говорил о приверженности убунтуйцев к своему дистрибутиву, всё же немалое их число мигрировало по тем или иным причинам на Debian.

Так что к рубежу нулевых и десятых классовые противоречия между этими дистрибутивами можно считать вчерашним днём. А временами возникающие конфликты можно отнести скорее всего к противостоянию отцов и детей – а от этого не было избавлено ещё ни одно человеческое поколение.

Однако к тому же времени – рубежу нулевых и десятых – относится обострение конфронтации на другой линии: RHEL/Fedora против Ubuntu. Ибо в Ubuntu замахнулись на святое – на сервера и прочий корпоратив, начав выпуск «долгоиграющих» (LTS) релизов. Не то чтобы Ubuntu Server вдруг в одночасье стал прямым конкурентом для серверов на RHEL. Более того, отношение к Ubuntu в амплуа сервера было ещё более скептическим, чем поначалу – к Ubuntu в роли пользовательского десктопа. По крайней мере, на Linux-ресурсах было хорошим тоном иронизировать по этому поводу. Кстати сказать, кое-где иронизируют и по сей день.

Но в Red Hat сидят люди серьёзные, и им было не до иронии. Может быть, потому, что они вспомнили историю, начавшуюся в 1995 году. Ей посвящено следующее отступление, которое предназначено для тех, кому не довелось жить в то интереснейшее время.

Отступление. Всё началось с того, что была выпущена Windows 95. К которой, как и к Ubuntu, поначалу никто не относился серьёзно: она воспринималась как платформа для запуска игрушек. Даже для всамделишной офисной работы резонные люди консервативного склада отдавали предпочтение старой, не очень доброй, но досконально известной Windows 3.1/WfW 3.11. Прогрессисты же склонялись к OS/2. Что же до серверов на Windows 95 – такое могло привидеться в кошмарном сне с большого перепоя.

Нет, у Microsoft была в загашнике и самая настоящая ОС – Windows NT, от которой по прямой линии происходят все варианты всех современных Windows. Но как серверная платформа и она и близко не была тогда конкуренткой не только с UNIX'ам, но даже OS/2. А на рабочих станциях применение NT тормозилось интерфейсом,

унаследованным от Windows 3.1, который в считанные месяцы после выхода 95-ой стал казаться старомодным.

Однако, быстро оккупировав домашние компьютеры, Windows 95 постепенно утвердилась на рабочих местах различных контор. А затем... затем Microsoft в очередной раз всех напарила, выпустив Windows NT 4 с интерфейсом в стиле modern, то есть a la Windows 95. И именно с неё началось распространение NT-серверов и рабочих станций.

В результате в 1997 году – а кто не помнит, это был год рождения массового российского Интернета, – некоторые московские провайдеры впервые стали предлагать хостинг не только на UNIX-машинах, но и на NT-серверах. Причём последний стоил дороже. Что мотивировалось привычностью интерфейса для wb-мастера. Судя по тому, что эта услуга пользовалась спросом, аргумент действовал.

Так вот, Ubuntu тоже начала свой путь с оккупации пользовательских десктопов. В том числе десктопов школьников и студентов. А поскольку, как я уже говорил, пользователи Ubuntu, в отличие от Mandriva, уже показали завидное постоянство своих привязанностей, резонно ожидать, что со временем эти самые школьники и студенты принесут её и на рабочие места, которые Red Hat с давних пор полагал своей вотчиной. Надо было принимать меры – и они были приняты в двух направлениях. Здесь я остановлюсь только на первом, нетехнологическом.

Оно выразилось в агитации и пропаганде, достойной лучших учеников товарища Ульянова в скобках Ленина. Когда всё наше заранее объявляется прогрессивным, а всё не наше – устаревшим и маргинальным. Большевистский лозунг – «Кто не с нами – тот против нас!» – неожиданно прозвучал в исполнении тех, кто считал себя (и считает до сих пор) оплотом свободы.

Не обошлось и без прямых уколов в адрес Ubuntu, дескать, недостаточно внимания уделяющего разработке ядра и вообще системных компонентов Linux. А уж что до мелких подковык относительно перекрашивания обоев или перетаскивания кнопок – то это просто смешно. Особенно если не забывать о том, что стратегия «перекрашивания обоев и перетаскивания кнопок», говоря очень фигурально, за несколько лет дала Ubuntu больше пользователей, чем

всем остальным дистрибутивам – самые передовые патчи ядра за двадцать лет.

Итог

Пора попытаться в первом приближении ответить на вопрос: так в чём же феномен Ubuntu?

Ubuntu начинала как сугубо десктопный дистрибутив для конечного пользователя, и, не смотря на наличие аналогов в виде СБР, ориентированных на ту же нишу, очень быстро преуспела на этом поприще. Однако, не останавливаясь на достигнутом, она тихо и незаметно расширяет сферу своей деятельности в двух противоположных от десктопа направлениях.

Первое – это серверные решения, реализуемые в виде периодически выходящих «долгоиграющих» (LTS) версий. Второе же – прямо противоположное: разного рода гаджеты, планшеты и прочие смартфоны. И если в серверной сфере Ubuntu тащилась в хвосте не только за Red Hat и SUSE, но даже за прародительским Debian'ом, то здесь она оказалась в числе передовиков производства. В том числе и потому, что Ubuntu одной из первых всерьёз занялась адаптацией самой себя для альтернативных процессоров – ARM'ов всякого рода. Причём как организованно, так и частным порядком.

«Кратко резюмирую сегодняшний базар»: если раньше пользователь в основном приспособливался к миру Linux'a, то с появлением Ubuntu он впервые почувствовал, что и Linux-мир стал приспособливаться к нему.

Определить феномен Ubuntu короче у меня не получилось.

Глава двадцатая. *Linux*: история применителей

Прошлая глава закончилась на том, что Ubuntu суждено было стать первым дистрибутивом Linux'a, который не требовал от пользователя приспособляться к нему. А, напротив, сам приспособлялся к пользователю. Осталось только определить, кто же он – потенциальный пользователь Linux'a? Для чего нам опять придётся погрузиться в древность настолько седую, что даже и сказать страшно – в эпоху бронзы.

Очень древнее вступление

Где-то близ середины второго тысячелетия до нашей эры, на просторах Ближнего Востока появились «колесничные народы» – индоевропейцы, выходцы из евразийских степей. Наиболее известны из них арии, давшие, с одной стороны, правящую династию царства Митанни, с другой – цивилизацию Вед в Индии. Но к тому же общевосточному колесничному сообществу принадлежали и касситы – покорители Вавилонии, и гиксосы, завоевавшие Нижний Египет, в него органично вписались хетты Анатолии и микенские греки-ахейцы. А отклики колесничной культуры достигли кельтов на Западе, индоариев Индии на Юге и Китая эпохи Чжоу – на Востоке.

»Колесничные народы» повернули ход мировой истории. Именно от них идет дискретная (но, тем не менее, никогда не прерывавшаяся полностью) традиция индивидуального мастерства. Если раньше (да, в большинстве случаев, и по сей день) исход боевых действий определялся численностью участников с той и другой стороны, то тут впервые вступил в игру фактор интеллектуальный – технологическое превосходство и личное умение.

Действительно, давайте посмотрим, что такое боевая колесница. Это – предельно высокотехнологичное, по меркам того времени, изделие: каждая ее деталь – платформа, ось, обод и спицы колеса, – сделана именно так, как предписано техзаданием. Что требовало мастера – изготовителя.

Однако: колесница без движущей силы мертва. И потому ей требовалась пара лошадей совершенно определенного экстерьера. Но и этого мало: эта пара должна быть обучена и выезжена должным образом. И потому мастерство конской выездки – вторая составляющая успеха «колесничных народов».

Высокотехнологичная колесница и идеально обученная конская запряжка – хорошо. Однако техника в руках дикаря – это кусок металла. И любые технические средства требуют того, кто мог бы ими управлять – и управлять умело. То есть третий компонент колесничного войска – это мастер-возница: не случайно их имена фигурируют в «Илиаде» Гомера наравне с именами базилевсов.

И, наконец, конечный пользователь этого аппаратно-программного, как сказали бы мы сейчас, комплекса: колесничный боец. Ведь это ради него, ради его успеха при выполнении боевой задачи строится колесница, растятся и обучаются кони, ради эффективности его действий управляет ими возница. А потому боец обязан соответствовать по своим качествам тем усилиям, которые затратило на все эти дела общество – в лице представителей описанной «технологической» цепочки.

К чести его, колесничного бойца, заметим: все исторические источники говорят, что он предъявляемым требованиям соответствовал. Подтверждением чему будут и так называемый Эпос Пентаура Египта, и Махабхарата, повествующая о почти фантастических колесничных поединках между Пандавами и Кауравами, и Записки о Галльской войне Цезаря – он застал последних колесничных бойцов Британии, и свидетельства ирландских скел.

Отступление. Самым знаменитым сражением древности, в котором в массовых количествах участвовали колесничные бойцы, была битва при Кадеше (Сирия), случившаяся в 1274 году до н.э. между египтянами во главе с фараоном Рамзесом II и хеттами, предводительствуемыми царём Муваталлисом. Сначала хеттские колесница полностью уничтожили один из египетских корпусов и изрядно погромили второй, предводительствуемый лично Рамзесом – так, что последний чуть ли не в единственном числе (про возникшего источник скромно умалчивает) оказался окружённым врагами. Однако в этой ситуации фараон продемонстрировал своё боевое искусство, сумел вырваться из окружения, собрать остатки своего потрепанного

воинства и организовать оборону. С наступлением темноты стороны разошлись «при своих», хотя каждая не замедлила выступить с победными реляциями.

Кто победил в этой битве – египтологи и хеттологи до сих пор спрят между собой с ожесточением, по сравнению с которым «священные войны» Linux vs Windows и им подобные кажутся детсадовской вознёй. Однако в контексте нашей темы она интересна тем, что являет документированный пример действий колесничного бойца-индивидуала (пусть даже и фараона) против организованной военной машины противников. Или, проводя аналогию с дальнейшим, пользователя «персоналки» против легиона корпоративных рабочих станций.

Если присмотреться к компонентам, определившим успех «колесничных народов», можно увидеть картину, знакомую по IT-индустрии? В основании которой окажутся:

- производители аппаратных средств;
- разработчики движущей их силы – программного обеспечения;
- сисадмины, обеспечивающие бесперебойное функционирование и харда, и софта.

А венец пирамиды – конечные пользователи, применяющие всё это... не буду бросаться громкими словами типа блага человечества, скажем так: для пользы дела. И вот об этих-то пользователях – применителях в основном и пойдёт речь дальше. Для чего из седой древности вернёмся... нет, ещё не в наши дни, а в недавнее прошлое, 80-90-е годы минувшего века.

Становление сословия пользователей

«Когда машины были большими» и работали в пакетном режиме, люди, тем или иным образом связанные с компьютерами, делились на два антагонистических класса. Те, кто имели дело непосредственно с вычислительным комплексом, назывались операторами. Прочие же, те, кто обеспечивал их работой... да вроде никак они специально не назывались. Ибо могли они быть и большими начальниками, отдающими руководящие указания о необходимости разработки того-то и того-то, и программистами, реализующими указания начальников,

или просто заказчиками, которым требовалось кое-что обчислить для решения собственных задач, никак с вычислительной техникой не связанных. Например, вычисления состава мантийного источника для выплавления базальтовой магмы на основе распределения редкоземельных элементов в кристаллизовавшихся из неё базальтов.

Как не трудно догадаться, из всех категорий «прочих» – начальников, реализаторов и заказчиков, – «наиболее прочими» обычно оказывались именно последние (если, конечно, не принадлежали к начальственному сословию или не оказывались востребованными оным в качестве реализаторов). То есть те, ради решения задач которых эти самые вычислительные комплексы по идее и создавались. Впрочем, это обычные бюрократические заморочки, к нашей теме имеющие лишь то отношение, что они-то и оказались прототипом позднейших пользователей.

Появление систем разделения времени и терминального доступа ликвидировало операторов как класс. Но вместо этого вызвало к жизни кастовое деление – на системных администраторов, имеющих доступ к святой святых – управляющему терминалу, иначе консоли, и простых людей, вынужденных довольствоваться терминалами обычными. За этими самыми простыми людьми со временем закрепилось имя пользователей, хотя по жизни они могли быть и программистами.

Окончательно сословие пользователей сформировалось с появлением архитектуры клиент-сервер, состоящих, как легко догадаться из названия, из клиентских машин, именуемых рабочими станциями, на которых выполнялись прикладные задачи, и серверов, обеспечивающих взаимодействие между ними и предоставление общих ресурсов. За рабочими станциями работали пользователи, деятельностью серверов управляли администраторы.

Подчеркну, что все пользователи и систем терминального доступа, и клиент-серверных систем были пользователями профессиональными. Вот только профессии их иногда (в первом случае) или очень часто (в случае втором) могли не иметь ни малейшего отношения к вычислительной технике: это были инженеры-конструкторы, геологи и геофизики, специалисты по финансам... да в общем представители всех сфер человеческой деятельности, в которых требовалась цифровая обработка данных. Разумеется, за рабочими станциями работали и программисты – разработчики как

прикладного, так и системного софта. Но в иерархии клиент-серверной системы они были такими же пользователями, как экономист или геофизик.

В заключение раздела подчеркну главное: применение рабочих станций осуществлялось исключительно в рамках «корпоративной модели» в самом широком смысле слова: под «корпорацией» в данном контексте следует понимать государственные учреждения и частные фирмы, университеты и общественные организации, и так далее. Об индивидуальном их использовании в те годы никто и подумать не мог.

Начало «персонализации»

Параллельно миру терминальных, а затем и клиент-серверных систем зарождается и развивается мир систем персональных – естественно, со своими пользователями. Чем же они отличались от пользователей рабочих станций? А именно персональным использованием своей техники и отличались.

Отступление. Название «персональный компьютер» (первоначально IBM PC – имя собственное первой персоналки производства одноимённой фирмы, в последующем просто PC или ПК) исторически закрепилось за машинами с процессорами архитектуры x86. Однако первым их представителем по праву считается Apple II (см. следующую врезку) с процессором MOS Technology 6502. Да и появившиеся позднее Macintosh'и по способу использования ничем не отличались от IBM PC. Так что в настоящей статье жаргонный термин «персоналка» применяется ко всем компьютерам универсального назначения, ориентированным на использование «в личных целях». Антитеза ему – рабочая станция, то есть специализированная машина для решения одной определённой задачи.

Типичным пользователем рабочей станции, как уже говорилось, был профессионал в своей области, трудящийся в организации, где выполнял определённый круг обязанностей, определяемых его «должностной инструкцией»: проектировал самолёты в CAD'e, генерировал трёхмерный рельеф для «облёта территории» в ER Mapper, обрабатывал данные сейсморазведки, проектировал базы данных результатов Международной программы глубоководного бурения или обеспечивал наполнение их контентом, и так далее.

Отступление. Начало эры персональных компьютеров обычно исчисляют с момента выхода Apple II в 1977 году. Однако, строго говоря, за точку отсчёта надо брать 1979 год – год появления первого табличного процессора VisiCalc. Именно он превратил «игрушку для гиков» в рабочий инструмент инженеров и научных работников, с одной стороны, и в средство для финансовых расчётов и делопроизводства – с другой.

Разумеется, не все пользователи рабочих станций занимались столь глобальными задачами – было немало и более приземлённых, типа бухгалтерского учёта или подготовки всякого рода документации. Однако для любой из них в данном учреждении существовала единственная специализированная программа. Сверх чего требовалось разве что средства для сочинения материалов бюрократического характера и работы со служебной корреспонденцией. Напомню, что от обязанностей администрирования даже своей локальной машины пользователь рабочей станции был не просто освобождён, а скорее даже отстранён: это была вахта системного администратора.

Конечно, я несколько утрирую, и существовали ситуации, когда всё было не совсем так (а иногда и совсем не так). Но в общем случае рабочая станция была машиной «одной задачи», а её пользователь, говоря словами генерала Карпентера из известного рассказа Альфреда Бестера, – «закалённым и отточенным орудием» для её выполнения.

Персональные же компьютеры почти с первых дней своего появления стали в массовом количестве применяться пользователями-индивидуалами. Почти – потому что на самом деле широкое распространение персональных компьютеров началось с того времени, как для них были разработаны основные пользовательские приложения, позволяющие применять эти машины в производственных целях. Ибо первые пользователи «персоналок» были точно такими же профессионалами, как и пользователи рабочих станций, и, подобно последним, далеко не всегда связанными с компьютерными технологиями. Среди них были научные работники и инженеры, врачи и юристы, редакторы и переводчики, наборщики и верстальщики, а иногда даже и программисты.

Общим для всех этих категорий пользователей было применение компьютеров в производственных целях и в индивидуальном порядке – именно в те годы и появилось выражение «кустарь-одиночка с

персональным компьютером». Который, в отличие от Виктор Михалыча Полесова, занимался задачами отнюдь не кустарными, а теми, что раньше требовали мощи корпоративных или государственных вычислительных комплексов.

Отступление. Принято считать, что широкое распространение IBM PC было вызвано их открытой архитектурой, позволяющей производить их в массовых количествах по доступным ценам. Это правда, чистая правда, но не вся правда, ибо массового спроса на «персоналки» не было бы без приложений, позволяющих их эффективно использовать. И такими приложениями стали текстовый процессор WordPerfect, портированный под DOS в 1982 году, и табличный процессор Lotus 123, разработанный специально для этой ОС в 1983 году, после чего он быстро вытеснил VisiCalc. Именно эти две программы позволили кустарям-надомникам эффективно использовать IBM PC в самых различных целях – от юриспруденции до инженерии. Чем и обеспечили востребованность платформы.

Появление Macintosh'a, а затем и Amiga расширила круг пользователей персоналок. В их число влились представители «творческой интеллигенции» и «артистической богемы» – от писателей до музыкантов. Однако структурно этот круг не изменился – его представители были по-прежнему индивидуалами, приобщавшимися к информационным технологиям в порядке личной инициативы. А использовали они свои машины в первую очередь для того, чтобы «не ходить на службу».

Разумеется, началось внедрение персоналок и в корпоративном секторе: PC-совместимые компьютеры широко использовались в делопроизводстве, Macintosh'и – в книгоиздании и так далее. Но этот «корпоративного» использования персоналок лежит далеко за рамками настоящей статьи.

Прародитель «персонализации» Linux'a

Во предыдущих строках ни слова ещё не было сказано о Linux'e. В оправдание могу отметить, что в описанные времена его ещё не существовало. Да, рабочие станции работали часто под управлением UNIX'ов разного рода – но обычно проприетарного. А пользователи развивавшихся в направлении «персоналок» BSD-систем, будучи

преимущественно сотрудниками университетов (LXF, #146), принадлежали скорее к «корпоративному» сословию, нежели к пользователям-индивидуалам.

Зато совершенно классическим примером «кустаря-одиночки с персональным компьютером оказывается» ни кто иной, как Линус Торвальдс. В главе четвёртой, посвящённой возникновению Linux'a, собственно об истории его создания практически не говорилось – ибо только личности с непреодолимым отвращением к писательскому труду, вроде героя Джерома К. Джерома, её ещё не пересказывали. Но у нас нынче разговор идёт об истории не Linux'a, а его пользователей. Так что не будет большим грехом вспомнить в этом ракурсе отдельные её моменты, основываясь на самом авторитетном свидетельстве – книжке самого Линуса, название которой (Just for Fun) Максим Отставнов некогда предложил перевести как «Для прикола».

Так вот, вся история эта началась с того, что, во-первых, Университет Хельсинки приобрел в 1990 году DEC MicroVAX – рабочую станцию с ОС Ultrix, одним из тогдашних проприетарных UNIX'ов, разработанным той же DEC. Во-вторых, в том же году в этот университет поступил горячий финский парень Линус Торвальдс. И в-третьих, этот самый горячий парень сгоряча купил PC с процессором 80386, позволявшем установить и запустить на нём 32-битную операционку, то есть UNIX. Каковой, в лице ОС MINIX разработки профессора Таненбаума, и был им на него водружён.

И всё было хорошо, но MINIX разрабатывалась в качестве «студенческой операционки», и имела массу функциональных ограничений. Частично их можно было компенсировать патчами сторонних разработчиков (в первую очередь Брюса Эванса). Но для их получения требовался выход в онлайн. Вспомним, что Интернета в те годы не было, обмен информацией между разработчиками осуществлялся через телеконференции, BBS, почтовые рассылки, ftp-сервера. Доступ к которым требовал подключения к университетской машине по телефонной линии через терминальную программу. В MINIX она, по свидетельству Линуса, была «хуже всего». А поскольку для него она была критически важной – пришлось писать собственную программу эмуляции терминала. Которая со временем превратилась в ядро Linux, а затем и в одноимённую ОС (вопрос, не GNU ли этот Linux – мы уже обсуждали, см. LXF, #146).

Таким образом, вся ОС Linux была лишь побочным следствием того, что её создатель возжелал получить терминальный доступ к университетской UNIX-машине, не выходя из дому. То есть Линус действовал по той же схеме «персонализации», что и многие другие пользователи-индивидуалы до него, одновременно с ним и после него, в том числе и автор этих строк – вне зависимости от профессии и выдвигаемых ею задач.

Поскольку Линус оказался также изобретателем велосипеда, известного под названием «метода Тома Сойера», то первыми пользователями его системы стали её же собственные разработчики, в основном такие же индивидуалы, как и основатель Linux'a. То есть, казалось бы, «персонализация» этой сферы обещала пойти в том же направлении, что и общая «персонализация» десять лет назад. И за разработчиками системы потянутся разработчики приложений, а вслед за ними и широкие массы пользователей – профессионалов в некомпьютерных областях. Тем более, что последние в те годы могли бы выступить в роли предпоследних – то есть сочиняли бы всякие программки для решения собственных задач: шелл-сценарии Linux'a давали для этого не меньше возможностей, чем BASIC-кодирование на предыдущем этапе общей «персонализации». А с появлением Интернета к тому отпали последние препятствия в виде существовавших ранее коммуникационных ограничений.

Провал первой «персонализации»: технологические причины

Однако этого не произошло, и тому было несколько причин, которые можно сгруппировать в два круга – технологический и психологический. Суть причин первого выражалась сакраментальной фразой, в последующие годы повторяемой как мантра: Linux не был готов для десктопа. С одной стороны, эта неготовность была обусловлена относительной «сложностью» установки любого из существовавших тогда дистрибутивов (а какие они тогда были – описано в главе одиннадцатой). Кавычки в предыдущем предложении я поставил потому, что на самом деле никаких особенных сложностей при инсталляции не было – но для успешного её осуществления требовались некоторые предварительные знания, которыми

пользователи-некомпьютерщики часто не обладали, а с источниками информации была изрядная напряжённость. Напомню, что речь идёт о 95-98 годах прошлого века – когда массовый Интернет как таковой только возникал, был не быстр и дорог.

Но вот после установки часто начинались настоящие сложности, не зависящие от степени информированности пользователя. И одной из главных была настройка графического режима работы, то есть тех самых Иксов, о которых речь пойдёт в части третьей. Ибо тут дело упиралось в отсутствие драйверной поддержки со стороны производителей графических карт. А дальше начинались проблемы со всякого рода периферией – модемами, принтерами и сканерами. Ибо пользователь-индивидуал, в отличие от «корпоративщика», не мог рассчитывать на разделяемые ресурсы «головной организации».

