

программирование денег лопатами

крис касперски ака мыцх, по email

*...это не ты владеешь деньгами,
это они владеют тобой.*

восточное

говорят, что наука — это удовлетворение собственного любопытства за чужой счет. программирование, между прочим тоже! графический интерфейс аляповатых приложений скрывает огромные денежные вложения и если вовремя подсуетиться, можно кое-что отхватить от этого куска. хотите узнать, какова ваша рыночная стоимость и как не продешевить на продаже продуктов своего труда?

введение

счастье не в деньгах, а в умении их зарабатывать

...современный человек испытывает глубокое недоверие практически ко всему, что не связано с поглощением или испусканием денег. внешне это проявляется в том, что жизнь становится все скучнее и скучнее, а люди — все расчетливее и суше. отсюда и знакомое любому чувство, что все упирается в деньги

Виктор Пелевин. "Поколение П"

Стоит ли вообще стремиться к деньгам? Быть вовлеченным в безжалостный круговорот потребления и выделения. Зарабатывать и накапливать. Накапливать и тратить. Тратить и приближаться к иллюзорной картине коллективной галлюцинации, называемой "счастьем". Скажите честно — оно вам надо? Для счастья не нужно ничего, кроме сознания, а деньги приносят только пустоту и страдание, но обычно это понимаешь лишь тогда, когда их (денег) становится слишком много и уже поздно что-либо менять. Жизнь прожита, а вложенное в деньги время не вернуть назад. Человек потребляющий в конечном счете живет и работает на унитаз. Может, настала пора остановиться и подумать о чем-то более возвышенном?

Посмотрите "Бойцовский Клуб" (Fight Club), найдите и прочитайте "Поколение П" Виктора Пелевина и "99 франков" Фредерика Бегбедера. **Вставляет крепче любого драпа вставляет всерьез и надолго**. Пошлите карьеру, деньги шмотки **наххх [на хрен]** и займитесь, наконец, самим собой. Тем, чтобы вы с удовольствием делали и бесплатно, тем что наполняет жизнь смыслом, удерживает у монитора до самого утра, тем что вам по-настоящему интересно, а я покажу как обернуть этот интерес в деньги. Никакого противоречия здесь нет. Плохо — когда деньги становятся самоцелью и хорошо если они средство (существования) или инструмент (реализации своих идей). Вот об инструментальных средствах мы сейчас и поговорим.

Несогласным лучше сразу сменить профессию и заняться чем-то другим (торговлей бананами, например). Программирование — не лучший способ для быстрого обогащения (во всяком случае в России) и что еще хуже — требующий колоссальных начальных вложений. самого ценного и к тому же невозполнимого ресурса. Времени. Придется много-много учиться, а научившись — расставаться с накопленными знаниями и навыками и вновь переучиваться. Ведь прогресс не стоит на месте и с каждым годом бежит все быстрее и быстрее.



Рисунок 1 раз, два, три, четыре пять, вышел доллар погулять

куда податься?

Нашим предкам было хорошо. Они сидели в пещерах, укрывались шкурами и находились на полном обеспечении у государства, решающего все кадровые вопросы за тебя. С приходом свободы, а, значит, и возможности выбора, позиция "самых сильных программистов в мире" значительно пошатнулась. Никто не собирается оплачивать из своего кармана поиск недокументированных возможностей в недрах операционной системы или программирование в чистом машинном коде. Преемственность программисткой школы нарушена, старшие уже не учат младших, преподаватели вузов бесконечно далеки от проблем практического программирования и безнадежно отстают от прогресса.

Как обрести мастерство? Очень просто — устроится на постоянную работу в фирму, где еще сильны программистские традиции и имеются профессионалы, способные научить молодежь, передав ей часть своего опыта (только помните, что ремеслу не учатся, ремесло воруют). Трудовой договор (он же "контракт") бывает двух типов — ограниченный или бессрочный. В России чаще всего практикуется второй (устроились на работу и хрен вас потом уволишь, в "нормальных" фирмах легче терпеть безделье ничего не делающего человека, чем затевать волокиту с его увольнением, короче полная лафа — не так важно знать программирование, как законы). Зарубежные фирмы в своей массе нанимают специалистов лишь на очень короткий срок (например, на год), после чего контракт может быть либо возобновлен, либо послан пинком под зад. Причины? Специалист не оправдал возложенного на него доверия, или проект завершен и его услуги фирме больше не требуются. Тысячи специалистов, работающие над "Боигном" были выброшены, после того как птичка начала летать. Короче говоря, чем больше вкалываешь, тем быстрее тебя увольняют. Но на любую хитрость найдется свой лом. Умудренные опытом специалисты затягивают сдачу проекта всеми силами, каждый раз "находя" новый дефект, требующий доработки. Птиц вроде бы и летает, но в тоже время и нет. Работодатель нервничает, ругается матом, называет всех суками и дураками, но... вынужденно продлевает контракт.



Рисунок 2 рабочие будни программиста

Сейчас всюду требуют знания приплюсного Си и OLE, а как быть если вы его ненавидите любой ненавистью? Или, что еще хуже, живете в глубокой провинции, где в основном занимаются поддержкой и внедрением, а программирование задвинуто в глубокий подвал с робкими попытками нарисовать что-то на DELPHI и впарить неразборчивому клиенту за полцены? Программировать, в принципе можно и удаленно. "В принципе", потому что большинство нормальных фирм крайне настороженно воспринимают такой способ сотрудничества, предпочитая Интернет-технологиям старый добрый офис и постоянный штат. К тому же оплата удаленных программистов в несколько раз ниже, зато и выбор работы шире. Трудоустраиваются даже те, кто знает Фокс-доместос или даже древний Кобол, не говоря уже о системном программировании, машинных кодах и ассемблере. К тому же, одновременно можно устроиться в трех-четырех местах. Совокупная зарплата практически не ограничена. Главное — работоспособность иметь!

Чисто технически, можно присоединиться к любому готовому проекту. На форумах рассказывают кучи историй о том, как молодой программист нашел к известному компиляторе багу и затем стал бета-тестером на постоянной основе. Или как журналист написал документацию к продукту, устроившись техническим писателем. Судя по всему, это легенды, выдуманные для самоуспокоения. С бета-тестерами чаще всего расплачиваются новыми бета-версиями, а что же до технических писателей... а вы попробуйте написать документацию удаленно! Без программиста под рукой! Есть менюшка, но что она делает — непонятно. А половина возможностей вообще скрыта в "недокументированных" сочетаниях горячих клавиш. Недокументированных потому, что программист просто забыл их включить в документацию, но сам он (и все его коллеги) пользуются ими постоянно...

