

работа, деньги, интернет

крис касперски ака мыщъх, no-email

чужой опыт редко бывает полезным, особенно если он касается таких индивидуальных аспектов как выбор операционной системы или... поиск работы. универсальных рецептов нет — люди не взаимозаменяемы! у каждого из нас свои сильные и слабые стороны, свои цели и свои средства их достижения. описывая сложную систему взаимоотношений нанимателей с работодателями, автор ни за что не агитирует и ни к чему не призывает. просто... показывает один из многих путей выживания в этом мире.

введение

Поиск работы это всегда компромисс между размером заработка и величиной интереса к ней. Только не путайте деньги с возможностями, которые они предоставляют. Деление жизни на "время работы" и "время отдыха" делает человека и бедным, и глубоко несчастным одновременно. Чтобы впоследствии не жалеть о бесцельно прожитых годах, работать следует из удовольствия, полностью (или частично) отождествляя собственное "Я" с тем, что это "я" делает, воспринимая деньги как нечто само собой разумеющееся.

А деньги в IT-индустрии вращаются воистину огромные и когда другие говорят о растущей безработице, у нас наблюдается устойчивый кадровый кризис. Куда ни плюнь — везде полным-полно незанятых мест, а специалистов на всех не хватает. Зарабатывать > \$1000 для программиста средней руки — вполне реально, правда и вкалывать придется так, что... окружающий мир перестанет существовать и в сознании останется только клавиатура и монитор. Естественно, москвичи и жители других крупных городов находятся в более выигрышном положении, чем периферия, а в штатах даже юниоры получают на порядок (двоичный ;-) большие суммы.

Кто-то, соблазнившись открывающимися перспективами (причем, не только финансовыми), перебирается из родной глубинки в столичные округа или даже в даже пересекает океан, надеясь встретить на той стороне вечно голубое небо, пальму и белый пароход (не Титаник). Что ж, каждый выбирает свой путь сам, но существует тысяча мотивов, чтобы никуда не уезжать, а жить там, где ты родился. Интернет дает уникальную возможность удаленной работы из любой, даже самой глухой и отдаленной деревни. А что? Свежий воздух, экологически чистые продукты, частный дом, не идущий ни в какое сравнение с тесными городскими квартирами... Впрочем, дело даже не в деревне, а в образе жизни. Удаленная работа пользуются популярностью и в Москве, избавляя творчески настроенных людей от необходимости каждое утро таскать свое тело в офис, и расширяет границы охотничьих "владений" до размеров земли (а в скором будущем — и орбитальных станций).

за и против

Ассоциация с "охотой" возникла не случайно. В отличии от "firm job", удаленная работа зачастую вообще не имеет никакого юридического статуса, превращая нас в хищников, бегающих по просторам Интернета в поисках добычи. А хищника, как известно, ноги кормят. Сейчас, по прошествии десяти лет, проведенных на "вольных хлебах", я оглядываюсь назад и думаю — какое же это счастье иметь стабильную работу, на которую можно прийти и... тупо делать то, что тебе скажут (или... не делать, если не скажут). Где думают за тебя, решают за тебя, и креатив не является основой выживания.

Но нет! Это все-таки не для меня! Есть люди, которые в любой ситуации привыкли полагаться _только_ на самих себя. И я — один из них. Выполнять чужое распоряжение, заранее зная, что оно неправильное (неоптимальное) — это же какая сила воли (или пофигизм) требуется! Не говоря уже о том, чтобы являться в офис строго по расписанию, вступая в сложную связь с коллегами, начальством и политическими решениями руководства. Только крупные фирмы могут позволить отдельным (наиболее талантливым) работникам реализовать свой творческий потенциал, не требуя от них моментального "выхлопа". Во всех остальных случаях — работа становится синонимом слова "несвобода", превращая человека в унифицированный зубец большого колеса, которое в свою очередь является лишь малой частью огромного механизма.

Это вовсе не призыв к "революции". Это просто такая экономическая модель. Популярная, но не единственная. Возможность выбора уже есть, только далеко не каждый стремится выбраться из клетки на свободу, поскольку, свобода требует ответственности и умения постоять за себя и свое право оставаться свободным и впереть. Вместо регулярной зарплаты — нестабильные доходы из серии "то густо, то пусто". Вместо поступательного продвижения по карьерной лестницы — хаотичное броуновское движение, где понятие "статуса" отсутствует как класс.

Удаленная работа — это не райские куши и агитировать за нее могут только те, кто зарабатывает на жизнь агитацией. Прежде, чем покидать клетку и отправляться в первобытные джунгли, следует, по крайней мере, совершить несколько предварительных вылазок, убедившись, что, во-первых, офисная жизнь вас действительно тяготит и, во-вторых, вы уже откопали альтернативный источник доходов, запаса которого хватит на достаточно продолжительное время, позволяющее спокойно осваиваться в новом мире, не хватаясь за первый попавшийся заказ (который, зачастую, является и последним, ведь "увязнуть" в единственном проекте и в самом деле очень легко, особенно если одним из условий является его развитие и поддержка).

