

Пол Грэм

Первоклассные хакеры

КОМПЬЮТЕРРА. ОПУБЛИКОВАНО 19 АВГУСТА 2004 ГОДА

"Хакеры", вынесенные в заголовок, - это вовсе не компьютерные преступники. Грэм употребляет это слово в изначальном смысле: хакер - это высокопрофессиональный и любопытный программист, способный к нетривиальным решениям [ИР]. Некоторые идеи Грэма могут показаться спорными, но тем-то они и интересны. - Редактор

Перевод - Илья Этингоф. Оригинал статьи можно найти по адресу <http://www.paulgraham.com/gh.html>

Несколько месяцев назад я выпустил книгу*, в отзывах на которую часто встречались слова "провокационная" и "противоречивая", чтобы не сказать "идиотская".

*) Речь идёт о книге "[Хакеры и живописцы](#)". Читателям "Компьютерры-Онлайн" знакома одна из её глав - "[Языки программирования через сто лет](#)".

Я не собирался писать противоречивую книгу. Скорее наоборот, я хотел сделать книгу сжатой и конкретной. Я не хотел бы тратить время читателей на изложение уже известных им вещей, а написать лишь о том, что, как мне кажется, могло бы оказаться новым и интересным для моих читателей. Но, по-видимому, такой подход к делу грозит обернуться чуть ли не скандалом.

Эдисоны

Среди рецензентов нет противоречий по поводу того, какая из идей в книге наиболее противоречива. Очевидно, это предположение о том, что расслоение общества по уровню доходов, должно быть, вовсе не такая великая проблема, как представляется на первый взгляд.

Я не утверждаю, что значительный разрыв в благосостоянии сам по себе хорош. Однако он может быть признаком правильной организации общества. Как, например, нет ничего хорошего в мигрени, но если речь идет о сотрясении мозга, то головная боль есть первый признак выздоровления.

Различия в уровне доходов могут отражать различия в производительности, и это хорошо, поскольку, если в обществе все работают одинаково, то это, скорее всего, не потому что каждый там талантлив как Томас Эдисон, а скорее, потому что в таком обществе нет ни одного Эдисона.

В традиционных, невысокотехнологичных сообществах не обнаруживается заметных различий в производительности его членов. Возьмем к примеру какое-нибудь первобытное племя охотников-собирателей. Насколько продуктивнее окажется самый лучший собиратель? Ну, скажем, раза в два. А вот с появлением совершенных инструментов, вроде компьютеров, оказывается, что возможности отдельных людей возрастают неимоверно!

Все это известно давно, в 1974 году Фред Брукс опубликовал результаты исследования на данную тему, проведенного еще в 1968 году. Сейчас мне кажется, что он недооценил различия, существующие среди программистов. Он оценивал продуктивность программистов в количестве строк кода, которые те были способны произвести, или во времени, потраченном на решение задачи - лучшие программисты могли решить задачу в десять раз быстрее коллег. Но что, если задача не поставлена вовсе? В программировании, как и во многих других областях человеческой деятельности, проблема заключается не столько в решении задач, а в их формулировке. Трудно точно оценить развитость воображения, но на практике именно эта оценка производительности программиста оказывается более точной.

В любой области деятельности продуктивность людей различается, однако в наше время существуют несколько родов занятий, в которых несоответствия могут быть разительны. В программировании такие количественные различия переходят в качественные, хотя, по-видимому, не только в нем. Где бы то ни было, технология наращивает разброс в уровне производительности, а в программировании - в первую очередь. Таким образом, мне кажется, что постепенно продуктивность людей, занятых в других

областях, будет разниться все больше. И в конце концов, успех компаний и даже стран будет все больше зависеть от того, как те справляются с этими разбросами в продуктивности.

По мере того как продуктивность различается все больше благодаря современным технологиям, вклад наиболее успешных сотрудников окажется не только непропорционально велик, но и продолжит со временем расти. Достигнув состояния, когда 90% продукции обеспечивается 1% населения, сообщество становится уязвимо, если какие-либо неожиданности (вроде набега викингов) снижают продуктивность наиболее полезных членов сообщества.

Если мы желаем выжать максимум пользы из самых продуктивных людей, нам следует повнимательнее к ним присмотреться. Что движет ими в их деятельности? В чем они нуждаются для выполнения работы? Как распознать самых продуктивных людей? Как привлечь их к работе? И, наконец, как самому стать одним из продуктивнейших людей сообщества?