Подчеркну, что в то время под декстопом понималась машина производственного назначения, пусть и домашнего. До переосмысления этого термина как развлекательного медиацентра оставались годы. И потому настройка всякого рода медийного «железа» воспринималась как очень не обязательная опция. Впрочем, проблем с настройкой звуковых карт или телетюнеров в те времена хватало и пользователям Windows.

Все перечисленные проблемы были вполне реальны, и для «нетехнического» пользователя доставляли массу головной боли. Однако при доброй его воле (включающей умение и желание читать соответствующие материалы) в большинстве случаев были или преодолимы, или обходимы, или, наконец, при достаточно сильной мотивации, просто игнорируемы, как проблемы медийные – повторяю, речь шла в первую очередь о том, чтобы ехать, а не о «шашечках».

Однако, распутавшись со всеми этими заморочками, пользователь-некомпьютерщик оказывался перед другими: в свежееинсталлированной системе он обнаруживал в своём распоряжении массу средств разработки, утилит системного администрирования, серверных служб и... считанные пользовательские приложения привычного ему облика. Причём большинство из них были или весьма низесреднего качества от природы, или функционально отставали от своих Windows-, OS/2-, Mac- и даже DOS-аналогов на пару-тройку поколений.

С другой стороны, для решения привычных задач Linux предлагал множество непривычных инструментов. Так, для работы с текстами вместо word-процессора, с которым «текстовики» сроднились со времён WordPerfect'a, напрашивалось применение текстового редактора (которых имелось более дюжины, разной степени продвинутости, как консольных, так и Иксовых), дополняемого набором утилит командной строки, для подготовки оригинал-макетов, вместо программ вёрстки, – текстовые процессоры, начиная с groff и заканчивая TeX'ом, визуализируемые LyX'ом, для обработки изображений, вместо растровых редакторов (PhotoShop в те времена ещё не вытеснил всех своих конкурентов), – imagemagick, дополняемый GIMP'ом с его развитым скриптингом (или наоборот).

Что же до работы в Интернете, что в те годы стало очень актуальным, то вместо универсального клиента, вроде неуклюжего Netscape Navigator'a, существовал обширный ассортимент программ – монофункциональных, но отточенных и закалённых: системы доставки почты, почтовые агенты, браузеры текстового режима, строго соблюдающие спецификации W3C. А уж число web-редакторов до тех пор, пока программы этого класса были востребованы, приближалось к числу текстовых редакторов просто. Не говоря уж о том, что многие из последних (vim, emacs, joe, NEdit) лёгким движением руки превращались в специализированные редакторы html-кода.

Были, конечно, и области, в Linux-мире практически не охваченные – например, работа с векторной графикой. Но тут уж, как говорится, их пользователи ошиблись дверью при выборе платформы. Для многих же иных пользователей инструментов имелось вдоволь – вопрос стоял только об изменении стиля работы, что при должной мотивации и любознательности препятствием не являлось. Однако здесь на первый план выступал круг причин психологического характера.

Провал первой «персонализации»: психологические причины

Как уже говорилось в главе одиннадцатой, первой сферой практического применения Linux'a за пределом круга его собственных разработчиков стали сетевые службы, а его первыми, хоть и не

персональными, пользователями – системные администраторы оных. Каковые, как только прониклись величием идей Linux'a, мгновенно расслоились на два сословия – «элитарщиков» и «эгалитарщиков», к одному из которых примыкали и некоторые разработчики (впрочем, в те времена разработчиками и админами подчас были одни и те же люди). Запомним – их противостояние пройдет красной нитью через всю историю Linux'a и в конце концов скажется роковым образом на его «desktopной» судьбе. Почему?

Опять напомним – что идет о середине – второй половине 90-х годов прошлого века: Linux только вышел за пределы машин своих собственных разработчиков, общедоступных печатных источников информации очень мало (а на русском языке – ещё меньше, от слова «нет вообще»), сайты Linux-тематики в Интернете единичны, СМИ внимания этой системе почти не уделяют. Конечно, Linux, как и любой другой UNIX, сопровождается собственной обильной документацией. Однако для её изучения нужно знать о существовании этой ОС – раз, знать о её документированности – два, и иметь веские мотивы для изучения man- и info-страниц – три. То есть – иметь некий минимум первичной информации общего характера. Который распространялся изустно – как в прямом смысле этого слова, так и через форумы (о существовании которых тоже следовало узнать заранее). И распространялся он именно представителями упомянутых сословий, что накладывало отпечаток на его содержание.

Для админов – «элитарщиков», как легко догадаться, превозидение Linux'a стало основанием для причисления себя к сливкам информационного общества, клубу избранных, куда вход пресловутым «простым» (то есть профессионально с IT не связанным) пользователям следует запретить. И с их лёгкой руки в народе широко распространилось мнение о невероятной сложности установки и настройки Linux'a, а также его пригодности только для серверов, но не для десктопов.

Админы же – «эгалитарщики», напротив, видели своё призвание в приобщении к Linux'у широких народных масс, в том числе и посредством распространения знаний об этой ОС и сферах её применения. Более того, многие энтузиасты эгалитаризма реально пытались внедрять Linux среди «подучётного контингента». И, как правило, не просто безрезультатно – а с результатом негативным.

Потому что, с одной стороны, не всегда представляли себе задачи, стоящие перед пользователями, и не учитывали того, что многие из них средствами Linux'a тогда не решались (а некоторые не решаются и по сей день). С другой стороны, будучи энтузиастами, они с переходом на Linux обещали «златые горы». Что, после знакомства с реальным положением дел, не могло вызвать у пользователей ничего, кроме отторжения.

В результате к концу тысячелетия в массовом пользовательском сознании сложился устойчивый, хотя и противоречивый, образ линуксоида – хама и грубияна, на любой вопрос по делу отвечающего заклинанием RTFM в обрамлении обценной лексики. Но при этом беспрерывно бухтящем о «космических кораблях, бороздящих просторы Большого театра». Причём многие пользователи, подвергнутые принудительной линуксизации, совершенно искренне считали, что такое бухтение прописано в должностной инструкции линуксоида. И очень удивлялись, если его не слышали.

Самое смешное, что образ этот сложился к тому моменту, когда Linux впервые стал пригоден к использованию в мирных (то есть не разработческих и не административных) целях, а именно к 1998 году. Но прежде чем говорить об этом, необходимо сделать небольшое отступление.

Linux: «персонализация снизу»

Итак, к концу 90-х годов прошлого века казалось, что все попытки «персонализации» Linux блистательно провалились. Как провалились незадолго перед тем попытки «персонализации» проприетарных UNIX'ов, о чём говорилось в главе девятой. Однако не тут-то было: в 1998 году начинается новый виток этого процесса. И связан он с выходом первой версии Linux Mandrake, созданной Гаэлем Дювалем сотоварищи. Именно тогда Linux впервые обратился лицом к «простому» пользователю. И в ответ «простой» пользователь впервые обратился в Linux'у – уже сознательно, а не по принуждению сисадмина.

Конечно, не только появление Mandrake было тому причиной, но и множество более иных факторов. О них говорилось в соответствующих статьях исторического цикла, так что повторяться не

буду. Подчеркну лишь, что их кумулятивный эффект вызвал Linux-бум рубежа тысячелетий, нашедший своё отражение, в частности, многочисленными публикациями в компьютерных СМИ, писавших о Linux'e как о «UNIX'e с человеческим лицом». Что привлекло к этой ОС внимание многих «некомпьютерных» пользователей «персоналок», которые отныне могли полагаться не только на мнение «одного знакомого компютещика», но и на источники информации, казавшиеся независимыми.

Увы, среди авторов этих источников по прежнему было немало энтузиастов «человекомордия» Linux'a и его любвеобилия к пользователям. В результате чего у последних часто складывалось впечатление, что Linux – «та же Windows, только бесплатная». Какое рассеивалось при первом же столкновении с реальностью, оставляя осадок типа: заманили и бросили. После чего новообращённые линуксоиды столь же массово разбегались в разные стороны, подобно хлопцам пана атамана Грицько Таврического.

Однако разбегались не все. Многие «некомпьютерные» специалисты, убедившись на практике в эффективности инструментов Linux'a для решения их задач, оставались. И среди оставшихся, как ни странно, были не только инженеры и прочие «технари», но и гуманитарии. В первую очередь – юристы и переводчики. Что, впрочем, легко объяснимо: именно они наиболее чувствительны к «легальности» своего производственного инструментария.

Отступление. Так чем же привлекал Linux десктопных пользователей? Во-первых, конечно, ореолом новизны и даже таинственности: сейчас трудно воспроизвести чувство, с которым читались первые «толстые» книги об этой ОС. Как, впрочем, и о UNIX вообще – резкой грани между ними ещё не проводилось. И авторы-линуксоиды всегда считали обязательным подчеркнуть, что Linux – это в первую очередь одна из UNIX-подобных систем, а потом уже всё остальное. Так вот, чувство это можно было сравнить только с тем, которое испытываешь при чтении хорошего детектива ранее неизвестного автора.

Однако была и вторая, практическая, причина. Меня, например, Linux некогда подкупил эффективностью работы с текстами. Во второй половине 90-х годов прошлого века мне по долгу службы приходилось составлять много проектов, отчётов и тому подобных

«научно-административных» документов, вся фактографическая сторона которых была мною написана и опубликована в виде журнальных статей, в сборниках трудов всяческих совещаний и так далее. Естественным желанием было воспользоваться методом «ножниц и клея», или, говоря современным языком, *copy-and-paste*. Средства предлагавшиеся текстовыми процессорами, выглядели в этом плане очень убого, требуя массы лишних телодвижений (впрочем, и нынче ситуация ничуть не изменилась). А вот комплекс утилит типа *find*, *grep*, *cat*, *split* и *sed*, плюс любой развитый текстовый редактор, решали эту задачу легко и непринуждённо.

Таким образом, следствием Linux-бума было формирование сообщества «простых» пользователей, применяющих персоналки с Linux'ом для решения своих производственных задач. И на протяжении всей первой половины нулевых годов оно росло и крепло, пока...

Применители и потребители

... пока компьютерный мир не начал меняться. То есть начал меняться он давно, с экспансией сначала мультимедии, затем массового Интернета и, наконец, вследствие широкого распространения компьютерообразных устройств, которые в народе ласково именуют гаджетами.

Отступление. Как известно, термин *гаджеты*, в отличие от многих других, имеет однозначный перевод на русский язык, и означает: «Гад же ты!» Именно эти слова сказала своему мужу жена одного компьютерщика, когда услышала, сколько стоит его новый наладонник.

А параллельно менялся и мир окружающий. Профессия «кустаря-одиночки с персональным компьютером» отмирала – в причины этого явления сейчас вдаваться неуместно. Однако остаётся фактом, что одни бывшие линуксоиды-надомники начинают переквалифицироваться в офисных менеджеров, другие находят применение своим знаниям и умениям в администрировании корпоративных систем, третьи же, кому повезло больше – становятся штатными сотрудниками организаций своего профиля.

Соответственно, у них сменяется система приоритетов. На работе они работают в той системе, в которой прикажут (иногда эта система может оказаться Linux'ом). Дома же... дома они просто отдыхают. И

компьютер из производственного инструмента превращается в медийно-развлекательный центр.

Нет, конечно, линуксоиды-одиночки, продолжающие индивидуальную производственную деятельность, до конца пока не вымерли, но сильно уменьшились в числе. Настолько, что ныне, когда речь заходит о компьютере для дома, подразумевается не машина для работы с текстами, графикой, не компьютер в буквальном смысле – то есть «решатель» счётных задач, а нечто, призванное проигрывать музыку, показывать кино, выходить в Интернет и общаться в социальных сетях. С наступлением десятых годов эта тенденция стала явной, требовалось лишь её терминологическая фиксация. И появилось деление всех тех, кого ранее гуртом записывали в пользователи, на потребителей контента и его создателей.

С первым термином трудно не согласиться: назвать человека, околокомпьютерные интересы которого сводятся к социальным сетям и просмотру Youtube, пользователем персонального компьютера – как-то язык не поворачивается. Более уместно для него именование просто «потребитель». Ибо он потребляет специально предназначенный для него контент точно так же, как пиво с попкорном.

А вот второй термин вызывает активное неприятие, ибо открытым текстом подразумевает, что все те, кто не «потребители», только и озабочены лабанием контента им на потребу. И потому последних могижан-индивидуалов можно назвать «применителями» – ибо они заняты применением персоналок со всякими Linux'ами для решения своих пока ещё производственных задач.

Только что я сказал о линуксоидах-применителях в контексте, допускающем установки между ними тождественного равенства. Разумеется, это не так: мне прекрасно известно о существовании применителей Windows или MacOS. Я даже догадываюсь, что их несколько больше, чем применителей-линуксоидов. И рискну высказать ещё одну крамольную мысль: в ближайшее время их, относительно последних, может стать ещё больше.

Обоснование этого утверждения потребовало бы отдельной главы, которая ещё не написана – и которую я пока писать не собираюсь, ибо пока надеюсь на лучшее. Однако перед многими действующими линуксоидами-применителями реально может встать выбор одного из трёх вариантов:

1. «лучшие из лучших» пользователей Linux (то есть, если следовать логике кальвинистской морали, самые богатые) обзаведутся Mac'ами; собственно, уже обзаводятся;

2. «лучшие из худших» (то есть в меру обеспеченные и законопослушные граждане, которых, надеюсь, среди читателей этого сочинения большинство) приобретут лицензионную Windows;

3. «худшие из худших» – те, что, подобно попам, студентам и офицерам, вистуют на девятерной (соответственно, из жадности, от бедности или по пьяни), вернутся на Windows пиратскую.

И не только потому, что их к этому подвигнет принудительная сила реальности. Нет, к этому подталкивают сами разработчики и майнтайрнеры многих современных дистрибутивов. Ибо в момент, когда Linux на определённом витке диалектической спирали доходит до состояния готовности к созданию законченных систем для применителя, они начинают активное внедрение очередных новшеств, выводящее весь пар через гудок.

В своё время Андрей Боровский в своём блоге написал материал с умышленно провокационным заглавием – «Почему Линукса нет и не будет на десктопах», главный тезис которого сводился к тому, что Linux битву за десктопы проиграл. Это не так. Linux не проиграл битву за десктопы. Он от неё постоянно уклонялся именно в те моменты, когда, следуя завету Бонапартия, в драку следовало ввязываться – «а там будет видно».

Вот уже полтора десятилетия по Сети гуляют оценки количества пользователей Linux'а по отношению к числу «персоналок» вообще. И все эти годы оценки они не меняются, составляя 1-2%. Что рассматривается как свидетельство отсутствия роста «десктопной» доли этой ОС. На счёт «доли» – согласиться можно, хотя каковы источник таких оценок – тайна сия велика есть. Но предположим, что источники эти вполне компетентны. Хотя эмпирические наблюдения показывают, что абсолютное число «настольных» пользователей Linux'а растёт. Только среди моих личных и виртуальных знакомых оно за те же пятнадцать увеличилось минимум на порядок.

Объяснение этому парадоксу легко найти, если вспомнить, что за то же время число пользователей персональных компьютеров вообще увеличилось порядка на два. Причём число потребителей росло опережающими темпами по сравнению с количеством применителей.

Превратившись в бытовой прибор, персональный компьютер стал своего рода предметом мебели современной квартиры, используемым для разговоров по скайпу, сидению в социальных сетях, потребления всякого рода мультимедии и тому подобных занятий, далёких от производственной деятельности и не очень располагающих к применению Linux'а. Так что остаётся только удивляться сохранению процента линуксоидов на протяжении времени, столь длительного и столь насыщенного переменами в отношении к «персоналкам».

Возникает глубокое подозрение, что большинство разработчиков Linux и майнтейнеров его дистрибутивов панически боятся того, что Linux станет по настоящему десктопным. То есть в тотально придёт на машины применителей. Ведь тогда им пришлось бы отвечать за тех, кого приручили. А не с азартом чукчи хирурга восклицать: «Опять ничего не получилось!» И начинать перекраивать всё заново и по-живому. Обрекая тем самым Linux на участь испорченной и урезанной копии потребительских систем.

Заключение

Я планировал в этой главе говорить не о тенденциях развития Linux'а, а об эволюции состава его пользователей. И уж тем более не хотел заканчивать её на минорных тонах. Так что попробую найти поводы для оптимизма – их у нас (пока?) есть. Потому что, во-первых, не все майнтейнеры занимаются кардинальными «улучшениями» системы – некоторые уделяют внимание и развитию инфраструктуры, равно пригодной для разработчиков и применителей их разработок. Ярким примером таких дистрибутивов является Ubuntu, вокруг которой выросло большое количество «применительских» решений.

А во-вторых, потому,

Что есть ещё на свете горы, Куда так просто убежать.

То есть BSD-системы – последний рубеж применителей в мире UNIX-подобия.

Часть III. История интерфейсов

Глава двадцать первая. История shell'ов

Долгое время первым интерфейсом, с которым сталкивался применитель любой POSIX-совместимой ОС, был интерфейс командной строки: именно его он видел после ввода пароля и логина в текстовой консоли. И с ним же последним он расставался, выходя из системы с помощью одной из команд завершения работы. Нынче это обычно не так, вход в систему обычно выполняется в графическом режиме, через какой-нибудь дисплейный менеджер. Однако интерфейс командной строки (Command Line Interface, далее просто CLI) по прежнему остаётся Интерфейсом номер один для многих применителей. Почему его история в этой части книги и должна быть рассмотрена первой.

Конкретные реализации CLI представлены классом программ, именуемых командными интерпретаторами, командными оболочками или просто шеллами (shell – раковина, скорлупа). Большая часть командных оболочек делится, на основе синтаксиса интерпретируемого ими языка, на две группы – sh- и csh-совместимые. На самом деле различия между ними синтаксисом команд не исчерпываются, а лежат глубже – в подходе к обработке командных конструкций, но к истории, которую мы в данный момент обсуждаем, это отношения не имеет.

Появление первого шелла

Первой командной оболочкой для первозданного UNIX'a была программа, которая сначала именовалась, за отсутствием других, просто shell. Она была написана в 1978 году Стефаном Борном, вследствие чего за ней позднее закрепилось имя «шелл Борна» (Bourne shell). Язык её интерпретатора был основан на Algol 68, что определило мощные по тем временам возможности составления сценариев: достаточно сказать, что многие команды, позднее вошедшие в золотой фонд классических UNIX-утилит (породивших, в свою очередь, BSD- и GNU-утилиты), первоначально были реализованы именно как shell-скрипты. Примеры этому можно найти в

книге Рассела Сейджа *Приёмы профессиональной работы в UNIX*, изданной около тридцати лет назад, но в методическом отношении представляющей интерес и по сей день.

Однако, средства интерактивной работы в первозданном шелле Борна были, мягко говоря, не очень хорошо развиты.. В известной таблице сравнения командных оболочек для UNIX, некогда составленной Брайаном Блэкмором (Brian Blackmore), во всех её строках, имеющих какое-либо отношение к интерактивности, можно видеть красноречивое **NO** . То есть в ней не было почти ничего из того, к чему мы привыкли с молодых линуксовых ногтей: ни автодополнения, ни истории команд, ни возможности редактирования командной строки, ни даже возможности изменить вид приглашения.

Да, мы знаем, что потом всё это постепенно и по очереди будет появляться – сначала в шелле Корна (`ksh`), и в шелле Альмквиста (`ash`), затем изобильно в `bash`'е (возрождённом шелле Борна), и, наконец, в `zsh`, ставящем на сегодняшний день последнюю точку в развитии командных оболочек. Но сейчас-то мы находимся в далёких 70-х годах прошлого века, когда UNIX только вышел из родительских пенат AT&T (см. главу первую) и отправился в (почти) свободное плавание по университетам Америки, Европы и сопредельных стран вроде Австралии. Пришвартовавшись, в частности, и в университете Беркли, штат Калифорния.

Появление C-Shell

В Университете Беркли, как мы помним по главе второй, в это время всюду разрабатывался собственный вариант UNIX, который позднее получит имя BSD4.4 и ляжет в основу всех свободных операционок берклианского семейства. И один из перворазработчиков BSD, Билл Джой, которому мы обязаны также текстовым редактором `vi`, уже в 1979 году предложил свою командную оболочку, получившую имя C-shell (или просто `csh`).

Почему? Если Борн при создании `shell`'а опирался на язык Алгол, то Джой для языка своего шелла применил синтаксис, сходный с таковым языка Си, исконного для UNIX. Это сделало оболочки `sh` и `csh` несовместимыми на уровне сценариев. Но зато в `csh` было добавлено множество интерактивных возможностей – автодополнение,

буфер истории, средства навигации внутри командной строки и её редактирования, настройка вида приглашения, различие схемы настройки интерактивного и неинтерактивного шелла... Короче, всё то, что потом в той или иной мере инкорпорировали shell-совместимые оболочки, включая `bash` и `zsh`. И что ныне кажется нам неотъемлемым атрибутом любого шелла – всё это в конечном итоге происходит из `csh`.

Так что C-shell очень быстро стал неременной принадлежностью разрабатываемой в Брекли BSD-системы. Разработчики же коммерческих UNIX'ов пошли другим путём.

Как возрождался шелл

Как только что было сказано, по своим интерактивным возможностям шелл Борна, с одной стороны, оставлял желать лучшего. Со стороны же другой, перед глазами разработчиков проприетарных UNIX'ов был уже C-Shell, существенно более продвинутый в этом отношении. И потому у них появилась потребность обогатить `/bin/sh` средствами интерактивной работы.

Результатом воплощения этой потребности стала оболочка Корна (Korne Shell). Она сохранила совместимость с борновским шеллом на уровне синтаксиса, однако привнесла в него как дополнительные возможности интерпретации команд, так и приёмы, направленные на удобство интерактивной работы – образцом последних и послужил C-Shell. В итоге именно `ksh` лёг в основу стандарта POSIX, которому должны удовлетворять командные оболочки совместимых с ним систем.

Следует заметить, что соответствие этому стандарту – один из критериев отнесения некоей ОС к семейству UNIX или UNIX-подобных. В частности, именно такой стандартизированный шелл должен (был до недавнего времени) вызываться при исполнении общесистемных сценариев инициализации любой POSIX-системы. Сам же по себе POSIX shell – некая мифическая абстракция. Ближе всего ему соответствует `ash`, принятый в качестве стандартной оболочки пользователя во FreeBSD (под условным именем `/bin/sh`) и в NetBSD (а также широко используемая во всяких минидистрибутивах Linux).

Шеллы и Борна, и Корна не были свободно распространяемыми программами. Поэтому ни тот, ни другой не могли использоваться в таких ОС, как все BSD или Linux, хотя свободная реализация оболочки

Корна (`pdksh`) и была создана. Однако, как и ее прототип, шелл Корна, она, несколько разившись относительно первозданного Bourne shell, обладала интерактивными достижениями, уже далекими от идеала. Столь же слабы они были и в `ash`. И потому наиболее широкое распространение из всего `sh`-совместимого семейства получила оболочка `bash` (что расшифровывается как «ещё одна оболочка Борна», «заново рожденный шелл» и тому подобным образом), разработанная в рамках проекта GNU.

Популярность `bash` в немалой степени была обусловлена его интерактивными возможностями – он аккумулировал все достижения интерактивной мысли `sh`- и `cs`h-совместимых оболочек, прибавив к ним немало собственных. И умудрившись при этом сохранить базовую совместимость с POSIX shell. Конечно, многие его собственные особенности (так называемые «`bash`'измы») выходили за рамки этого стандарта. Однако для соответствия оному был предусмотрен режим совместимости – то есть `bash` был способен эмулировать стандартный POSIX shell.