виды удаленной работы

"Свободный программист" традиционно ассоциируется с клепателем шаровар. Профессия в наше время крайне распространенная, а потому не сильно привлекательная с коммерческой точки зрения. Рынок насыщен до предела и нащупать еще незанятую нишу очень трудно, особенно если речь идет о продуктах массового потребления. С одной стороны поджимают компании-гиганты, с другой — открытые проекты, созданные на голом энтузиазме и распространяемые на бесплатной основе.

Но даже имея на руках конкурентоспособный продукт, львиную часть времени придется тратить на техническую поддержку, документирование, разборки с (не)платежами, устранение конфликтов с новым (старым) железом и ОС. На творчество остаются лишь жалкие урывки и вместо удовольствия работа станет приносить одно сплошное разочарование. К тому же всегда существует риск, что проект провалится, а создание (и раскрутка) нового за один день не делается. Сразу же возникает вопрос — как дальше жить? А без еды долго жить не получится, увы! Так что необходимо заблаговременно позаботиться о создании НЗ, в который входят не только деньги, но еще и интеллектуальные ценности (типа программ, библиотек и прочих ингредиентов, готовых к немедленной продаже).

А вот другой путь — вместо того, чтобы создавать весь продукт целиком, лучше сосредоточиться на том, что вам действительно интересно и работать в одном или нескольких узких направлениях — оптимизация, разработка защитных механизмов, дизассемблирование, поиск ошибок и т. д. Большинство мелких компаний (и программистских коллективов) не могут позволить себе держать в штате кучу узких специалистов, большую часть времени которые все равно окажутся незадействованными. Можно, конечно, поручить защиту программы своему самому "продвинутому" сотруднику или воспользоваться одним из широко разрекламированных протекторов (вот только каким именно?), но не лучше ли обратиться к независимому специалисту, "окучивающему" исключительно эту тему и знающему о ней если не все, то гораздо больше самого "продвинутого" из широкопрофильных программистов.

Особенно это актуально для представителей редких (или вымирающих) специальностей. Представим себе человека, уверенно программирующего на ассемблере в реальном режиме и знающего все закоулки MS-DOS, но испытывающего устойчиво отвращение к Windows. Найти работу по своему профилю в родном городе и примыкающим к нему окрестностям крайне маловероятно, но! по всему миру работает огромное количество машин с MS-DOS, которые требуется обслуживать и удовлетворять растущие потребности их владельцев, но... только без перехода на Windows (поскольку, во-первых, Windows не является системой реального времени, а, во-вторых, стоимость перехода слишком велика, особенно если он включает в себя перепроектирование специализированных контроллеров, вставляемых в ISA-слот и слыхом не слышавших ни про какую там PCI).

Программистский рынок делает ставку на молодых специалистов, осваивающих новейшие языки, методы разработки, библиотеки и прочие "инновации", сменяющие друг друга каждые несколько лет. Но с каждым годом переучиваться становится все сложнее и сложнее. Возникает естественное желание, ухватиться за какой-то один отдельно взятый язык, например, Си, и писать программы лишь на нем одном. Это усиливает профессионализм и улучшает качество кодирования (с одной стороны), но создает практически непреодолимые препятствия

при поиск работы, на которую требуются люди "владеющие" Си++, С# и прочими модными словечками. И пускай, опытный Си-программист выполнит поставленную задачу в десять раз быстрее и в сто раз эффективнее, чем студент, прослушавший курс лекций по Си++, но так и не осиливший парадигмы процедурного программирования, от чего исходные тексты пестрят глобальными переменными и массивами, передаваемыми по значению, а не указателю. Главное, что он позиционирует себя как Си++ программиста!

Конечно, это несколько утрированный пример и такие "специалисты" в большинстве случаев отсеиваются еще на собеседовании, но... факт остается фактом. Подавляющее большинство программистских контор при наборе персонала руководствуется шаблонными стереотипами. Даже посредственный программист, идущий в авангарде прогресса, с высокой степенью вероятности найдет себе работу в любой точке мира (правда, при этом ему придется конкурировать с девушками, окончившими двухмесячные курсы по С# и знающие язык исключительно на уровне синтаксиса, а где есть конкуренция, там нечего надеяться на достойную зарплату).

Удаленное сотрудничество многократно расширяет ареал поиска работы. В конечном счете намного выгоднее искать возможности применения уже существующих знаний и навыков, чем заниматься непрерывным самообразованием. Чтение компьютерных (и околокомпьютерных) книг и журналов вызывает устойчивое ощущение, что мир сошел с ума и миллионы леммингов мигрируют на новые платформы задолго до их появления. Но при живом общении с реальными разработчиками, выпускающими реальные продукты (а не утилиты, написанные одним щелчком мыши) все становится на свои места. Основной программистский костяк стоит на трех китах (одним из которых является Си) и не собирается с них сходить.