Не только деньги

Когда я пытаюсь понять, что же общего у того небольшого числа первоклассных хакеров, с которыми я знаком, то первой схожей их характеристикой оказывается любовь к занятию программированием. Рядовые программисты работают, в основном, ради денег. Выдающиеся хакеры считают свою работу одним из величайших удовольствий, а то, что кто-то еще и платит им за работу, почитают за удачу.

Иногда говорят, что первоклассные хакеры безразличны к деньгам, но это не вполне верно. Правильно тут то, что приоритетом у выдающихся программистов является привлекательность задачи. С другой стороны, имея достаточно денег, можно было бы работать над любой, по собственному выбору, интересной задачей, а такая перспектива привлекает первоклассных хакеров. Однако пока приходится работать за деньги, хакеров более занимает интересность задачи, нежели размер зарплаты.

И это самый важный, с экономической точки зрения, факт, поскольку в таком случае нет необходимости платить первоклассному хакеру столько, сколько на самом деле стоит его работа. Выдающийся программист может быть в десять или сто раз продуктивнее, чем его рядовой коллега, но он будет счастлив зарабатывать хотя бы раза в три больше. Далее я поясню, что отчасти этот феномен связан с тем, что первоклассные хакеры просто не знают, насколько хорошо они выполняют свою работу. Но это все же еще и потому, что деньги - не главная их цель.

Так что же нужно хакерам? Как и любой мастер своего дела, они любят качественные инструменты. А в программировании это особенно важно. Хорошие программисты просто не выносят некачественных инструментов. Они, скорее, вовсе откажутся участвовать в работе над неправильно организованным проектом.

Когда я работал в одном стартапе, в офисе на доске объявлений среди всего прочего висела рекламная листовка фирмы IBM, на которой был изображен компьютер серии AS400, а надпись поверх гласила что-то вроде: "Презираем хакерами"*. Дело в том, что выбирая инструменты для работы над тем или иным проектом, вы принимаете не только и не столько техническое решение. Вы также выбираете еще и сообщество, которое сформируется для работы над проектом, а это, как представляется, еще важнее.

*) Честно говоря, я должен отметить, что IBM производит вполне приличные компьютеры. Я написал эту статью на ноутбуке IBM.

Представьте, что ваша компания решила разработать какую-то новую программу. Казалось бы, в наше время для новых разработок следует использовать язык Java. Но выбрав язык, мы автоматически выбираем и сообщество разработчиков. Программисты на Java, которых ваша компания сможет нанять для работы над проектом, окажутся, в среднем, хуже подготовлены, чем разработчики на языке Python*. А качество инженеров, работающих над проектом, вероятно, важнее используемого языка программирования. Между прочим, если говорить честно, тот факт, что первоклассные хакеры предпочитают язык Python языку Java, кроме всего прочего, еще и характеризует относительные качества этих инструментов программирования.

*) Когда Google приглашает на работу программистов на Java, они еще и требуют опыта программирования на языке Python.

Люди бизнеса предпочитают наиболее популярные языки, потому что рассматривают их как стандарты. Руководители компании едва ли пойдут по пути фирмы Sony, чей формат видеозаписи Betamax в свое время проиграл в конкурентной борьбе формату VHS, который, в отличие от Betamax, лицензировался многим независимым производителям. Но дело в том, что языки программирования - это не просто стандарты. Когда вам потребуется переслать биты данных из точки А в точку Б, то вы, вероятно, используете самый распространенный в наши дни способ, протокол TCP/IP. А вот с языком программирования ситуация сложнее, поскольку он не только инструмент, но еще и среда для творчества.

Я где-то читал, что язык Java не так давно стал самым популярным инструментом программирования, опередив по популярности "Кобол". Для языка, как стандарта, это, безусловно, большой успех. Но в качестве среды для творчества языку Java еще есть, к чему стремиться. Из всех знакомых мне выдающихся программистов я знаю только одного человека, который добровольно выбрал бы язык Java для работы. Да и то, в связи с тем, что этот человек работает в фирме Sun Microsystems.