Как вариант, можно заняться внедрением чужих продуктов у себя на местах. Но здесь больше бизнеса, чем, собственно, самого программирование (ну разве что вам передадут исходные тексты с подпиской о нераспространении) с целью доработки и адаптации под конкретные требования "аборигенов". Но для этого нужен солидный начальный капитал. Уж лучше Windows/UNIX устанавливать пользователям, благо что при нынешней компьютерной "грамотности" спрос на таких "программистов" необычайно велик и со временем он будет только расти. Но к обсуждаемой нами теме это не имеет никакого отношения.

Рынок "вольных хлебов", на которых пасутся свободные копеечники, могуч и широк. Он простирается от Аляски до глухих сибирских пенат. В отличие от сотрудничества на контрактной основе, предполагающего более или менее длительные отношения, свободный копеечник ориентирован на краткосрочный заказ. Защитить продукт от копирования. Перенести программу с Перла на Си и т. д. Услуги свободных копеечников резко полярны. Это либо высокоинтеллектуальная задача (как, например, в случае с защитой), с которой штатные сотрудники компании справляются не в состоянии, либо рутинная работа, которую дешевле перебросить на "пионеров", чем напрягаться самим (как, например, в случае с переносом). Тем не менее, пестрое племя свободных копеечников в основном состоит из профессиональных программистов, продающих свои знания по цене кокаина. Остальные в этом мире просто не выживают, уходя в "серьезные" фирмы на постоянную работу. Любой аналитик подтвердит, что 90% дохода фирма получает от 10% специалистов (а, если брать таких гигантов как Intel или

Microsoft, то соотношение и вовсе окажется 99:1). Но ведь этим 90% тоже нужно что-то платить! И это "что-то" приходится отрывать от специалистов.

Вместо того, чтобы работать на "дядю", некоторые программисты предпочитают трудиться на самого себя. Они в одиночку (или тесным коллективом) создают утилиты или даже целые программные комплексы, приторговывая ими через Интернет. Считается, что это самый прибыльный и наиболее перспективный путь. Чушь собачья! Просто так сеть и строчить шаровары не получится. Тут идеи нужны! А у кого они есть? Рынок программного обеспечения набит всевозможными утилитами до отказа. Все, что только было можно придумать уже придумано и реализовано в десятках конкурирующих меж собой софтин. Корпорации тоже не сидят без дела, постепенно прибирая новые ниши рынка к рукам. Еще недавно IE не умел сохранять страницы целиком на диск и не поддерживал докачку, что давало пишу шароварным программистам, но теперь... Старое поколение ставит ReGet скорее по инерции, чем по необходимости (V.92, DSL делают докачку все менее и менее актуальной), а WEB-страницу быстрее найти в Интернете, чем на своем замусоренном жестком диске.

К тому же, работающая программа еще не есть готовый продукт, предполагающий рекламу, маркетинг, внедрение, техническую поддержку, обучение пользователей и т. д. Компании-регистраторы, берут на себя лишь продажу и отчасти рекламу, но 99% потенциальных потребителей даже не подозревают о существовании такого продукта! Доход большинства шароварщиков невелик. Удачных проектов единицы и нечего на них ориентироваться (вы бы еще Била Гейтса в пример привели). Но дело даже не в этом. Разработка шаровар — широкая предметная область, охватывающая все спектры программистской деятельности: от дизайна до кодирования. Узким специалистам это не в ништяк. А широкими специалистами за два дня не становятся. Я не беру в расчет тех молодых людей, что пишут в своем резюме невообразимо длинный список языков программирования и программных пакетов в которых они как бы умеют работать. Без разделения и специализации труда, создать конкурентоспособный продукт навряд ли получится. Для этого нужна фирма. Пусть маленькая и мобильная, но все-таки фирма, а не пара небритых мужиков с горящими глазами и залитой пивом кейбордой. Еще нужны большие вложения (надо же на что-то жить, пока создается продукт), без гарантии их окупаемости. Мрачноватая картинка получается? Индийский вариант в России не удался. Валового продукта не появилось. Так... отдельные яркие утилиты и приложения: RAR, FAR, Fine Reader... Зато мы создаем лучшие в мире компиляторы (филиал Intel в Нижнем Новгороде), космические симуляторы (филиал Maris Multimedia в подмосковном Королеве), системы распознавания рукописного текста, нашедшие применение в Palm и т. д. Список можно продолжать бесконечно. Куда ни плюнь, всюду мы. Так что крест на себе ставить не нужно.



Рисунок 3 надпись гласит "готов программировать на хытымле за жратву"

чем заняться?

До недавнего времени, наиболее доходной областью было проектирование и программирование баз данных. Сначала это были простенькие приложения, затем они объединились в локальные сети, а сети — в геоинформационные системы. В России с ее вечно меняющейся законодательной базой, финансово-отчетные программы превратились в настоящую кормушку, питающую полчища разработчиков, лабающих кривой код на потоке. Теперь это в прошлом. Бухгалтера пересели на 1С и чистые программисты оказались невостребованными.

Зато сейчас интенсивно развивается телефония. Приложения для сотовых телефонов расходятся на ура (даже если это всего лишь "Питон" или "Тетрис" в стиле начала восьмидесятых). Учишь Java, скачиваешь SDK с сайта производителя и вперед! Создается множество мини-АТС на базе обычных персоналок со звуковым кодеком внутри. Берешь готовый кодек, написанный как правило на Си++, оптимизируешь его (чтобы втиснуть в быстродействие ЦП как можно больше каналов), клепаешь аппаратную часть и... железка готова!

А вот сетевое программирование приходит в упадок. Работа утекает в крупные конторы, повсеместно внедряются готовые решения, армия свободных копеечников ожесточенно конкурирует между собой и доходы, соответственно, падают. Технологии программирования сменяются одна за другой и приобретенные навыки быстро устаревают. Последующие несколько лет судя по всему пройдут под знаменем .NET'a (которому пророчат целую эпоху), правда, никаких гарантий на этот счет нет, так что вопрос о выборе средств разработки остается открытым.