Однако главным для популярности `bash` было то, что эта оболочка оказалась тесно интегрирована с операционной системой Linux: именно `bash` волею судеб стал одной из первых программ, которые Линус запустил поверх своего новосозданного ядра. И потому идеи `bash`-скриптинга пронизали Linux до самых его оснований. Ну а для совместимости использовался тот самый режим эмуляции: в Linux файл, запускающий POSIX shell (по стандарту он должен именоваться `/bin/sh`), являет собой жесткую или символическую ссылку на реальный `/bin/bash`. Последний же, будучи вызванным таким образом, полностью воспроизводит функционально стандартный POSIX shell (разумеется, путем утраты своих продвинутых функций).

История `tcsh`

Как уже говорилось, оболочка `cs`h привнесла в шеллы элементы интерактивности. Однако и её возможности в этом отношении были далеки от идеала. Что особенно почувствовалось в начале 90-х годов, когда, с одной стороны, произошло освобождение BSD-систем в лице NetBSD и FreeBSD от тяжёлого наследия проприетарного режима лицензирования. А с другой – началось победное шествие Linux'a с его

стандартной оболочкой GNU `bash`, обходящей древний `cs`h на несколько корпусов.

И тогда разработчики FreeBSD вспомнили об оболочке `tcsh`, которая, основываясь на синтаксисе C-shell, с давних времён разрабатывалась сначала Кэном Григом в университете Карнеги-Меллона, а затем Полом Плэйсвэем в университете Огайо. Чем она была примечательна? Сейчас увидим.

Изначально `tcsh` создавалась по образу и подобию командного интерпретатора операционной системы TENEX -- собственно, имя её и означает TENEX `cs`h. А особенностью TENEX -- древней, ещё до-UNIX'овой, операционки (из недр которой, кстати, происходит и знаменитая «собака» в адресах электронной почты) были чрезвычайно длинные команды, да ещё и с избыточными словами «для ясности». С такими командными директивами было бы трудно работать без развитых средств навигации и редактирования командной строки. Каковые и стали отличительными особенностями её командного интерпретатора, получившего имя TENEX C-shell.

Разработчики FreeBSD, взяв за основу TENEX C-shell, адаптировали его для своей операционной системы. В которой он и утвердился в качестве стандартного login shell администратора. В OpenBSD же и DragonFly `tcsh` изначально стал шеллом по умолчанию «для всех». Применяется он и в таком юзер-ориентированном отпрыске BSD-клана, как PC-BSD. Более того, первое время `tcsh` выступал шеллом по умолчанию в MacOS X, пока его не заменили на `bash`.

Однако и поныне имя `/bin/csh` (как и соответствующие ему конфигурационные файлы `/root/.cshrc` и `$HOME/.cshrc`) сохранилось в файловой иерархии всех BSD-систем как реликт предшествующей эпохи. Однако теперь оно соответствует не самостоятельной оболочке, а было лишь жесткой ссылкой на тот же `tcsh`. Поведение же последней по умолчанию абсолютно идентично изначальному C-Shell. Это вызвано не эмуляцией предшественника, как в случае с `bash`, запускаемых в качестве `/bin/sh`, а исключительно настройками — точнее, почти полным отсутствием таковых по умолчанию.

Наконец, Z-Shell

Как известно, Z — последняя буква латинского алфавита. И её наличие в имени следующего нашего персонажа, Z-Shell (или просто `zsh`) призвано символизировать то, что эта оболочка представляет собой последнюю ступень в развитии командных оболочек вообще. Хотя на самом деле происхождение её иное. Она разрабатывалась первоначально, начиная с 1990 года, Паулем Фальстадом (Paul Falstad), в ту пору студентом Принстонского университета. Аспирантом же одного был некто Zhong Shao, логином которого в университетской сети был `zhong`. Вот отсюда-то, как гласит легенда, и взялась буква Z. То есть можно предполагать, что, как это часто бывало в истории свободного софта, она появилась тут "для прикола".

Тем не менее, и трактовать её в первом смысле тоже есть основания. Ибо существует мнение (и не только мое), что в `zsh` нашли свое воплощение все прогрессивные тенденции таких развитых оболочек, как `bash` и `tcsh`. И, ознакомившись с его возможностями, с этим трудно не согласиться — в `zsh` есть все, что было хорошего в тех обеих оболочках, но, если так можно выразиться, в превосходной степени.

Ныне Z-shell развивается в рамках самостоятельного проекта сообществом энтузиастов (Zsh Development Group) при координации Петера Стефенсона (Peter Stephenson), являющегося также автором большей части документации проекта. В отличие от `bash`, прямого (как, впрочем, и косвенного) отношения к GNU `zsh` не имеет, и распространяется под собственной лицензией BSD-стиля, а, следовательно, является полностью свободной программой.

Глава двадцать вторая. Из истории Иксов

Хотя, как было отмечено в предыдущей главе, CLI в полной мере сохраняет своё значение в современных UNIX-подобных системах, жизнь как «обычного пользователя», так и типичного применителя оных протекает в основном в рамках интерфейсов графического режима. А во всех этих операционках в основе его лежит оконная система X (X Window Sysytem), именуемая в народе просто Иксами. Так что настала пора обратиться к её истории.

Вступление

Нынешний Linux в пользовательской своей ипостаси не мыслим без современной инкарнации оконной среды X – Xorg. Немыслим настолько, что начинающие пользователи этой ОС часто ставят между ними знак равенства, не отдавая себе отчёта в том, что, как говорится в народе, X – он и в Африке... Икс. Точнее, един во всех UNIX'ах и UNIX-подобных операционных системах.

В своё время немало было сломано копий о тему: как должна называться операционная система, одна из главных героинь этой книги – Linux или GNU/Linux. И арьергардные бои на этом ристалище возникают временами и по сей день. Автор сих строк немало высказывался на эту тему – в том числе и в главе четвёртой данного сочинения. Так что в рамках темы главы нынешней замечу только: если уж так неймётся прикрутить к имени Linux какой-либо префикс, то с куда большим основанием это может быть префикс X.

Ибо теоретически вполне можно представить дистрибутив этой системы без компонентов, разработанных в рамках проекта GNU – есть и практически близкие к этому реализации, например, Tiny Core Linux. А вот без оконной системы X, менеджеров её окон, интегрированных десктопов и приложений графического режима говорить о каком-то десктопном Linux'e просто нелепо – как сказала поэт-линуксоид Алиса Деева,

Это разговор не в пользу бедных -- В пользу продавцов клавиатур.

И это – первая причина поговорить об истории Иксов. Вторая же – то, что разработчики проекта Wayland действительно обещают сделать Иксы достоянием истории. Которую надо рассказать, пока мы ещё чего-то помним и нас ещё не замочили.

Доисторический период

Я не буду закапываться в глубокую праисторию и вспоминать о первых мышах, бегающих по закоулкам графического интерфейса в Исследовательском центре Hewlett-Packard в Пало-Альто – редкий лентяй не осветил этот вопрос всесторонне.

Не буду я рассматривать и детективную проблему – потибрили ли из Пало-Альто идею графического пользовательского интерфейса (по нашему, по бразильскому – GUI, именно этот термин и будет для краткости применяться в дальнейшем). И если потибрили – то кто же сделал это первым. Потому что по сему поводу исчерпывающе высказался лирический герой Венечки Ерофеева в поэме «Москва-Петушки»:

Никто этого не знает, и никогда теперь не узнает. Не знаем же мы вот до сих пор: царь Борис убил царевича Димитрия или же наоборот?

А начну я доисторический период с первых графических интерфейсов UNIX. Которая, как известно, зародилась и долгое время развивалась в качестве чисто текстовой. Настолько чисто, что сама идея прикручивания к ней какого-либо GUI воспринималась как ересь. Но, однако, идея эта была реализована.

Первой на это поприще вступила фирма Sun, о чём говорилось в главе третьей), изначально тесно связав свою SunOS со средствами поддержки GUI собственной разработки. Превым таким GUI'ём была SunView (Sun Visual Integrated Environment for Workstations, первоначально называвшаяся SunTools) – оконная система, возникшая едва ли не одновременно с самой SunOS в первой половине 80-х годов.

От всех последующих оконных систем, ориентированных на UNIX, она отличалась тем, что большая её часть поддерживалась ядром. Она включала в себя набор стандартных пользовательских приложений, таких, как текстовый редактор, почтовый клиент, инструменты для настройки. И, наконец, она являлась частью стандартной поставки операционной системы. По-видимому, именно в SunOS впервые была реализована интеграция операционной системы, GUI и пользовательских приложений, получившая дальнейшее развитие не только в мире UNIX, а и в более иных операционках, доживших до наших дней (а имена их читатель и сам ведаёт).

На смену SunView в середине 80-х годов прошлого века пришла NeWS (Network extensible Window System) – более сложная система, основанная на PostScript. Однако получить широкого распространения она не успела.

Наконец, в конце 80-х годов для SunOS была разработана оконная среда OpenWindows. Сначала она поставлялась как отдельное дополнение, а потом была включена в состав операционной системы. Однако и она не снискала популярности.

Ибо тогда же, в конце 80-х годов начинается смена аппаратной платформы для SunOS: процессоры Motorola 68XXX в серверах и рабочих станциях заменяются на «камни» собственной разработки и производства, RISC-процессоры SPARC. Для которых разрабатывается и новая операционная система, названная Solaris 2. А с её приходом происходит и отказ от всех GUI собственной разработки. Они сменяются ставшими стандартными для UNIX мира оконной системой X и рабочей средой CDE. О последней речь пойдёт в главе про десктопы. А вот к истории первой обратимся сейчас.

Рождение Иксов

Программные средства обеспечения работы в графическом режиме можно условно разделить на две части – ту, что обеспечивает взаимодействие с аппаратурой, и собственно интерфейс с пользователем. Разделение это не строгое – только что я говорил о SunView, в которой эти части были неразрывны. Однако магистраль развития GUI лежала в области разделения этих частей, что наиболее

последовательно было реализовано в оконной системе X (X Window System).

Кстати, о названии. Как гласят легенды древности, некогда существовала операционная система V. Поверх неё работала оконная система W, созданная в Стэнфордском университете Полом Асентэ (Paule Asente) и Брайаном Рейдом (Brian Reid). В дальнейшем она была портирована на UNIX, хотя и работала там, как говорят, очень медленно. Поэтому, когда под UNIX была разработана новая оконная система, её просто маркировали следующей буквой латинского алфавита.

Иксы начали разрабатываться в 1984 году сотрудниками MIT Робертом Шейфлером (Robert Sheifler) и Джимом Геттисом (Jim Gettys), к которым скоро присоединился Рон Ньюмен (Ron Newman). Идея разработки заключалась в том, чтобы создать графическую систему, полностью независимую как от аппаратной платформы, так и от операционной системы, даже родительской UNIX, которая бы обеспечивала студентам и преподавателям MIT доступ ко всему тому зоопарку компьютеров, который существовал в стенах этого заведения.

Официальным названием новой системы было – X Window System, причём слово Window появилось в ней более чем за год до его употребления в имени одной графической оболочки для DOS, которую тогда никто не воспринимал всерьёз ввиду непригодности к практическому использованию. И для которой оно стало ни чем иным, как торговой маркой, охраняемой законом.

Тем не менее позже в кругах, далёких от UNIX, некоторое распространение получило словосочетание X Windows. Однако, говоря словами всё того же героя Венечки, «это позорно и преступно». И истинный линуксоид до такого никогда не опустится – и не из-за пиетета перед законами. А вот более краткое X System, несколько жаргонное X11 (почему именно 11 – станет ясно со временем) и, особенно, просто X, не только допустимо, но и широко используется краткости для.

На русский язык официальное название X Window System переводилось столь же официально – Оконная система X. Однако используется оно редко – только в особо торжественных случаях. Ибо практически сразу было вытеснено кратким словом «Иксы» – пусть и жаргонным, но адекватным. Его я использовал в этой книге ранее и

буду использовать впредь, поскольку официоз в ней кажется уж совсем неуместным.

Работа над Иксами проводилась ударно-стахановскими темпами. Первая версия новой системы вышла в мае 1984 года, а вышедшая в январе 1985 года версия получила порядковый номер 6. Это привлекло к Иксам внимание фирмы DEC, и Иксы были портированы на её VAX'ы.

Версии Иксов, именовавшиеся X#, сменялись с калейдоскопической быстротой: уже во второй половине 1985 года появляется X9 – первая версия, распространяемая свободно, на условиях так называемой лицензии MIT (до этого институт распространял её за плату). А на рубеже 1985-1986 годов выходит версия X10 – подряд в нескольких вариантах, так называемых реализациях: X10 – декабрь 1985 года, X10R2 – январь и X10R3 – февраль 1986.

Версия X10R3 получает широкое распространение. Кроме VAX, она портируется на машины Hewlett-Packard, рабочие станции Apollo, Sun и ряд других. В связи с этим возникла необходимость сделать её окончательно независимой от аппаратуры, что и было выполнено с привлечением сил DEC в рамках версии X11.

Разработка версии X11 заняла необычно много времени для этого проекта – более полугода: финальный вариант её вышел аж в сентябре 1986 года. Ход работы над ней широко обсуждался в Сети благодаря открытым спискам рассылки. Это был пример первого в истории масштабного проекта, разрабатываемого распределенным коллективом программистов, связанных лишь Интернетом. Таким образом, разработка X11 послужила прообразом современных крупномасштабных проектов FOSS.

Длительность разработки X11, тщательность тестирования и широта обсуждения проекта привели к тому, что эта система стала уникальным для софтверной индустрии долгожителем: номер главной версии с тех пор не менялся ни разу, добавлялись только номера реализаций. С формальной точки зрения те Иксы, с которыми мы имеем дело сейчас (то есть Xorg), также являются представителями ветки X11. Именно поэтому для X Window System широко распространилось сокращённое именование – X11.

Иксы в своей 11-й ипостаси продолжали свое триумфальное шествие по «железным» архитектурам и их операционным системам. Только что я говорил, что ради Иксов Sun прекратила собственные разработки в области графического интерфейса. Та же судьба постигла и все иные графические системы для UNIX – даже имена их ныне забылись. Иксы стали стандартным средством обеспечения работы в графическом режиме всех без исключения UNIX'ов и UNIX-подобных операционок.

Кроме того, Иксы были портированы на VAX/VMS фирмы DEC, OS/2 от IBM и даже, страшно сказать, на Windows. Хотя о последнем факте и не любят говорить вслух.

Проект приобрёл такие масштабы, что для управления им в 1988 году потребовалось создать специальную организацию, названную Консорциум X MIT (MIT X Consortium), которая объединила в своём составе как университетские круги, так и компании, разрабатывающие коммерческие решения. В 1993 году на смену ему пришёл X Consortium, выпустивший в мае 1994 года X11R6, реализацию, просуществовавшую более 10 лет: следующей, X11R7, суждено было появиться только в конце года 2005! И различные минорные её версии (последняя по времени – X11R7.7, появившаяся в июне 2012 года) используются в дистрибутивах Linux и BSD-системах по сей день.

XFree86 и другие

Сама по себе оконная система X версии X11 представляла из себя набор открытых спецификаций (reference implementation), доступных под лицензией MIT, на основе которых предлагалось создавать конкретные реализации – X-сервера с сопутствующими компонентами. Которые уже могли распространяться на любых условиях, в том числе и коммерческих.

Этим предложением не замедлили воспользоваться разработчики программного обеспечения для UNIX-систем. В результате чего появился ряд проприетарных X-серверов, обеспечивавших графический режим в столь же частнособственнических UNIX'ах. Однако наибольшую известность получила свободная реализация Иксовых референсов – XFree86.

Прототип проекта XFree86 зародился в 1991 году в рамках X11R5: соответствующий спецификациям этого релиза X-сервер был разработан Томасом Роэллом (Thomas Roell) и Марком Снайтили (Mark W. Snitily) для 32-битных платформ на процессорах Intel, получив вполне ожидаемое имя X386. Этот сервер распространялся под проприетарной лицензией фирмы SGCS (ныне SGI), сотрудником которой являлись его авторы.

Это, однако, не помешало Дэвиду Векселблату (David Wexelblat), Гленну Лэю (Glenn Lai), Дэвиду Доуэсу (David Dawes) и Джиму Тсилласу (Jim Tsillas) в 1992 году внести в его исходники коррективы столь существенные, что у них были основания выделить их в отдельную версию, названную X386 1.2E и распространять её уже свободно.

Такое положение создавало определённую путаницу. И потому после обсуждения и обмена мнениями стороны постановили: переименовать новый проект в XFree86. Что по созвучию намекало и его на связь с корнями (X-three-eighty-six), и на характер распространения (X-free-eighty-six), а цифры явным образом указывали также на архитектуру целевых процессоров – x86.

Не пропало и дело исходного проекта: X386 был переименован в Accelerated-X, и Томас Роэлл создал фирму Xi Graphics, которая на протяжении многих лет занималась его продажами, а поддержку осуществляет чуть ли не по сей день. У нас ещё будет повод вспомнить про этот X-сервер.

А пока вернёмся к XFree86. Во вступлении я уже говорил, что своей популярностью среди широких пользовательских масс Linux во многом обязан Иксам – и конкретно именно свободной их реализации. Но, с другой стороны, XFree86 повсеместным своим распространением обязана Linux'у в не меньшей степени. Ибо усилиями Ореста Зборовски (Orest Zborowski) она заработала на Linux'е чуть ли не со дня своего рождения, а именно с апреля 1992 года.

Вообще взаимоотношение XFree86 с Linux'ом на том этапе развития обеих систем в источниках освещено довольно противоречиво. Остаётся не вполне ясным: создавалась ли свободная реализация Иксов уже с прицелом на юную, но активно развиваемую операционку? Или ещё под абстрактный UNIX, а на Linux была

Орестом лишь портирована? Прямых ответов на эти вопросы мне найти не удалось.

Так что в эти детали я вдаваться не буду. Тем более в рамках нашей темы сейчас важнее другое: в октябре 1992 года XFree86 была включена в комплект, который можно назвать первым в истории настоящим дистрибутивом Linux. Им стал SLS (Softlanding Linux System), разработанный Питером Мак-Дональдом (см. главу одиннадцатую). И с тех пор судьбы XFree86 и Linux'a были неразрывно связаны на протяжении 12 лет, вплоть до событий «Великого раскола», о которых я расскажу в следующих разделах.

Однако Linux не была единственной операционкой, в которой реально использовалась XFree86. В недрах архивов проекта можно обнаружить списки издревле поддерживаемых им ОСей. В их числе мы увидим такие, ныне забытые, имена, как Amoeba (первая попытка Танненбаума построить «настоящую» микроядерную ОС) и BSDi (коммерциализированная сестричка FreeBSD), Mach, ушедший в тень MacOS X, и «игрушечную» Minix, благоздравствующую, но «незнаменитую» NetBSD и скандально ослабившуюся SCO, а также SVR3 и SVR4, давно ставшие абстракциями учебников по Computer Science.

А вот следов поддержки FreeBSD в ранних версиях XFree86 найти не удаётся: похоже, что в её портах свободные Иксы появляются во второй половине 1994 года (собственно, одновременно с самой системой портов). Однако с этого момента XFree86 оказывается связанной с FreeBSD не менее тесно, чем с Linux'ом – и со временем мы увидим, что эта связь окажется особенно важной для нашей страны.

Как уже говорилось, изначально XFree86 была основана на X11R5, и так продолжалось во всех версиях 1-й и 2-й веток (нумерация их не имеет никакого отношения к версиям и релизам собственно Иксов). Последней представительницей этой линии была XFree86 2.1.1, вышедшая в мае 1994 года. Однако почти одновременно появляется и новая спецификация Иксов – X11R6, которая ложится в основу новых реализаций X-серверов. В их числе была и XFree86 3.0, увидевшая свет в конце августа 1994 года.

И это были те самые Иксы, с которыми впервые столкнулось большинство из ныне действующих применителей Linux первого и

второго призыва. Те самые, которые появляются в портах FreeBSD. Иначе говоря, те самые, которые стали, как принято говорить, стандартом *de facto* для всех существовавших тогда свободных UNIX-подобных ОС. А со временем – и для несвободных тоже, но это будет не очень скоро.

XFree86: немного о внутренностях

Говоря о XFree86 как об одной из реализаций X-сервера, я был не совсем точен: это была не одна из реализаций, а множество реализаций оного. Откуда их столько взялось? Ответить нетрудно.

Ныне, когда речь заходит о видеоподсистеме и её поддержке в Linux'e, он обычно сводится к священной войне между приверженцами Nvidia и ATI/AMD. От которой дистанцируются резонные применители, любовно поглаживающие в сторонке свои встроенные Intel'я. И подчас вспоминая даже о существовавших не так давно SiS и Matrox. Однако представить себе пестроту видеорешений, сосуществовавших каких-нибудь 15-20 лет назад, может только тот, кто видел её своими глазами.

Ибо это была пестрота восточного базара. На котором можно было встретить видеокарты на любом чипе из выпускавшихся в те годы: и откровенно рабоче-крестьянском Trident, и плебейски-претенциозном Cirrus Logic, и S3 – символе мещанского благополучия, и аристократическом ATI, скромном трудяге – Tseng и блистательном Imagine128, 3DLabs, знаменующем высоты профессионализма, и многих других, имена которых стёрлись в памяти даже ветеранов. И чуть ли не каждому производителю видеочипов в составе XFree86 полагался свой персональный X-сервер, а некоторым их доставалось по два (как S3), а то и по три (как ATI).

Впрочем, производители видеочипов пренебрежительно относились к оказанному им уважению. И мало того, что сами манкировали поддержкой своей продукции, но и не предоставляли информации о ней независимым разработчикам. Что само по себе было объяснимо – конкуренция среди «видеочипмейкеров» тогда была бешеная. Но не оправданно – и безвременная кончина почти всех «фигурантов дела» из предыдущего абзаца стала тому

подтверждением. А выживших заставила в той или иной форме «делиться»...

Но это будет ещё не скоро. А пока работа многих свободных X-серверов – результата «слепого» реинжиниринга фирменных решений – часто оставляла желать лучшего. Вплоть до того, что иногда графический режим просто не удавалось запустить с «родным» сервером. Правда, в составе XFree86 имелись «всечиповые» сервера VGA и SVGA, работающие на любых картах, поддерживающих соответствующие стандарты. Однако наблюдать на дисплее, подключённом к крутейшей видеокарте, что-нибудь вроде 640x480 при 16 цветах доставляла тогдашним пользователям не много радостей.

Конечно, не всё было так мрачно, и большинство карт на типовых чипах (особенно не гипермодерновых) с типовыми же X-серверами из поставок XFree86 работали вполне справно. Однако вопрос поддержки видеосистемы свободными Иксами в 90-х годах стоял весьма остро. И попытки его кардинального решения тем или иным образом предпринимались постоянно.

Первым направлением таких попыток было, разумеется, совершенствование самих X-серверов. На этом поприще особенно прославилась фирма S.u.S.E. – создатель одноимённого дистрибутива Linux, героя главы семнадцатой. Благодаря тесным контактам с фирмой Elsie – производителем высококлассных видеокарт, в том числе профессиональных, её разработчики очень быстро получали информацию о новинках «видеожелеза» и столь же оперативно вносили патчи в свои реализации X-серверов.