Проблема России в том, что крупные программные комплексы у нас уже давно не создаются (немногочисленные успешные проекты типа FineReader не в счет), а мелкие утилиты можно писать на чем угодно, хоть на DELPHI, хоть на С#. Им неведомо бремя совместимости, они совершенно не знают, каково сопровождать гигабайты кода, первые строки которого были заложены еще в начале девяностых и по которым можно проследить эволюцию развития процессоров, языков программирования, компиляторов и операционных систем. Переписывать все это хозяйство на С# никто не будет хотя бы уже потому, что к моменту завершения столь грандиозного процесса, о С# быть может никто и не вспомнит, т. к. появятся новые языки. Кстати говоря, слухи о смерти Фортрана (и других древних реликтов) сильно преувеличены. Существует огромное количество кода и множество библиотек. Выпускаются компиляторы, оптимизированные под современные процессоры и т.д. и т.п. А это значит, что специалисты по языку по-прежнему остаются востребованными (пусть и не в таких количествах как десять-двадцать лет тому назад).

В практическом плане это значит, что если вы собираетесь сотрудничать с отечественными компаниями, то с вероятностью близкой к единице, в список обязательных требований к кандидату попадет .NET. Практически нигде вы не встретите объявлений, о розыске "чистых" Си-программистов, знатоков процессора и ассемблера и других "доисторических" языков. За рубежом картина иная. Во-первых, здесь создается множество встраиваемых устройств с микропроцессорами (причем, "микро" в прямом смысле слова), для программирования которых используется либо чистый ассемблер, либо смесь ассемблера с Си. Во-вторых, даже в мире РС, сплошь и рядом встречаются высокопроизводительные библиотеки (или графические движки) написанные на Си (или на Си++ в стиле Си) с ассемблерными вставками.

Количество проектов, использующих более половины возможностей, предоставляемых языком Си++, исчисляется единицами и большинство из них находятся на грани провала. Программисты в авральном порядке переписывают код, возвращаясь к истокам: проще! быстрее! надежнее! только ведь на собеседовании не скажешь: ознакомившись с Си++ и попытавшись применить его на практике, я убедился, что подавляющая часть предоставляемых им возможностей мне совершенно ни к чему и не упрощает, а усложняет кодирование, ресурсоемкость, сопровождение и отладку.

Другими словами, если в окрестностях вашего обитания найти работу по специальности никак не удастся, почему бы не попробовать посотрудничать с зарубежными компаниями через Интернет?

О КОМПАНИЯХ

Компании бывают разные — жидкие, твердые, газообразные... Одни не приемлют удаленного сотрудничества в принципе (это корпоративная политика у них такая и против нее

не погрешь), для других же — это норма. Некоторые на 90% из удаленных сотрудников и состоят (в особенности, это характерно для Open Source проектов). Сразу два совета (многократно проверенных и перепроверенных на личном опыте).

Если компания не практикует удаленного сотрудничества по тем или иным причинам, не стоит ломиться в закрытые двери — все равно не проломишь. Это может быть и "бзик" руководства, и принятый уклад организации труда, и специфика обусловленная родом деятельности. Далеко не все можно передать через Интернет: документы, полученные самой компанией под подписку о неразглашении (по-ихнему — NDA), финансовые данные, обрабатываемые программой (которые так же нельзя передавать на сторону и которые очень сложно "правдоподобно" смоделировать), не говоря уже о контроллерах и прочем аппаратном обеспечении. Опять-таки, удаленный сотрудник может внедрить в программу закладку, практически ничем не рискуя, поскольку компании (даже сильной и могучей) будет очень трудно до него дотянуться. К тому же, возникают определенные финансовые и юридические сложности при заключении контракта и переводе денег. А зачем бухгалтерии лишняя головная боль? Короче, если вам скажут "нет", то это "нет".

Совет номер два: Open Source проекты (даже те из них, что распространяются по лицензии BSD, не требующей отчислений в случае коммерческого применения) бесплатны только для "народа", а всем остальным приходится платить. Взять хотя бы кодек XviD, поддерживаемый практически всеми современными автономными видеоплеерами. Что дешевле: скачать исходные тексты и засунуть их в плеер своими силами или нанять на более-менее постоянную работу одного из участников проекта? Причем, это необязательно должен быть ведущий разработчик. Подойдет любой, кто разобрался в структуре машинного кода и готов заняться его оптимизацией под конкретный процессор (а в аппаратный видеоплеерах они, как правило, достаточно маломощные). Отсюда следует важный вывод: совершенно необязательно быть активным участником Open Source проекта, достаточно просто в нем разбираться хотя бы в общих чертах и, естественно, присутствовать в списке developer'ов (иначе, вас просто не смогут найти!).