Дырявость сетевого обеспечения и его неумелое использование притягивает как хакеров, так и тех, кто с ними борется. В высших учебных заведениях уже появилась такая специальность как "безопасность". Чему там учат — непонятно. Хакер это вам не прыщавый подросток! Это высоквалифицированный специалист с огромным опытом, развитой интуицией, нетривиальным мышлением, просчитывающий поведение машины на несколько шагов вперед. Роберт Моррис (известным своим червем) до этого занимался тем, что переписывал системы безопасности для больших компьютеров и с дырами был знаком не понаслышке. Предотвратить нападение червя мог только другой Моррис, а не "эксперт по безопасности", прослушивавший курс лекций, но никогда не заглядывавший в исходный код sendmail'a. Чтобы защититься от хакеров, необходимо иметь выдающуюся эрудицию и просчитывать все на десять-двадцать шагов вперед. Все что может сделать "эксперт" — это выявить грубые ошибки конфигурации системы (типа дырявый сервер, неправильно сконфигурированный брандмауэр). Заниматься консалтингом прибыльно, но рискованно. Иной клиент может и по морде дать. Во всей физической прямоте этого слова. Ну да! С него взяли деньги, а через некоторое время атаквали! Брать оплату за конкретно обнаженные дыры намного проще. Особых познаний и навыков здесь не требуется. Можно, например, дизассемблировать приложения на предмет поиска ошибок переполнения, просматривать исходные тексты, проверять скрипты или специализироваться на TCP/IP протоколах: подделке обратных адресов, перехвате трафика, обходе брандмауэров и т. д. Это увлекательно и высоко оплачиваемо (в среднем по \$500 за дыру, если предложат меньше — пошлите их наххх).

Защита программно обеспечения похоже переживает свои последние дни. Программные решения уступают место аппаратным, а демократия сменяется жестким тоталитарным режимом, расценивающим пиратство как терроризм, а терроризм — это пожизненное заключение или расстрел. Тем не менее, хорошие специалисты без работы не останутся и всегда найдут себе применение. Взять те же лазерные диски. Обычная такса за защиту: ~5% от стоимости тиража. Разумеется, приведенная цифра очень условна. Тут все зависит от уровня защиты, стоимости дисков и размеров самого тиража. За массовый продукт можно взять и 1% — не прогадаешь. А 300 — 500 экземпляров можно и не защищать. Не окупится. Разве, что поставить типовую защиту, но тогда ее сразу взломают.

Вовсю идет дизассемблирование ПЗУ и программных модулей с целью похищения оригинальных алгоритмов, восстановления структуры файлов данных или протокола обмена. Эта работа неплохо оплачивается, хотя она и не совсем законна. Наше законодательство дизассемблирование не запрещает, а его результатами все равно будете пользоваться не вы, так что ответственность ложиться на нанимателя. Правда, после этого в Америку можно въехать только чучелом или тушкой. Есть риск, что вас арестуют прямо в аэропорте. Уж лучше

отправляйтесь в Южную Корею. Там наших любят. Не в том смысле, чтобы "скушать", а как специалистов по разработке 3D шутеров и вообще.

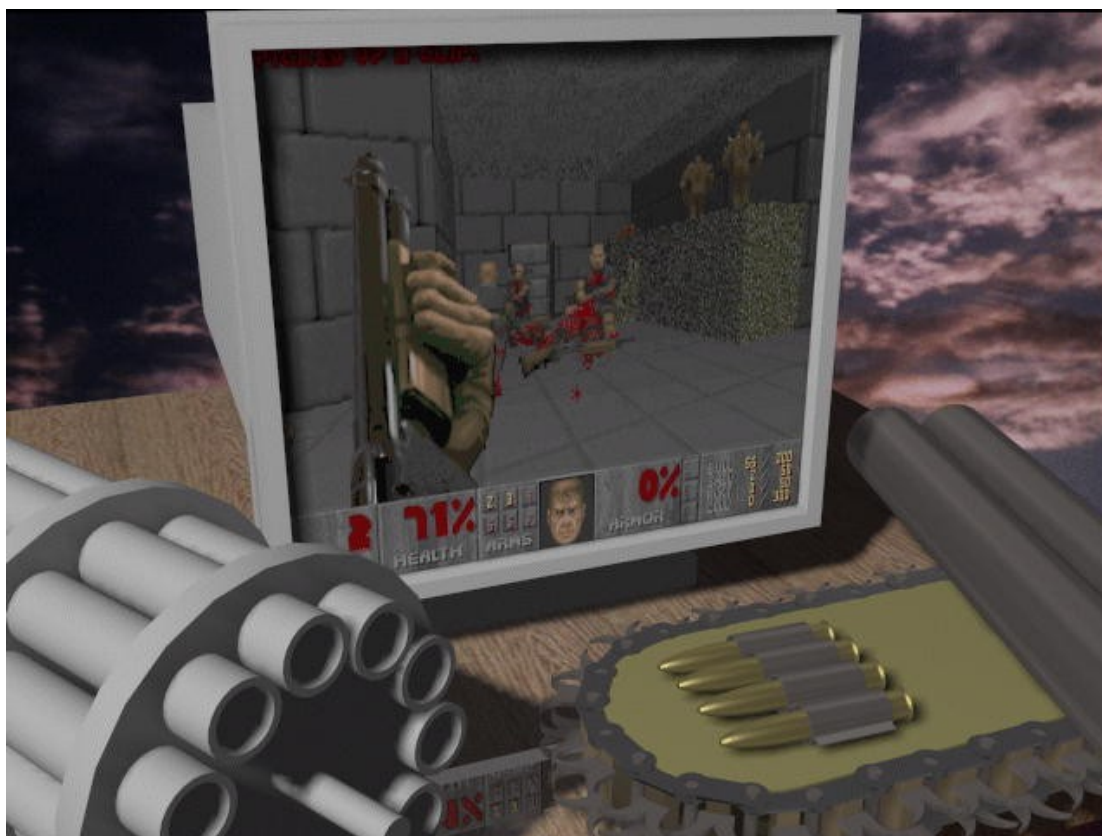


Рисунок 4 "...там где мы работаем, другие отдыхают" (разработчики игр)

Программирование игр — весьма увлекательное и к тому же неплохо оплачиваемое занятие. Игры естественно должны быть трехмерными и фотореалистичными, иначе на фиг они не будут никому нужны. К тому же быстродействие современных процессоров не бесконечно, и эффективность исполнения кода вырывается на первый план. Опять-таки, необходимо хорошо знать физику и математику. Без этого можно запрограммировать разве что крестики-нолики. А отсюда недалеко до разработки систем моделирования (движение звезд в галактике, развитие атмосферных циклонов и фронтов, развитие напряжений внутри сварочного шва). Заказчиками выступают либо институты, либо коммерческие корпорации. Кстати говоря, моделирование — это как раз та сфера, в которой Россия всегда была традиционно сильна. У индусов кишка тонка.

Восстановление данных — еще одна перспективная область, гарантирующая, что без куска хлеба специалист не останется. Даже в масштабах небольшого уездного городка проблемы с жесткими дисками и оптическими носителями случаются регулярно. Конечно, для восстановления данных на физическом уровне необходимо весьма дорогостоящее оборудование, но в подавляющем большинстве случаев разрушения носят логический характер и для их восстановления достаточно иметь редактор диска плюс пару-тройку утилит собственного написания. Автоматизированные доктора типа Easy Recovery это **херня**. Для домашнего использования вполне сойдет, но брать за такое "восстановление" деньги...