Вторым направлением было использование коммерческих X-серверов, вроде упоминавшегося выше Accelerated-X. Сами по себе они не были ни свободными, ни открытыми, но на определённых условиях могли распространяться (почти) бесплатно. Благодаря этому их нередко включали в состав дисковых наборов Linux-дистрибутивов и софта для них, вроде упомянутых в главе про *Linux на Руси* боксов от Walnut Creek и InfoMagic. Правда, тут возникали свои сложности, как лицензионные, так и чисто технические. А для Руси этот путь оказывался закрытым напрочь, потому что тот же Accelerated-X в принципе не поддавался кириллизации.

Наконец, третий путь продемонстрировала в 1998 году маленькая и неприметная тогда фирма Nvidia. Выпустив незадолго перед тем

«народную» 3D-карту Riva 128, она вскоре сопровождала её и фирменным драйвером для работы в Linux'e – драйвером, работавшим безукоризненно.

Я не буду выстраивать причинно-следственных связей. Однако то, что звёздный час Nvidia начался с Riva 128 и её Linux'ового драйвера, остаётся медицинским фактом. На констатации которого я и поставлю запятую в своей истории.

От XFree86 к Xorg

Хронологически привязанную часть прошлых разделов этой главы я закончил на появлении в августе 1994 года XFree86 версии 3.0 – той самой, которой суждено было со временем лечь в основу десктопного Linux'a. В нынешней же главе предлагаю посмотреть, что было со свободной реализацией Иксов дальше.

Между «тройкой» и «четвёркой»

А дальше на протяжении 1994-1996 годов плавно сменились версии 3.1 и 3.2, не прославившиеся какими-либо громкими новшествами. А затем в мае 1997 года выходит версия 3.3, ознаменовавшаяся появлением в ней XFree86 Acceleration Architecture (ХАА). Архитектура эта, как явствует из названия, обеспечивала ускоренный вывод графики – пока ещё двухмерной: как это ни трудно нынче представить, но тогда последняя ещё нуждалась в акселерации. Именно на её почве расцвели пышным цветом те самые X-сервера, привязанные к видеочипам, о которых шла речь в заключительной части прошлой статьи.

За два с половиной года активного развития версии 3.3 она пережила более полудюжины минорных релизов, характеризовавшихся, однако, весьма мажорными нововведениями. Описывать их последовательно было бы скучно, да нынче и проследить их поверсионно сейчас уже нелегко. Поэтому остановлюсь на тех из них, которые сыграли важную роль в постановке под названием «десктопный Linux».

О постоянном совершенствовании X-серверов, обеспечивавших базу для всего остального, я уже говорил. Далее, важным оказалось развитие поддержки шрифтов. Первоначально в XFree86 применялись, во-первых, растровые шрифты (форматов bdf и pcf), во-вторых,

векторные шрифты формата PostScript (ATM Type 1). Сфера применения первых была ограничена – они использовались преимущественно в терминальных программах и текстовых редакторах. Назначение же вторых было, теоретически, вполне универсально. Практически же их широкому использованию мешало. во-первых, низкое качество самих шрифтов, во-вторых – не лучший их рендеринг.

До решения второй проблемы было ещё далеко. А вот в отношении первой в описываемое время произошли определённые подвижки, выразившиеся в появлении поддержки шрифтов True Type. Что сначала дало возможность использовать шрифты от Microsoft, штатно применявшиеся в Windows, начиная с версии 3.1 (Arial, Times New Roman, Courier New и так далее), по чьему-то недосмотру оказавшиеся в почти свободном доступе, хотя и в несколько извращённой форме. А потом оказалось толчком для разработки качественных ttf-шрифтов, уже полностью свободных (Bitstream Vera, DejaVu, Liberation).

Теперь надо сказать пару слов о поддержке устройств ввода – то есть клавиатур и мышей.

О самих по себе клавиатурах говорить нечего – как устройства они поддерживались всеми вариантами Иксов испокон веку. А вот с раскладками для различных языков (точнее, наборов символов) связана забавная история, касающаяся, разумеется, нашей любимой кириллицы.

Ввод кириллицы в кодировках ISO 8859-5 (по историческим причинам изначально принятый в русских вариантах проприетарных UNIX'ов, в частности, в SunOS) и KOI8-R (утвердившейся, благодаря Андрею Чернову aka ache, во всех свободных UNIX-подобных системах), поддерживался ещё в XFree86 2.X, однако методом, считавшимся идеологически неправильным. Полноценная поддержка кириллических раскладок стала возможной только в версиях 3-й ветки – правда, ценой несовместимости с более старыми русифицированными приложениями, например, Netscape Navigator (легендарный предшественник современных Firefox'a и Thunderbird'a).

Параллельное использование более чем одной раскладки клавиатуры (впрочем, в те времена их число ограничивалось двумя) требовало средства переключения между ними. И такое средство ещё в

версиях 2.X было жёстко встроено в саму раскладку. В частности, переключение с латиницы на кириллицу и обратно было привязано к клавише **CapsLock** (а функцию перевода алфавитных клавиш в верхний регистр выполняла комбинация **Shift+CapsLock**).

В XFree86 3.X появилась возможность переназначения переключателя раскладок, чем народ немедленно воспользовался: для переключения с латиницы на кириллицу пошли в ход Windows-подобные комбинации типа **Control+Shift** и **Alt+Shift**, и даже приснопамятные «два шифта».

Очередное отступление, или *Легенда о двух Shift'ax*. Во времена самостийных русификаций DOS и её приложений переключатели латинской и кириллической раскладок бывали самыми причудливыми. Однако, когда русский язык начал официально поддерживаться этой ОС (начиная с версии 4.01), в этом качестве утвердилась комбинация левого и правого **Shift**'ов. И ей суждена была долгая жизнь — но благодаря не DOS'у, а Linux'у.

В первых версиях Red Hat, самостоятельно, без участия российских пользователей, поддерживавших русский язык, выбор его на стадии установки автоматически приводил к тому, что умолчальной становилась кириллическая раскладка (причём об этом нигде не говорилось). Это на стадии создания аккаунтов делало невозможным ввод корректных паролей администратора и пользователя, если не переключиться на латиницу. Переключение это программе установки (и только в ней) осуществлялось комбинацией обоих **Shift**'ов — но это нигде не было документировано. И не одно поколение пользователей Red Hat, а затем и Fedora, имело повод поупражняться в солдатской смекалке, угадывая сначала, почему в ответ на ввод пароля выдаётся сообщение об ошибке, а потом — о способе её устранения. Ибо последние рецидивы данного бага (давно превратившегося в фичу) я наблюдал ещё в Fedora 11.

При этом поначалу наличие «заказного» переключателя раскладок не исключало и использование переключателя встроенного, то есть всё той же клавиши **CapsLock**. Что, казалось бы, никому не мешало — как и наличие бочек с красной и чёрной икрой при входе в магазин Рабиновича на Дерибасовской улице. Однако было сочтено излишеством — и в версии XFree86 3.4 встроенный переключатель раскладок исчез. Что скоро, с появлением средств автоматического

конфигурирования Иксов, доставило много дополнительных развлечений не одному поколению начинающих линуксоидов и берклианцев.

Однако прежде чем отсутствие встроенного переключателя раскладок проявило себя во всей красе, произошла та самая забавная история, о которой я недавно упоминал. Суть её была в том, что в один прекрасный момент, с выходом XFree86 версии 3.3.2, русская раскладка для кодировки KOI8-R без всяких видимых причин просто перестала работать. Точнее, она делал вид, что работала, но выводе получалась абракадабра из русских букв.

Дело оказалось в том, что раскладка для кодировки KOI8-R подменялась таблицей для ISO 8859-5. А причина заключалась в банальной опечатке в исходниках главной Иксовой библиотеки – xlib. Более ни на что эта опечатка не влияла, и потому её выявление заняла довольно много времени. Это было сделано Иваном Паскалем, который и написал соответствующий патч для версии 3.3.2, штатно включённый в следующий релиз (3.3.3).

Отступление. Выше я упомянул о той роли, которую сыграло появление Иксов во FreeBSD для нашей страны (точнее, её государственного языка). Так вот, Иван Паскаль был вовсе не линуксоидом, как наверняка подумали многие. Нет, он был применителем FreeBSD, автором фундаментальных записок об этой ОС, сохранивших свою актуальность и поныне. Но, к сожалению, в момент сочинения этих строк недоступных.

Тем не менее, XFree86 в рамках ветки 3.3 от версии к версии всё совершенствовалась, пока система не приобрела законченный вид в релизе 3.3.6, вышедшем в декабре 1999 года. Она тут же была включена во все дистрибутивы Linux, и использовалась на протяжении долгого времени.

Однако на этом развитие 3-й XFree86 ветки фактически прекратилось – уже в марте 2000 года появляется первый релиз ветки 4-й, кардинально переработанной. Правда, разработчики XFree86 совершенно чётко позиционировали версию 4.0 как экспериментальную, и майнтейнеры всех распространённых дистрибутивов вняли их предупреждению, включая её в состав своих сборок в качестве опции: как основная графическая система ещё долгое время применялась версия 3.3.6.

Хроника поступательного движения

главным новшеством XFree86 4-й ветки была ликвидация зоопарка X-серверов, расплотившихся по принципу «один видеочип – один сервер». Отныне герой (то есть X-сервер) остался один. А вся чипо-специфическая его часть выносятся в отдельные модули (по старой традиции называемые драйверами).

Далее, в XFree86 4.X появляются собственные средства автоконфигурирования. Если раньше к таковым понятие «авто» можно было применить чисто условно (см. следующие отступления), то теперь запуск команды

```
# X -configure
```

давал неизменно превосходный результат в виде конфигурационного файла /etc/X11/XF86Config компактного, но при этом исчерпывающе прокомментированного. А самое главное – почти всегда работающего без всяких дополнительных телодвижений. Лишь в нашем кириллическом окружении он требовал минимальной ручной доводки.

Настройка XFree86: как это делалось встарь Говорят, что во времена совсем былинные настроить Иксы можно было только лобовым созданием их конфигурационного файла в текстовом редакторе. Однако затем появились механизированные средства для решения этой задачи, и первое из них носило имя xf86config, издревле входившее в штатную поставку XFree86. Оно запускалось из командной строке и работало в интерактивном режиме. То есть – требовало нудного ответа на множество вопросов. И любая ошибка могла быть исправлена только одним способом – выходом из конфигуратора и повторением процедуры с самого начала. Однако это средство обладало одним несомненным достоинством – при должной аккуратности позволяло добиться результата (в виде сгенерированного файла /etc/XFree86.conf), почти не требующего ручной доводки (за исключением некоторой докрутки кириллицы и, для некоторых CRT-мониторов, юстирования частотных характеристик развёртки).

Настройка XFree86: как это делалось в Одессе В XFree86 появились меню-ориентированные программы настройки: Xconfigurator с псевдографическим интерфейсом, XF86Setup, работавшая в графическом режиме VGA, и xf86cfg, допускающая как

текстовый, так и графический варианты запуска, причём в последнем она пыталась максимально точно определить параметры видеоподсистемы. Разумеется, работать через меню, да ещё и (в графических вариантах) управляемом мышью, было куда проще, чем отвечать на многочисленные вопросы. Да и вернуться к ошибочно определённом пункту было куда легче, чем перезапускать заново, с нуля, `xf86config`. Однако объём однако объём ручной правки после работы любой из перечисленных утилит, был куда больше, чем при использовании последнего.

Настройка XFree86: как это делалось в Linux'ax В конце 90-х годов ряд дистрибутивов, проявлявших наибольшую любовь к пользователям, штатно включили средства настройки Иксов в свои инсталляторы. В числе первых на этом поприще отметились Mandrake и Caldera OpenLinux – приоритет сейчас уже не определить. Да он и не очень важен. Более существенно то, что средства настройки Mandrake были унаследованы сначала его Russian Edition, а потом Altlinux'ом. Где были расширены, углублены и закалены в боях с кириллицей (или, правильней сказать, за кириллицу в Иксах). А под идейным влиянием дистрибутива от Caldera развивался инсталлятор ASPlinux, выпускавшийся одноимённой фирмой. А поскольку она тоже имела соплеменное происхождение, то и он доблестно показал себя на ниве кириллизации Иксов.

Разумеется, сказанным усовершенствования «четвёрки» не исчерпывались: тут были и возможность работы с двумя мониторами, и автоматическое распознавание большинства моделей «колесатых» мышей (ранее для их поддержки требовался отдельный пакет `imwheel`), и способность работать с графическими планшетами «рисовального» назначения.

Все эти «улучшательства», постепенно накапливавшиеся в ходе коротких жизненных циклов минорных релизов 4.0.X, привели к тому, что релиз 4.1, вышедший в июне 2001 года, широко распространился в дистрибутивах Linux в качестве «Иксов по умолчанию». А появлением в начале 2002 года релиза 4.2 «тройка» из их состава практически исчезла, уцелев разве что в ультраконсервативных системах типа Debian'a.

На этом поступательное развитие XFree86 не остановилось: в сентябре 2002 года выходит корректирующий релиз 4.2.1, а за ним, в

феврале 2003, – «полумажорный» 4.3. В рамках последнего продолжают накапливаться усовершенствования, призванные воплотиться в будущем релизе 4.4, уже совсем «мажорном». Однако тут происходит поворот судьбы, значение которого для дальнейшего развития и Иксов, и Linux'a трудно преувеличить – хотя и до сих пор не ясно, как оценить.

Предпосылки Великого раскола

Все указанные усовершенствования, хотя и существенно улучшали жизнь пользователей, но никак не могли рассматриваться в качестве революционных. Что не нравилось некоторым участникам проекта XFree86, в первую очередь – одному из его основоположников, Кейту Паккарду. И о дальнейших событиях существуют противоречивые свидетельства.

Основная часть разработчиков проекта утверждала тогда, что Кейт втихаря начал разработку проекта параллельного. И в результате в марте 2003 года его исключают из «партии XFree86» с закрытием доступа к «партийному распределителю» – дереву исходников. Сам же Паккард отрицал партизанский характер своих действий, хотя и активно критиковал бывших коллег за медлительность в рассмотрении патчей от сторонних разработчиков и общий консерватизм. И действительно, после изгнания «из рядов», создаёт собственный форк Иксов – тот самый, которому суждено было стать современным проектом Xorg. Обещая, что он будет развиваться более динамично и «инновационно».

Насколько эти обещания были выполнены на первом, ещё «безымянном», этапе развития нового проекта, судить трудно. Потому что это не был ещё сам Великий раскол: в итоге основу форка Xorg составил один из пре-релизов XFree86 – 4.4 RC2. А поводом для собственно раскола послужили лицензионные противоречия. Но это – совсем другая история, которая принадлежит скорее политике.

Повод к Великому расколу

Что потом началось – не опишешь в словах Владимир
Высоцкий

Прошлый раздел завершился тем, что на базе последнего кандидата в релизы XFree86 4.4 RC2 был создан его форк. А вслед за этим происходит событие, приведшее к Великому расколу. Точнее, послужившее его непосредственным поводом – причины его, как мы только что видели, были гораздо глубже.

В феврале 2004 года выходит долгожданный релиз XFree86 4.4, аккумулирующий все новшества предшествующих корректирующих и «полумажорных» релизов. Однако – под скорректированной же лицензией. Если ранее XFree86 распространялась под стандартной «разрешительной» лицензией MIT, то в новой версии появился пункт, несколько напоминающий пресловутую «оговорку о рекламе» из первой версии BSD-лицензии, в последующем изъятую (подробности см. в главе второй).

Хотя, если вчитаться в текст лицензии, выясняется, что всё её новшество сводится к требованию включать в документацию дистрибутивов, использующих XFree86, такую фразу:

Данный продукт включает программное обеспечение, разработанное The XFree86 Project, Inc (<http://www.xfree86.org/>) и его сотрудниками.

То, что в просторечии обычно называется лицензией MIT, правильней именовать лицензией X11. Ибо в разных проектах MIT использовалось несколько видов лицензий, большинство из которых со временем вышли из употребления. Существующий же текст лицензии, под которой распространялась XFree86 вплоть до последнего кандидата в релизы версии 4.4, был сочинён специально на ранней стадии разработки Иксов и именно для них.

Казалось бы – чего страшного? Почему бы лишний раз не помянуть добрым словом разработчиков хорошей (и на тот момент времени практически незаменимой) системы? А вот именно её незаменимость и вызвала, думается, претензии: фактически каждый дистрибутив Linux'а общего назначения или операционная система BSD-семейства включали в себя Иксы – а альтернатив XFree86 к тому времени не было, ибо прочие X-сервера, о которых вскользь упоминалось в одном из предыдущих разделов, практически вымерли. И такое требование было воспринято сообществом (сначала –

некоторыми, но влиятельными его членами) как беззастенчивая реклама проекта XFree86 за счёт разработчиков всех остальных, взаимосвязанных, но независимых проектов.

Так что дальнейшие события можно описать цитатой из классика, чьи слова приведены и в качестве эпиграфа раздела:

Тут поднялся галдёж и лай..

Новая лицензия была названа несовместимой с принципами свободы такими авторитетами, как Ричард Столлмана и разработчики Debian'a. К ним постепенно присоединились остальные ведущие разработчики, включая Тео де Раадта (OpenBSD) – его высказывания в этом плане были, пожалуй, наиболее резкими.

Правда, все эти разработчики поначалу пошли своими путями. В одни системы (Debian, OpenBSD) была включена XFree86 под последней «чистой» MIT-лицензией, большинство же переключилось на её форк – Xorg, первый официальный релиз которого появился в апреле 2004 года. Который вскоре и стал магистральной линией развития Иксов.

Конечно, были и отдельные «голоса из ветвей». Так, Патрик Фолькердинг высказался в том смысле, что его все эти политико-юридические игры не интересуют, и ничего крамольного в новой лицензии XFree86 он не видит. Однако был вынужден, в целях совместимости с прочими дистрибутивами Linux'a, перейти на Xorg.

А вот разработчики NetBSD вообще сочли, что новая лицензия – вполне нормальна с точки зрения BSD-стиля, и использовали XFree86 в своей ОС вплоть до её версии 4.0 включительно (вышла в самом конце 2007 года). Правда, в версии 5.0 (апрель 2009) и им пришлось от неё отказаться, ибо разработка XFree86 к этому времени фактически прекратилась: последний её релиз, 4.8, появился в конце 2008 года.

Конец XFree86...

That is the end of Solomon Grandy. *Английское народное*

Причин прекращения разработки XFree86 было несколько. Здесь можно назвать и её не востребованность в связи с переходом

большинства, а затем и всех дистрибутивов Linux и BSD-систем на Xorg, и миграцию многих, если не большинства, бывших её разработчиков в теми же направлении. Но главной, на мой взгляд, причиной было то, что в XFree86, в сущности, стало нечего разрабатывать: уже к 2004 году она представляла собой устоявшуюся экосистему, кардинальные улучшения в которую можно было внести путём столь же кардинальных изменений. Так что участникам проекта, в сущности, оставалось только отлавливать баги и вносить коррективы в соответствии с изменениями остальных базовых компонентов. То есть осуществлять банальную техническую поддержку конечного продукта. А это для разработчиков Open Source, следующих курсу *Just for Fun*, что серпом по... ушам. Вот хлопцы и разбежались в разные стороны. Точнее, в сторону Xorg.

Однако в последнем случае нельзя исключить и влияние активной пропаганды порочности новой лицензии XFree86 с точки зрения идеалов свободы и демократии. А насколько такого рода пропаганда может быть действенной в сообществе Open Source, мы имели возможность убедиться совсем недавно – на примере тотальной systemd'изации всея Linux'а. Так что уделим обсуждению этого вопроса ещё несколько минут.

Как уже было сказано, с точки зрения здравого смысла и человеческой порядочности новая формулировка лицензии XFree86 не может вызвать никаких возражений: упоминание автора всегда считалось хорошим тоном в мире фундаментальной науки. А разработка Open Source, как ещё давно показал Николай Безруков, есть разновидность фундаментальной науки. По крайней мере, должна ею быть...

Интересно, что наиболее активные из тогдашних критиков новой лицензии XFree86 (не будем лишним раз помянуть их все) и по сей день не упустят случая напомнить, что ОС Linux на самом деле должна называться GNU/Linux. Ибо Linux – это только ядро, а всё пользовательское окружение – достижения проекта GNU.

Но, если следовать этой логике, то термин вроде XFree86/Linux имел 10 лет назад куда больше прав на существование. Как сейчас более правомерен был бы термин Xorg/Linux. А во все времена и у всех народов – просто X/Linux. Ибо, с одной стороны, так называемое пользовательское окружение ядра Linux состоит не только из GNU-

компонентов. Более того, существуют дистрибутивы (например, Tiny Core), где их нет совсем. А вот без Иксов в том или ином их проявлении ни один дистрибутив общего назначения не обходился никогда.

Со стороны же другой, современный пользователь Linux может (то есть имеет не только право, но и возможность) даже не подозревать о пресловутом пользовательском GNU-окружении. Ибо часто работает исключительно в графической среде, то есть в Иксах и надстраивающих их оконных менеджерах или интегрированных десктопах. А потому без XFree86/Xorg ни о каком десктопном Linux'e не может идти и речи.

Так что новая лицензия XFree86 была только попыткой «легализовать» существующее положение дел. Причём без претензий на замену устоявшихся имён операционных систем, а только на уровне обязательности упоминания одного из важнейших, с точки зрения пользователя, их компонента. И, более того, не во всяких там копирайтах и копилэфтах, а исключительно в документации, сопровождающей дистрибутивы. Тем не менее, она была дружно признана несовместимой с духом изначальных Иксов (Тео де Раадт), свободного софта вообще (Ричард Столлман) и буквой «самой свободной» из всех свободных лицензий, то есть с GPLv2. И потому подверглась всеобщему ostracismu.

Новая лицензия проекта XFree86 по научному называлась *Лицензия XFree86 версии 1.1* (под версией 1.0 следует понимать лицензию X11). Интересно, что GNU/FSF, в своё время столь категорически её осудившая, ныне признаёт её совместимость с лицензией GPLv3 – той самой, которую FSF в настоящее время считает наиболее правильной и всячески рекомендует её к употреблению. Однако с GPLv2 лицензия XFree86 версии 1.1 несовместима по прежнему – из-за требования упоминания её имени в документации.

Впрочем, ныне это уже не актуально. И может быть приведено только в качестве примера того, как юридические формулировки, придуманные, казалось бы, с самыми лучшими намерениями – ради защиты свободы – вступают в противоречие со здравым смыслом. А в конечном счёте – и с той же самой свободой. Ещё один такой пример мы уже видели при рассмотрении истории файловой системы ZFS (см. главу десятую).

... и начало Xorg

Если историю XFree86 уже в 2004 году можно было считать законченной (хотя она существует и доступна по сей день), то история Xorg только начиналась. И начиналась она весьма бурно.

Дабы отрешиться от старого мира, в проекте Xorg отказались от продолжения нумерации версий XFree86. И для начала продолжили нумерацию спецификаций оконной системы X вообще: первый релиз проекта (апрель 2004) именовался просто X11R6.7.0. Напомню, что предыдущая «общеейсковая» версия, X11R6.6, появилась на свет в апреле 2001 года. А спецификации «мажорные», то есть X11R6, на протяжении многих лет лежащие в основе XFree86, уходят в далёкий майский день 1994 года.