Выдающиеся хакеры, как правило, стремятся использовать ПО с открытым кодом. Не потому, что качество таких программ выше, а потому что исходные тексты дают разработчикам практически полный контроль над процессом разработки, отладки и эксплуатации программ. А хорошие программисты желают иметь максимально полное управление в разрабатываемой системе. Ведь именно полный контроль делает простых разработчиков выдающимися хакерами, давая им возможность самостоятельно одолеть ту или иную проблему в системе.

Несколько лет назад один мой приятель, инвестор, рассказал мне о планах фирмы, с которой был связан. Их перспективы выглядели вполне радужно. Спустя некоторое время, когда я снова встретил этого человека, оказалось, что руководство компании решило разрабатывать свой продукт для Windows NT, и что они наняли опытного NT разработчика в качестве ведущего инженера проекта. Тогда я подумал, что эта компания обречена. Во-первых, даже опытейший разработчик для Windows NT не может быть первоклассным хакером, так как для того, чтобы набраться опыта программирования для Windows NT, необходимо добровольно и неоднократно использовать эту платформу, а я не могу себе представить выдающегося хакера за этим занятием. И во-вторых, даже если окажется, что программист для Windows NT действительно хорош, то ему окажется непросто подобрать команду первоклассных хакеров для участия в программировании для Windows NT*.

*) Они и впрямь были обречены. Та компания закрылась через несколько месяцев.

Последний оплот

Говоря об инструментах первоклассного хакера, следующим по важности после программ, по-видимому, оказывается его офис. В больших компаниях место сотрудника в офисе подчеркивает его положение в служебной иерархии. Но для хакеров офис - это еще и место, где они работают над решением задач. Заставлять первоклассных хакеров работать в шумной и отвлекающей обстановке - все равно, что красить автомобили в запыленном помещении.

Герою комиксов Дилберту было что сказать о кубиклах*. Все мои знакомые хакеры не любят их. Одна только перспектива в любой момент оказаться оторванным от решения сложной проблемы каким-нибудь проходящим мимо сотрудником не дает хакерам начать сложную работу в кубикле. В результате, им остается работать дома или приходить в офис до или после рабочего дня или в выходные, когда в офисе никого нет. Удивительно, что крупные компании не замечают этой проблемы. Ведь офис предназначался для комфортной работы, а вовсе не для создания дополнительных помех сотрудникам.

*) Кубиклами (cubicle) называются секции в офисе, разделённые невысокими перегородками на крохотные рабочие места на одного сотрудника. [Дилберт](#) - герой весьма едких комиксов на темы офисной жизни корпоративной Америки. Примечание переводчика.

Большие компании, такие как Cisco Systems, гордятся тем, что каждый их сотрудник, включая генерального директора, имеет собственный кубикл. Увы, они не так прогрессивны, как им представляется: по-видимому, они до сих пор рассматривают кубиклы как атрибут социального положения. Заметим, между прочим, что Cisco известна небольшими объемами собственных разработок. Вместо этого, Cisco приобретает небольшие молодые компании, которые разработали ту или иную технологию. По-видимому, эти компании как раз и есть тот тихий омут, в котором водятся первоклассные хакеры.

Существует, однако, одна большая компания, в которой понимают, что нужно хакерам - это Microsoft. Однажды я встретил объявление о приеме на работу в Microsoft. На листовке была изображена дверь, а надпись гласила: работайте у нас, и мы дадим вам возможность работать там, где вы сможете выполнить свою работу. А знаете, что Microsoft отличается от других больших компаний тем, что им удается создавать значительную часть ПО собственными силами? И хотя результат мог бы быть и лучше, но все-таки, он достаточно хорош.

Если компании хотят от хакеров максимально производительной работы, то им следует посмотреть, как хакеры работают у себя дома. Ведь дома человек организует свое рабочее место наилучшим для себя, максимально эффективным образом. Дома хакеры избегают открытых, шумных мест; они работают в комнатах за закрытыми дверями, в уютных, расположенных по соседству помещениях. Когда им бывает нужно обсудить что-либо друг с другом, они отправляются куда-нибудь погулять, вместо того чтобы встречаться в стеклянных "аквариумах", нарезанных на манер мест для парковки автомобиля. Дома хакеры устраивают себе место для отдыха, где они могут вздремнуть, а не сидеть на рабочем месте в состоянии комы, делая вид, что работают. И над ними не вьется команда уборщиц с пылесосами каждый вечер, отвлекая от работы в самое продуктивное время. Дома у них нет совещаний, корпоративных мероприятий и упражнений по укреплению навыков работы в команде. А если вы посмотрите, что хакеры делают на своем домашнем компьютере, то обнаружите лишь подтверждение того, о чем я говорил ранее в главе об инструментах. Хакеры могут использовать Java и Windows на работе, но дома, в ситуации свободного выбора, вы, вероятнее всего, встретите на их компьютерах Perl и Linux.

