

## Крах иллюзий

<http://www.computerra.ru/think/240888/>

Порой кажется, что пора бурного развития вычислительной техники осталась позади. Unix написан 35 лет назад, все основные идеи GUI были заложены в семидесятые, до сих пор активно используются C и C++, придуманные несколько десятилетий назад, и даже Java скоро стукнет 15 лет. Статья Роба Пайка как раз об этом. Ему застой должен быть еще очевиднее: он много лет трудился в Bell Labs и участвовал в разработке того самого первого Unix, в 1981 г. написал первую оконную систему для Unix, а потом разрабатывал операционную систему Plan 9, которая должна была стать следующим поколением Unix, но не стала. Статья была написана в 2000 году (делайте на это скидку), только вот что изменилось с тех пор? Ничего. Разве, что сам Пайк ушел из Bell Labs. Теперь он работает в Google.

Исследования системного программного обеспечения потеряли смысл.

Все нижесказанное – отражение моих самых пессимистических ощущений от нынешнего состояния исследований систем. Не стану говорить о том, что внушает оптимизм. Этим и без меня есть, кому заняться. Ныне все восторгаются компьютерной индустрией. Не исключено, что все на самом деле не так мрачно, как мне представляется.

Однако, с моей точки зрения, ситуация сложилась крайне тяжелая, и требуются незамедлительные действия.

### Определим термины в первой фразе статьи

- исследования – преимущественно, академические изыскания, которыми занимаются университеты и несколько отраслевых лабораторий;
- системного – операционные системы, сети, языки – все то, что объединяет программы между собой;
- программного обеспечения – комментарии излишни;
- потеряли – в данный момент, не десять лет назад и, хотелось бы верить, не на последующие десять лет тоже;
- смысл – не оказывают никакого влияния на отрасль.

### Тезис

Исследования системного программного обеспечения приобрели второстепенный статус для бурно развивающейся компьютерной отрасли.

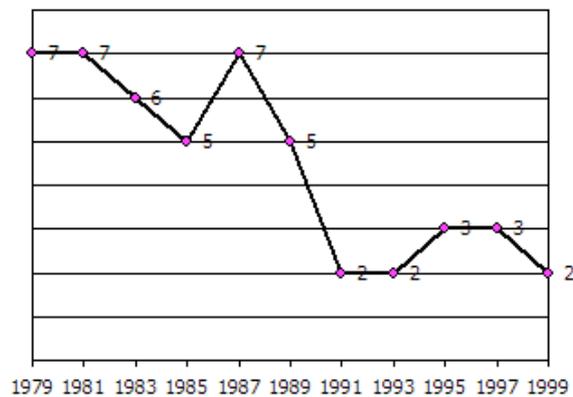
Как это ни смешно, в то время как компьютерные технологии практически отождествляют с инновациями, исследования и в области программного обеспечения, и в области аппаратных средств как в научных, так и в коммерческих кругах стали изолированными, закоснели и утратили важность.

Причин тому множество. Некоторых можно было избежать, а часть из них носят эндемический характер.

Возможно, есть пути к выправлению ситуации, но они потребуют общих усилий всего ИТ-сообщества.

### Упадок

На графике изображено количество новых операционных систем, представленных на Симпозиуме по принципам операционных систем (SOSP).



"Да кому вообще сейчас нужны новые операционные системы?" – спросят меня. Может быть, и никому, но это лишь подтверждает мои слова.

"Но сейчас выпускается огромное количество статей по файловым системам, производительности, безопасности, веб-кэшированию и т. п.", – скажут мне. Да, вот только кто-нибудь вне исследовательской сферы обращает на них хоть какое-то внимание?

### **Вклад системных исследований в бум ИТ-сферы**

Сравните программное и аппаратное обеспечение мощной рабочей станции в 1990 и в 2000 гг. Железо изменилось радикально, а в софте полный застой: те же операционные системы, те же программы, те же языки программирования (добавились только Java и немного Perl).

### **Где же инновации?**

Главным образом, в Microsoft. Для примера, сравните программы Microsoft десять лет назад и теперь. Если вы скажете, что это не инновация, а копирование, я отвечу, что Java по отношению к C++ – это то же самое, что Windows по отношению к Macintosh, ответ индустрии на интересную, но небезупречную в техническом отношении системную программу.