как надо и как не надо составлять резюме

О резюме скажу предельно коротко. Составлял их я мало. Читал еще меньше, поэтому, основной опыт, который я приобрел, это опыт отрицания. То есть, теперь я знаю путь, куда ходить не надо, но куда идут толпами мои коллеги, глядя на которых так и хочется прибить к столбу табличку: "осторожно! грабли!". Но мы же ведь не о граблях говорим, а о резюме.

А резюме это сплошной официоз. Это прокрустово ложе, пытающее загнать векторную величину (коей и является интеллект) в безликий скаляр. Возьмем двух людей: Петю, участвовавшего в десяти проектах, все десять из которых завершились полным провалом (потому, что Петя, переоценив свои возможности, полез совсем не в ту степь, получив вместе с крахом бесценный опыт по его преодолению) и Васю, успешно поддерживающего три проекта, и не имеющего никакого представления о существовании тех проблем, которые Петя чует за версту и обходит на форсаже задолго до их появления на горизонте. Не знаю как вы, а вот я с Васей в горы не пошел, но... при обработке резюме секретарем далеким от программирования, Петя получает сто очков, а Вася... идет в горы и все лесом, лесом, лесом...

Другая проблема — честность. Точнее, ее отсутствие у некоторых из нас. Довольно часто спрашивают — можно ли врать в резюме? Не с моральной точки зрения, а вообще? Ну это как сказать... Перечень большого количества языков, средств разработчики, библиотек и операционных систем с которыми кандидат якобы умеет работать обычно дальше корзины не идет. Иногда, правда, могут дернуть на собеседование — посмотреть: что это за зверь такой, но надеяться на это не стоит. Если вы действительно знаете весь спектр языков и операционных систем, лучше составить несколько независимых резюме, пользуясь тем замечательным фактом, что русские имена на английский манер можно писать слегка по-разному. Это существенно увеличивает вероятность быть замеченным.

Опыт работы с языком — весьма провокационный вопрос. Я никогда не понимал, что он в действительности означает и от чего предполагается вести отчет: с момента первой откомпилированной программы типа "hello, world!", с момента овладения языком без поминутного подглядывания в учебник/MSDN, с момента постижения стандарта и преодоления барьера переносимости (т. е. умения писать программы, которые компилируются и работают не только на машине их создания), или же... с момента осознания того факта, что проверка корректности указателей зачастую (но не всегда!) несет больше вреда, чем пользы? В результате мы получаем "вилку" в десяток лет, а то и больше.

Любой кандидат может написать, что имеет гигантский опыт работы с Си++ и что он начинал программировать на нем еще тогда, когда Си++ был всего лишь front-end'ом к Си. Естественно, подобная наглая лож немедленно выявляется на собеседовании (а, может, это и не лож вовсе! просто человек `_так_` программирует, некоторые русские в Америке и по тридцать лет живут, а знают английский хуже самоучки, прошедшего двухмесячный курс по методу Ильи Франка). Так что понятие "опыта работы с языком" применимо только к людям со средними способностями и если ваши способности отличаются от таковых, рекомендуется сделать перерасчет в ту или иную сторону.

Что же касается умышленного завышения опыта, то я скажу так — врать не стоит, а вот приврать сам мышцх велел, только главное не завратся, а то может очень легко получиться, что ваш опыт работы с языком на несколько лет превысит время существования самого языка (правда, тут всегда можно открутиться, сославшись на то, что вы начинали с самых первых альфа-версий, не анонсированных ни в каких пресс-релизах и доступных только по предварительной подписке — навряд ли работодатель захочет проверить этот факт, тем более, что проверить его весьма затруднительно).

Самое противное, что заполняя резюме, мы предоставляем массу сведений о себе, при этом не имея никакой (или практически никакой) информации о работодателе. Особенно, если резюме отправляется в кадровое агентство (или вывешивается на сайт), а не вручается конкретному работодателю персонально. Часто приходится сталкиваться с такой картиной. Составляешь резюме, сидишь ждешь и вдруг звонок: "мы рассмотрели Ваше резюме, все ОК, Вы нам подходите, мы Вас берем!". Кто это "мы"?! Смотришь... а там фирмочка размером чуть больше собачьей конуры и какая-то совершенно непрофильная (например, я указывал в резюме: x86 ассемблер, Си, а они клепают сайты на PHP). Такое впечатление, что они берут всех, кого только удастся загрести, а резюме читают штрихпунктиром по диагонали.