Тогда, возможно, та статистика о популярности языка Java покажется вам не вполне корректной. Если мы хотим выяснить, какие же из инструментов и в самом деле самые качественные, нам стоит посмотреть, что же выбирают хакеры, когда у них есть выбор, то есть в их собственных проектах. Тогда может оказаться, что операционные системы с открытым кодом уже захватили большую часть рынка, а самый распространенный язык программирования - это, вероятно, Perl.

Правильная работа

Помимо качественных инструментов, хакеры хотят работать над интересными задачами. Так что же делает проект интересным для хакеров? Разумеется, высокотехнологичные разработки, вроде самолетов-невидимок или моделирования спецэффектов, были бы им интересны. И хотя, в принципе, любая задача может оказаться привлекательной, если она требует решения новых технических проблем, не так просто определить, какими проблемами заинтересуются хакеры. Ведь задача может стать для них интересной, лишь если в какой-то момент будет найдено некое новое решение. До того как компания ИТА (которая разрабатывала ПО под эгидой фирмы Orgitz) занялась разработкой ПО для резервирования билетов на авиарейсы, программисты считали, что нет на свете более скучной задачи. Однако ИТА переформулировала проблему, и в менее однозначной редакции та стала интереснее.

Мне кажется, что-то подобное произошло в компании Google. В те времена, когда была основана Google, разработчикам так называемых "порталов" задача поиска казалась несущественной и скучной. Но хакеры из Google не считали проблему поиска скучной, и потому у них все получилось так хорошо.

Это одна из областей, где управленцы могут на что-то повлиять. Подобно родителю, который держит с ребенком пари на то, что тот не сможет убрать свою комнату за десять минут, менеджеры иногда могут переформулировать проблему в более интересном ключе. Кажется, Стив Джобс в этом особенно преуспел - в части поддержания высокой планки. В те времена, когда фирма Apple начинала свой

бизнес, рынок уже был насыщен недорогими персоналками. Так вот, задача, в формулировке Джобса, была - сделать симпатичную персоналку. И это, вероятно, подействовало на инженеров Apple не хуже, чем морковка или палка на осла.

И у них получилось! Когда только появился первый "Мак", не нужно было даже включать его, чтобы понять, что это отличная разработка. Это было видно по корпусу. Несколько недель назад я прогуливался по Кембриджу, и в чьем-то мусорном ведре я увидел то, что оказалось сумкой для "Мака". Я открыл ее и увидел Mac SE. Я отнес его домой, включил, и он загрузился. Улыбка "Макинтоша", а затем и его файловый менеджер Finder. Ох, это было так просто. Просто, как... Google.

Высококласным хакерам нравится работать с людьми, которые устанавливают высокую планку. Однако одной педантичности недостаточно. Необходимо также следовать правильным техническим решениям, что означает, что менеджер сам тоже должен быть высококласным хакером. Мне встречались статьи о том, как управлять коллективом программистов. На самом деле должно быть две статьи по этому поводу - для программистов и непрограммистов. Так вот, последняя статья может содержать только три слова: ничего не выйдет.

Работа менеджера команды хороших программистов заключается не столько в каждодневном управлении: хорошие программисты, как правило, вовсе не требуют руководства, так как сами успешно управляют своими проектами. А вот непрограммисту трудно понять, насколько хорошо работает тот или иной его подчиненный. Похожая ситуация возникла с американскими автомобилями - я называю ее парадоксом дизайна. Допустим, вы задались целью создать какой-нибудь замечательно красивый продукт и хотели бы для начала пригласить талантливого дизайнера. Так вот, если допустить, что у вас нет художественного вкуса, то как определить, какой дизайнер действительно талантлив? Судить по его прежним работам вам не дано по определению. Ориентироваться на награды или прежние места работы бессмысленно, поскольку в дизайне, как и во многих других областях, эти вещи, в первую очередь, зависят от моды и частных договоренностей и лишь потом - от художественных качеств. Из этого парадокса нет выхода: вам не удастся создать прекрасную вещь, не понимая, что такое красота и гармония. Так вот, американские автомобили не красивы, потому что американскими автомобильными компаниями управляют люди с плохим вкусом.