Если бы исследования в сфере системного ПО играли какую-либо существенную роль, мы бы наблюдали внедрение новых операционных систем и языков программирования не реже, чем в семидесятые и восьмидесятые годы.

Вместо этого мы видим процветающую софтверную индустрию, которой, в общем-то, наплевать на исследования, и академическое сообщество, предпочитающее писать научные статьи, а не программы.

### **Linux**

Инновация? Нечто новое? Ничего подобного, очередная копия все того же старья.

Старья. Сравните разработку программ под Linux и под Microsoft Visual Studio или с использованием какого-нибудь инструментария IBM Java/Web.

Успех Linux, на самом деле, может быть одним из самых сильных аргументов в пользу моего тезиса: воодушевление, вызванное клоном тридцатилетней давности операционной системы, отражает запустение, которое исследователям оказалось нечем заполнить.

Новизна Linux не в программе, а в организации работы программистов, и едва ли это можно называть триумфом академической компьютерной науки (особенно в области разработки программного обеспечения).

### **Что представляют собой исследования системного ПО сегодня?**

Веб–кэширование, веб–серверы, файловые системы, задержки сетевых пакетов и все такое. Производительность, периферия, приложения, но только не ядра и даже не приложения пользовательского уровня.

По большей части, это масса измерений, ложная трактовка и неверное применение научного метода.

Перебор с феноменологией: изобретения сменились наблюдениями. В результате, мы видим статьи со сравнениями задержки обработки прерываний в Linux и Windows. Это может быть интересным и даже существенным вопросом, однако к исследованиям это никакого отношения не имеет.

В бестолковом стремлении к наукообразию авторы перегружают статьи измерениями: деталями о производительности и дурными графиками.

С новыми языками или системами все наоборот. Они способны менять ощущения от работы с машиной, поразить и дать почувствовать новизну. Сегодня, однако, это делают крутые веб–сайты или рост мощности процессоров или всякие хитрые маленькие устройства.

Искусство ушло.

Часть проблемы как раз и заключается в том, что искусство – не наука. Исследования в области системного ПО не могут быть исключительно наукой. Тут должно найтись место и технике, и искусству.

### **Что же в итоге получилось?**

Многое:

**PC.** железо стало дешевым, а дешевое железо стало качественным. В конечном счете, если оно не работает на PC, то от него нет проку, потому что среднестатистический компьютер – это PC.

Еще в восьмидесятые изрядная часть работ в области системного ПО вертелась вокруг новых процессорных архитектур (RISC, iAPX/432, Lisp). Теперь все уже не так. Основной источник новых проблем и (потенциально) интересных решений просто иссяк.

Опять–таки, большая часть изысканий в системной области велась с целью добиться переносимости программного обеспечения. Однако в условиях, когда аппаратная часть всегда одинакова, такой проблемы просто не возникает.

Все дело в PC как в аппаратной архитектуре. Что касается софта, то и тут наблюдается то же самое.

**Microsoft.** На эту тему уже много сказано. Microsoft – легкая мишень. Этой корпорации досталась роль козла отпущения, но источник основных проблем вовсе не она.

**Веб.** Когда в начале девяностых случился Веб, научное сообщество было удивлено не меньше, чем коммерсанты. WWW заняла главенствующее место в качестве предмета дискуссий, а что толку? Сетью теперь правит бизнес.

Вклад исследователей тут невелик, несмотря на тонны статей о кэшировании, прокси, серверной архитектуре и тому подобном.

**Стандарты.** Чтобы компьютерная система была жизнеспособной, она должна следовать огромному списку массивных и часто меняющихся стандартов: TCP/IP, HTTP, HTML, XML, CORBA, Unicode, POSIX, NFS, SMB, MIME, POP, IMAP, X и т.д. Работы невпроворот, но если не уважать стандарты, останешься на задворках. На уровне железа ту же роль играют системы команд процессоров, шины и т.д. При таком количестве "внешнего груза" для новаторства места остается немного.

Хуже того, коммерческие компании, которым эти стандарты "принадлежат", например, Microsoft и Cisco, преднамеренно усложняют следование всем стандартам, с тем чтобы окоротить конкурентов. Академические круги тут выступают в роли случайной жертвы.