По моим наблюдениям (которые, возможно, окажутся и неверны), нормальные фирмы, резюме не читают, а набирают сотрудникам по другим каналам, вербуя известных (или подающих надежды) специалистов из конкурирующих компаний или программистов-одиночек. Так же, я бы хотел предостеречь от заполнения резюме просто "по приколу". Если вы не собираетесь устлаваться в такую-то компанию, не пытайтесь посмотреть, что будет если кликнуть мышью по "вакансиям" на фирменном сайте и предложить им свое тело (вместе с мозгами) в обмен на пачку вечнозеленых. Последствия могут быть весьма неблагоприятные (особенно, если указан контактный телефон). Привяжутся — не отстанешь, а если скажешь, что пошутил, и устлаваться совершенно не собирался — в будущем может сильно аукнуться. Ведь мир, как известно, тесен.

Короче, подытожив сказанное, резюме мы отмечаем как тупиковый вариант.

стратегия поиска работы

В жизни каждого программиста рано или поздно наступает такой момент, когда не ты ищешь работу, а она тебя, что порождает совершенно неведомую дотоле проблему: отказать клиенту, одновременно с этим не потеряв его. Не отказывать нельзя, иначе многолетне наслаения заказов прогнуться под собственной тяжестью, возбуждая бурную тектоническую активность заказчиков и окончательный съезд крыши исполнителя с глубоким нервным истощением, граничащим с высадкой на дур. дом. Какое же это счастье — иметь непрекращающийся поток заказов, но какая же эта пытка осознать, что он не закончится `_никогда_` и выкроить недельку-другую отдыха удастся разве что в той жизни. Правда, когда молод, полон энергии и сил, об этом не задумываешься и ничего кроме перспективы не замечаешь. А перспектива, как известно, штука коварная. Параллельные рельсы сходятся в точке на горизонте, но по мере продвижения вперед эта точка все удаляется, удаляется и удаляется...

Отсюда — при выборе работе предпочтение следует отдавать краткосрочным проектам, не векующим за собой хвост всевозможных обязательств (типа поддержки, сопровождения и т. д.). Такие проекты могут приносить немалый доход вначале, но требовать огромных (и практически неоплачиваемых) усилий в конце. Вести несколько проектов одновременно — тяжелое испытание. Они затягивают словно болото и не отпускают, а все попытки вырваться обычно только усугубляют и без того незавидное положение. Удовольствие от работы — исчезает, времени на "поковыряться" в новых языках/библиотеках/осях нет и не предвидится. Кругозор сокращается в точку, стремительно коллапсирующую в черную дыру, откуда еще никому не удавалось выбраться.

Гораздо лучше работать по принципу: "получил ТЗ, въехал в тему, сдал проект, забыл". Конечно, в этом случае работать придется еще больше, но зато работа будет разнообразной и практически всегда можно выбрать время для отдыха или своих собственных некоммерческих проектов.

"Американская мечта", описанная во многих книжках по "шароваростроению", как некий никому не известный программист, написал одну программу (угробив на это свой отпуск), а потом до конца жизни ничего не делал, только пил пиво и вносил незначительные изменения в код — это миф, если не наглая брехня. Конкретных примеров таких программ нет и не будет! Компьютерная индустрия слишком быстро развивается, вынуждая переписывать значительные куски кода каждые несколько лет просто для сохранения совместимости с существующим окружением, не говоря уже о том, что требования к продуктам (и в особенности, к их интерфейсам) неуклонно ужесточаются, а если популярный продукт стоит на месте и не развивается, его тут же обходят конкуренты.

Но это что касается профессионалов. А как быть новичкам? Они любой работе рады! И в этом-то и заключается их главная ошибка. Каждый выполненный заказ на раннем этапе развития должен работать на ваш авторитет, чтобы его было не стыдно упомянуть при поиске всех последующих. Причем выбирать заказы следует на пределе своих возможностей или даже чуть-чуть выше их. Во-первых, сложные заказы это престижно, во-вторых, "повторение пройденного" (т. е. написание простых программ) отнимает время, но не дает никакого нового опыта взамен. А для успешного "старрапа" важно за минимальное время освоить максимум технологий, причем, "освоить" не в смысле "прочитать учебник", а в обязательном порядке опробовать все это на практике. Книги, бесспорно, отличный учитель, вот только доверять им следует с большой осторожностью. Типичный случай: "когда я узнал о конструкции X, я стал применять ее везде, где только можно, в результате чего поимел столько проблем, что решил навсегда отказаться от X, но через несколько лет, я все-таки пришел к выводу, что X — хорошая штука, если применять ее с умом".

Лучший способ научиться программировать — дополнить чтение учебников разбором чужих исходных текстов, пытаясь ответить на вопрос: почему они написаны именно так, а не иначе? А как бы это написали вы? Естественно, это должны быть достойные исходные тексты, а не сочинения кота Мурзика, который начал программировать, раньше чем писать. Где брать достойные исходные тексты? Ну это, в общем-то, не вопрос. Существует куча Open Source проектов по одному виду которых можно сказать отстой это или не отстой.