Многие люди в Америке считают вкус чем-то иллюзорным или даже пошлым, но это не так. Менеджер, от которого зависит дизайн продукта, должен стать самым капризным и требовательным пользователем. А если повезет, и у нашего руководителя окажется хороший вкус, как он обнаружился у Стива Джобса, тогда наша задача может привлечь первоклассных хакеров.

Дурацкая работа

Совершенно очевидно, какие задачи будут неинтересны нашим хакерам: это те, в которых, вместо решения нескольких значительных и ясно определенных проблем, инженер сталкивается с кучей небольших и неестественных. Один тяжелый случай такого рода - создание интерфейса к программе полной ошибок. Или, например, адаптация какого-нибудь ПО к сложной и неправильно определенной модели отдельно взятого пользователя. Для хакера, такие задачи - медленная и мучительная смерть.

Отличительная черта такого рода задач в том, что они ничему не учат человека, занимающегося их решением. Например, создание компилятора языка интересно хотя бы потому, что такая работа позволяет разобраться в том, как устроены компиляторы. А вот написание интерфейса к неправильно работающей программе не научит ничему, поскольку ошибки несистематичны. Так что хакеры избегают дурацкой работы не только из чистоплотности и снобизма. В большей степени это вопрос самосохранения, поскольку борьба с подобными искусственными проблемами ведет к деградации инженера. Первокласные хакеры избегают их из тех же соображений, что и манекенщицы избегают чизбургеров.

(Кстати, мне кажется, что когда люди говорят о "смысле жизни", они имеют в виду именно это. Само это выражение звучит странно; жизнь - это ведь не высказывание, которое может иметь смысл. Но жизнь может определяться качеством, которое очень похоже на "смысл". В работе над компилятором

инженер решает много отдельных задач, но все они попадают в ту или иную категорию типовых, что похоже на выделение полезного сигнала. Соответственно, решение несистематических, нетиповых задач больше напоминает шум.)

Разумеется, некоторые задачи, по определению, относятся к категории дурацких. При благоприятной конъюнктуре, работа над некоторыми такими задачами может быть весьма прибыльна. Компания, которой удастся привлечь к работе над дурацкими задачами первоклассных хакеров, может оказаться очень успешной. Но как это сделать?

Одно из мест, где такое возможно - это стартапы. В нашей фирме системным администратором работал Роберт Моррис*. Это было все равно, что пригласить Rolling Stones на празднование Бар-Мицвы. Нанять специалиста такого высокого класса на такую неинтересную работу совершенно невозможно. Но ради развития своей собственной компании люди готовы на любую рутину**.

*) Роберт Моррис - профессор в Массачусетском технологическом институте (MIT). В 1988 году написал так называемый "червь Морриса", который считается первым в истории компьютерным червём, распространившимся по интернету.

**) Эйнштейн занимался проектированием нового типа холодильника (и даже получил патент).

Большие компании решают эту проблему, разделяясь на части. Они привлекают толковых людей, приглашая их в отдельный исследовательский отдел, куда не добираются конечные пользователи со своими неинтересными проблемами*. При такой организации исследовательский отдел выполняет роль генератора идей, которые, возможно, смогут использовать остальные отделы компании.

*) Не так просто сказать, чем именно занимаются исследователи в компьютерной области. Для начала, я бы сказал, что они пишут программы, в которых нет очевидной потребности. Не думаю, что эта статья направит первоклассных хакеров на поиски работы в исследовательских отделах компаний. Едва ли им захочется потратить три часа на встречу с менеджером по продуктам по вопросу об интеграции корейской версии программы Word билда 13.27 с говорящей скрепкой.