**Ортодоксия.** В научных кругах используют Unix, X, Emacs и Tex. Это их мирок, зачастую единственное в компьютерном мире, к чему они когда-либо прибегали для работы. Двадцать лет назад студент сталкивался с множеством операционных систем, со всеми их плюсами и минусами. Новые сотрудники нашей лаборатории (Пайк имеет в виду Bell Labs – прим. переводчика) приносят свой мирок с собой или ожидают обнаружить его здесь. В этом есть смысл, однако были времена, когда работа в новой лаборатории означала шанс познакомиться с новыми методами работы.

Ограниченность опыта ведет к ограниченности воображения.

Ситуация с языками чуть лучше: во многих учебных заведениях в расписание включено изучение функциональных языков и т.д. Однако и тут есть своя ортодоксия – C++ и Java.

В науке принято самые большие почести оказывать тем, кто доказал, что мы были не правы. Но только не в компьютерной науке.

**Смена масштаба.** Когда внешних ограничений так много, и столько всего уже сделано, самое интересное требует крупномасштабных усилий. Для того чтобы написать современную, реалистичную систему, понадобится не один человекогод. Для большинства университетов это непосильная задача.

Временная шкала тоже велика – от замысла до окончательной версии может пройти и пять лет. Опять же, для большинства аспирантов это неподъемный труд.

В итоге, индустрия склоняется к разработке крупных, основополагающих проектов – операционных систем, инфраструктуры и т.д., а исследовательские группы помельче предпочитают заниматься менее масштабными задачами.

В результате имеем три тенденции:

1. Не создавать, а измерять (феноменология, а не новаторство).
2. Стремиться не к ширине, а к глубине (микроспециализация, а не системная работа).
3. Брать что-то уже сделанное и заниматься его изменением.

Думаю, этим и объясняется спад в количестве операционных систем.

**Unix.** Новые операционные системы в наше время зачастую оказываются очередными попытками переработки Unix. Если они обладают новаторской архитектурой (а у некоторых она есть), первое, что делают разработчики – пишут слой эмуляции Unix.

Какой смысл во всех исследованиях в сфере системного ПО, если в результате все операционные системы оказываются неотличимыми друг от друга?

В конце семидесятых – начале восьмидесятых говорили, что Unix убил исследования в области операционных систем, поскольку никто больше не хотел браться ни за что другое. Тогда я в это не верил. Сегодня с сожалением вынужден признать, что это похоже на правду (несмотря на Microsoft).

Жертва собственного успеха: мобильность ПО привела к его повсеместному распространению. Это означало, что архитектура особого значения не играла, и теперь она осталась в одиночестве.

Linux – нечто новое и интересное, но это все тот же Unix.

**Linux – Microsoft Windows для ученых.** Святая троица: Linux, gcc и Netscape. Естественно, это просто еще одна ортодоксия. Они приобрели статус икон не из-за того, чем являются, а от того, чем они НЕ являются, а именно, продукцией Microsoft.

Однако в сугубо техническом плане они отнюдь не представляются чем-то выдающимся. А Microsoft, между тем, усердно трудится, и я возьмусь утверждать, что во многих аспектах (не во всех, впрочем), продукты MS, соответствующие вышеупомянутой троице, технически более совершенны. И становятся лучше с каждым годом.

Linux может угодить в ловушку Macintosh: чопорная изоляция приводит к (практически полному) устареванию.

К тому же, исследования в области системного ПО ничем особенным улучшению троицы не способствуют.

**Стартапы.** Начинающие компании – главные соперники ученых в плане идей, финансирования, кадров и учащихся. Но исследования, финансируемые государством и бизнесом, ориентированы на скорейший возврат инвестиций.

В результате, приоритеты оказываются другими. Во-первых, исследуется только то, что может дать большие деньги (или выход на биржу) за год. Во-вторых, горизонты слишком ограничены для долгосрочной работы. В-третьих, источники финансирования и сами испытывают то же давление, так что получается порочный круг.

Само мерило ценностей ошибочно.