Естественно, программисты не безгрешны и роюсь в исходных текстах мы наверняка найдем баги или просто неэффективные фрагменты кода, которые так и просятся, чтобы их переписали. Отправляя исправленные версии разработчикам, мы скорее всего получим приглашение вступить в их ряды и подписаться на закрытую рассылку. Как правило, большинство остальных участников проекта уже работает в других компаниях (или даже возглавляет свои), поэтому знакомство с ними будет весьма не лишним. Дело тут даже не в заказах (которых может и не быть). Это отличный канал для получения документации и разных программ, распространяемых либо за деньги (слишком большие для начинающих программистов), либо же только через NDA, но... из чисто дружеских побуждений, зарубежные коллеги охотно делятся этим добром с остальными, особенно если живут под лозунгом "информация должна принадлежать миру". Тоже самое (только в меньшей степени) относится и к закрытым проектам. Если вы имеете идеи по улучшению продукта, а еще лучше — написанный плагин (или захученную DLL/EXE), попробуйте списаться с разработчиками. Быть может, они и примут вас в свой тим.

Другой способ заявить о себе — это форумы, блоги и прочие электронные (а еще лучше — бумажные) публикации. Форумы, естественно, англоязычные, причем достаточно серьезные. Общие правила поведения — в полемику не вступать, не пытаться показаться слишком умным, не лезть в те дебри в которых не разбираешься, отвечать обстоятельно и по делу. Естественно, все это требует времени, не гарантируя никакого эффекта. Блоги в этом отношении намного более предпочтительнее. Выбираем блог продвинутого программиста и начинаем постить, но только ни в коем случае не отзывы на статьи, а уж тем более — сообщения об ошибках. Еще неизвестно, что это за человек и как он отреагирует на критику (неважно — конструктивную или нет). Лучше всего — делиться своими идеями, дополняющими статью или пытаться вызывать автора на обсуждение некоторых технических проблем, которое переходит либо в перепалку, либо в очередной заказ.

Заводить свой собственный блог стоит только в случае избытка свободного времени, поскольку моментальной отдачи он все равно не принесет, а поддерживать его в "живом" состоянии быстро надоест и из развлечения превратится в пытку. Огромную отдачу приносит

издание книг за рубежом. Заказы валят косяками, даже если сама книга практически не продается (а она и не будет продаваться, если рассчитана на специалистов, коих единицы. Напротив, бойко раскупаемая книга, ориентированная на массовую аудиторию, не принесет ничего, кроме гонорара, да и тот навряд окупит время, потраченное на написание).

Еще большую отдачу дает выступление на зарубежных конференциях. Предложения о сотрудничестве начинают поступать чуть ли не сразу после схождения с подиума. Ну тут я сужу уже не по своему опыту, т. к. сам никогда ни на каких конференциях не был и бывать не собираюсь. Я же мышцх! Серый пещерный зверь! И вытащить меня из норы очень трудно!

Достаточно многие форумы имеют "коммерческий" раздел, публикующий предложения о работе. При отсутствии заказов стоит залезть туда и посмотреть на "раздачу". Не всегда, но достаточно часто таким путем можно "подцепить" хороших клиентов, с которыми потом завязывается продолжительное и весьма плодотворное сотрудничество, однако, поиск на форумах — это крайняя мера, к которой я когда-либо прибегал. Во-первых, большая аудитория создает конкуренцию, ощутимо сбивающую цену, а, во-вторых, большинство предложений — провокация или развод чистой воды.

требования к кандидату

Программы, окружающие нас, написаны такими же людьми, как и мы (многие из которых только учатся программированию) и если вы не умеете превращать байты в золото, ничего страшного в этом нет. В конечном счете, главное — это умение преподнести себя, а программистские способности с уровнем зарплаты связаны очень слабо. В IT-индустрии сплошь и рядом можно встретить людей, получающих свыше 100.000\$, разъезжающих на "Феррари", одевающихся в пиджаки индивидуального покроя с бэджиком "эксперт", но при этом знающие только одно: на сервере должен стоять RAID. Проблемы, создаваемые этим RAID'ом, их совершенно не волнуют (они ведь не знают, что ту же самую надежность можно обеспечить и другим путем), а восстановление RAID'a (в случае краха) все равно поручат не "эксперту", а технику в робе с куда более скромной зарплатой. Но зато этот техник знает кучу файловых систем, множество физических интерфейсов и переставляет чипы с закрытыми глазами да так, что следы пайки видны только в лупу (а то и не видны совсем).

Все очень просто. Одни люди стремятся к знаниям, другие же тем временем делают карьеру. Но не стоит думать, что сделать карьеру это легко. Тут определенный тип личности нужен, каким-то седьмым, восьмым, девятым чувством угадывающий что нужно лизать и кому. Но все это лирика. Пора перейти к суровым реалиям.