Однако это не единственно возможный подход. Методика программирования "снизу вверх" предлагает иной способ специализации внутри компании: можно привлечь толковых инженеров к производству инструментария для создания продукта компании. В таком случае 99% кода конечного продукта окажется написано профессионалами, но вместе с тем, они, как и в случае с выделенным исследовательским отделом, будут почти полностью изолированы от конечных пользователей. Производители инструментария будут иметь конечных пользователей, но ими окажутся собственные разработчики компании*.

*) Что-то похожее долгое время происходило в области строительства. Если Ваш дом был построен несколько сот лет назад, то тогдашние строители делали все сами. Но постепенно строители переходили на сборку дома из готовых компонентов, изготовленных третьими фирмами. Как и в случае с макетированием книг на персоналке, это дало людям свободу для эксперимента, иногда разрушительного, но, в целом, подобная специализация, очевидно, эффективнее.

Возможно, если бы в Microsoft использовали подобный метод разработки, то их программы не содержали бы такого количества ошибок, поскольку наиболее критичные части кода (вроде управления памятью) были бы написаны наиболее толковыми инженерами. Вместо того чтобы писать программу Word полностью на языке C, можно было бы создать инструмент - специальную высокоуровневую среду разработки, и собрать Word из кусочков, подобно конструктору Lego.

Кучкование

Помимо интересной работы, первоклассные хакеры любят работать вместе. Хорошие инженеры сами собой собираются в группы, порой весьма значительные, как это случилось в компании Хегох*. По этой причине у Вас не получилось бы привлечь тем больше первоклассных хакеров, чем лучше условия Вы бы создали для них. Эта тенденция собираться в колонии, меняет линейную зависимость количества хакеров от качества условий на квадратичную. В конечном счете, возникает ситуация "все или ничего".

В любой момент времени на рынке существуют десятки мест, где первоклассные хакеры хотели бы работать. Если ваша компания не в их числе, то у Вас не просто будет меньше хакеров, их Вам не достанется вовсе.

*) Графический интерфейс пользователя в том виде, в котором мы его знаем (окна, иконки, меню), появился в исследовательском центре Xerox PARC в семидесятые годы.

Однако одно наличие в команде первоклассных хакеров не гарантирует успех компании. Например, у Google и ITA дела идут отлично, в этих компаниях в данный момент собрались первоклассные хакеры, но вот у компаний Thinking Machines и Xerox все не так здорово. Некоторое время хорошо работала компания Sun, но если выбрана неправильная бизнес-модель, то даже лучшие хакеры не смогут спасти компанию.

Тем не менее, я считаю, что при прочих равных условиях компания, которой удалось привлечь первоклассных хакеров, получает громадное конкурентное преимущество. С этим согласятся не все. Когда мы обсуждали перспективы молодых компаний в девяностых годах прошлого века, некоторые говорили, что программистские фирмы никогда не добьются успеха на рынке, даже если будут разрабатывать отличное ПО. Выиграют, как считали эти люди, те, кто раскрутит торговые марки, организует мощные каналы сбыта и заключит выгодные контракты.

Они действительно верят в такую модель и, мне кажется, я знаю почему. Возможно, множество инвесторов надеются найти, по крайней мере, подспудно, новую фирму Microsoft. Разумеется, если ваш кумир - Microsoft, вам не стоит инвестировать в компании, которые создают качественное ПО. Но эти инвесторы заблуждаются: молодой компании едва ли удастся пойти по пути Microsoft, если только какая-нибудь другая компания не готова в нужный момент лопухнуться и стать, таким образом, очередной IBM.

Было бы ошибкой ориентироваться на Microsoft, потому что их успех базируется на счастливом стечении обстоятельств. В этом смысле история Microsoft не характерна. Если не учитывать присутствие Microsoft на рынке, то окажется, что по-настоящему хорошие продукты превалируют на рынке. Потому инвесторам следует искать новые Apple или Google.

Думаю, что Билл Гейтс все это знает. И беспокоит его не то, что Google не принадлежит Microsoft, а то, что в Google работают первоклассные хакеры*.

*) Google гораздо опаснее для Microsoft, чем Netscape. Возможно, это даже самая опасная компания из всех существовавших до сих пор. Не в последнюю очередь, потому что Google полна решимости бороться. На их страничке отдела кадров написано, что одна из их главных ценностей - не быть монстром. Для компании, занимающейся продажей соевого масла или горнопроходческого оборудования подобное заявление выглядело бы эксцентрично. Но мне кажется, что всем нам, людям из ИТ-отрасли, совершенно очевидно, что означает это заявление.