Наблюдается растущий спрос и на программирование микроконтроллеров, в которых доминирует ассемблер, си, форт и машинные коды. Работа приносит такое удовольствие, что брать за нее деньги становится просто стыдно. Но ведь дают! Правда, непостоянно. Иной раз за месяц не поступает ни одного заказа. Тогда на пропитание приходится зарабатывать сборкой домашних кинотеатров (они сейчас популярны в народе) — просто берем slim-корпус, отрываем от него мышь, монитор и клавиатуру, пишем простенький загрузчик Линуха, автоматический распознаватель формата диска и подключаем к плееру пульт управления по ИК. Да много чего сконструировать можно — была бы фантазия! Возможностей для самореализации — море. Выбирай на вкус. Слухи о безработице и невостребованности программистов сильно преувеличены. Отмирают одни специальности, но на их месте расцветают другие, подтверждая естественный круговорот. Так что работу себе не найдет только гурман или ленивый.

Что же до администрирования, сборки и ремонта компьютеров — это перспективные и притом бурно развивающиеся области, но к программированию они не относятся, а потому в данном обзоре не затрагиваются. К слову сказать, администратор — это в первую очередь хороший хозяйственник и только потом знаток сетевых протоколов и ОС. Гуру, настраивающие систему так, чтобы она работала и не падала, вынуждены конкурировать с большой армией начинающих "администраторов", даже не дотягивающих до звания продвинутых пользователей. При всех недостатках Windows, она ставится с полпинка и даже как бы работает. Необходимость в администрировании осознается начальством лишь тогда, когда его конкретно поймут, а это случается не со всеми и не всегда... То же самое и со сборкой. Выткнуть материк в корпус — большого ума не надо. А грамотно сбалансированная, дешевая, безглючная и высокопроизводительная конфигурация никому не нужна. То есть нужна, конечно, но переплачивать за это вам никто не будет. Ремонт электроники экономически не выгоден. Поломанный блок легче выкинуть и заменить новым, более современным, чем горбатиться над паяльником. "Обслуживание" компьютеров, сводящееся к переустановке Windows, вытесняет качественный сервис, когда упавшая система поднимается без радикальных перемен. Да что там говорить... Специалисты нужны лишь на гребне волны. На острие прогресса. На передовой линии фронта. В тылу им делать нечего. В тылу обирает MS Word и сексапильные секретарши.

сравнительный анализ прибыльности языков программирования

...ты спрашиваешь, действительно ли опытные российские программисты очень ценятся в США? и какие языки и операционки стоит знать в первую очередь? и так далее по пунктам?... вот что я тебе скажу, парень. люди, которые задают такие вопросы, не могут называться "опытными российскими программистами". таким людям не светит трудоустройство даже в ЮАР!

Перси Шелли

"Как перестать программировать и начать жить"

Начинающие испытывают большие трудности с выбором "перспективного" языка программирования. На самом деле все это туфта. Умение программировать и умение зарабатывать деньги — это **разные** умения, никак меж собой не пересекающиеся. Билл Гейтс был задвинут на Бейсике, но это не помешало ему стать самым богатым человеком в мире. Вот хорошая статья на эту тему "Как перестать программировать и начать жить" (www.fuga.ru/articles/2001/09/programmer.htm).



Рисунок 5 программистская валюта

Трагедия нашего поколения в том, что квалифицированные специалисты оказываются невостребованными и всюду доминирует "индийский вариант", когда неграмотные недоучки создают программные комплексы, работающие кое-как, требующие немыслимых аппаратных ресурсов и падающие от малейшего ветерка. Забудьте об оптимизации, надежности, удобстве управления и прочих абстрактных вещах. Главное — аляповатый, перегруженный фенечками интерфейс плюс передовые технологий программирования — OLE/ActiveX/DCOM/XML и прочие торговые марки, засоряющие окружающую среду и отравляющие программистскую жизнь. Вот такой **блин [блядь]** рынок мы имеем на сегодняшний день.

Но что есть рынок? Мусор, плавающий на поверхности. Основной средой разработки здесь становится Visual Basic, DELPHI, C++ Builder или Visual C++ плюс MFC. DELHI/Builder

предпочтительнее тем, что существенно упрощает перенос приложений на Линух (в идеале просто перекомпилировал и все!). Visual Studio генерит компактный и эффективный код (в некоторых случаях это критично), но требует большего времени для разработки. А на DELPHI зачастую программировать и вовсе не приходится! Идем в сеть, находим нужный компонент (а готовых компонентов для DELPHI просто море), перетягиваем мышью на форму, вот и все! Минимум телодвижений, максимум прибыли.



Рисунок 6 арабский вариант добывания денег

Зато если требуется запрограммировать что-то очень сильно нестандартное, все преимущества DELPHI испаряются как утренний туман и на первый план выдвигается Visual C++, который на самом деле никакой не Visual, а самый что ни на есть приплюснутый Си с неплохим IDE (впрочем, MultiEdit все равно круче). А плюсы это в первую очередь проектирование и только потом программирование. Visual C++ катастрофически невыгоден при решении типовых задач вроде склада или учета, но вот если потребовалось рассчитать степень сжатия металла при презентационном литье (кто видел какие там формулы — тот поймет), лучшим выбором будет Си (как с плюсами так и без) а то и древний Фортран — под них по крайней мере можно найти достойные математические библиотеки и неплохие оптимизирующие компиляторы. Графические приложения и системы потоковой обработки данных однозначно программируются на Intel C++ — опять-таки из-за наличия мощных библиотек, содержащих кучу готовых компонентов на все случаи жизни. Писать это на DELPHI или матерясь переносить на Visual C++/C++ Builder — пустая трата времени, а ведь время — это деньги.

Доход приплюсных программистов в среднем в два-три раза выше, чем у их дельфийских коллег, но и знать они должны намного больше, к тому же плюсы требуют совсем иного типа мышления. Так что выбирайте язык не по прибыльности, а по душе, но только не трогайте Visual Basic! По сложности своего освоения он вплотную приблизился к DELPHI, унаследовав все недостатки своего прародителя, как-то: отсутствие нормальных средств разработки, контроля и отладки приложений, привязанность к среде и т.д. Загляните на форумы — большинство V-Бейсковцев не решает проблемы, а пытается их обойти. Это настоящая война с языком! На большинство нанимателей слово "Бейсик" действует как красная

тряпка на быка и найти хорошо оплачиваемую работу можно лишь в узко специализированных областях, например, в финансовой сфере. Там Бейсик любят! Программы управления отчетностью под Excel или Access расхватываются как горячие пирожки. Однако, тут требуется быть больше бухгалтером, чем программистом.