Знание английского разговорного `_без_` словаря — при сотрудничестве с зарубежными компаниями в общем-то необязательно, но очень, очень желательно. По фильмам (с оригинальной звуковой дорожкой и субтитрами) за два-три месяца язык осваивается до вполне приемлемого уровня и легко распознается на слух. Богатый словарный запас для разговорной речи не понадобится. Напротив, по моим наблюдениям он только мешает. Важно не количество слов, которые ты знаешь, а умение выразить свои мысли теми словами, которые сейчас крутятся в голове. За произношение волноваться не надо. Я неоднократно слышал как американцы (причем весьма коренные и респектабельные) произносят "Core" как "Корэ", а "Work" как "Ворк". Не знаю, то ли они умышленно коверкают слова на манер нашего "удава", то ли действительно американская почва позволяет себе подобные вольности с английским языком, но факт остается фактом — вас поймут! А вот без навыка быстрого письма (без ворд'a и словаря) придется туго, поскольку писать придется много. Впрочем, язык легче всего осваивается в процессе общения, так что заблаговременно учить его "вхолостую" без конкретного приложения — означает понапрасну терять время.

Языки программирования, процессоры и операционные системы — это по выбору. Главное не бояться ответить "нет", "не знаю", "это мы не изучали". Нормальные компании охотнее берут человека, способного обучаться, чем энциклопедический словарь. Ни в коем случае не стоит врать, утверждая, что вы знаете то, чего вы не, поскольку, элементарная проверка поставит вас в тупик, и у работодателя сложится впечатление, что и все остальное вы знаете точно так же, то есть никак.

Важное замечание на счет ООП. Есть два типа людей. Одни считают, что парадигма программирования (равно как и господь бог) должна быть одна на всех, другие же рассматривают ООП не как венец эволюции, а рядовой, да к тому же весьма специфичный инструмент с ограниченной областью применения. Поэтому, если вам ближе по душе процедурное программирование в духе старого доброго Си, так об этом и говорите. Отказать в приеме на работу могут, но на костре во всяком случае не сожгут.

трудоустройство

Удаленное сотрудничество часто не подкрепляется никакими бумагами, ограничиваясь устной договоренностью, действующей по принципу "деньги — стулья". И хотя при этом вас могут элементарно "кинуть", ничего не заплатив, заранее настраиваться на это не надо. Прием на работу или раздача заказов происходит достаточно буднично и без юридической волокиты. Но это — в идеале.

Достаточно многие фирмы не успокоятся, пока не увидят того, кого они нанимают живьем или, по крайней мере, не услышат его по телефону (звонок, естественно, оплачивается за их счет). Еще могут попросить переслать какую-то загадочную сигнатуру в pdf с договором. Так вот, эта "сигнатура" по нашему будет роспись. Расписываемся на бумаге, прогоняем ее через сканер и вставляем в pdf любой утилитой, которая только есть под рукой. Маразм, конечно. Ведь так можно подделать любой документ, но против чужих нравов не погрешь. Так же могут потребовать отфаксить паспорт (у меня требовали только один раз, но я отказал, в пылу спора меня сначала послали, но на следующий день мы пришли к компромиссу: я не посылаю паспорт, а они не посылают меня).

Договоры (если таковые все-таки составляются) не факсятся практически никогда. Вместо этого они печатаются на лазерном принтере в двух-трех экземплярах, после чего вы расписываетесь на каждой странице и отправляете договор по почте, а через некоторое время ловите ваш экземпляр с подписью работодателя и печатью. Прекрасное средство личной гигиены надо сказать!

Да! На счет гигиены. Если кто думает, что договор перед подписью нужно показать юристу, то он здорово ошибается, поскольку договор в 9 из 10 случаев типовой и каким бы то ни было изменениям не подлежит. В тексте договора может быть все что угодно. Любая нелепица. Вплоть до передачи компании прав собственности на вашего домашнего кота Мурзика. Но попытки исключить подобные пункты из договора ни к чему (хорошему) не приводят, поскольку человек, отправляющий текст договора вам на подпись, обычно не уполномочен менять этот самый договор ни на бит. С другой стороны, договор — это для юристов. Реальный расклад задается устной договоренностью. Даже если в договоре черным по белому написано, что вы не имеете использовать компоненты разработанного вами продукта в других проектах, вы _можете_ делать это, договорившись с фирмой по хорошему. И если не сильно наглеть, на договор можно вообще не смотреть.

В особо "ответственных" случаях потребуются вылет за рубеж или в российское представительство компании. Зарубежный перелет (и все связанные с ним расходы) в 100% случаях оплачивает компания, которая вас нанимает. Поездку в российское представительство, как правило, приходится оплачивать самостоятельно, но можно не ехать, а позвонив по телефону, попробовать договориться как человек с человеком. Во всяком случае у меня это получалось (с некоторого времени я стал тяжел на подъем и уже давно никуда не езжу, а раньше довольно много мотался по России, останавливаясь в номерах "люкс", оплачиваемых, естественно, фирмой. эх... молодость... но потом все это надоело).