Стэнфорд теперь агитирует студентов идти работать в начинающие компании, потому что преуспевшие руководители этих компаний потом помогают университету деньгами. Новый президент Стэнфорда, кстати, весьма успешный ИТ-предприниматель.

**Бабушки за компьютером.** Это значит, что ИТ-отрасль ориентируется в разработке систем и сервисов на "обычных людей". Все внимание сосредоточено на приложениях и устройствах, а не на инфраструктуре и не на архитектуре, исконной вотчине исследований системного софта.

Причины, в основном, лежат в рыночной плоскости и в колоссальном количестве несовместимых устройств. На софте денег не сделаешь, только на железе, а потому давайте будем делать нишевые штуkenции, и черт с ней, с Великой Новой Идеей.

Программируемость – некогда Великая Идея вычислительной техники – оказалась на обочине. И опять изыскания в области системного ПО в безнадежном проигрыше.

## **Как быть**

Стартапы слишком зациклились на кратких сроках и практических результатах, чтобы пробовать что-то новое. Большие корпорации, в свою очередь, уперлись в существующие приоритеты, чтобы, опять-таки, братья за нечто новое. Стартапы сосут энергию из исследований. А от золотых лихорадок, как известно, остаются города-призраки. Готовьтесь скоро очутиться там.

Давайте вернемся к обдумыванию и построению систем. Узкопрофильность тут неуместна и малозначима, зато важна широта, и в этом суть системы.

Давайте попробуем поразмыслить над тем, как системы функционируют, и как они себя ведут, и не будем ограничиваться сравнением их между собой. Сосредоточимся на интерфейсах и архитектуре, а не только на программировании. Будем храбрыми и попробуем разные вещи, поэкспериментируем.

Финансовые структуры должны вкладывать деньги смелее, особенно в долгосрочные проекты. Университеты, в свою очередь, должны всегда оставлять студентам возможность работать на дальнюю перспективу.

Надо мерять успех идеями, а не только статьями и деньгами. Заставьте индустрию интересоваться вашей работой.

## Что делать

Вокруг уйма полезных, интересных и ценных замыслов. Вот маленькие примеры в доказательство. Если поле истощено, это не от недостатка возможностей.

Только один графический интерфейс был серьезно опробован, да и то его лучшие идеи были заложены еще в семидесятые. В некоторых отношениях GUI даже стал хуже. Наверняка есть другие возможности (кстати, интерфейс Linux даже до уровня Windows не дотягивает).

Множество разговоров велось вокруг компонентной архитектуры, но успешной была только одна – конвейеры Unix (Unix pipes). Однако, вполне вероятно, можно строить интерактивные и распределенные приложения по модульному принципу.

Будущее за распределенными вычислениями, однако сообщество языков программирования не слишком много внимания уделяет этим возможностям.

Сеть уже определяет, как системы представляют и используют информацию: принцип сводится к вынужденному взаимодействию, пользователь должен тянуться за информацией. Может быть, лучше гору к Магомету подтащить? Чтобы все-таки информация сама шла к пользователю?

Системное администрирование тоже остается сложной проблемой. Не самой главной, но есть масса возможностей для нововведений (и не без коммерческих перспектив).

## Выводы

Мир решает, какие компьютеры ему нужны. Исследователи системного ПО кое-как на это решение повлияли, однако в весьма незначительной степени, а теперь они просто вышли из дискуссии.

Сейчас дошло до того, что уже вообще сомнительно, чтобы кто-то взялся финансировать даже самые блестящие проекты, а если деньги и появятся, то едва ли найдутся подходящие для работы организации. Шансы на успех всегда были невысокими, теперь же они практически упали до нуля.

Сообщество – университеты, студенты, отрасль, финансовые организации – должны сменить приоритеты.

Сообщество должно принимать и исследовать неортодоксальные идеи.

Сообщество обязано отделить исследования от рыночной капитализации.

P.S. Все это похоже на истину. Ну и что? Сдаться?. Не дождетесь!

<http://www.rsci.ru/company/innov/more.html?MessageID=965>

Как указано в <http://is.ifmo.ru/guestbook/> студентом СПб ГУИТМО Виктором Лещиковым "Кризис — это великий шанс" для нас. И этим надо пользоваться, а не опустить руки!

А.А.Шалыто