Со временем параллельно ей стала использоваться нумерация по версиям собственно сервера Xorg. В релизах X11R6.7.0 и X11R6.8.X в неявном виде подразумевалось, что он имеет номер версии 1.0. А далее к нему прибавлялась «мажорная» (1.X) или «минорная» (1.X.Y) единица. В настоящее время именование по версиям сервера Xorg является основным. Так, текущая его версия на момент сочинения этих строк – 1.15.

Итак, в апреле 2004 года появляется первая версия собственно Xorg – X11R6.7.0, основанная на исходниках XFree86 4.4 RC2 и мало чем от последней отличавшаяся. Точнее, не отличавшаяся ничем – имел удовольствие сравнивать их вживе. А далее версии сменяются с быстротой, заставляющей вспомнить ранние времена первозданных Иксов: 8 сентября 2004 – X11R6.8.0, 17 сентября 2004 – X11R6.8.1, февраль 2005 – X11R6.8.2.

В версии X11R6.8.0 впервые появляются такие ныне привычные вещи, как Composite – предварительная прорисовка изображения для вывода его на экран в уже готовом виде, прозрачность окон и прямая поддержка конфигураций с несколькими мониторами. «Минорные» же версии носили корректирующий характер.

Далее наступает некоторое затишье, продолжавшееся до 21 декабря 2005. Зато этот день ознаменовался выходом сразу двух версий – X11R6.9 и X11R7.0. Нет, это было не раздвоением личности, а окончательным разрывом со старыми традициями: переходом от Imake – системы автоматизации сборки, унаследованной от XFree86, к

Autotools – аналогичной системе, развиваемой в рамках проекта GNU. Что вызвало и переход от монолитной сборки к модульной. В результате чего на одной кодовой базе и были созданы монолитная версия 6.9 и модульная версия 7.0. Во всех последующих релизах Xorg использовалась только система Autotools и, соответственно, все они были модульными.

Традиционно исходные коды XFree86 распространялись в виде нескольких крупных тарбаллов (в разных версиях – от трёх до семи). Которые при монолитной системе сборки и в бинарном виде собирались как серия крупных пакетов, таких, как Xbin, Xlib, Xxserv и так далее. Разумеется, бинарники можно было собрать и более дробно, и майнтайнеры ряда дистрибутивов, таких, как RedHat и Debian, прибегали к этому со стародавних времён. Но поначалу в Xorg штатно такая возможность не использовалась. Как не применялось «дробное» пакетирование и в дистрибутивах, придерживавшихся соответствия пакетов «авторских» и «дистрибутивных» – а такими на протяжении долгого времени были Slackware, Gentoo и идеологически близкие к ним, не говоря уж об LFS.

С переходом к модульной системе сборки бинарников Xorg как огромного множества небольших макетов стала нормой во всех дистрибутивах. Более того, и сами исходники Xorg стали распространяться в виде изобилия мелких тарбаллов «целевого назначения», что обеспечивало существенно большую гибкость при построении конечной системы.

Впрочем, чтобы, как говорится, почувствовать разницу, достаточно сравнить древа исходных текстов версий X11R6.9 и X11R7.0 на официальном сервере проекта.

Следующая веха в истории Xorg – версия X11R7.3 (X-сервер 1.4), вышедшая в сентябре 2007 года: в ней, среди прочего, получает дальнейшее развитие автоопределение оборудования, в том числе и горячего подключения. Тогда это делалось через HAL (Hardware Abstraction Layer) – и делалось вполне успешно, и, что характерно, в сборках Иксов не только для Linux, но и для BSD-систем. В частности, автор этих строк неоднократно использовал HAL во FreeBSD.

Тем не менее, в версии X11R7.6 (X-сервер 1.8.0, декабрь 2010) в управлении устройствами подсистема HAL была заменена менеджером устройств udev. Что, с одной стороны, привело к

определённому прогрессу в этом деле. А с другой, поскольку udev – инструмент, специфический для Linux'a, отгородило последующие версии Xorg от остальных UNIX-подобных систем. Но на этом я поставлю точку: мой рассказ подошёл к своему логическому завершению. И вместе с ним близится и конец Иксов вообще: на горизонте маячат Wayland, с одной стороны, и Mir – с другой. Но это уже дела дней сегодняшних и грядущих.

Глава двадцать третья. Управители окон: извлечения из истории

Предыдущая глава была посвящена истории X Window System и её свободных реализаций, XFree86 и Xorg. Однако ни слова не было сказано об истории того, как эти протоколы, спецификации и реализации претворялись в те самые графические интерфейсы, с которыми непосредственно имеет дело пользователь.

Терминологическое введение

Протоколы, спецификации и реализации претворялись в виде двух классов программ – оконных менеджеров, иногда именуемых также диспетчерами окон (WM – Window Manager), и интегрированных графических рабочих сред (Graphic Deskton Environment), которые называют также средами рабочего стола (DE – Desktop Environment) или, в просторечии, десктопами.

Чтобы в дальнейшем не путаться среди многочисленных реализаций графических интерфейсов, дам краткую вводную установку. Оконный менеджер – это программа, обеспечивающая внешний вид и функциональность элементов графического интерфейса. Под видом интерфейсных элементов понимается оформление окон, включая панель заголовка, рамки, линейки прокрутки, управляющие кнопки и меню. Функциональность же – возможность минимизации, максимизации, сворачивания окон, их перемещения, изменения размера, вызова различных наборов контекстных меню.

По большому счёту от оконного менеджера ничего иного и не требуется. Однако, как мы скоро увидим, со временем развитые представители этого класса обрастали наборами привязанных к ним утилит и приложений, а также собственными средствами настройки и наборами предопределённых тем. Откуда оставался один шаг до интегрированных рабочих сред, то есть десктопов.

Десктопы также включают в себя средства оформления окон и управления ими, то есть оконные менеджеры, собственные (как в KDE

и Xfce) или заимствованные (как в GNOME и LXDE). Однако средства собственного конфигурирования, наборы тем и стилей, штатные утилиты и приложения входят в них уже в обязательном порядке. Хотя количество последних может быть различным – от всеохватного в KDE до весьма скромного в Xfce или совсем уж бедного – в LXFE. Важно, что все штатные программы десктопов характеризуются единством интерфейса, настраиваемого собственными конфигураторами среды.

Настоящая статья, как и следующая за ней, будут посвящены истории менеджеров окон. В статьях же дальнейших я расскажу о драматической истории десктопов и о разворачивавшихся вокруг них «священных войнах».

Я не берусь рассказать про возникновение и развитие всех оконных менеджеров, когда-либо существовавших на свете – это задача непосильная для одного человека. Поэтому ограничусь только теми представителями семейства, с которыми работал непосредственно, или которые внимательно рассматривал на предмет практического применения.

Предтечи управителей

Практическая работа в X Window System без менеджера окон почти невозможна, или, по крайней мере, очень неэффективна. Тому, кому доводилось видеть «голые Иксы», понятно, почему: это просто серое поле с курсором мыши в виде крестика. И никакое щёлканье мышными кнопками не влечёт за собой ни малейшего результата.

Конечно, в пользовательском конфиге Иксового сеанса можно прописать автоматический запуск каких-либо приложений – например, собственного эмулятора терминала, `xterm`, и работать в его командной строке, как в текстовой консоли. Можно было открыть при старте Иксов много терминальных окон для одновременной работы. И я знаю применителей, которые вполне серьёзно полагали это главной сакральной функцией графического интерфейса – ибо ни в чём, кроме терминалов, не нуждались. Можно было, наконец, открыть много приложений – но радости это приносило мало: исключительно иксовыми средствами окна их нельзя было ни перемещать, ни масштабировать, ни даже закрывать, не говоря уж о сворачивании в пиктограммы.

Так что можно предполагать, что оконные менеджеры возникли одновременно с первыми реализациями X Windows System, однако память о них затёрлась. И первое о них упоминание обнаруживается только в X10 (конец 1985 года) под именами `xwm` и `xnwm`, но сведений о том, что они собой представляли, мне обнаружить не удалось. По косвенным данным можно предполагать, что управлялись они не мышью, а с клавиатуры комбинациями с участием клавиши **Meta**, и не имели средств конфигурирования.

В том же 1985 году компания DEC разрабатывает оконный менеджер `uwm` (Ultrix Window Manager). Он предназначался для её собственной реализации UNIX для платформы VAX – Ultrix (в последующем Tru64 UNIX), в которой Иксы не использовались. Однако был быстро портирован на них, и уже в X11R3 стал стандартным средством управления окон. Это был первый оконный менеджер, в котором с помощью файла конфигурации можно было настроить поведение кнопок мыши и привязать к ним меню управления окнами – функции, которые нынче кажутся столь тривиальными, что в гипермодернистских средах типа GNOMEShell и Unity они почти редуцировались.

У истоков управления окнами: twm...

Следующим этапом в развитии оконных менеджеров стал `twm`, разработанный Томом Ластрэнджем (Tom LaStrange) в 1987 году и включённый в качестве стандартного компонента в Иксы, начиная с X11R4 (декабрь 1989 года). Откуда он и попал в XFree86, появившуюся, как помнит читатель, в феврале 1991 года (LXF#168).

В отличие от ранее упомянутых оконных менеджеров, `twm` могли бы наблюдать многие из ныне действующих линуксоидов. Хотя развитие его прекратилось, `twm` до недавнего времени в качестве стандартного средства управления окнами входил практически во все сборки Иксов – как в XFree86, так и в Xorg. А некоторым и довелось наблюдать его: именно `twm` запускался по умолчанию в ответ на команды `startx`, если в пользовательском конфигурационном файле Иксов не было определено ничего другого.

Создатель `twm`, Том Ластрэндж (Tom LaStrange), разрабатывал этот оконный менеджер для себя – и, вполне естественно, назвал его

собственным именем: первоначальной расшифровкой аббревиатуры было: Tom's Window Manager. Такая практика в те годы была обычной (вспомним, например, Joe – Joe's Own Editor, то есть Личный Редактор Джозефа Аллена, его создателя) и отражала не манию величия или стремление увековечить своё имя. А напротив, как бы говорила: эту программу я сделал для себя. Подразумевая в скобках: а подойдёт ли она вам – не знаю.

При включении twm в штатный комплект Иксов Том передал права на своё произведение X-Консорциуму, стоящему в то время у руля управления графическими интерфейсами в UNIX'ах. И twm перестал быть его личным инструментом, а стал всенародным достоянием (на условиях X-лицензии, разумеется). К тому же новые разработчики добавили в него функцию объединения заголовков окон в единую панель с закладками (позднее нечто подобное будет реализовано во FluxBox'e, а сама идея закладок нашла применение в бессчётном количестве прикладных программ; говорят, что не так давно закладки появились даже в Internet Explorer). Так что twm с полным на то правом был переименован в Tabbed Window Manager.

А через четверть века после своего возникновения получила распространение более иная расшифровка имени twm: Timeless Window Manager. Что, применительно к случаю, я перевёл бы как Оконный Менеджер Всех Времён (а возможно, и народов).

Ныне место «аварийного» оконного менеджера в ряде дистрибутивов Linux занял IceWM. Однако и twm до сих пор сохраняется во многих сборках, например, его можно обнаружить в стандартной установке openSUSE.

Основным средством управления окнами в twm окончательно стала мышь. С её помощью, по щелчку правой кнопкой, вызывалось контекстное меню рабочего стола для запуска приложений и прекращения их работы, выполнения системных действий, таких, как выход из сеанса или рестарт машины, манипулирования открытыми окнами. Впрочем, управление окнами можно было осуществлять и действиями в строке заголовка – опять же с помощью мыши.

Собственного средства конфигурирования или каких-либо тем и стилей twm не имел: все настройки осуществлялись правкой единственного и весьма простого по устройству конфига – twmrc. Что,

тем не менее, позволяло добиваться весьма причудливого и эффектного внешнего вида.

Современный оконный менеджер и, тем более, десктоп трудно представить себе без виртуальных рабочих столов – некоего аналога виртуальных терминалов консольного режима. Однако в twm их ещё не было. Зато допускалось применение виртуального разрешения экрана, и во времена, когда преобладали мониторы с физическим разрешением 640x480, а режим 800x600 считался предметом роскоши, это было более чем востребовано).

Отступление. Виртуальное разрешение экрана – функция не оконного менеджера или рабочей среды, а X-сервера, и задаётся в его конфигурационном файле. Оно может быть в полтора-два раза больше максимального физического разрешения монитора. Собственно, верхний его предел определяется только объёмом видеопамяти. При включении виртуального режима на экране видна только часть рабочего пространства (например, четвертинка, если виртуальное разрешение задано вдвое большим, чем физическое). Для доступа к невидимым его участкам достаточно подвести курсор мыши к правому или нижнему краю экрана, чтобы плавно переместиться как бы за его пределы.

Ныне, в эпоху больших LCD-мониторов, виртуальное разрешение экрана устанавливается редко, обычно для каких-то специальных задач, и многие современные лиунгксоиды даже не подозревают о его существовании. Однако во времена, когда самым ходовым размером экрана было 14, много 15 дюймов, это была палочка-выручалочка при использовании таких нагруженных интерфейсными элементами рабочих сред, как KDE.

Кстати, оверлейный режим GNOME Shell, которые создатели «третьегнома» продают как последний писк прогресса, – ни что иное, как гальванизированная идея виртуального разрешения. Воистину, если не всё, то многое новое – это основательно испорченное старое.

В сущности, уже в twm почти в современном виде сложилась парадигма работы в графической среде, которая и поныне применяется в большинстве оконных менеджеров и рабочих сред. И, хотя развитие его прекратилось очень давно, ещё в прошлом тысячелетии, он оставил несколько прямых потомков и немало косвенных. А идейное

его влияние в той или иной мере прослеживается абсолютно во всех оконных менеджерах и рабочих средах.

... и его линия

Как уже было сказано, twm не поддерживал виртуальных рабочих столов. Однако эта функция появилась в его ближайших прямых потомках – vtwm (Virtual TWM) и tvwm (Tom's Virtual Tab Window Manager – опять же разработка Ластрэйнджа для личного пользования). Которые, в сущности, только ею и отличались от родителя.

Из косвенных потомков twm наибольшее распространение и известность приобрёл FVWM, что изначально расшифровывалось как Feeble Virtual Window Manager (то есть хилый виртуальный менеджер окон), однако в дальнейшем значение первой литеры забылось, и она стала восприниматься символически. Он был создан в 1993 году Робертом Нэйшном (Robert Nation). И сначала – также для личного применения. Однако, будучи также автором эмулятора терминала gxvt, он начал распространять их совместно – и FVWM был принят народом «на ура».

Вскоре Роберт прекратил заниматься FVWM, и его на посту основного разработчика сменил Чарльз Хайнс (Charles Hines), который внёс изменения в формат конфигурационного файла, дополнив его рядом новых возможностей. Получившийся в результате оконный менеджер стал известен в народе как FVWM2, хотя до сих пор оба названия часто употребляются как синонимы.

FVWM, особенно в варианте FVWM2, прославился своей поистине безграничной настраиваемостью как функционала, так и внешнего вида. Правда, достигалась она кропотливой правкой конфигурационного файла, причём далеко не очевидной, требующей вдумчивого чтения документации. В этом отношении его можно сравнить с позднейшим KDE: хотя в нём настройка выполняется через собственный графический интерфейс, однако он тоже далеко не прозрачен и требует освоения. Впрочем, в обоих случаях затраченные усилия окупаются неизменно превосходным результатом.

Вследствие своей конфигурабельности FVWM2 завоевал большую популярность, и до сих пор широко распространён в узких

кругах терпеливых и аккуратных применителей. И почти неизвестен за их пределами. Однако он сыграл большую роль в становлении следующей волны оконных менеджеров.

По стопам Windows 95

Летом 1995 года появляется Windows 95 со своей сакраментальной кнопкой Пуск (она же Start). И сразу обретает бешенную популярность среди windows-профов (windows-профи в это время продолжают применять Windows 3.1/3.11 для офисных задач и NT 3.X – для задач всамделишных).

По закономерной случайности в это же примерно время Linux делает первую попытку обратиться лицом к пользователю в корпоративном его исполнении (LXF#150), а энтузиасты-линуксоиды начинают первую волну пропаганды своей любимой системы частным порядком, среди широких народных масс. А так как последние уже вкусили от прелестей кнопки Пуск, предпринимаются попытки обеспечить их таковой и в графических оболочках Иксов.

Отступление: про win-профи и win-профов. Вопреки устоявшемуся мнению, победное шествие Windows той линии, которую в фольклоре именовали «оболочкой дешёвой», началась не сразу с появлением Windows 95. Свидетельством чему – появившаяся в «Компьютере» колонка тогдашнего её главного редактора, Георгия Кузнецова, о профи и профах. В которой прозвучала пророческая фраза (цитирую по памяти): «Профы поустанавливают Windows 95» дома, потом притащат её на работу, и профи придётся с этим разбираться.

Так оно и случилось. Хорошо помню время, когда на моей тогдашней службе проходила массовая замена незадолго перед тем внедрённой в качестве «фирменного стандарта» OS/2 на Windows 95 и затем на Windows 98. И проходила она по пожеланиям трудящихся. А поскольку среди трудящихся велик был процент не только простых докторов наук, но и академиков с член-коррами, пожелания эти по силе своей были близки к армейскому приказу.

Кстати, именно тогда мы впервые на практике массово применили Linux в условиях производственного десктопа. Оказалось, что самый

быстрый и простой способ искоренения OS/2 – загрузка Linux'a с дискеты и запуск в нём команды `# dd if=/dev/zero of=/dev/hda`

В результате в 1996 году на свет божий появляется fvwm95 – весьма причудливая имитация интерфейса Windows 95. Она была образована из несколько облегчённого FVWM с прикрученной к нему панелью задач в win-стиле и, разумеется, кнопкой **Пуск**. Собственных средств его конфигурирования по прежнему не было, но настраивать стало легче, стало веселей. Потому что конфигурируемых параметров стало меньше, по сравнению с прототипом. В общем, на меня некогда этот оконный менеджер произвёл впечатление откровенной пародии – причём и на FVWM, и на Windows 95. Судя по тому, что он очень быстро сошёл со сцены – я был не одинок в своём мнении.

Однако дело кнопки **Пуск** не пропало. Его подхватили... и так далее, по Ульянову в скобках Ленину, разработчики других GUI, начиная с IceWM и кончая героями... Но о героях речь пойдёт в следующей главе. А здесь поговорим о IceWM.

Если fvwm95 выглядел поделкой, слепленной на скорую руку на потребу win-профам, IceWM, разработка Марко Мачека (Marko Maciek), начатая им в 1997 году, производил впечатление продукта, сделанного не только с умом, но и с любовью. Кроме того, он был написан «с нуля», а не основывался на коде FVWM, хотя без идейного влияния последнего и не обошлось (в частности, в плане гибкости конфигурирования).

Основных особенностей, выделявших IceWM в ряду прочих оконных менеджеров (а я выше упомянул далеко не всех его современников – одних только эпигонов FVWM тогда существовало с полдюжины), изначально декларировалось три:

- интуитивно понятный интерфейс (кто видел умолчальный FVWM, поймёт, о чём речь) с возможностью гибкой индивидуальной настройки (в этом отношении если IceWM и уступает FVWM, то не намного);

- доступность полного функционала интерфейса с клавиатуры, без использования мыши, что для большинства современников было затруднительно: в те времена мне приходилось слышать высказывания, что в Иксах без мыши работать вообще невозможно; так вот, IceWM был живым их опровержением;

- быстродействие и минимизация в использовании ресурсов.

Все эти цели были достигнуты – и плюс ещё ряд дополнительных. Например, хотя исходно предусматривалась только настройка IceWM традиционным путём правки конфигов (их довольно много, но они просты и понятны), очень быстро он был дополнен средством собственной настройки – графической утилитой IcePref. А простота создания тем для IceWM привела к тому, что таковых было создано много: кроме умолчальной стилизации под Windows 95, в сети можно было найти множество тем, воспроизводящих внешний вид OS/2, Motif и многих других систем. При наличии желания и толики свободного времени не составляло труда сделать и собственную тему.

Оконный менеджер IceWM продолжает своё развитие до сих пор: очередные его версии выходят не часто, ибо кардинально улучшать в нём уже нечего, но регулярно – в соответствии с необходимостью подгонки под новые версии библиотек, с которым он связан зависимостью. И, как уже было сказано, в сборках Xorg некоторых некоторых дистрибутивов IceWM сменил twm в качестве «умолчального» (или «аварийного») оконного менеджера, который может быть запущен, вне зависимости от наличия или целостности любых других графических сред.

Однако, как мы увидим в следующей статье, свет клином не сошёлся на линии FVWM и Windows-подобии. Напротив, наиболее яркие представители семейства оконных менеджеров представляли собой линии более иные.

По следам легенды

Вторая половина 90-х годов – период бурного развития оконных менеджеров: все ныне существующие их группы (за единственным исключением, о котором я упомяну в конце статьи) возникли в это время, в том числе и самые яркие, по моему мнению, представители семейства.

Кто же не помнит старика Крупского? Пардон, старика NeXT'a? Увы, наступил угар НЭПа, и имена героев информационной революции постепенно забываются. Так что как раз NeXT в качестве аппаратной архитектуры и NeXTStep в роли операционной системы для неё ныне вспоминают не часто. А ведь эта платформа стала легендой ещё при жизни...

О судьбе аппаратной составляющей платформы я скажу пару слов в отступлении. Здесь же речь пойдёт о продолжении дела программной составляющей – ОС NeXTStep. Каковое имело место быть отнюдь не в проприетарной OpenStep – совместном детище компаний NeXT и Solaris: ей суждено было стать жертвой аборта на ранней стадии беременности. И даже не в MacOS X – не смотря на общее происхождение, родства на генетическом уровне между ними оказалось не так уж много. А о продолжателях дела старика NeXT'a из мира свободного программного обеспечения.

Отступление. Операционная система NeXTStep изначально разрабатывалась для аппаратной платформы NeXT, созданной в 1987 году одноимённой фирмой, основанной и возглавлявшейся Стивом Джобсом в период его развода с Apple. Компьютер NeXT, сердцем которого был пламенный мотор от Motorola за номером 68040, выглядел тогда как пришелец из далёкого будущего: футуристический «чёрный кубик» (что было его народным названием) в качестве непременных компонентов включал в себя мощный видеоадаптер, привод компакт-дисков и звуковую карту. То, о чём в те годы рядовой пользователь не только РС, но и Mac'a не мог даже мечтать.

К сожалению, стандартное богатство комплектации NeXT'a влекло за собой стоимость, чрезмерную даже по тем былинным временам, когда до массовых дешёвых комплектующих от продолжателей дела Великого Мао было ещё далеко. И потому спрос на него был далёк от ожидаемого во всех сферах. Кроме, разве что, сферы научной: сочетание мощной аппаратуры и удобных средств разработки позволяло, например, биологам и химикам создавать специальные приложения для моделирования структуры сложных органических молекул.

Однако учёные не принадлежат к самым богатым слоям прогрессивного человечества, и с золотым запаом у них часто напряги. И в итоге развитие NeXT как аппаратной платформы прекратилось в 1993 году. – именно по причине недостаточного объёма продаж Однако уход NeXT'a с «железного» рынка трудно назвать иначе чем триумфальным: последние месяцы продаж «чёрного кубика» ознаменовались ажиотажным на него спросом. И как раз со стороны научных учреждений, в том числе и российских, которые тогда всюю

начинали переживать свои не лучшие времена (продолжающиеся по сей день). Можно сказать, что NeXT «ушёл на дно, не опуская флаг»...