финансовые вопросы

Неразвитость отечественных платежных систем создает значительные проблемы при регулярном переводе денег из зарубежных стран в Россию. Лично меня механизм перевода волновал меньше всего. Поговорив со знающим человеком, я выяснил, что наибольшей популярностью пользуются два пути: открытие валютного счета в коммерческом или сбербанке и чеки, пересылаемые по почте. Чеки идут от нескольких месяцев до полугода, причем банки их то принимают, то отвергают без объяснения причин (хорошо, если не обвиняют в попытке подделки или мошенничестве), а регулярные переводы на валютный счет вызывают подозрения у всех спецслужб, которые только есть в нашей стране. И не только налоговой (с налоговой как раз проще всего: заплатил налоги — и спи спокойно). Экс-КГБ может запросто позвонить с визитом и поинтересоваться: а вы случайно не Родину на секреты продаете? Можете, вы шпион? А вы точно доступа к секретной информации не имеете? Ну это вы говорите, что не имеете, а если проверить? Вдруг какой-то секрет просочился в Интернет (ведь у вас есть Интернет, правда?), а вы его того. В смысле взяли и продали.

Надежнее всего перевод на рублевый сбербанковский счет, упоминание о котором приводит иностранец в ужас, граничащий с суицидом, и они приводят тысячи доводов, почему рублевый сбербанковский счет это бэд. Но если твердо стоять на своем, то... куда они денутся. Переведут. Правда, могут содрать до 10% в качестве моральной компенсации, но это уже расплата за удобство. Жителям больших городов, возможно, удобнее всего получать деньги на

кредитную карточку. Ну тут я не знаю. У нас всего один банкомат, да и тот находится в слегка разобранном состоянии.

как избежать "кидалова"

Начнем с грустного. Полностью застраховаться от обманов и невыплат при удаленной работе практически невозможно. Ни имя компании, ни наличие договора — это еще не гарант. Меня кидали и крупные, и мелкие компании всеми способами какие только есть. Самый распространенный вариант это когда вам говорят, что вот сейчас вы получите совсем немножко, чтобы не умереть с голоду, а потом — пойдут отчисления от продаж продукта, объем которых проконтролировать практически невозможно, а даже если было бы и возможно — что с того? Деньги-то ведь все равно не заплатят.

Менее распространенный, но гораздо более гадкий способ — найти в вашей программе пару мелких багов и поднять вселенский вопль, что из-за них компания понесла огромные убытки и теперь имеет все основания для подачи на вас в суд (брехня, конечно, но что мы можем сказать в ответ?) поэтому, ни о каких выплатах речь вообще не идет. Как вариант, компания может постоянно вносить значительные изменения в ТЗ, давая минимум времени для их реализации, а за всякую задержку — штрафовать. Так что к концу работы будет непонятно кто кому платить должен. А могут просто взять программу, сказать "спасибо" и... не заплатить.

На первых порах (пока ни имени, ни опыта сотрудничества еще нет), остается надеяться только на удачу и проявлять максимум осмотрительности в выборе партнеров. Как правило, компании среднего звена в этом плане наиболее честные, поскольку если они будут кидать программистов направо/налево, об этот быстро узнают все остальные и тут же разорвут партнерские отношения. Мелким компаниям это не грозит, поскольку им нечего терять. Они просто закроются и тут же откроются вновь под новым именем. Крупные компании, которые правят балом, могут вытворять все, что им вздумается, в том числе и не оплачивать разработку продукта. Монополисты... К таким и на бульдозере не подъедешь!

Вот почему сотрудничество с новой (для вас) компанией следует начинать с небольших проектов, и только потом, убедившись в ее честности, переходить к по-настоящему серьезным вещам, работа над которыми может занять несколько лет.

Что же касается меня, то с непроверенными заказчиками, я работаю либо из спортивного интереса, либо при 100% предоплате. Естественно, не все заказчики на это соглашаются, но... лично меня это не сильно волнует, поскольку заказов и так выше крыши... надо же как-то эту крышу расшищать...

заклучение

Удаленная работа — это реально! Это хороший способ совместить приятное с полезным и заработать деньги, не выходя из дома. Кстати, "дом" — это ключевой момент. Далеко не каждому понравится подобное (и к тому же добровольное) "затворничество". Если одиночество вас угнетает — ходите в офис, работая среди людей! Или... совмещайте оба вида работы сразу, если, конечно, "корпоративный устав" это позволяет. Компании, в которых мне довелось работать, крайне ревностно относились к интеллектуальной собственности и считали, что все написанное программистом в рабочее время принадлежит им и если программист продает свой труд на сторону, он должен платить "мзду". Что ж! Вполне логично! Именно поэтому, я там больше не работаю, а... сотрудничаю с ними через Интернет! На гораздо более выгодных (для меня ;-) условиях.