Серьезные программные комплексы пишутся преимущественно на Си (не плюсах!) и ассемблере. Использование готовых компонентов от сторонних производителей здесь традиционно сводится к минимуму. Почему? Чужие ошибки обходятся слишком дорого. Даже если вы и докажете, что баг не ваш, нанимателю от этого не станет легче. Я знаю людей, которые писали собственный стек TCP/IP после того, как попытка использования уже "готового" привела к огромным потерям. Черви, дыры, отказ в обслуживании — ну в общем, вы меня понимаете. Оплата труда таких разработчиков абстрактно-астрономическая (мне известны случаи, когда на полном серьезе предлагали до 200\$ в _час_), однако, просто взять и прийти в этот бизнес с улицы не получится. Тут требуется особый инженерный дар. У меня он отсутствует и максимум что я могу — служить подмастерьем у гуру. Знание ассемблера (или даже машинных кодов) само по себе ничего не дает и потребности в таланте не отменяет. Русским языком владеют миллионы, а гениальные романы удаются считанным единицам, поэтому и говорить о рыночной стоимости подобных специалистов некорректно и глупо (рынок подразумевает массовость, а где она здесь).

Во встраиваемых системах в основном используется Ассемблер, Форт, Си и эта гнусная Java. Количество микроконтроллеров, окружающих нас, поистине огромно и хотя в большинстве своем они создаются и программируются за границей, в последнее время начали появляться и отечественные разработки. Тоже самое относится и к драйверам. Всякий драйвер программируется под какое-то конкретное железо, и чтобы его запрограммировать это железо необходимо подержать в руках. Так что удаленная работа на Тайвань отпадает сразу и остается только рыхлый сегмент отечественного рынка "танкостроения". Специалистов, умеющих держать паяльник в руках и готовых программировать на низком уровне практически нет и не будет. Спрос на них невелик да и доход скорее мал, чем огромен, поэтому встраиваемыми системами промышляет либо те, кто бесконечно любит ассемблер, либо те кто навечно остался в восьмидесятых и кроме Турбо-Паскаля ничего не знает. Мизерная стоимость старых материнских плат позволяет использовать их в ряде встраиваемых устройств, так что специалист по MS-DOS работу себе всегда найдет, хотя и будет получать плевки в след.



Рисунок 7 доллар и его тень — погоня за призраком

free lancer vs steady job

...дают зарплату за то, что хожу на работу. а если при этом я еще и работаю, тогда — премию.

Юрий Никитин

Стоит ли искать постоянную работу или лучше оставаться на вольных хлебах? Все зависит от психотипа личности. Кто-то предпочитает кочевую жизнь, а кто-то оседлую. Не будем делать программистов на "правых" и "виноватых", а лучше расскажем о каждой из сторон

поподробнее, попутно отмечая некоторые не вполне очевидные проблемы, с которыми вам придется столкнуться в пути.

Свободных копеечников (они же free lancer'ы) можно встретить в любой точке мира — от столичной квартиры до глухой провинциальной норы. Интернет стирает границы и уравнивает город с деревней в правах. Даже находясь в шалаше, вы можете скачивать стандарты и спецификации, общаться с коллегами по всему миру, оставаясь при этом запертыми в четырех стенах, которые что там, что тут — одинаковы. Свободный график (точнее, полное отсутствие такового), богатый ассортимент начальства (если шеф не нравится, не раздумывая посылаем его нахxxx), никакой карьерной лестницы со свойственными ей пороками и извращениями. Понятие "надо" кажется абстрактным и бесконечно далеким, уступая место "мне нравится" и "вот это кайф". Для творческих людей с широким спектром скачкообразно меняющихся интересов лучших условий работы нельзя и придумать.

Теперь перейдем к недостаткам. Чтобы удержаться на плаву свободный копеечник должен быть подвижным как ртуть. Заказов много — только успевай, но на блюдечке с голубой каемочкой их никто не принесет (во всяком случае на первых порах). Значит, нужны обширные связи. Тематика заказов самая разнообразная — от микроконтроллеров до телефонии и узким специалистам приходится туго. К тому же в последнее время наметилась неприятная тенденция оттока заказов в корпорации и крупные программистские фирмы, у которых свой укомплектованный штат.

Вместо райских садов нас встречают дикие джунгли, живущие по принципу "волка ноги кормят". Обычно заказы ходят косяками — то вообще никакой халтуры нет, а то кааак сыпанет! Нахватаешь ее на радостях (после месяца ничегонеделанья любая работа встречается с трудно скрываемым энтузиазмом), а потом чешешь репу и думаешь — когда же я все это делать буду? Какой там сон! Хорошо если вздремнешь на клавиатуре часок-полотора. Какая еда! Жуешь бутерброд вместе с промасленной распечаткой, попутно обдумывая архитектуру будущего проекта и гоняя мышью свободной рукой! Это на постоянной работе можно прийти, лениво почитать журналчик, пофлиртовать с секретаршей, а потом, закрывшись у себя в кабинете, поDOOMать или початится до конца трудового дня.

У свободных копеечников расклад совершенно иной. Времени на личную жизнь практически ни у кого не хватает, что часто приводит к жестоким размолвкам в семье. Вы пробовали программировать при жене? Я — пробовал. Вынеси — то. Подай это. Сделай все наоборот. Ты меня совсем не любишь и т. д. В общем развелся. Не могу удержаться, чтобы не привести еще одну цитату "*...молодой инженер уходил на работу к восьми утра, работал без перерыва на обед, уходил с завода часов в семь, приезжал домой, полчаса играл с ребенком, ужинал с женой, ложился в постель, быстро занимаясь с ней любовью, затем вставал и, оставив ее в темноте, уходил на два-три часа за свой стол, чтобы поработать над парой вещей, которые взял с собой. Он мог, уходя с завода, заглянуть в Wagon Wheel и выпить пива... Вернувшись домой в девять, когда ребенок уже уснул, ужин холодный, а жена еще холоднее, он пытался объяснить ей что-то, а в голове у него вертелись совсем другие мысли: LSI, VLSI, альфа-поток, прямое смещение, паразитические сигналы... К тому же в Wagon Wheel он встретил сексапильную пышку из Signetics, и она его прекрасно понимает*" — Тим Джексон "INTEL — взгляд изнутри".

Эту книгу читать обязательно! Желание связываться с крупными корпорациями сразу же пропадет и мысль устраивается к ним на работу покажется просто бердовой. Сильнее всех страдают русские программисты, не имеющие никакого иммунитета против чумы западного мира. Если говорить кратко (но книжку вы все-таки прочитайте!), там берут специалиста, прогоняют через соковыжималку и выбрасывают как ненужный хлам на свалку. Если же он одумается и попытается свалить из компании в собственный бизнес — его разоряют. Мрак. Полый.