Как распознать первоклассного хакера

Так кто же из множества людей работающих в ИТ-области, относится к настоящим хакерам? Как распознать первоклассного хакера при встрече? Оказывается, не так просто. Даже сами хакеры не всегда могут сказать. Сейчас я совершенно уверен, что мой друг Тревор Блэквел - первоклассный хакер. Вы могли прочитать на [Slashdot](#) заметку о том, как Тревор [разработал и сделал](#) свой собственный Segway. Замечательная деталь этого проекта кроется в том, что все ПО для самоката Тревор написал за один день (на языке Python, кстати). Для Тревора это нормально. А когда я первый раз встретил его, мне показалось, что он полный идиот. Я увидел его в офисе Роберта Морриса бормочущим то о том, то о другом, и я помню, как стоял позади Тревора и жестами пытался намекнуть Роберту, что пора бы выгнать его. Потом Роберт рассказал мне, что и он поначалу ошибся насчет Тревора. Ко времени их первой встречи Тревор как раз начинал проектировать новую схему, причем ход работы он записывал в блокноте, который всегда носил с собой. К тому же, он только что вернулся из Канады, прихватив с собой сильный канадский акцент и модную причёску в стиле "кефаль".

Задача распознавания хакера усложняется еще и тем, что, несмотря на репутацию ребят замкнутых и нелюдимых, хакеры иногда стараются выглядеть умно. Когда я учился в университете, я время от времени проходил мимо лаборатории искусственного интеллекта в MIT. Там было неуютно. Все говорили очень быстро и непонятно. Через некоторое время, я понял в чем тут дело. Нет необходимости соображать также быстро, достаточно просто произносить слов больше, чем необходимо для изложения сути дела.

При таком низком отношении сигнал/шум трудно понять, кто есть кто на самом деле. Да я и сейчас этого не знаю. Трудно судить и по резюме. Похоже, что единственный способ оценить возможности инженера - поработать вместе с ним над какой-нибудь задачей.

Вот почему высокотехнологические оазисы возникают вокруг университетов. Активная составляющая там не столько профессура, сколько студенты. Небольшие новые компании образуются в кампусах благодаря тому, что университеты привлекают молодых и талантливых людей и дают им возможность работать сообща. Наиболее талантливые находят друг друга и объединяются для работы над своими собственными проектами.

Поскольку нам не удастся узнать первоклассного хакера иным способом, кроме как поработав вместе с ним, сами хакеры не могут оценить, насколько хорошо они работают. До некоторой степени дело обстоит так и в прочих областях человеческой деятельности. Я заметил, что выдающиеся люди не особенно высоко оценивают свои собственные возможности, но не перестают поражаться некомпетентности окружающих. Люди, которых я встречал, и которые безупречно выполняли свою работу, как правило, не считали, что работают хорошо. Они, как правило, считали себя ленивыми и несообразительными, жаловались на то, что их голова работает как следует только один день из десяти, и что их увольнение - лишь вопрос времени.

Хакерам особенно трудно определить свой уровень из-за сложностей сравнения их работы. В большинстве областей с этим все просто. В соревнованиях по бегу на 100 метров через 10 секунд Вы узнаете, кто лучший. Даже в математике, как кажется, существуют согласованные оценки сложности тех или иных проблем и качества тех или иных решений. Но программирование больше похоже на литературу. Кто может сказать, какой из двух романов лучший? Уж наверняка не авторы.

В случае с хакерами, по крайней мере, другие хакеры могут оценить уровень работы, поскольку, в отличие от писателей, хакеры сотрудничают друг с другом в работе над задачей. Но хакер не может оценить уровень своей работы, так что если мы спросим первоклассного хакера, насколько хороши результаты его деятельности, то он, скорее всего, ответит, что не знает. И не потому что скромничает, он действительно не знает.

И никто из нас не знает, за исключением людей, когда-либо работавших с нами. Все это ставит нас в странную ситуацию - мы не знаем своих героев. Хакеры, получившие известность, оказались знаменитыми случайно, вследствие неожиданного пиара. Иногда мне требуется привести пример первоклассного хакера, и я совершенно теряюсь. Первое, что приходит в голову - это мои друзья, но использовать их имена как-то неудобно. Может, мне стоит назвать имена Ричарда Столлмена или Линуса Торвальдса или Алана Кея или что-то в этом духе. Но, строго говоря, я не знаю этих людей, и мне трудно сказать что-либо об их профессионализме.