Интерфейс операционной системы NeXTStep отличался, с одной стороны, функциональностью, с другой – элегантностью, так с тех пор и непревзойдённой (на мой взгляд). И потому он послужил сначала образцом для подражания в виде целой линии оконных менеджеров. Первым из них стал AfterStep, разрабатываемый с 1996 года. Он был основан на коде FVWM, но внешний вид его был приведён в соответствие с таковым NeXTStep. Казалось бы, процедура, аналогичная проделанной ранее с fvwm95 – однако результат был не сопоставимым. И, хотя AfterStep и не снискал большой популярности – он развивается до сих пор, и вокруг него сложилось небольшое, но преданное сообщество.

Отступление. Оконный менеджер AfterStep оказал влияние и на мир Windows: в 1997 году Франсис Гастеллу (Francis Gastellu) разработал его клон для платформы Win32 – LiteStep. Первоначально он настолько точно воспроизводил внешний вид прототипа, что невозможно было поверить в существование лежащей под ним банальной Windows 95/98. В дальнейшем он эволюционировал в сторону конструктора, позволяющего воспроизвести поверх Windows разного рода (в том числе и линии NT/2000/XP etc.) интерфейс любой рабочей среды для Иксов или создать интерфейс собственный. Оболочка LiteStep активно развивается по сей день: в частности, в ней реализована и поддержка Windows 8. Насколько широко она используется «записными подоконниками» – судить не берусь. Но ряд лично знакомых мне линуксоидов активно применяли её во время вынужденной работы в Windows.

Если AfterStep имел в своей основе код FVWM, то второй последователь NeXTStep, WindowMaker, разрабатывался «с нуля» Альфредо Коджимой (Alfredo Kojima), начиная с 1997 года. И первоначально этот оконный менеджер предназначался для кросс-платформенной среды GNUstep – попытке свободного воспроизведения OpenStep, того самого нерождённого дитяти от союза NeXT и Sun, которое поминалось выше.

Сама среда GNUstep угодила в долгий ящик – время от времени появлялись только её реализации на ядре Linux (что, впрочем, было свойственно всем амбициозным проектам, до которых антилопа GNU

дотягивалась своими копытами). А WindowMaker же, как оконный менеджер для Иксов, вследствие своих несомненных достоинств (элегантность и, при некоторой привычке, удобство интерфейса, быстрота, нетребовательность к ресурсам) быстро завоевал заслуженную популярность.

Не последнюю роль в распространении WindowMaker'а сыграло то, что изначально он включал в себя утилиту настройки, работавшую в графическом режиме: необходимости в ручной правке конфигов больше не было. Хотя и запрета на неё тоже не налагалось. Кроме того, для него был разработан и комплекс служебных программ, что знаменовало первый шаг в направлении интегрированных десктопов (правда, дальнейших шагов в эту сторону не последовало – WindowMaker так и остался менеджером окон).

В нынешнем тысячелетии WindowMaker несколько захирел. В том числе и потому, что, когда всё прогрессивное человечество начало в массовом порядке переходить на UTF8, долгое время оставался верен восьмибитным кодировкам. Правда, в середине нулевых годов вышел релиз 0.95.0 с поддержкой юникода – но затем многие годы о WindowMaker'е не было слышно ничего. По традиции он входил в штатный набор графических сред ряда дистрибутивов, присутствовал в их репозиториях, официальных или дополнительных, но о былой популярности его говорить не приходилось.

Казалось, что WindowMaker обречён на тихую и незаметную кончину. Как вдруг случилось чудо: в январе 2012 года новой командой разработчиков было объявлено о реанимации проекта и выходе нового релиза – 0.95.1. А вслед за тем очередные версии этого оконного менеджера начали выходить регулярно – последняя на сегодняшний день (0.95.5) датируется августом 2013.

Начинание разработчиков этого оконного менеджера получило поддержку со стороны майнтейнеров некоторых дистрибутивов. И в начале июня 2013 года свет увидел LiveCD на базе Debian'a, в котором WindowMaker выступает в качестве рабочего окружения.

А в период стагнации WindowMaker оказал несомненное влияние на две самых современных рабочих среды современности: режьте меня на куски, но идея больших объёмных кнопок на панели запуска приложений вдоль боковины экрана в Unity и GNOME Shell ведёт своё начало от него. Хотя разработчики обеих сред не любят говорить об

этом вслух. И, дабы окончательно обрубить концы преемственности, переместили эту панель справа (где она имела место быть в WindowMaker'e по умолчанию) налево.

Линия боксов

В основе интерфейса всех оконных менеджеров, о которых говорилось в предыдущей статье, лежал какой-нибудь прототип, «родной» (как первозданный twm) или пришедший из «другого мира» (Windows, NeXTStep). Однако в семействе их имеется линия абсолютно оригинальная – по крайней мере, прообразов для неё я не видел никогда и нигде. Это – линия так называемых *kbox'ов.

Прародитель семейства, Blackbox, был разработан Брэндли Хьюгсом (Bradley Hughes) в 1997 году как своего рода неявный ответ на IceWM: то есть как ещё более лёгкий с точки зрения потребления ресурсов, ещё более минималистичный по своему интерфейсу, ещё более простой в настройке и использовании, не несущий к тому же никаких следов чужеродного воздействия. Иными словами – истинное воплощение True UNIX GUI в превосходной степени.

И надо сказать, что поставленную перед самим собой задачу Брэндли выполнил с блеском, придав своему решению предельно простой и строгий дизайн. Который сводится к двум элементам: узенькой управляющей панели, служащей для переключения рабочих столов и запущенных приложений, и рабочего пространства, с которого щелчком правой кнопки мыши вызывается контекстное меню, позволяющее запускать приложения и выполнять первичную настройку Blackbox'a. Более тонкая настройка выполнялась путём коррекции конфигурационных файлов.

Blackbox быстро приобрёл популярность и, как следствие стал обрастать дополнениями в виде многочисленных тем рабочего стола и интерфейсных элементов (например, средства вывода запускающих пиктограмм на рабочий стол). Появилось и средство собственного конфигурирования – bbconf. Однако в нём самом, после первого периода «бури и натиска», фактически ничего не менялось, и всё по той же причине, которая вызвала «захирение» многих свободных проектов: что-то кардинально улучшить без смены парадигмы в нём

было уже невозможно. А смена парадигмы привела бы к тому, что Blackbox перестал быть самим собой.

В результате в первой половине нулевых годов развитие этого оконного менеджера прекратилось – последняя его версия (0.70.1) на официальном сайте датируется ноябрём 2005 года. Однако сам по себе он не умер: майнтейнеры большинства популярных дистрибутивов держат его в своих официальных репозиториях, заодно поддерживая совместимость его с новыми версиями библиотек (благо зависимостей у Blackbox'a не мало, а очень мало).

Продолжал развиваться Blackbox и другим образом – в виде своих потомков. Из них до сего дня дожили два – Fluxbox и Openbox. Оба они в целом сохранили минималистический интерфейс родителя, но обогатили его некоторыми новшествами. Для Fluxbox'a (чистого клона Blackbox'a), возникшего на рубеже тысячелетий, главным из них была возможность объединения совместно используемых приложений (например, терминала, текстового редактора и браузера) в «группы по интересам». И перемещаться внутри них с помощью закладок – уже настоящих табов, а не тех их прототипов, что были в twm. Кстати, особенность эта до сих пор остаётся уникальной не только для оконных менеджеров, но и для десктопов.

Появившийся несколько позже (в 2002 году) Openbox также поначалу был клоном Blackbox'a, то есть основывался на его кодовой базе. Однако затем он был переписан на чистом C (Blackbox и Fluxbox написаны на C++), чем приобрёл самобытность, хотя и сохранил минимализм интерфейса предтечи. Однако главная составляющая его самобытности – это графическое средство конфигурирования, ObConf. Оно обеспечило ему место оконного менеджера в рабочей среде LXDE, у которой с собственными средствами настройки была (и сохраняется до сих пор) некоторая напряжённость. Но об этом – в следующей главе.

И всё же ещё минималистичней...

Казалось бы, интерфейс минималистичней, чем у Blackbox'a, придумать трудно. Но предела совершенству нет ни в каком направлении – ни в усложнении, ни в упрощении. Что мы сейчас и проиллюстрируем.

Был некогда такой оконный менеджер – wm2 (что расшифровывалось просто – Window Manager 2). Разработанный Крисом Каттамом (Chris Cannam) в 1996 году, он отличался не просто простотой, а, я бы сказал, простецкостью. Ибо обеспечивал только перемещение окон, изменение их размера, скрывание и закрытие. Никаких других функций у него не было – ни виртуальных десктопов, ни средств запуска приложений, ни иконок, ни инструментов конфигурирования (кажется, и параметров конфигурирования не было тоже). И потому вид его был предопределён изначально. В частности, фирменной особенностью его была вертикальная ориентация строки заголовка.

Вероятно, автору этих возможностей (или, скорее, невозможностей) хватало. А вот Биллу Спитзаку (Bill Spitzak) – нет, хотя ему также были близки идеи минимализма и нравилась вертикальная ориентации строки заголовка. И потому он добавил в wm2 необходимые функции – расширенные средства управления окнами, средство запуска приложений из контекстного меню рабочего стола, поддержку виртуальных десктопов в неограниченном количестве. В результате чего получился FLWM (Fast Light Window Manager).

Появилось в FLWM и средство конфигурирования контекстного меню запуска программ, не требующее даже правки конфигурационных файлов. Достаточно было в каталоге `~/.wmx/`, создать подкаталоги, соответствующие пунктам меню любой желаемой структуры (до десяти уровней вложенности). И поместить в них символические ссылки на исполняемые файлы необходимых приложений. После чего в контекстном меню появляются новые пункты.

Последняя авторская версия FLWM (1.02) датируется 2006 годом. Однако заложенные в нём идеи минимализма получили дальнейшее развитие в самом минималистичном дистрибутиве Linux – Tiny Core, включающем его версию, усовершенствованную разработчиком последнего, Робертом Шингледекером (Robert Shingledecker). Именно в таком виде FLWM входит в репозитории ряда дистрибутивов (например, openSUSE и Ubuntu).

Оступление. В начале этой главы я писал, что работать в Иксах без оконного менеджера практически невозможно. Однако некогда это

было не совсем так. Офисный пакет StarOffice позволял обходиться без всяких управителей окон – достаточно было обеспечить автоматический запуск терминала при старте иксового сеанса, а уже из его командной строки вызвать Desktop Manager этого офисного пакета. Который обеспечивал все необходимые средства управления окнами – правда, только для входящих в него приложений, то есть текстового процессора, электронной таблицы и так далее. С окнами любых других программ StarOffice Desktop Manager работать не умел. Но не это ли вековая мечта любого руководителя – чтобы его сотрудники всё своё рабочее время занимались работой, а не играли бы в игры, слушали музыку и сидели в социальных сетях?

Способность работать без оконного менеджера была унаследована и первыми, после обретения свободы, версиями OpenOffice, но этот куплет – уже из другой песни.

Управители тайлингом

Как я уже сказал, «период бури и натиска» в развитии оконных менеджеров пришёлся на вторую половину 90-х годов. И на рубеже тысячелетий стало казаться, что все идеи в этом направлении исчерпаны. Идеи разумные были реализованы в удачных оконных менеджерах, получивших распространение и достигших той стадии совершенства, когда «хорошее улучшать – только портить». А оконные менеджеры, основанные на идеях неразумных, или просто неудачные, тихо сошли со сцены, и даже память о них затёрлась.

К тому же массовый приток новых пользователей из мира Windows (потому что больше им просто неоткуда было братья) вызвал снижение интереса к оконным менеджерам вообще – наступала эра интегрированных графических сред, выглядевших для мигрантов-«подоконников» более привычно. Да и линуксоиды первых призывов, вдоволь наигравших с редактированием конфигов и гс-файлов, всё чаще причаливали в тихой десктопной гавани. А оконные менеджеры всё больше становились инструментом энтузиастов.

Однако энтузиасты, как известно, потому так и называются, что ко всему относятся с энтузиазмом. В том числе и к интерфейсам. Им стало скучно в очередной раз реконфигурировать box'ы и FLWM'ы. И в первой половине нулевых годов они придумали новую парадигму

управления окнами – тайлинг, реализовав её к их середине в виде многочисленных тайловых (или фреймовых) оконных менеджеров.

В основе тайловых менеджеров лежит та же идея, что и у консольной утилиты `screen`: расщепление на ряд независимых областей (фреймов), подобное покрытию пола кафелем (*tiling*), в каждой из которых запускается какое-либо приложение. Собственно, к программам этой группы понятие «оконный менеджер» применимо достаточно условно: они обеспечивают управление не столько окнами, сколько теми самыми областями экрана, в которых эти окна открываются. Каковое может быть статическим, с жёстко определёнными размерами фреймов, и динамическим, при котором размеры фреймов изменяются при масштабировании окон запущенных в них приложений.

Распространение больших широкоформатных LCD-мониторов сделало идею тайлинга очень актуальной, и тайловые менеджеры получили широкое распространение. А элементы тайлинга были задействованы и в некоторых интегрированных средах (`Xfce`, `Unity`, особенно активно – в `Cinnamon`'е). Однако я тайловых менеджеров практически не использовал – для моих задач больше подходит принцип «один десктоп – одно окно». Так что описать их историю не могу – надеюсь, что кто-нибудь из знатоков и любителей тайловых менеджеров восполнит пробел в моём историческом обзоре.

Заключение

Подводя итог истории оконных менеджеров, процитирую великого русского поэта А.К. Толстого:

«К чему твоя баллада?» Иная спросит дева. – О жизнь
моя, о лада, Ей-ей, не для припева!

Так вот, и я сочинил обе заметки на заданную тему, дабы чтобы развеять одно распространённое заблуждение: будто бы разработчики графических интерфейсов Иксов только и делали, что заимствовали и копировали решения из `Windows` и операционных систем для `Macintosh`'а (мало кто нынче помнит, что до появления `MacOS X` они назывались очень просто – `System` с добавлением номера версии).

Дело обстояло как раз наоборот: если не считать общих корней GUI, произраставших из Xerox PARC, все остальные атрибуты современных графических интерфейсов, представляющиеся сейчас самоочевидными, впервые получили распространение именно в оконных менеджерах для X Window System. Это и активное использование всех трёх кнопок мыши, и множественные виртуальные рабочие столы, и виртуальное разрешение экрана, и управляющие панели, и контекстные меню, и многое другое. В послужной список Windows можно вписать только сомнительную честь изобретения кнопки Пуск. А к вящей славе Mac'овских систем всегда служило не создание новых парадигм, а умелая и успешная реализация существующих.

Глава двадцать четвёртая

В настоящей главе мы поговорим об истории комплексов программ, задумчиво именуемых интегрированными, или графическими, рабочими средами, они же – окружения (по английски – Graphic, или Intergated Desktop Environment). Впрочем, в народе их величают гораздо короче – просто десктопами или даже аббревиатурой DE.

Вступление

Если функции оконных менеджеров, о которых рассказывалось в предыдущей главе, сводятся, как следует из их названия, к управлению окнами, то задачи, стоящие перед десктопами, гораздо шире. Они в обязательном порядке включают в себя средства конфигурирования как самих себя, так и штатных приложений. Некий неперенный круг таких приложений, более или менее обширный, обладающих похожим по виду и функциональности интерфейсом, настраиваемых одним и тем же образом, также является неотъемлемой принадлежностью десктопов – почему они и носят имя интегрированных сред.

Средства конфигурирования десктопов изначально работали в графическом режиме, прямое редактирование настроек в них требовалось лишь в исключительных случаях. Ибо, вспоминая словам маршала Ланна, можно сказать, что GUI, который не может настроить себя своими GUI'евыми средствами – не GUI, а... нехорошее слово.

Что же до штатных приложений, то они, кроме оконного менеджера (оригинального или заимствованного) и всяких служебных утилит типа часов, регулятора звука, средств мониторинга, непременно включали в себя малый джентльменский набор, необходимый любому применителю: файловый менеджер (своего рода сердце среды), эмулятор терминала, простой текстовый редактор. Остальными приложениями разные десктопы комплектовались по разным принципам – от полного минимализма до собственного браузера и даже офисного пакета.

Оба эти фактора – простота и «всеобщность» конфигурирования, во-первых, и набор приложений, более или менее достаточный для начала работы большинства применителей, да ещё и выглядевших единообразно и столь же унифицированно настраиваемых, и обусловили мгновенную популярность интегрированных сред в широких народных массах – и не только среди начинающих пользователей. По сути дела, чистые оконные менеджеры остались уделом последователей Агафьи Тихоновны, то есть любителей

... губы Никанора Ивановича да приставить к носу Ивана Кузьмича, да взять сколь-нибудь развязанное, какая у Балтазара Балтазаровича, да, пожалуй, прибавить к этому ещё дородности Ивана Павловича...

Так что история десктопов, хотя и более короткая, чем у оконных менеджеров, оказалась весьма насыщенной событиями и куда как более драматической. Впрочем, и предыстория их также глубоко уходит в дебри прошлого.

Предыстория десктопов

Разумеется, в этой статье будет говориться только о тех десктопах, которые работают в открытых и свободных UNIX-подобных системах, и сами принадлежат к миру FOSS. И потому за точку отсчёта времени в истории интегрированных сред можно принять осень 1996 года – начало работы Маттиаса Эттриха (Matthias Ettrich) над средой KDE. Однако рассмотрение предыстории вопроса опускается куда глубже, в недра проприетаризма.

Отдалённым прототипом современных интегрированных сред были первые, ещё до-иксовые графические интерфейсы фирмы Sun – SunView (Sun Visual Integrated Environment for Workstations, изначально SunTools), затем NeWS (Network extensible Window System) и, наконец, OpenWindows, о которых я упоминал в главе про историю Иксов. Они предназначались для операционной системы SunOS – фирменного варианта UNIX: именно здесь впервые была реализована идея интеграции операционной системы, GUI и пользовательских

приложений, получившая дальнейшее развитие не только в мире UNIX, но и далеко за его пределами.

Именно графические интерфейсы, пришедшие из совершенно других миров – не UNIX'овых и не свободных, миров OS/2, Macintosh'a и Windows, оказали определяющее влияние на десктопы, о которых вскоре пойдёт речь.

Ныне мало кто помнит, но OS/2 не включала графический интерфейс пользователя как непременный атрибут операционной системы, однако обладала оригинальной и весьма совершенной графической оболочкой, Presentation Manager, позднее Workplace Shell (WPS). Она отличалась исключительной аккуратностью и вылизанностью, благодаря чему и послужила образцом при создании первого собственно Иксового десктопа – среды CDE.

Среда CDE (Common Desktop Environment) была разработана в 1993 году под эгидой The Open Group и при участии Hewlett-Packard, IBM, Novell и Sun. Она основывалась на библиотеках Motif и включала в себя оконный менеджер VUE (или HP-VUE – Visual User Environment), ранее применявшийся в HP-UX. Она быстро стала стандартным графическим интерфейсом для всех проприетарных UNIX'ов.

Как легко догадаться, CDE (как и лежащие в её основе библиотеки Motif) не была ни свободной, ни открытой. Именно потому влияние её на дальнейшее развитие десктопов, за единственным исключением (и, разумеется, самой идеи интеграции), оказалось не очень большим. Не изменило ситуации и открытие в начале десятых годов исходников и Motif'a, и CDE, ибо состоялось оно по принципу: возьми, боже, что мне не гоже. В то время, когда DOSS-десктопы давно превзошли своего первопредка и по функциональности, и по удобству.

Отступление. Среда CDE основывалась на библиотеке Motif – в те времена стандартном наборе для графических интерфейсов проприетарных UNIX. На нём же основывалось и множество приложений графического режима, некоторые из них были открытыми. Но, поскольку сами библиотеки были закрытыми, свободными эти приложения быть не могли.

Поэтому был изобретён свободный аналог – библиотека OpenMotif. Правда, функционально обеднённый – и не все Motif-приложения могли быть с ним собраны.

В дальнейшем, уже в начале текущего десятилетия, и библиотека Motif, и среда CDE были последовательно открыты для народа. Но к тому времени они вышли из моды, морально устарели, и казалось что народу не нужны. Однако буквально в то время, как в эту книгу вносилась последняя правка (1 марта 2014 года), мир облетела весть о выходе первой, со времён открытия исходников (и, следовательно, первой свободной), версии CDE, за номером 2.2.1. Пока она доступна только в виде исходников, адаптированных, как говорят разработчики, для сборки в любом дистрибутиве Linux или BSD-системе. Что из этого получится – проживём, увидим.

Ранние версии Windows, с версии 1-й по 3.X, представляли собой обычные графические псевдонногозадачные надстройки над DOS, хотя нынче об этом и не любят говорить вслух. Имя графическим DOS-надстройкам тогда было – легион: достаточно вспомнить такие среды, как GEM (много лет служившую для запуска Ventura Publisher), DesqView, GEOS и GeoWorks. И это не говоря уже о том, что многие популярные приложения, вроде Lotus 123, QuattroPro или WordPerfect, располагали собственными графическими оболочками. Влияние этих Windows за пределами своего мира также было невелико. Звёздный час Windows наступил в 1995 году, с выходом версии его имени. Но к этому вопросу мы ещё вернёмся.

В отличие от OS/2 и DOS/Windows, операционная система Macintosh'a, именовавшаяся в те времена просто и скромно – System 4, 5, 6 и так далее, – изначально в качестве неотъемлемого компонента включала в себя графическую среду, неотделимую от собственно операционки. На десктопы свободных UNIX-подобных систем она поначалу оказала влияние сугубо косвенное. Хотя и в те времена, как и по сей день, выступает в качестве своего эталона графического интерфейса пользователя.

Очень важной вехой в развитии графических интерфейсов оказалась система NeXTStep, о которой упоминалось в предыдущей главе. Однако это стало ясно годы спустя, после того, как её достижения были инкорпорированы в недра MacOS X.

Воистину судьбоносным в истории графических сред стал август 1995 года – момент выхода Windows 95. Именно в ней впервые появляется центральный элемент большинства графических интерфейсов последующих лет – сакраментальная кнопка **Start** (она

же **Пуск**), располагающаяся на главной управляющей панели и вызывающая каскадное меню приложений.

Справедливости ради надо отметить, что прообраз этой кнопки, нёсший на себе изображение надкусанного яблока, появился в Macintosh'евских System. Управляющая панель в Windows 95 также в значительной мере унаследована от графической среды Macintosh'a, хотя нельзя исключить и влияние ранних оконных менеджеров Иксов. Наконец, из Иксов же, прямо или косвенно, в «девяностопятке» заимствуется идея контекстных меню рабочего стола.

Я надеюсь, никто не заподозрит меня в излишних симпатиях к Самой Великой ОС всех времён и народов. Однако, вопреки расхожему мнению, я не склонен считать, что разработчики Windows 95 взяли и просто так потибрили все перечисленные компоненты откуда бы то ни было. Во-первых, многие из них восходят к далёким временам экспериментальных графических интерфейсов, разрабатывавшихся в Исследовательском центре Пало-Альто компании Hewlett-Packard (PARC) и оказавших огромное влияние на все последующие графические системы и среды без исключения.

Во-вторых, интеграция всех указанных компонентов в виде, доступном для восприятия тем самым пресловутым простым пользователем, о котором так любят говорить и в проприетарной сфере, и в мире FOSS – это неоспоримая заслуга разработчиков Windows 95.