Рисунок 8 бизнес машина коммерческого программирования

грибные места халтурных заказов

не бойся показаться дураком. наоборот, бойся показаться очень умным. потому что тогда немедленно возникнет вопрос: если ты такой умный, то почему нанимаешься на работу, а не нанимаешь на нее?

Пелевин, "Поколение П"

Никогда не ходите на сайты типа job.ru! Свободным копейщикам там нечего делать. Наиболее лакомые куски обычно перепадает от знакомых или знакомых знакомых. Знакомства обычно заводят на форумах, там же можно найти и заказы, поступающие как от частных лиц (других free lancer'ов), так и от программистских фирм, либо нуждающиеся в услуге, которую сами не могут реализовать (написать ассемблерный модуль, дизассемблировать программу, спроектировать контроллер, разработать драйвер и т. д.), либо передают часть своих заказов — иногда безвозмездно, иногда за определенный процент.

Свободными копейщиками чаще всего становятся специалисты редких профессий (ассемблерщики, фортисты и т. д.). Держать их в штате мелким компаниям невыгодно, лучше переплатить за краткосрочный контракт под конкретный заказ.

Существует два диаметрально противоположных способа оплаты: единовременная выплата и отчисление определенного процента с продаж программного пакета (аппаратно-программного комплекса) в создании которого вы принимали участие. Наниматели охотнее идут на единовременную оплату, размер которой в зависимости от специфики заказа колеблется от сотен до десятков тысяч долларов. Величина отчислений (так называемых royalty), редко превышает 10% от розничной/оптовой стоимости одного экземпляра ПО, но даже 1%-3% лучше чем своим ничего. Тут все зависит от раскрутки продукта и объемов продаж. На отчислениях можно неплохо заработать, однако можно и потерять (попробуй проконтролировать сколько

копий продано — одна или миллион!), в то время как сумма единовременной выплаты — гарантирована. В общем, royalty — это журавль в небе, а единовременная выплата — синица в руке. Лично мне журавли нравятся больше, хотя здесь не обходится без разочарований и обманов.

Вот, из Пелевина: "...если один из десяти проектов кончается успешно, это уже большая удача. В бизнесе никогда не следует проявлять поспешности, иначе сильно сбавляешь цену, а это глупо: продавать самое святое и высокое надо как можно дороже, потому что потом торговать будет уже нечем."

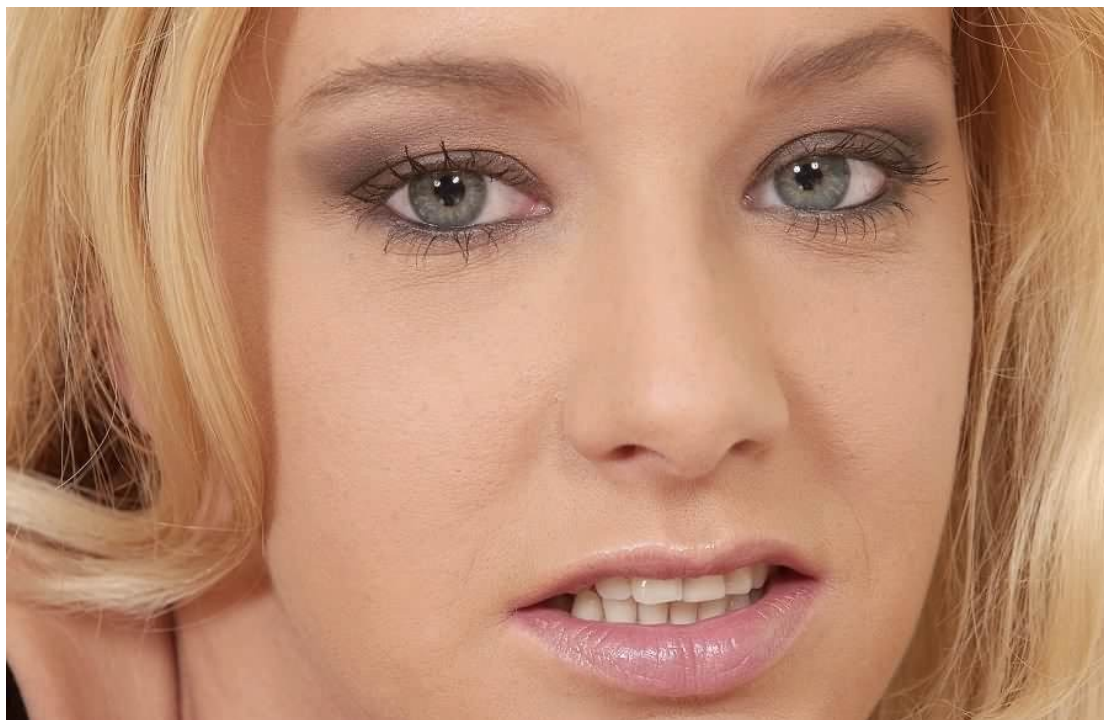


Рисунок 9 в мире есть не только программирование но и вот такие вот женщины

коммерческая природа некоммерческого Open Source

любой имидж имеет четкое денежное выражение. если даже он подчеркнуто некоммерческий, то сразу возникает вопрос, насколько коммерчески ценен такой тип некоммерциализованности

Виктор Пелевин, "Поколение П"

В открытых исходных текстах больше идеологии и пропаганды, чем, собственно, самой технологии. Можно ли на этом заработать? Традиционная формулировка адептов Open Source: "программное обеспечение должно быть бесплатно, деньги берутся только за поддержку" не выдерживает никакой критики. Присмотритесь к Microsoft и другим корпоративным производителям. Платный софт и бесплатная поддержка. Это наиболее устойчивая бизнес-схема, проверенная временем. Да! На поддержке можно заработать, но... только отнюдь не самим разработчикам открытого продукта. Допустим, вы выпустили компилятор и он стал популярным. Постепенно, вокруг него начинает формироваться целое сообщество — кто-то пишет книги, кто-то консультирует пользователей, кто-то приторговывает библиотеками и т. д. Возникает децентрализованная инфраструктура, проворачивающая миллионы долларов, но не желающая делиться с создателями.