Если и существует свой Майкл Джордан в программировании, то никто, включая его самого, об этом не знает.

Как стать первоклассным хакером

И наконец, вопрос, которым, должно быть, задаются все программисты: как стать первоклассным хакером? Не знаю, возможно ли воспитать самого себя в таком ключе. Очевидно, что раз человек может сам загубить свою карьеру, то наверно, он может и построить ее, став первоклассным хакером.

По-видимому, самое главное для хакера - это интересная работа. Вспоминая моих знакомых хакеров, я осознаю, что еще одно из их общих качеств - чрезвычайная неохота выполнять работу, которая им не нравится. Не знаю, что это - причина или следствие. Возможно и то и другое.

Чтобы сделать что-либо хорошо, обязательно любить эту работу. До тех пор пока Вам удастся сохранять любовь к программированию, результаты будут высокими. Старайтесь сохранить свой интерес к программированию в том же состоянии, каким он был в 14 лет. Если Вам кажется, что Ваша нынешняя работа не развивает Вас профессионально, то это, вероятнее всего, именно так и есть.

Лучшие хакеры, как правило, умные люди, но ведь это справедливо и для других профессий. А существует ли какое-либо уникальное качество, характерное только для хакеров? Я спрашивал об этом друзей, и первое, что они называли, было любопытство. Я всегда предполагал, что умные люди любопытны; это любопытство - прямое следствие их знаний. Но судя по всему, хакеры особенно любопытны, в первую очередь, насчет того, как все устроено. И это объяснимо, поскольку программы, по сути, гигантские описания того, как работают те или иные устройства.

Несколько из моих респондентов отметили свойство хакеров концентрироваться на решении задачи. Как сказал один из моих друзей, они умеют изолироваться от внешнего мира. Мне кажется, это так. Я слышал о нескольких хакерах, которые не могли работать уже после пол-кружки пива. Так что, возможно, программирование требует особой способности фокусироваться на одной проблеме. Вероятно, первоклассные хакеры могут поддерживать в собственной голове несколько контекстов так, что, обдумывая строчку кода, они представляют себе работу всей программы. Джон Макфи писал, что успехом в баскетбольной карьере Билл Брэдли обязан, прежде всего, экстраординарному периферийному зрению. Глаз нормального человека имеет угол обзора в 47 градусов по вертикали. У Билла Брэдли было 70: он мог видеть корзину когда смотрел в пол. Может быть, первоклассные хакеры имеют аналогичные врожденные способности. (Тут я слегка лукавлю, используя упрощенный язык, который трудно оспорить в суде.)

Все это объясняет проблему с кубиклами. Может быть, тем людям, которые организуют работу офиса, не требуется особая концентрация для выполнения своих функций, и потому они не понимают, что для хакера работа в кубикле едва ли производительна.

Еще одно отличие хакеров от просто умных людей, которое я подметил, кроется в том, что хакеры менее политкорректны. Только находясь в своей среде, конечно, когда они чувствуют, что могут свободно излагать свою точку зрения, не рискуя быть побитыми камнями окружающими. И мне кажется, я понимаю, почему политическая некорректность может оказаться полезным качеством программиста. Дело в том, что программы могут иметь очень сложную структуру, а в руках хорошего хакера они очень динамичны. В такой ситуации полезно иметь привычку проверять то, чему все привыкли верить.

Можно ли воспитать в себе все эти качества? Не знаю. Во всяком случае, можете попытаться не подавлять их в себе. Итак, мой единственный совет: если это вообще возможно, сознательно стать первоклассным хакером, заключите сами с собой соглашение: Вы более не обязаны братья за дурацкую работу (при условии, конечно, что ваши родные от этого не пострадают), а взамен обещайте себе всегда выполнять свою работу качественно. Все первоклассные хакеры, которых я знаю, похоже, заключили с собой подобное соглашение. Хотя, возможно, у них просто не было выбора.

Автор хотел бы поблагодарить Джессику Ливингстон, Роберта Морриса и Сару Харлин за вычитывание ранних версий этой статьи