Так или иначе, Windows 95 создал тот образ графического интерфейса, который быстро завоевал популярность среди широких народных масс. И с этим образом приходилось считаться разработчикам любых десктопов, ориентированным на ту же целевую аудиторию.

Рождение KDE

Не стал исключением и Маттиас Эттрих – создатель первого интегрированного десктопа для свободных UNIX'ов, когда он в 1995 году приступил к разработке KDE. Маттиас, в то время студент университета в Тюбингене, уже приобрёл известность как разработчик LyX – системы компьютерной вёрстки, которую с некоторой условностью можно считать надстройкой над TeX. И первые шаги по

сбору команды для работы над KDE он предпринял в списках рассылки своего проекта.

Согласно легенде, Эттрих занялся разработкой интегрированного десктопа – делом тогда ещё новым и неосвоенным, для того, чтобы его девушка чувствовала себя в Linux'e так же комфортно, как и в Windows 95, к которой она, видимо, успела привыкнуть.

Так что, создавая KDE, Эттриху пришлось включить в неё и управляющую панель в стиле Windows 95, и кнопку Start. Однако было бы неправильно считать, что он тупо скопировал интерфейс Самой Великой ОС, о чём любят говорить неосознательные (или неосведомлённые) граждане. Ибо гораздо больше в интерфейсе KDE было унаследовано от старых добрых оконных менеджеров для Иксов. В ней были и развитая система контекстных меню, и множественные виртуальные десктопы.

Как мы помним по прошлой главе, про оконные менеджеры, всё это в совокупности имело место быть и в оконных менеджерах, воспроизводивших внешность Windows 95, таких, как FVWM95 и IceWM. Однако в KDE имелось и кое-что ещё, и кое-что другое, а именно – набор штатных приложений. Набор этот включал в себя файловый менеджер kfm в стиле пресловутого Windows Explorer, эмулятор терминала konsole, сразу два текстовых редактора – KEdit типа Notepad'a и более «продвинутый» KWrite. Все эти приложения имели интерфейс в едином стиле, хорошо вписываемый в среду.

Отступление. Как показывают многочисленные вопросы на форумах, нынче мало кто помнит расшифровку названия KDE. А расшифровывалась эта аббревиатура очень просто: Kool Desktop Environment. Впрочем, это не прибавляет ясности и помнящим те времена почти былинные. Ибо никто так и не знает, что же вкладывалось создателем в слово Kool – его нет ни в английском языке, ни в немецком. Разве что если воспринимать его как попытку записать по немецким правилам английское слово Cool? Намёки на что можно найти в ранних обзорах этой среды.

Видимо, разработчики быстро поняли это, и ныне литера K в имени этого десктопа не означает ничего, кроме самой себя. То есть его название можно перевести на русский язык как «Рабочее окружение некоего гражданина K» – и не более того.

Самое же главное – в KDE имелось средство тотальной настройки, KDE Control Center (KCC), позволяющее настраивать параметры как самой среды, так и всех штатных приложений – этого в столь интегрированном виде ни у одного из оконных менеджеров для Иксов тогда не было и в помине.

Отступление. В традициях раннего KDE было называть все приложения для неё именами, начинающимися с заглавной буквы K, или, в редких случаях, ею оканчивающихся. В случае с эмулятором терминала это послужило основанием для многолетней путаницы, продолжающейся по сей день.

KDE создавалась как среда не исконно немецкая, а интернациональная, и потому имела английский интерфейс. По-английски текстовая консоль, или виртуальный терминал (console) и терминальная программа из KDE (Konsole) пишутся несколько по-разному. По-немецки же они пишутся абсолютно одинаково (konsole) и столь же одинаково произносятся. Кстати, в русском языке ситуация аналогичная.

В результате многие пользователи, и не только начинающие, явно или не явно, но не делают различия между работой в текстовой консоли и в командной строке эмулятора терминала Konsole.

С самого начала KDE задумывалось как среда самодостаточная, снабжённая комплексом приложений, необходимых для практической работы. Создание вокруг проекта сообщества разработчиков позволило реализовать эту идею на ранних его стадиях.

Кроме перечисленных ранее приложений, ней очень быстро появились собственные средства для работы с сетью, комплекс мультимедийных программ, кстати, едва ли не первый в своём роде, наборы для украшения рабочего стола.

Всё это было очень логично организовано в несколько «авторских» пакетов, таких, как kdelibs – библиотеки, дополняющие основополагающую библиотеку Qt, kdatabase – базовые приложения среды, kdenetworks – сетевые средства, kdemultimedia очевидного назначения, kdeartworks – набор украшений, и так далее. А вскоре для KDE был создан даже офисный пакет KOffice, развиваемый и поныне под именем Calligra.

Среда KDE, как и все её приложения, основывалась на библиотеках Qt, разработанной незадолго до этого (также в 1996 году)

норвежской фирмой Trolltech. Библиотека эта распространялась с открытыми исходниками, но не была свободной в понимании GNU/FSF. Ибо имела две версии – платную для коммерческого использования и бесплатную – для использования некоммерческого. Последняя и была положена в основу KDE и её собственного набора библиотек – kdelibs.

Не вполне свободный характер библиотеки Qt и послужил завязкой всего последующего сюжета, стержнем которого стало противостояние KDE и GNOME. Не смотря на то, что вскоре (осенью 1998 года) некоммерческий вариант Qt/X11 стал распространяться под лицензией GPL, работа над GNOME уже началась. Так что если бы Эттрих выбрал бы для своего десктопа любую другую библиотеку (а дальнейшие события показали, что он сделал лучший выбор), был бы найден более иной повод для создания «истинно свободной» интегрированной среды.

На самых ранних стадиях разработки KDE вокруг проекта собралась небольшая, но сплочённая группа товарищей, в основном вполне студенческого возраста. Они создали и некоммерческую организацию KDE e.V. (eingetragener Verein – зарегистрированное объединение, по немецки), уставной фон для которой набрали из карманных денег. И, как свидетельствуют очевидцы, например, один из соучередителей, Маттиас Далхаймер (Matthias Kalle Dalheimer), процесс разработки KDE проходил весьма весело, в лучших традициях университетских буршеншафтов.

К слову, Маттиас Далхаймер в те далёкие годы работал в фирме Star Division – той самой, в которой разрабатывался кроссплатформенный офисный пакет StarOffice, предшественник современных OpenOffice и LibreOffice. И занимался он там как раз его портированием на Linux – исходно тот был предназначен для OS/2. Но эту историю я расскажу как-нибудь в другой раз.

А ещё Маттиас – один из авторов последних изданий знаменитой книжки «Запускаем Linux» (Running Linux). Написанная в 1995 году Ларом Кауфманом (Lar Kaufman) и Мэттом Уэлшем (Matt Welsh), она, обрастая соавторами, за десять лет выдержала 5 изданий. И стала настольной книгой многих поколений линуксоидов, как начинающих, так и действующих.

KDE – в жизнь

Вследствие не вполне свободного характера библиотеки Qt, основанная на ней среда KDE была настороженно принята столпами дистрибуции, в первую очередь ревнителями идеологической чистоты, такими, как Debian и Red Hat.

Так что поначалу KDE самостоятельно собиралась исключительно энтузиастами в рамках более иных дистрибутивов. Однако звёздный час её был недалёк: в июле 1998 года выходит первая версия первого по настоящему юзерофильного дистрибутива – Mandrake Linux, потомком которого является современная Mandriva и ряд её дериватов.

Первая версия Mandrake, как ни странно, носила номер 5.1, ибо представляла собой достаточно точный клон Red Hat 5.1, появившегося на свет незадолго до того, весной 1998 года (кажется, с тех пор и пошла традиция для клонов – наследовать номера версии прародительской системы). Но её «юзерофилия» как раз и заключалась в том, что она штатно включила в себя среду KDE с её штатными приложениями, в том числе графическими и мультимедийными.

Более того, среда KDE была в первом Mandrake десктопе по умолчанию. А сам дистрибутив этот оказался первым, вообще получившим умолчальный десктоп. Ранее такого понятия просто не существовало – и тому было много причин, одна из которых – отсутствие свободных десктопов вообще (оговорки на счёт Qt будем считать юридическим крючкотворством, тем более что и повод для них скоро пропал).

Вслед за Mandrake, осенью 1998 года, среда KDE в качестве умолчальной была включена в версию 5.3 SUSE. И с тех пор судьба обоих этих дистрибутивов оказалась тесно с ней связанной. Хотя, разумеется, ни в том, ни в другом KDE не был единственным десктопом, да и простыми оконными менеджерами они не оскудели. Но, как говорится, оба они «затачивались» в первую очередь под KDE.

А затем, на рубеже тысячелетий, поднялся первый вал «Linux'ов с человеческим лицом», или, точнее, первых систем быстрого развёртывания, начиная со StromLinux'a. За которым последовал вал второй – VectorLinux, MEPIS и другие, ряд которых существует и сегодня. И во всех этих системах в качестве десктопа, теперь уже не просто умолчального, а единственного штатного, задействовалась

среда KDE. Хотя в то время она была уже не единственным представителем своего класса, имелись и альтернативы, о которых пойдёт скоро пойдёт речь.

Среда для холериков

Как было только что сказано, к рубежу тысячелетий доминирование KDE в ряду графических интерфейсов свободных UNIX-подобных систем. Однако KDE существовало не в безвоздушном пространстве, в чём мы сейчас и убедимся.

Буквально через несколько месяцев после выхода KDE, появляется среда Xfce. Говорят, что первоначально название её расшифровывалось как XForms Common Environment – разделяемая среда на библиотеке XForms. К тому времени, как я её впервые увидел, та же аббревиатура трактовалась как The Cholesterol Free Desktop Environment – «обезжиренный» свободный десктоп. Хотя мне больше нравилась трактовка названия как среды для холериков – её действительно отличала не просто быстрота реакции на действия пользователя, но именно порывистость. Однако давно уже в Xfce не осталось и следов от библиотек XForms, да и того взрывного впечатления, как некогда, она уже не производит. Так что, подобно большинству аббревиатур FOSS-мире, ныне Xfce не расшифровывается никак.

Среда Xfce была создана французом Оливером Форданом, и не столько под впечатлением и по примеру KDE, сколько как попытка создания свободного аналога CDE. И в интерфейсе её ранних версий чётко прослеживается влияние первой рабочей среды UNIX'ов.

И строилась она по совершенно другим принципам, нежели KDE. Если последняя стремилась стать самодостаточной средой со всеобъемлющим набором пользовательских приложений, то в Xfce, кроме средств управления окнами и сквозной настройки, имелся лишь очень небольшой комплект базовых утилит и приложений, типа файлового менеджера и эмулятора терминала. Благодаря этому Xfce была средой очень компактной и быстродействующей.

Среда Xfce основывалась на наборе библиотек XForms весьма сложного происхождения, но восходившем в конечном счёте к IRIS GL, проприетарной графической библиотеке компании Silicon Graphics

(ныне SGI). Да и сама XForms в то время не была свободной, а только лишь бесплатной для личного некоммерческого использования (настоящую свободу в понимании FSF она обрела годы спустя). И потому отношение к ней дистрибьютеров было не менее настороженным, нежели к KDE. То есть её существование игнорировалось и Red Hat'ом, и Debian'ом. А поскольку по своей функциональности Xfce не намного превосходила развитые оконные менеджеры, то и большого интереса энтузиастов она тоже не вызывала.

Отчётливо маргинальное положение занимала среда Xfce во время жизненного цикла своей 1-й и 2-й версии. Версию же 3-ю Фордан весной 1999 года полностью переписал с использованием библиотек Gtk – тех самых, которые, как мы скоро увидим, перед тем легли в основу среды GNOME. В результате этого «освобождения» Xfce стала включаться как дополнительный десктоп в ряд дистрибутивов, в частности, в ту же Mandrake и её русскую редакцию. Однако дистрибутива, в котором она выступала бы десктопом по умолчанию, не существовало вплоть до середины нулевых, когда появились первые «лёгкие» системы быстрого развёртывания, такие, как Zenwalk, в которых она пришлась очень ко двору. Но это уже переход к следующему этапу нашей истории.

А пока – несколько слов об Xfce версии 3. Именно в отношении её название среды трактовалось «с холерическим уклоном» – и к тому были все основания. Она работала чрезвычайно быстро и, так сказать, «импульсивно». Ощущение это передать словами почти невозможно, но оно осталось незабываемым – больше нигде и никогда мне такого испытать не довелось.

Да и интерфейс её тогда сильно отличался от уже примелькавшихся к тому времени KDE и GNOME, не обнаруживая ни малейшего намёка на «Windows-подобие», и скорее вызывая в памяти WPS из OS/2 эпохи «Кривого Мерлина».

Завершая этот раздел, ещё раз подчеркну: появление Xfce не оказало большого влияния на мир рабочих сред для свободных UNIX-подобных систем. Ибо ось развития их проходила по линии противостояния KDE, о котором уже шла речь, и GNOME, к рассмотрению истории которого мы сейчас перейдём.

Disclaimer, он же – отмазка. Однако прежде чем перейти к истории GNOME я должен принести слова покаяния. Настоящая

книга, как неоднократно декларировалось, написана, по мере возможности, в соответствии с заветом Тацита — «без гнева и пристрастия». Один раз, в главе девятнадцатой, я уже отступил от этого завета. И ныне вынужден сделать это вторично.

Ибо, когда речь заходит о среде GNOME, мне этого завета придерживаться трудно. Разумеется, никакого гнева против этого десктопа я не испытываю. Нет и особого пристрастия или противострастия — одно время я даже был его пользователем. А все предыдущие годы честно (хотя и безуспешно) пытался найти те самые его несравненные достоинства, о которых столько говорили его приверженцы.

Нет, речь идёт не о гневе или пристрастии, а скорее о чувстве недоумения: почему такая эклектическая, неорганизованная и даже не очень оригинальная в идейном плане среда завоевала столь широкую популярность. Разумеется, к тому были некоторые объективные предпосылки, о которых я скажу по ходу дела. Но во главе угла всё-таки стоял субъективный фактор — тот самый, который можно назвать кредитом доверия пользователей. И который, как мы увидим, оказался фактически безграничным, хотя разработчики GNOME раз за разом подвергали его испытанию на прочность. Причём таким, на какие не решалась даже компания Microsoft, чья Windows даже на заре своего существования имела несоизмеримо большую пользовательскую базу.

В общем, говоря об истории GNOME, мне очень трудно сохранять объективность — что и прошу учитывать читателя.

Радетели свободы

Однако вернёмся к истории. Пока и KDE, и Xfce базировались на не вполне «идеологически выдержанных» библиотеках, истинные радетели свободы готовили удар по проприетаризму. Во главе этих радетелей встали Мигель де Икаса (Miguel de Icaza) и Федерико Мена-Кинтеро (Federico Mena-Quintero), основавшие в 1997 году проект GNOME (GNU Network Object Model Environment). Его можно рассматривать как ответ «твёрдых искровцев»... пардон, истинных Freesoftware'овцев прихлёбным плюралистам из проекта KDE.

Целями проекта, насколько можно понять замысел его создателей, были (в порядке убывания приоритетов):

1. создание полностью свободной, в терминах FSF, интегрированной рабочей среды;

2. создание интегрированной среды, не похожей ни на одну из существующих сред;

3. создание интегрированной среды, доступной для всех пользователей, вне зависимости от уровня их технической подготовки.

Достижение первой цели требовало выбора соответствующей библиотеки, на которой базировалась бы новая среда. А единственной по настоящему свободной графической библиотекой в то время была базовая xlib из штатной поставки XFree86, функционала которой для построения интегрированного десктопа было явно недостаточно. Более развитые же библиотеки, о которых я упоминал – Motif, XForms, Qt, – имели те или иные ограничения в использовании распространении.

И тут на помощь пришёл случай. Начиная с 1995 года Спенсером Кимбеллом и Питером Маттисом велась разработка растрового графического редактора, который получил название GIMP, что первоначально означало General Image Manipulation Program (букровка G приобрела значение GNU в 1997 году, года эта программа удостоилась чести войти в данный проект).

В ходе своей работы авторы программы столкнулись с острой потребностью в высокоуровневой графической библиотеке. Каковую они и создали, назвав её Gtk – GIMP ToolKit. Её первая стабильная версия была выпущена весной 1998 года под архисвободной лицензией LGPL. Библиотека эта создавалась с вполне конкретной, и достаточно узкой, целью. И в этом качестве она показала себя самым лучшим образом. Но как основа для интегрированной среды она не планировалась. Однако требование свободы оказалось сильнее технологических соображений, да и выбора-то особого у создателей GNOME не было: все остальные высокоуровневые графические библиотеки (Motif, Xforms, Qt) в то время несли на себе пятно тлетворного влияния проприетаризма. И это наложило отпечаток на всё дальнейшее развитие среды.

Второй целью, не намного менее важной, нежели свобода, было создание среды, ни на что не похожей. По этой причине в качестве оконного менеджера GNOME был выбран Enlightenment – невероятно

эффектный, но не очень эффективный в работе и очень неудобный в настройке.

Так что третья цель – создание простой в использовании рабочей среды – по уровню приоритености оказалась где-то тридцать третьей. Да, первая версия GNOME, вышедшая весной 1999 года, была фантастически красива, загружаясь с заставкой в виде анимированного театрального занавеса. Вот только настраивать её было не очень просто. Да и до настройки обычно дело не доходило. Ибо среда эта в те годы хорошо умела делать только одно: свопировать саму себя на винчестер. По крайней мере, при работе в GNOME индикатор его активности обычно не гас ни на мгновение.

Тем не менее, GNOME получил активную поддержку со стороны дистрибьютеров – тех самых, которые в штыки встретили KDE и просто проигнорировали Xfce. Практически сразу после выхода первого релиза он становится десктопом по умолчанию в Debian'e (март 1999 года) и в Red Hat'e (апрель 1999 года). Не побрезговали им и разработчики Mandrake, включив GNOME в качестве опциональной рабочей среды – именно в составе русской редакции этого дистрибутива я и пытался ознакомиться с его первыми версиями.

Однако, насколько я могу судить, объявление GNOME умолчальным десктопом в Debian'e и Red Hat'e было скорее теоретическим (или, если угодно, политическим). Среди лично мне знакомых в те годы дебианистов и редхатчиков пользователей этой среды я не припоминаю. Ну а в Mandrake он вообще присутствовал для галочки – чтоб было. И причина была в практической непригодности GNOME к реальному применению. Именно в те годы и родилось высказывание:

KDE разработали обычные программисты для удобства собственного и пользователей. А GNOME создали крутые программеры для доказательства собственной крутизны.

Надо сказать, что в те годы для такого утверждения были все основания. Да и ныне, во времена GNOME 3 и GNOMEShell, оно не выглядит совсем беспочвенным.

Великое противостояние

Разумеется, так продолжаться не могло: одно из знамён свободного софта практически не применяется его сторонниками. И разработчики GNOME вынуждены были, поступившись своей крутизной, пойти по пути упрощения.

Началось всё со смены оконного менеджера – место красивого, но непрактичного Enlightenment'a занял непритязательный, но пригодный к употреблению Sawfish. А поскольку собственного оконного менеджера у GNOME не было никогда, смена «управителей окон» с тех пор стала в этой среде традиционной.

Следующим шагом на пути упрощения стал выход 2-й версии GNOME, в которой простота настройки и использования были объявлены высшими приоритетами. А поскольку эталоном простоты в некоторых кругах в то время считалась Windows, в сети прозвучал лозунг:

Сделать GNOME большей Windows, чем сама Windows.

Правда, популярности этот девиз не снискал, и о нём постарались поскорее забыть. Однако репутация среды, в которой может работать даже кухарка, за GNOME закрепилась прочно. Хотя функционал штатных приложений его оставлял желать лучшего. А большинство хороших приложений, включавшиеся майнтейнерами в состав GNOME, разрабатывались независимо от него.

С другой стороны, KDE, при всех его многочисленных недостатках (а старым верным пользователям этого десктопа они были известны лучше, чем любым критикам), обладал тремя несравненными достоинствами:

1. обильным функционалом в отношении управления окнами и приложениями;
2. абсолютной настраиваемостью всего и вся;
3. набором штатных приложений и приложений от примкнувших сторонних разработчиков.

За средства настройки KDE подвергалась критике. И местами справедливой: они были запутанными, интуитивно не очень понятными, настройки связанных, казалось бы, параметров, часто оказывались раскиданными по противоположным пунктам меню. Но они были – и охватывали абсолютно все опции системы. Причём всё

это поддавалось конфигурированию штатными средствами: необходимости в ручной правке конфигов практически не возникало. Разве что, как это часто бывало в истории (и неоднократно повторялось впоследствии – в том числе и в GNOME), она требовалась при настройке русской раскладки клавиатуры и её переключателей.

В силу перечисленных факторов первой половине нулевых годов вокруг KDE сплотилось сообщество пользователей-прагматиков, нуждавшихся в качественном софте для решения своих задач в среде, поддающейся стопроцентной индивидуальной настройке. В том числе и пользователей, никакого отношения к компьютерам не имеющих – юристов, переводчиков и прочих гуманитариев. Нельзя сказать, что таких не было среди пользователей GNOME – однако в этом сообществе до некоторого момента всё-таки доминировали компьютерные гики, вынашивающие планы изменения мира.

Отступление. В шутке про изменение мира есть только доля шутки. Одна бывшая наша соотечественница, будучи уволенной из NASA чуть ли не по подозрению в шпионаже (в пользу русской мафии, надо полагать), в поисках работы обратилась в том числе и в GNOME Foundation. Где на собеседовании её спросили: хотите ли вы участвовать в изменении мира? Вопрос этот показался ей смешным, и в результате в этот самый фонд её не взяли. Пришлось идти на работу в маленькую заштатную конторку, которая тогда называлась (да и сейчас называется) Google. А после капитализации последнего пришлось стать очень богатой женщиной. Так что политика GNOME подчас благотворно отражается на судьбах людей, с ней соприкоснувшихся.

Не случайно все опросы, касающиеся используемого десктопа, приносили в те годы чистую победу KDE – и с большим, подчас двухкратным, перевесом. Правда, в таких опросах, как правило, не участвовали представители так называемого корпоратива, использующие, как правило, не тот софт, который хочется, а который прикажут. И потому картину эту нельзя считать совсем точной. Ибо с момента зарождения GNOME его интенсивно поддерживала самая мощная Linux-компания – Red Hat, в те годы бесспорный лидер корпоративного поля. А в 2003 году, после того, как фирма SUSE, выпускавшая одноимённый дистрибутив, была приобретена компанией Novell, за внедрение GNOME взялся игрок номер два. Ибо

после расщепления этого дистрибутива на две ветки – коммерческую SLE (Suse Linux Enterprise) и развиваемую сообществом openSUSE, в первой как десктоп по умолчанию был принят GNOME.

Почему – можно только гадать: ведь, как уже говорилось в предыдущем разделе, именно SUSE была одним из основных оплотов KDE. В качестве одной из причин указывалось то, что незадолго перед тем Novell прикупила компанию Ximian, занимавшуюся разработкой Mono – свободной реализации Microsoft .NET Framework. А поскольку основателем Ximian'a был Мигель де Икаса, технологии Mono были тесно завязаны на GNOME.

Не скидывая со счетов этого факта, можно предложить и более простое объяснение. Приобретя SUSE, Novell ориентировала коммерческую ветку этого дистрибутива на корпоративный сектор американского рынка. Где, как уже было сказано, главным игроком была компания Red Hat со своим RHEL (Red Hat Enterprise Linux) и его десктопом GNOME. Так что Novell просто пришлось равняться на конкурента.