Без щедрого спонсора здесь не обойтись. А кто может выступить в такой роли? Компании составители дистрибутивов, исследовательские организации и институты, конкурирующие фирмы и корпорации, рекламодатели, филантропы и т. д. С хорошим знанием английского и широко подвешенным языком выбить субсидии для своего проекта в общем-то несложно. Как поступают китайские парни? Сначала они в муках рожают минимально

работающий продукт, затем показывают его богатым дяденькам и просят денежку. Русский программист... Ого! Это совсем другой тип. Никакого продукта у него нет и в лучшем случае присутствует только проект, а все остальное занимают широко расставленные пальцы и грандиозные планы по сдвигу земной оси в неизвестном науке направлении. За пятизначную сумму в твердой валюте он с сотоварищами готов весь мир перевернуть, ну если не перевернуть так озолотить. При достаточной напористости (мамой клянусь, да!) запрошенная сумма все-таки выделяется. На эти деньги снимается офис, покупается джип и устраивается грандиозный кутеж с резней в Квейка с вечера до утра. Когда же приходит срок сдачи проекта (ломаете кайф, мужики!) и в офисе появился хмурое лицо инвестора, русский программист демонстрирует дохлый макет и говорит, что для окончательного завершения проекта ему требуется миллион долларов и еще один год времени, но это будет такая крутая программа, которой вообще ни у кого нет (гладит, стирает и даже летает) и которая принесет инвестору миллиардные прибыли (Бил Гейтс и рядом не лежал). Инвестор хмурится еще больше (куда наживаться, вернуть бы свое!), но в трех из пяти случаев деньги все-таки предоставляет (американцы — они такие наивные). Через год, когда инвестор приходит опять (а никаким продуктом в воздухе и не пахнет), русский программист говорит возмущенно: "а что, вы, собственно, хотели за такие деньги? программные комплексы такого масштаба создаются не с полпинка, а требуют миллиардных вложений!"

Вот почему с русскими программистами никто не хочет сотрудничать и получить деньги можно только влившись в состав забугорной команды, или создав фиктивную фирму на тихоокеанских островах (шутка, конечно, но для истины в ней все-таки есть). Еще вам потребуется хотя бы минимально работающий продукт. А где его взять? Создавать в одиночку — проблематично. Работать с командной — рискованно. Где гарантия, что один из членов команды не упрет исходные тексты и не сольет их на сторону? Или, когда программа вот-вот заработает, лидер группы не пошлет всех спецов к чертовой матери, наймет десяток-другой бестолковых студентов и не поручит им финальный "косметический ремонт"? И с тем, и с другим автору этой статьи приходилось сталкивался неоднократно.

Ладно, не будем о грустном. Представим, что вы работаете в одиночку или в группе близких друзей, которым верите как себе самому (я знаю программистскую семью, в которой жена, дед, отец и сын — все программисты). Вы пишете бесплатную версию какого-нибудь коммерческого продукта, в надежде заставить его создателей раскошелиться, ведь это прямой урон ихнему бизнесу. И действительно, через некоторое время к вам поступает предложение от которого нельзя отказаться. Продукт либо купят за единовременную сумму, либо предложат перейти в штат на постоянный оклад (с условием прекращения развития продукта). При желании можно и поторговаться, но главное не перегнуть. Если конкурента нельзя купить, в России его убивают (ну для начала могут и избить), а за границей вручают иск за нарушение авторского и патентного права. А в том, что вы его нарушили — можете не сомневаться. При этом истец может беззастенчиво воровать у вас лучшие куски программного кода и хрен вы что докажете!

Короче, если подходить к открытым источникам с чисто коммерческой точки зрения, выискивая наивных спонсоров и переводя деньги на номерные счета, в месяц выходит от \$1.000 до \$10.000 (при этом даже не требуется запускать компилятор!). Честным трудом в лучшем случае заработаешь долларов пятьсот, ну или от силы пару тысяч. Отсюда: заниматься Open Source имеет смысл лишь для приобретения программистского опыта, выгодного трудоустройства или по глубоким идейным причинам.

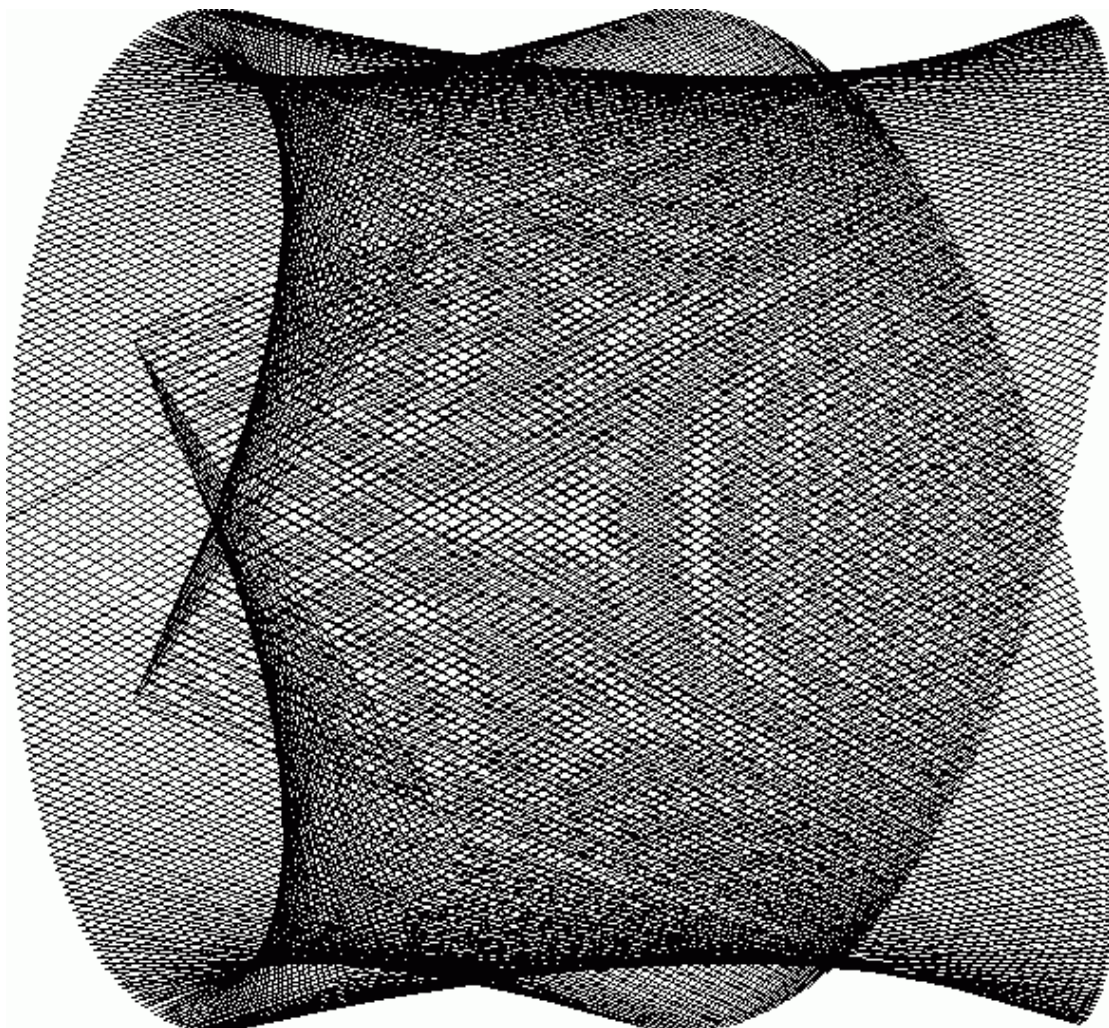


Рисунок 10 граница соприкосновения парадоксов

условно бесплатные заработки условно бесплатного обеспечения

альтернативная музыка — это такая музыка, коммерческой эссенцией которой является ее предельно антикоммерческая направленность. так сказать, антипопсовость. поэтому, чтобы правильно просечь фишку, альтернативный музыкант должен прежде всего быть очень хорошим поп-коммерсантом, а хорошие коммерсанты в музыкальный бизнес идут редко. то есть идут, конечно, но не исполнителями, а управляющими

Виктор Пелевин, "Поколение П"

Плодотворная нива условно-программного обеспечения держится уже на пределе. Ну не в состоянии она прокормить армаду прикладных программистов, клепающих безвкусные программы, гребящиеся при первом запуске. Массовый рынок выжат досуха и чтобы найти свою нишу требуется неординарная фантазия, мгновенное озарение, инженерная закалка наконец! Не писать же в самом деле очередной клон Norton Commander'a, Панели Быстрого Запуска или Менеджера Закачек. Все это уже написано за нас и, чтобы столкнуть укоренившихся фаворитов с пьедестала (или хотя бы заставить их потесниться), потребуется развернуть колоссальный маркетинг — заниматься не столько программированием, сколько рекламой и продвижением. Собственно говоря, в маркетинг все и упирается. Удачные технические решения поигрывают грамотному позиционированию. Зачем напрягаться и чего-то

там кодить? Берем готовый продукт, слегка дорабатываем его напильником (чтобы не был так узнаваем) и начинаем раскрутку под новым именем. Ведь в конечном счете конкурируют не конкретные продукты, а их абстрактные имиджи. Но грамотных маркетологов среди программистов что-то не наблюдается.

Формально, шароварщик относится к свободным копейщикам, но при ближайшем рассмотрении между ними обнаруживается огромная разница. Свободный копейщик в основном программирует крошечные фрагменты большого продукта (защитный механизм, ассемблерный модуль, аппаратный контроллер и т. д.), в то время как шароващик вынужден брать на себя весь стадию разработки целиком — от проектирования интерфейса до составления документации. Но это еще что! Спихнув заказ, свободный копейщик тут же забывает о нем, а шароващик вынужден поддерживать однажды родившееся дитя чуть ли не всю сознательную жизнь. Вести более трех независимых проектов на таких условиях просто немыслимо. Кому-то интересно из года в год ходить по протоптанному кругу, а кто-то требует постоянной смены обстановки, новых горизонтов и перспектив.

Доход шароващиков в среднем в несколько раз меньше, чем у free lancer'ов. Грубо это можно пояснить на следующем примере: заклинил у одной компании электродвигатель. Как ни пыhtели штатные инженеры, так ничего и не нашли. Позвали free lancer'a. Незлобно пнув агрегат ногой, тот одним ударом молотка восстановил его работоспособность, потребовав за свою работу \$1.000. Обосновал он это так — \$1 за удар, \$999 за знание места куда нужно было ударить. Вывод — свободный копейщик продает свой интеллект в чистом виде, а шароващик — грубый физический труд. Интеллект, естественно, ценится намного выше.

Типичная программа на 90% состоит из дебильного кода, который может написать любой студент, и создавать шаровары в одиночку — это все равно что заниматься умственным онанизмом. Лучше разработайте "вычислительное" ядро (так же называемое движком) и продайте его "студентам" (они же "пионеры") — нехай дописывают все остальное. Собственно говоря, свободные копейщики именно так и поступают. Но чтобы знать по какому месту ударить, необходимо пропустить через свои руки тысячи двигателей! Об этом нельзя просто взять и прочитать в журнале! Профессиональные навыки вырабатываются годами. Образно говоря, свободный копейщик это повзрослевший шароващик. Молодые программисты в условиях дикой природы долго не выживают.

Когда шаровара готова ее остается только продать. Лучше всего поручить это дело фирме-регистратору. Она возьмет на себя и финансы, и маркетинг, регулярно пополняя ваш банковский счет новыми суммами. Естественно, это делается не бесплатно и типичная такса составляет 10%-20% от розничной стоимости программы. Розничная стоимость обычно назначается самим регистратором и может непредсказуемым образом меняться в зависимости от конъюнктуры рынка и динамики спроса. Перечень популярных регистраторов лежит на сайте www.regshare.com.

По своему опыту и опыту моих коллег могу сказать, что доход от \$300 в месяц для начинающего шароващика это уже удача. Преодолеть \$500-\$1.000 барьер удастся лишь единицам, хотя это очень субъективные оценки. Возможно, вам повезет больше.



Рисунок 11 счастье не в деньгах, а том, чтобы иметь на кого их тратить...

заклучение

*вот многие говорят: водка, вино, ганджубас, другие
веселящие снадобья. а ведь, положя руку на половой
орган, нет ничего круче, чем как следует выспаться.*

Гоблин

Так все-таки можно зарабатывать на программировании или нет? Да как вам сказать... В глубинке, молодой специалист получающий порядка \$500 в месяц и не обремененный подхалимажем перед вечно недовольным начальством, вызывает у окружающих смесь зависти с восхищением. Но и работать ему приходится ой-ой-ой! В столице и промышленных центрах страны, эта цифра вызывает снисходительную улыбку с одобряющим похлопыванием по плечу: "...может тебе одолжить, а? У тебя вид какой-то запущенный".

Профессия программиста уже утратила свой мистический ореол и ее популярность тает прямо на глазах. Случайных людей здесь становится все меньше и меньше. В программирование идут преимущественно те, кому интересно проектировать структуры данных, листать потрепанную документацию, ковыряться в отладчике... Чем качественнее код, тем ниже его доходность, однако, погоня за личным обогащением опускает вычислительную технику в глубокую яму. Чтобы многолетние отложения глюкавого когда не рухнули окончательно, нужно забыть о деньгах и вспомнить, что Россия это страна с традиционно высокой инженерной культурой и неординарными людьми.