Тем не менее, при несомненных успехах RHEL и SLE в корпоративном секторе, на пользовательских десктопах результаты их деятельности по продвижению GNOME были не очень заметны. А началом коренного перелома в этом продвижении стал конец 2004 года, когда в мировом масштабе развернулось распространение дистрибутива Ubuntu, в качестве рабочей среды по умолчанию использовавшей GNOME. Однако предпосылки этого перелома закладывались ещё ранее, году в 2002, в то время, когда Марк Шаттлворт только приступал к реализации своего проекта. И когда он принял судьбоносное решение – использовать в новом дистрибутиве не KDE, а GNOME.

О возможной причине этого я уже говорил в главе шестнадцатой. К сказанному там добавлю только одно. Настройка KDE усложнялась от версии к версии (пропорционально росту функционала), тогда как GNOME как раз в это время резко сменил вектор развития: как недавно было сказано, если его позиционировали как крутейший десктоп от крутейших программистов, то теперь было решено сделать простейший десктоп, который может настроить и кухарка. А поскольку Ubuntu ориентировался (в том числе и) на эту категорию пользователей, выбор GNOME представлялся вполне логичным.

Однако главным фактором для предпочтения GNOME, как мне кажется, было всё-таки стремление выделить Ubuntu из ряда развивавшихся тогда (и, как казалось, не без успеха) Систем Быстрого Равёртывания (СБР), в качестве десктопа использовавших KDE.

Так или иначе, но вследствие сочетания обоих факторов – массовой бесплатной рассылки дисков с Ubuntu и «упрощения» GNOME – началось широкое распространение последнего. Причём первый фактор был, безусловно, более весом: в ряды линуксоидов влилось свежее пополнение, которое ничего, кроме Ubuntu и GNOME, в глаза не видело, и для них эти понятия стали, по крайней мере по первости, тождественны понятию Linux. Однако и второй фактор нельзя сбрасывать со счетов: если бы GNOME показался неофитам слишком сложным, диски с Ubuntu были бы выброшены ими на помойку без всякого сожаления, ибо халява. Так что тогдашний расчёт Марка оказался точным. Хотя более поздние события показали неоднозначность его последствий.

От KDE 4 до GNOME 3

Следующее судьбоносное событие, повлиявшее на распределение пользователей по десктопам (или десктопов по пользователям?) произошло в начале 2008 года. Это был выход KDE 4.0.

Эта версия KDE разрабатывалась примерно столько же времени, сколько хватило, чтобы смениться KDE 1, KDE 2 и KDE 3. И в анонсах на ранних стадиях разработки обещались если и не золотые горы, то по крайней мере серебряные. В частности, там содержались толстые намёки на то, что KDE 4 будет обходиться без Иксов (хотя каким образом – не уточнялось). Правда, со временем анонсы становились всё скромнее (в частности, из них исчезло упоминание об отказе от X-сервера), но по прежнему оставались многообещающими.

Тем большим было разочарование, я бы даже сказал, шок, по выходе релиза KDE 4.0. Вместо стабильной, функциональной, настраиваемой среды с привычным интерфейсом мы увидели нечто глюкавое и падучее, лишённое половины функций и настроек. Причём уменьшение вариантов конфигурирования вовсе не сопровождалось его упрощением – напротив, оно стало ещё более запутанным. Но зато

всё это было обвешано, как новогодняя ёлка, всякими красотами в виде плазмоидов.

Надо сказать, что разработчики KDE 4.0 честно предупреждали, что эта версия предназначена ещё не для практического применения, а лишь для ознакомления с новыми возможностями и их тестирования. Но сделали они это достаточно невнятно, и майнтейнеры дистрибутивов дружно бросились включать её в свои новые релизы (обычно, правда, параллельно с предыдущей, 3.5.X). А пользователи, приняв всё это за чистую монету, столь же дружно стали «нулевую четвёрку» устанавливать.

Результат оказался более чем предсказуем. Поскольку было вполне очевидно, что дни 3-й ветки сочтены, начался массовый отток пользователей KDE на запасные аэродромы. Для кого им стал XFce, кто вернулся ко временам информационной юности – на менеджеры окон. Но немало бывших пользователей KDE попросило политического убежища в стане GNOME.

Отступление. Сейчас, по прошествии нескольких лет, становится понятно, что разработчики KDE почти всё сделали правильно. Допустив лишь одну маленькую тактическую ошибку: они обозвали релизом достаточно сырую тестовую версию. И никакие объяснения, что это не совсем таки настоящий релиз, не в силах были преодолеть магию Слова. Хотя, с другой стороны, не пойдя KDE'шки на такой отчаянный шаг, разбудивший здоровую рабочую злость и пользователей-тестируйщиков, и сторонних разработчиков – кто знает, может быть, 4-я версия KDE отлаживалась бы и по сей день в тестовом режиме.

Тут-то и выяснилось, что в GNOME образца 2008-2009 годов жить можно. Конечно, огорчал подход к конфигурированию среды, когда в интерактивном режиме был доступен лишь самый минимум, необходимый, по мнению разработчиков, «простому» пользователю, а для реализации более сложных (по их же представлениям) опций следовало лезть в нечто вроде реестра, ещё менее интуитивно понятного, чем опции настройки KDE, и гораздо более бедного. Видимо, разработчики GNOME руководствовались старым советским принципом:

Что позволено парторгу – не позволено бичу.

Поначалу удручало убожество убожество штатных приложений – например, Gedit'a в сравнении с Kate или Nautilus'a – в сравнении с Konqueror'ом старого образца. Однако в этом оказались и свои плюсы: в GNOME органично интегрировались любые сторонние приложения, основанные на Gtk. И Gedit для всамделишной работы легко заменялся Geany, а Nautilus – дополнялся PCManFM'ом. Кроме того, GNOME оказался органично интегрированным с такими дистрибутивами, как Fedora.

Так что на рубеже первых десятилетий нынешнего века между KDE и GNOME был достигнут примерный паритет. Хотя KDE и сохранял некоторое преимущество – за счёт а) стойких его приверженцев и б) энтузиастов новаторства. Тем более, что на месте эта среда не стояла, а постепенно допиливалась до юзабельного состояния, сама по себе, во-первых, и портированием на 4-ю ветку «трёшечных» приложений – во-вторых. Я, утратив в то время всякий интерес к KDE, не следил за этим процессом. Но, по слухам, примерно к версии 4.4 KDE стал вполне похожим на настоящий как сам по себе, так и в отношении своих штатных компонентов.

Тем не менее, отток пользователей от KDE казался необратимым, и GNOME утвердился с ним на одной ступеньке пьедестала почёта, может быть, ну на полступеньки ниже. Повторяю, заслуги самого по себе GNOME в приближении к вершине пьедестала были минимальны, ибо главные роли в этом спектакле сыграли уже упомянутые факторы – массовое распространение Ubuntu и ошибки команды KDE в продвижении новой версии своей среды. К этому надо добавить неожиданный рост популярности дистрибутива Fedora, тесно интегрированного с GNOME. Причём рост этот начался не благодаря GNOME, а скорее вопреки ему: многие пользователи готовы были смириться с этой средой ради достоинств самого дистрибутива. Который как раз в это время временно отказался от своей роли «тестовой площадки» коммерческого RHEL'a и обратился лицом к пользователю.

Отступление. Была и попытка реанимации KDE 3 в виде, привычном для пользователей этой ветки (и всех её предшествовавших версий). Это – проект Тимоти Пирсона (Timothy Pearson) под названием Trinity KDE. Он представлял собой

дальнейшее развитие 3-й ветки с некоторыми заимствованиями из 4-й – теми, что не шли в разрез с исконной идеологией KDE.

Увы – начатый во второй половине 2010 года, он явно запоздал. За два с половиной года, прошедшие с момента выхода KDE 4.0, эта среда в своём традиционном виде растеряла изрядную часть своих некогда преданных поклонников: одни, скрепя сердцем, погрузились в мир плазмоидов 4-й ветки, другие же мигрировали на более иные десктопы. А новые пользователи воспринимали «четвёрку» как данность – тем более, что к этому времени она стала пригодной к употреблению. И Trinity оказалась просто невостребованной широкими народными массами. Хотя некоторое число приверженцев и обрела.

Сказанное выше касалось распространения GNOME в мировом масштабе. А в нашей стране дело его укрепилось благодаря успеху проекта Russian Fedora, деятельность которого воплощена была в сборках RFRemix, кардинально улучшавшихся от версии к версии, вплоть до RFRemix 14 – апофеоза отечественного дистроения. Который (и это не только моё мнение) можно было смело рекомендовать русскоязычным пользователям любой сферы деятельности и почти любого уровня подготовки.

Казалось бы, вот он – тот самый готовый к десктопу Linux, наделённый человеческим лицом в виде интегрированной среды GNOME. То есть то самое, о чём говорили со времён первой версии Mandrake с его KDE. Однако – увы, это только казалось.

Ибо свежий ветер перемен затронул и GNOME, да так, что обрушился с ураганной силой. Потому что в апреле 2011 года принёс он с собой GNOME 3 с его GNOME Shell. Если и раньше, во времена GNOME 2, среда эта не утомляла своего пользователя избытком настроек, даже в собственном реестре, то с переходом к 3-й ветке настройки практически исчезли как класс – или, позднее, в очень ограниченном количестве стали доступны благодаря сторонним утилитам-твикерам. Что же до интерфейса – он выглядел настолько революционным, что от привычных рабочих сред, казалось бы, не осталось просто ничего.

Однако большинство нового – это хорошо испорченное старое. Так, оверлейный режим GNOME Shell'a вызывал в памяти времена, когда не было ещё фиксированных множественных рабочих столов, но

был зато единый рабочий стол с виртуальным разрешением в два и более раз выше физического (см. главу двадцать третью). А вертикальный ряд пиктограмм вдоль левого края экрана выглядели как реплика Window Maker'a, который, в свою очередь унаследовал эту идею от NeXTSTEP'a.

Не известно, позаимствовали разработчики GNOME Shell'a эти идеи у предшественников, или изобрели свои велосипеды независимо от них. Но в любом случае – ничего ультра-революционного в плане интерфейса эта среда не содержала. Разве что слияние виртуального рабочего стола с вертикальными пиктограммами. В результате чего получилось бы, возможно, вполне удачное нишевое решение – например для планшетов и прочих устройств с сенсорным экраном. Да вот беда – никто из производителей таких гаджетов использовать GNOME 3 не торопился: ниша эта была прочно оккупирована Android'ом. А для использования на обычном десктопе или ноутбуке решение это было мало пригодно. По крайней мере, для тех применителей, которые на десктопе или ноуте преимущественно работали, а не развлекались.

Однако разработчики GNOME Shell'a с этим не согласились. И попытались представить своё нишевое решение как универсальное. Причём с такой агрессивностью, с какой я не сталкивался за полтора десятка лет соприкосновения с миром Open Source. На любую критику в адрес своей среды у её фанатиков (не фэнов, не фанатов, а самых натуральных фанатиков) существовал универсальный ответ: всё это гениально и прогрессивно, а кто того не понимает --- ретроград и обскурант.

Было среди пользователей GNOME и другое течение – страсготерпцев. Не будучи приверженцами модерна, но сохраняя приверженность GNOME, они полагали, что надо смириться и привыкать к новомодному GNOME Shell'у, по принципу «Стерпится – слюбится». И надо сказать, что не все они смирились с положением «терпил»: некоторые стали активно создавать всякого рода твикеры для настройки «третьегнома» – то есть того самого, чем его собственные создатели пренебрегли, вполне цинично возложив эту миссию на «саких утопающих».

Наконец, возникло и третье течение – приверженцев старого и доброго GNOME. Тех, кто не хотел привыкать к GNOME новому и

недоброму. И они пошли своим путём, да не одним, а сразу двумя.

Первый путь был ультраконсервативным и заключался просто в создании форма GNOME 2, получившего имя MATE. В отличие от близкого по идее проекта Trinity KDE, этот проект был начат практически сразу по выходе «третьегнома», когда запас спортивной злости пользователей его 2-й версии ещё не был растерян в дискуссиях на форумах и блогах. И потому MATE развивался (и развивается) вполне успешно.

Отступление. Название десктопа MATE происходит от слова *mate* (с ударением на первом слоге) – латиноамериканского (в первую очередь аргентинского и парагвайского) чаеподобного напитка, изготавливаемого из листьев Падуба парагвайского. В указанном регионе он пользуется очень большой популярностью. Антони Арлетти в своей книжке Трампеадор свидетельствует, что знатоки

...могут почувствовать разницу между мате «этим вечером на том же самом месте» и мате «муж что-то подозревает». Ооо, это дьявольский мате!

А вот почему название десктопа принято писать целиком заглавными буквами – покрыто тайной.

Второй же путь в рамках третьего течения был более радикальным. Ряд пользователей 2-го GNOME, понимая, что для него слова, некогда увиденные царём Валтасаром, уже написаны, занялись стилизацией интерфейса GNOME 3 под внешний вид предшествующей версии. И, надо сказать, тоже не без успеха. Однако проект этот, получивший имя Cinnamon, быстро перерос первоначальные рамки. Но это уже не история, а современность, о которой сейчас сочиняется цикл на Блогосайте.

Здесь можно было бы сказать и о среде Unity – однако и она ещё не стала историей.

А что же другие?

Этот фрагмент истории десктопов был начат кратким рассказом о Xfce и ею же уместно его и завершить. Мы оставили её на уровне 3-й

версии, которая, будучи переписана на основе библиотек Gtk, обрела статус свободной среды. Но на этом её история не закончилась.

При подготовке релиза 4.0 Xfce снова была полностью переписана – теперь на основе следующей, 2-й, версии Gtk. И не просто переписана изнутри, а кардинальным образом сменила интерфейс. От бывшего сходства с CDE не осталось и следа – внешности её по умолчанию было придано GNOME-подобие, а со временем, в версии 4.6, она обрела и своего рода реестр `xfconf`, также представляющий собой аналог `GConf` из среды GNOME.

Потеряла Xfce и свой «холерический темперамент», хотя по прежнему оставалась достаточно «лёгкой» и быстрой (хотя по этим двум пунктам можно найти и возражения). Однако её компактность трудно подвергнуть сомнению. И благодаря этому она стала десктопом по умолчанию ряда систем быстрого развёртывания, строившихся по принципу «одна задача – одно приложение». Среди таких систем наибольшую известность получили, пожалуй, Zenwalk и его родственник – Salix. А затем она была задействована в Xubuntu – одном из представителей семейства дистрибутивов Ubuntu. Хотя сборки с Xfce нынче можно видеть и в большинстве «больших» дистрибутивов, таких, как Fedora, openSUSE, Debian и ряде других.

Однако, как я только что сказал, разрастание Xfce вызывало нарекания – хотя, на мой взгляд, не вполне обоснованные. Тем не менее, в противовес ей были созданы ещё более «лёгкие» десктопы – сначала LXDE, а затем Razor-qt, основанные на библиотеках Gtk 3 и Qt, соответственно. Ныне их разработчики обозначили намерение слить оба проекта воедино. Но здесь мы уже окончательно покидаем историю и переходим к современности.

Послесловие

На этом я завершаю своё сочинение, посвящённое истории UNIX, Linux и их графических интерфейсов. В нём осталось немало пробелов – например, я почти не затронул историю ряда BSD-систем, таких, как NetBSD и OpenBSD, сыгравших немалую роль в становлении мира свободного софта в том виде, в каком он существует сейчас.

Не говорилось в этом цикле и об истории приложений для UNIX и Linux, хотя некоторые её аспекты были весьма драматичны – например, эволюция StarOffice, превратившегося в современные Libre Office и Apache Open Office. Немало поучительного было и в истории других офисных пакетов для UNIX и Linux, начиная с первых из них, Applix и AbiSuite (реликтом последнего является современный AbiWord). Весьма поучительна была попытка портирования на Linux пакета Corel Office, включавшего, в том числе, и легендарный WordPerfect.

Самое же главное – я ни слова не написал о втором туре истории Linux'a на Руси, проходившего в нулевых годах и захватившего начало нынешнего десятилетия. Тур, который, начался после единения с миром российских дистрибутивов, Altlinux и ASPLinux. Она связана постепенным угасанием последнего, с появлением дистрибутива ROSA и тесно переплелась с историей французской Mandriva в её «второй жизни», начавшейся после смены руководства и команды.

В последних строках главы тринадцатой я обещал дать этому объяснение. Во-первых, история эта ещё не закончена, и когда и чем будет закончена – сказать трудно. Во-вторых, я не очень хорошо её знаю. В-третьих...

... в третьих, остаётся только вспомнить слова Козьмы Пруtkова про объятие необъятного и попрощаться со своими читателями. Может быть – до тех времён, когда историей станут события современности. А возможно, что кто-нибудь другой захочет восполнить лакуны в моём повествовании.

Оглавление

[Предисловие 1](#)

Преамбула	1
Референсы	2
Реверансы	2
Часть I. История систем	3
Глава первая. Вопросы праистории	3
Вступление	3
Истоки и составные части	4
Computer Science и ARPANET	5
Зарождение UNIX	6
«Железные» предпосылки	8
Компьютеры для народа	9
Глава вторая. Берклиада	10
«Похищение Елены Прекрасной»	10
Начало свободного софта	11
Возвращение в Беркли	13
Интермедия	14
FreeBSD: рождённая свободной	15
Технологический детектив	16
Цена свободы	18
Глава третья. Возвращаясь к UNIX'ам	19
Серенада солнечной долины	19
В оковах проприетаризма	20
Битва за стандарты	22
Глава четвёртая. Рождение Linux	23
Предпосылка первая, «железная»	23
Предпосылка вторая, «денежная»	24
Предпосылка третья: «студенческая операционка»	24
Герой выходит на сцену	26
Глава пятая. Берклиада, тур второй	27
NetBSD и OpenBSD: первый форк в благородном семействе	27
FreeBSD: десятилетие спокойствия	29
FreeBSD: переломный год	31
Оглядываясь вокруг	32
Предварительный итог...	34
FreeBSD на Руси	35
Глава шестая. Последний взлёт индивидуального	ОСетворчества 36

Пролог	37
Чем была AtheOS	37
Как она получалась...	38
Как устанавливалась...	38
И как работала	39
Эпилог	40
<i>Глава седьмая. «Драгонада» Мэтта Диллона</i>	40
«Железная» ретроспектива	41
Софтверная перспектива	42
У истоков новой системы	44
<i>Глава восьмая. MINIX 3: вторая жизнь</i>	48
Вступление	48
О микроядрах	48
Собственно о MINIX 3	50
Чего нет у_рыб?	52
Заключение	53
<i>Глава девятая. UNIX: хождения в народ</i>	54
Зачем нужен десктопный UNIX	54
Блеск и нищета персональных Workstation'ов	55
Sun Solaris: глотки свободы	57
Мир без Солнца, или бессмертна ли мафия?	58
Дети Солнца	59
<i>Глава десятая. Из истории файловых систем</i>	60
Вводные слова	60
Дисковая разметка	61
Массивы и логические тома	61
Файловые системы	63
Из истории систем размещения	69
Начало истории ZFS	70
Из истории юриспруденции	71
ZFS on Linux: технология против крючкотворства	72
<i>Часть II. История дистрибутивов</i>	73
<i>Глава одиннадцатая. Начало дистрибуции</i>	73
Был ли дистрибутивом первозданный Linux?	73
Начало начал	74
Slackware: первый шаг к Linux'у для всех	74
Debian: второй шаг к пользователю	76

<u>Red Hat: совсем для всех? 77</u>	77
<u><i>Глава двенадцатая. Лицом к пользователю 79</i></u>	79
<u>Suse: по стопам Red Hat 79</u>	79
<u>Caldera: предыстория грядущего 80</u>	80
<u>Магия Linux 81</u>	81
<u>Назад, в будущее: Gentoo и другие 82</u>	82
<u>На пути к гармонии 83</u>	83
<u><i>Глава тринадцатая. Linux на Руси 84</i></u>	84
<u>Предыстория русского Linux'а 84</u>	84
<u>IP Labs Linux Team: начало русского Linux'а 86</u>	86
<u>Новые игроки на Linux-поле 88</u>	88
<u>Горизонты раздвигаются 88</u>	88
<u>Единение с миром 89</u>	89
<u><i>Глава четырнадцатая. Slackware и первые СБР 90</i></u>	90
<u>Slackware и потомки 90</u>	90
<u>Первые СБР 90</u>	90
<u>Путь Дзэн 91</u>	91
<u>Предназначение 93</u>	93
<u>И несколько последних слов 94</u>	94
<u><i>Глава пятнадцатая. Debian: история клонирования 94</i></u>	94
<u><i>Глава шестнадцатая. Ubuntu и его клан 95</i></u>	95
<u>Начало истории 95</u>	95
<u>Распространение 96</u>	96
<u>Появление «разновидностей» 98</u>	98
<u><i>Глава семнадцатая. SUSE в истории 99</i></u>	99
<u>Представление семейства 99</u>	99
<u>Из предыстории 99</u>	99
<u>Начало самостоятельного плавания 100</u>	100
<u>К технологическим высотам 101</u>	101
<u>Рождение openSUSE 102</u>	102
<u>Соглашение с дьяволом 103</u>	103
<u>Парадоксы бизнеса 103</u>	103
<u>Продажа бессмертной души 104</u>	104
<u>Последствие 104</u>	104
<u>SUSE на Руси 105</u>	105
<u><i>Глава восемнадцатая. Год великого перелома 106</i></u>	106
<u>Этапы Большого пути 106</u>	106

Почему 2005-й?	107
Последствия первые...	107
... и отдалённые	108
«Десктопизация» Fedora	109
Не Fedora единой...	110
Конец истории	110
Глава девятнадцатая. Феномен Ubuntu	110
Немного ретроспективы	111
Появление героя	112
Реакция	114
Итог	116
Глава двадцатая. Linux: история применителей	116
Очень древнее вступление	116
Становление сословия пользователей	118
Прародитель «персонализации» Linux'a	120
Провал первой «персонализации»: технологические причины	121
Провал первой «персонализации»: психологические причины	122
Linux: «персонализация снизу»	123
Заключение	126
Часть III. История интерфейсов	126
Глава двадцать первая. История shell'ов	126
Появление первого шелла	127
Появление C-Shell	127
История tcsh	129
Наконец, Z-Shell	129
Глава двадцать вторая. Из истории Иксов	130
Вступление	130
Доисторический период	130
Рождение Иксов	131
XFree86 и другие	133
XFree86: немного о внутренностях	134
От XFree86 к Xorg	135
Хроника поступательного движения	138
Предпосылки Великого раскола	139
Повод к Великому расколу	139
Конец XFree86...	141
... и начало Xorg	142

Глава двадцать третья. Управители окон: извлечения из истории 143

Терминологическое введение 143

Предтечи управителей 144

У истоков управления окнами: twm... 144

... и его линия 146

По стопам Windows 95 147

По следам легенды 148

Линия боксов 150

И всё же ещё минималистичней... 151

Управители тайлингом 152

Заключение 153

Глава двадцать четвёртая 153

Вступление 153

Предыстория десктопов 154

Рождение KDE 156

KDE – в жизнь 158

Среда для холериков 158

Радетели свободы 160

Великое противостояние 161

От KDE 4 до GNOME 3 163

А что же другие? 166

Послесловие 167



Деннис Ритчи – являлся, так скончался 8 октября 2011 года